Discovering Syntactic Phenomena with and within Precision Grammars

A thesis presented
by
Ned Letcher

to
The Department of Computing and Information Systems
in total fulfillment of the requirements
for the degree of
Doctor of Philosophy

The University of Melbourne
Melbourne, Australia
July 2018
Declaration

This is to certify that:

(i) the thesis comprises only my original work towards the PhD except where indicated in the Preface;

(ii) due acknowledgement has been made in the text to all other material used;

(iii) the thesis is fewer than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed: ____________________________ Date: ________________
Abstract

Precision grammars are hand-crafted computational models of human languages that are capable of parsing text to yield syntactic and semantic analyses. They are valuable for applications requiring the accurate extraction of semantic relationships and they also enable hypothesis testing of holistic grammatical theories over quantities of text impossible to analyse manually. Their capacity to generate linguistically accurate analyses over corpus data also supports another application: augmenting linguistic descriptions with query facilities for retrieving examples of syntactic phenomena.

In order to construct such queries, it is first necessary to identify the signature of target syntactic phenomena within the analyses produced by the precision grammar in use. This is often a difficult process, however, as analyses within the descriptive grammar can diverge from those in the precision grammar due to differing theoretical assumptions made by the two resources, the use of different sets of data to inform their respective analyses, and the exigencies of implementing a large-scale formalised analyses.
In this thesis, I present my research into developing methods for improving the discoverability of syntactic phenomena within precision grammars. This includes the construction of a corpus annotated with syntactic phenomena which supports the development of syntactic phenomenon discovery methodologies. Included within this context is the investigation of strategies for measuring inter-annotator agreement over textual annotations for which annotators both segment and label text—a property that traditional kappa-like measures do not support. The second facet of my research involves the development of an interactive methodology—and accompanying implementation—for navigating the alignment between dynamic characterisations of syntactic phenomena and the internal components of HPSG precision grammars associated with these phenomena. In addition to supporting the enhancement of descriptive grammars with precision grammars, this methodology has the potential to improve the accessibility of precision grammars themselves, enabling people not involved in their development to explore their internals using familiar syntactic phenomena, as well as allowing grammar engineers to navigate their grammars through the lens of analyses that are different to those found in the grammar.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Page</td>
<td>i</td>
</tr>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>ix</td>
</tr>
<tr>
<td>Dedication</td>
<td>xi</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Research Scope and Contributions</td>
<td>8</td>
</tr>
<tr>
<td>1.1.1 Research Scope</td>
<td>8</td>
</tr>
<tr>
<td>1.1.2 Research Contributions</td>
<td>8</td>
</tr>
<tr>
<td>1.2 Thesis Outline</td>
<td>10</td>
</tr>
<tr>
<td><strong>2 Background</strong></td>
<td>13</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>13</td>
</tr>
<tr>
<td>2.2 Setting the Scene</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1 Computational Linguistics</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 Describing Language</td>
<td>16</td>
</tr>
<tr>
<td>2.2.3 Corpus Linguistics</td>
<td>21</td>
</tr>
<tr>
<td>2.2.4 Situating this Research</td>
<td>22</td>
</tr>
<tr>
<td>2.3 Grammatical Phenomena</td>
<td>23</td>
</tr>
<tr>
<td>2.3.1 Linguistic Forms</td>
<td>24</td>
</tr>
<tr>
<td>2.3.2 The Function of Forms</td>
<td>28</td>
</tr>
<tr>
<td>2.3.3 Characterising Grammatical Phenomena</td>
<td>30</td>
</tr>
<tr>
<td>2.3.4 Syntactic Phenomena</td>
<td>33</td>
</tr>
<tr>
<td>2.3.5 Complexity</td>
<td>34</td>
</tr>
<tr>
<td>2.3.6 Frequency</td>
<td>37</td>
</tr>
<tr>
<td>2.3.7 Summary</td>
<td>39</td>
</tr>
<tr>
<td>2.4 Descriptive Grammars</td>
<td>39</td>
</tr>
<tr>
<td>2.4.1 Audience</td>
<td>40</td>
</tr>
<tr>
<td>2.4.2 Accessibility and Basic Linguistic Theory</td>
<td>41</td>
</tr>
<tr>
<td>2.4.3 The Anatomy of a Descriptive Grammar</td>
<td>43</td>
</tr>
<tr>
<td>2.4.4 Organisational Strategies</td>
<td>49</td>
</tr>
<tr>
<td>2.4.5 Text Collections</td>
<td>54</td>
</tr>
</tbody>
</table>
2.4.6 Summary ................................................................. 57
2.5 Treebanks ................................................................. 58
  2.5.1 Applications ......................................................... 59
  2.5.2 Querying Treebanks ................................................. 61
  2.5.3 Treebanking Methodology ......................................... 61
  2.5.4 Summary .............................................................. 63
2.6 Grammar Engineering .................................................... 64
  2.6.1 Applications .......................................................... 66
  2.6.2 Engineering Requirements ......................................... 71
  2.6.3 Workflow ............................................................... 74
  2.6.4 Linguistic Test Suites .............................................. 76
  2.6.5 Dynamic Treebanking .............................................. 79
  2.6.6 Multilingual Grammar Engineering ............................. 81
  2.6.7 Summary .............................................................. 87
2.7 Electronic Grammaticography .......................................... 87
  2.7.1 Overview ............................................................... 88
  2.7.2 Enhancements ......................................................... 89
  2.7.3 Taalportaal ............................................................ 95
  2.7.4 Summary .............................................................. 98
2.8 Projects Involving Grammatical Phenomena ........................... 99
  2.8.1 The TSNLP Project .................................................. 100
  2.8.2 A Theory of Syntactic Phenomena ............................... 105
  2.8.3 The ERG Semantic Documentation Project ..................... 108
  2.8.4 DepBank ............................................................... 112
  2.8.5 CCGBank .............................................................. 114
  2.8.6 Construction-focused Parser Evaluation ....................... 117
  2.8.7 Summary .............................................................. 120
2.9 Syntactic Phenomenon Discovery ..................................... 122
  2.9.1 Querying Pre-annotated Databases ............................. 123
  2.9.2 Signature-based Querying ......................................... 129
  2.9.3 Query-by-example .................................................. 136
  2.9.4 Summary .............................................................. 145
2.10 Chapter Summary ....................................................... 146

3 Resources ........................................................................ 150
  3.1 Introduction ................................................................ 150
  3.2 The LinGO English Resource Grammar ............................. 150
    3.2.1 Grammatical Phenomena ......................................... 151
    3.2.2 Robustness and Accuracy ....................................... 155
    3.2.3 High Quality Treebanks ....................................... 157
  3.3 DeepBank .................................................................. 159
  3.4 The Cambridge Grammar of the English Language ............... 160
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 Summary</td>
<td>161</td>
</tr>
<tr>
<td><strong>4 The Phenomenal Corpus</strong></td>
<td><strong>163</strong></td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>163</td>
</tr>
<tr>
<td>4.2 Data Desiderata</td>
<td>164</td>
</tr>
<tr>
<td>4.2.1 Corpus Data</td>
<td>164</td>
</tr>
<tr>
<td>4.2.2 Exhaustive Annotation</td>
<td>166</td>
</tr>
<tr>
<td>4.2.3 Character-level Annotation</td>
<td>167</td>
</tr>
<tr>
<td>4.2.4 Grammatical Framework-independent</td>
<td>167</td>
</tr>
<tr>
<td>4.3 Existing Phenomenon-annotated Resources</td>
<td>168</td>
</tr>
<tr>
<td>4.4 Selection of Phenomena</td>
<td>171</td>
</tr>
<tr>
<td>4.4.1 Relative clauses</td>
<td>173</td>
</tr>
<tr>
<td>4.4.2 Interrogative Clauses</td>
<td>174</td>
</tr>
<tr>
<td>4.4.3 Passive Clauses</td>
<td>176</td>
</tr>
<tr>
<td>4.4.4 Imperative Clauses</td>
<td>177</td>
</tr>
<tr>
<td>4.4.5 Complement clauses</td>
<td>177</td>
</tr>
<tr>
<td>4.4.6 Summary of Phenomena</td>
<td>179</td>
</tr>
<tr>
<td>4.5 Selection of Corpora</td>
<td>179</td>
</tr>
<tr>
<td>4.6 Data Collection</td>
<td>181</td>
</tr>
<tr>
<td>4.6.1 Methodology</td>
<td>181</td>
</tr>
<tr>
<td>4.6.2 Annotation Guidelines</td>
<td>182</td>
</tr>
<tr>
<td>4.6.3 Annotations Collected from the WSJ</td>
<td>186</td>
</tr>
<tr>
<td>4.6.4 Creating the Gold Standard Annotations</td>
<td>186</td>
</tr>
<tr>
<td>4.6.5 Error Analysis</td>
<td>187</td>
</tr>
<tr>
<td>4.6.6 Improving the Annotations</td>
<td>188</td>
</tr>
<tr>
<td>4.6.7 Data Packaging</td>
<td>191</td>
</tr>
<tr>
<td>4.7 Assessing Annotation Reliability</td>
<td>193</td>
</tr>
<tr>
<td>4.7.1 Chance-corrected Inter-annotator Agreement</td>
<td>193</td>
</tr>
<tr>
<td>4.7.2 Challenges Presented by this Annotation Task</td>
<td>196</td>
</tr>
<tr>
<td>4.7.3 Inter-annotator Agreement Results</td>
<td>203</td>
</tr>
<tr>
<td>4.7.4 IOBE Tagging Analysis</td>
<td>208</td>
</tr>
<tr>
<td>4.8 Discussion and Summary</td>
<td>212</td>
</tr>
<tr>
<td>4.8.1 Discussion of Quality Assurance</td>
<td>212</td>
</tr>
<tr>
<td>4.8.2 Discussion of Methodology</td>
<td>216</td>
</tr>
<tr>
<td>4.8.3 Summary</td>
<td>218</td>
</tr>
<tr>
<td><strong>5 Syntactic Phenomenon Discovery</strong></td>
<td><strong>220</strong></td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>220</td>
</tr>
<tr>
<td>5.2 Proposed Approach</td>
<td>222</td>
</tr>
<tr>
<td>5.3 DELPH-IN Grammar Architecture</td>
<td>224</td>
</tr>
<tr>
<td>5.4 Typedef Methodology</td>
<td>231</td>
</tr>
<tr>
<td>5.4.1 Type Extraction</td>
<td>232</td>
</tr>
</tbody>
</table>
Acknowledgments

Writing these words feels slightly surreal, as for much of this journey I’ve struggled to imagine actually getting to the end. Here I am though, and there are a number of people I need to thank for their support in getting me to this point.

First and foremost, I want to express my deepest gratitude to my supervisors, Tim Baldwin and Emily Bender. Without their unending support and commitment to helping me see this through, I certainly wouldn’t be writing these words. Were someone to ask me what I thought made for good supervisor material, I would simply point in their respective directions. I thank them both for their excellent mentorship and company that has been constant throughout my studies. A special thanks goes to Dan Flickinger, who graciously assisted me at a number of points in my research and indulged my predilection for cooking up tools that may or may not be useful to grammar engineers. Our ever enjoyable conversations played a significant role in shaping the direction of my research and helped me gain a better understanding of the craft of grammar engineering. I also want to thank the DELPH-IN community for being such a welcoming bunch of folk, and for playing such an important part of my academic adventures. In particular, I would like to thank the following people for their advice and discussions: Rebecca Dridan, Stephan Oepen, Francis Bond, Michael Goodman, Woodley Packard, Antske Fokkens, Glenn Slayden, and Sanghoun Song. Thanks also to Rachel Nordlinger, Steven Bird, Karin Verspoor, Paul Cook, Bob Carpenter, Jeremy Nicholson, and Harald Hammarström, whose support and discussions were most valuable. Thanks also to my excellent team of proof readers, Richard, Hannah, Stu, and David. I would also like to express my appreciation to the following institutions for providing me with funding for the various travels I was lucky enough to go on: The University of Melbourne, the University of Washington, the University of Oslo, and Stanford University.

One of the highlights of this PhD thing was the opportunity to live in Seattle for half a year. I want to send a massive thank you to everyone who helped make this as rewarding an experience as it was: Emily Bender for hosting me and providing the warmest of welcomes; Harmanas, for being such a delightful and unexpected intellectual companion; Michael, Ning, and Prescott for being such great neighbourhood friends; John, Bruce, Rick, Matt, and the rest of the Cascade Jugglers for being consistently good value every week; the excellent people who ran the machines at Ballard Coffee Works and Trabant Coffee in the U District; and Jim White and all the other people whose couches I stayed on until I found a home. A very special and heartfelt thank you goes to Charlie, Pepper, Kim, Michael, and Bianca, who welcomed me into their beautiful home and as part of their family.

Back in Melbourne, I want to thank the past and present members of the NLP group for many good discussions, seminars, and Bla-di-das; the participants of the venerable institution of Breakfast Club, for enabling the shared enjoyment of NLP, data science, fine food, and fine coffee; and Peter Kubas, for his all-encompassing infectious enthusiasm and for helping me care for my physical health while writing up. A special shout-out also to the many cafes of Melbourne that have accommodated my lurking habits and supplied me with what is probably a very large number of long
blacks that fuelled my writing. Much love goes to my dear friends Nick, Dan, Katie, Anna, and Katie, who are an ongoing source of emotional support and nourishment.

And finally, to my family, Libby, Marcus, and Toby: so much love and so much gratitude for always being there for me with support, even when I didn’t realise I needed it, and even when I struggled to reciprocate.
To my parents, Libby and Marcus
Chapter 1

Introduction

This thesis involves two types of grammars of natural language, both of which represent theories about the workings of specific human languages but are substantially different in nature. The first type includes hand-crafted computational grammars—sometimes referred to as precision grammars—which are based on formal models, and are capable of producing linguistically-precise syntactic and semantic analyses of text (Butt et al. 1999, Copestake 2002b). Precision grammars have strong practical value in the context of natural language processing applications such as question answering and information extraction, where the accurate extraction of semantic relationships is critical for performance. They also provide valuable theoretical contributions, facilitating the testing of linguistic hypotheses over large collections of text, which would not be possible without the aid of automation (Bender 2008b). The second type includes descriptive grammars, which are non-formalised, consist largely of prose text and structured description, and document the grammatical analysis of a particular language (Evans and Dench 2006). These grammars aim to provide a thorough treatment of the workings of their respective language and serve as indispensable references for linguists, as well as a range of other audiences. In this thesis, I explore techniques for making the linguistic knowledge contained within precision grammars more discoverable, so that they can be used to enhance the utility of descriptive grammars, in addition to improving the accessibility of precision grammars themselves.

Precision grammar development—the process of implementing, maintaining, and extending analyses of the grammatical phenomena of individual languages—is known as grammar engineering (Bender et al. 2011a). The approach taken towards analysis is typically based on a grammatical framework developed within theoretical linguistics, and the analyses themselves are informed by a range of data sources, including descriptive grammars, native speaker judgements, corpus data, and consultation with expert linguists. While precision grammars prioritise linguistic accuracy over robustness, some also achieve wide coverage over the language being modelled. The precision grammars used in this thesis are developed within the DELPH-IN consor-
Chapter 1: Introduction

Delph-In, an international collaboration dedicated to developing efficient and precise linguistic models for mapping text onto its underlying semantic structure. Delph-In grammars are based on the grammatical framework of Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag 1994) and share a joint reference formalism for implementation (Copestake 2002a), as well as a common toolchain for text processing, evaluation, and grammar development. The primary grammar used in this thesis is the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011), a broad coverage grammar of English that has been actively developed for over two decades.

Examples of grammatical analysis that are characteristic of the two different types of grammars—descriptive and precision—are shown in Figure 1.1 and Figure 1.2. These both pertain to the syntactic phenomenon of nominal complementation, a construction which involves a phrase combining with a noun, and in which the phrase forms an integral semantic contribution to the complete noun phrase. Figure 1.1 is taken from The Cambridge Grammar of the English Language (Huddleston and Pullum 2002:439), a comprehensive descriptive grammar of English, which features in the research presented in this thesis. This fragment illustrates the non-formalised nature of analysis that is characteristic of descriptive grammars, featuring prose descriptions and curated exemplars of the phenomenon. While not formalised, this type of analysis does contain information conveying structure, as can be seen in Figure 1.1, with the tabular annotation of exemplars indicating their subclassification. Figure 1.2 illustrates the formal nature of precision grammar analysis and the accompanying increase in complexity, presenting a fragment of source code from the ERG that includes three distinct grammar components, all associated with the analysis of nominal complements. Each of these corresponds to a subcategory of noun which allows specific complements to attach to it.

There is a considerable amount of overlap in the analytic content of the two types of resources since, for a precision grammar and descriptive grammar targeting the same language, they must both ultimately account for the same grammatical phenomena. Descriptive grammars may also be consulted to provide or inform the analyses found within the precision grammar. A notable example of this is Bender’s (2008a) implemented grammar of the Australian language Wambaya, which is based on Nordlinger’s (1998) descriptive grammar of Wambaya. This overlap can be seen within the two fragments of nominal complementation analysis. Looking at the three grammar components in Figure 1.2 from the source code of the ERG, each of these can be associated with specific examples from Figure 1.1. The first code block defines the identifier `n_pp_c-ns-obl_le`, which permits a prepositional phrase to attach to the noun and therefore corresponds to exemplar (i) from Figure 1.1. The second code block defines the identifier `n_cp_mc-optc_le`, which permits a subordinate clause to attach to the noun, corresponding to both (ii) and (iii). Finally, the identifier

---

1 http://www.delph-in.net
Post-head complements have the form of PPs or clauses:

1. *the journey to Rome* / back  
2. *the rumour that the city had been captured*  
3. *the question whether they were guilty*  
4. *the question ‘is God dead?’*  
5. *the decision to abandon the project*

Clauses in complement function are usually subordinate, but main clauses are found in direct reported speech or citation, as in [iv]. Nouns, unlike verbs, do not permit objects and hence do not normally take NPs as complement – NPs occur as oblique complements, related to the head by a preposition; compare, for example, the clause *They destroyed the city* with the NP *their destruction of the city.*

Figure 1.1: A fragment of grammatical analysis from page 439 of *The Cambridge Grammar of the English Language* describing nominal complementation.

```
n_pp_c-ns-obl_le := noun_empty_ppcomp_word &
   [ SYNSEM.LOCAL [ CAT [ HEAD.MINORS.MIN modable_nom_rel,
       VAL.COMPS < [ OPT -,
       LOCAL.AGR.PNG.PN #pn ] > ],
       AGR.PNG.PN #pn,
       CONT [ HOOK.INDEX.SORT non-time-sort,
       RELS <! relation !>,
       HCONS <! !> ] ] ].

n_cp_mc-optc_le := noun_word &
   [ SYNSEM basic_mass_count_noun_cpcomp_fin_synsem &
     [ MODIFD notmod,
       LOCAL.CAT.HEAD.MINORS.MIN norm_nom_rel ] ].

n_vp_c_le := noun_word &
   [ SYNSEM common_noun_vpcomp_synsem &
     [ MODIFD notmod,
       LOCAL.CAT.HEAD.MINORS.MIN norm_nom_rel ] ].
```

Figure 1.2: A fragment of the 1214 release of the ERG which contributes towards constraining the analysis of different kinds of nominal complementation.
\texttt{n_vp.c.le}, defined by the third block, permits a verb phrase to attach to the noun and so corresponds to (v). No grammar component is associated with (iv) as the ERG does not currently handle a quoted main clause as a complement of a noun.\footnote{This is based on the 1214 release of the ERG, which is the version primarily used in this thesis.}

The differing natures of these two types of resources means that they offer complementary sources of information. Precision grammars facilitate the automatic parsing of candidate sentences and the generation of corresponding fine-grained analyses that are guaranteed to be well-formed and consistent. Descriptive grammars, on the other hand, are designed to be human-readable rather than machine-readable and so cannot automatically analyse text. Their grammatical analyses are organised and presented in a way that maximises their accessibility and utility for the reader, contrasting with the internals of precision grammars, whose organisation is dictated by the needs of the primary audience—the machine. The ideal resource would be one which combines both these properties, providing a human-accessible treatment of the workings of a language, while also being able to perform automated grammatical analysis. One way of realising such a resource is through the synergistic combination of existing instances of these two resources, using the generative capacity of precision grammars and the linguistically precise analyses they produce, to enhance the utility of descriptive grammars.

Enhancing descriptive grammars with capabilities enabled through computational means is an area of open exploration, forming the focus of the emerging field of electronic grammaticography. A range of potential enhancements are identified within Nordhoff’s (2008) foundational articulation of the field as well as across the various articles appearing in the edited collection of Nordhoff (2012b). These enhancements include features such as non-linear navigation, dynamic layouts, incremental publication, and augmentation with digital linguistic resources such as corpora, lexica, glossaries, and audio and video recordings. The enhancements motivating the research presented in this thesis involve improvements to descriptive grammars that enhance the discoverability of syntactic phenomena—grammatical phenomena which pertain to how words and phrases combine to form grammatical sentences of language. These motivating enhancements can be summarised as:

1. Hypertext and non-linear navigation
2. Query facilities
3. Integration with text collections

The first enhancement involves enabling navigation of descriptive grammars by means of phenomenon-labelled links to example text that exhibit the relevant phenomena. This would serve to improve the discoverability of syntactic phenomena within descriptive grammars, with hyperlinked exemplars facilitating non-linear navigation of what are currently linear documents—a manner of interaction that is closer
to how people actually use descriptive grammars (Johnsen 1996). The second enhancement allows for additional examples of sentences containing a syntactic phenomenon to be retrieved from a supplementary corpus. Someone reading the section of Huddleston and Pullum (2002) describing nominal complementation, for example, may have residual questions regarding this phenomenon. The facility to retrieve further examples of the different types of nominal complementation from a corpus would present the reader with the opportunity to gain further insight. The third enhancement involves the application of query facilities to text collections that were used to develop the analyses found in the descriptive grammar, resulting in better integration of descriptive grammars with their text collections. The creation of such links between the descriptive grammar and the primary data upon which it is based would also improve the ability for readers to either verify or falsify the analysis put forth in the descriptive grammar (Mosel 2006, Thieberger 2008).

As argued by Bender et al. (2012), precision grammars represent an opportunity to facilitate these enhancements. By identifying the signature of specific syntactic phenomena within the representations of their analyses, the capacity of precision grammars to dynamically parse text and produce high-quality treebanks containing linguistically precise analyses can be leveraged to label and index phenomena contained within example sentences beyond those they exemplify, support the querying for and retrieval of instances of phenomena from a corpus, and generate links between the descriptions and supporting text collections. A demonstration of this enhancement through querying facilities is Bouma et al. (2015), who annotate a sample of exemplars and descriptions within an electronic descriptive grammar with queries that retrieve instances of the relevant phenomena from various corpora. These queries were developed with the aid of the Alpino Dutch precision grammar (van Noord 2006), which produces analyses consistent with those found in the treebanks, and have since been incorporated within Taalportaal (Landsbergen et al. 2014, van der Wouden et al. 2017), an online grammar of Dutch, which, I argue, currently represents the most fully realised example of an electronic descriptive grammar.

Since precision grammars are both machine-readable and designed with the express purpose of modelling the constraints of language, it may seem like identifying the signature of syntactic phenomena within the analyses they produce should be straightforward. The reality, however, is far from this. As linguistic abstractions, syntactic phenomena do not exist as Platonic Forms, but are always situated within a particular context of theoretical presuppositions. The task of identifying the signature of a syntactic phenomenon implies a particular characterisation of a phenomenon, which necessarily invokes a collection of theoretical presuppositions, either explicitly, as can be seen within descriptive grammars, or implicitly, potentially appealing to a shared body of linguistic knowledge.³ There are then two sets of analyses that

³The shared body of linguistic knowledge that provides common ground for couching grammatical analysis has been referred to as Basic Linguistic Theory (Dixon 1997), which I discuss in
Chapter 1: Introduction

need to be aligned: those assumed by the target syntactic phenomenon, and those found in the precision grammar. For a given language, the analysis of its grammatical phenomena across a given descriptive grammar and precision grammar pair are almost certain to diverge at times, whether because of differences in background theoretical assumptions, different sets of data used to inform the analyses, the practical constraints imposed by a large-scale engineering project, or simply because of the complexity inherent within language systems, which inevitably results in facets of analysis where the determination of what constitutes a parsimonious solution will involve some subjectivity.

These divergences of analysis can take the form of incompatibilities between the two resources, an example of which is reported by Bouma et al. (2015) in the context of using a precision grammar to align the representation of syntactic phenomena within a treebank with the analyses found within the Taalportaal electronic descriptive grammar. Taalportaal analyses nominal complements of adjectives as taking both genitive and dative forms, however within analyses produced by the Alpino precision grammar, the dative complements are instead analysed as indirect objects of the verb. Alternatively, the analyses of the two resources may be broadly compatible, but make different distinctions, making alignment more challenging. This can be seen concretely within the two fragments of analysis presented in Figure 1.1 and Figure 1.2. The second grammar component from the ERG fragment presented in Figure 1.2 permits nouns to have a subordinate clause as a complement, but does not distinguish between declarative and interrogative clauses, meaning that this component is responsible for analysing both (ii) and (iii) from Figure 1.1. In other words, the two different resources make different distinctions regarding the subtypes of nominal complementation. This can also happen in the other direction, with the ERG making distinctions that the descriptive grammar does not. The first grammar component appearing in Figure 1.2—which is responsible for handling the prepositional complement found in (i)—is but one of 31 grammar components the ERG uses to handle specific types of prepositional complements of nouns. Furthermore, the coverage of the implemented resource represents an additional complicating factor: the ERG does not currently handle a quoted main clause as a complement of a noun, and so does not provide a correct analysis for (iv).

In order to augment descriptive grammars with the capabilities of precision grammars it is therefore necessary to address the challenges of identifying the signature of target syntactic phenomena within precision grammars. I argue that a fruitful strategy for addressing this situation is the development of techniques for improving the discoverability of syntactic phenomena within precision grammars.

In addition to supporting descriptive grammar augmentation, the development of such techniques would also have the effect of improving the accessibility of precision grammars themselves. This would firstly be beneficial for researchers who want to
make use of a precision grammar but have not been involved in its development, for whom the grammar’s codebase and analyses it encodes can initially seem impene-
trable. Discovery techniques would present such users an opportunity to bootstrap their understanding of the grammar through syntactic phenomena they are familiar with. These techniques could also improve the accessibility of precision grammars to grammar engineers, allowing them to navigate their own grammars by means of syntactic phenomena. This would allow grammar engineers to view the constraints of the grammar through the lens of a particular characterisation of a syntactic phenomenon that is different to that invoked by the analysis found within the grammar. It could also help grammar engineers to more readily locate the various constraints associated with a phenomenon, which, due to the inherently interconnected nature of language systems and the constraints imposed by implementation, may be located within disparate parts of the grammar.

Motivated by these challenges, this thesis presents the development of an interactive methodology designed to assist users navigate the alignment of a particular characterisation of syntactic phenomena with corresponding components of a precision grammar, as well as the development of supporting resources and techniques. This methodology is oriented towards precision grammars that are based on HPSG principles, but has the potential to be extended to grammars informed by other grammatical frameworks. Rather than attempting to derive a generalised cross-linguistic solution to this problem, I focus on establishing the utility of the discovery methodology against the ERG and The Cambridge Grammar of the English Language, which represent a compelling precision grammar and descriptive grammar pair due to the considerable size and coverage over syntactic phenomena exhibited by both resources. The methodology itself however makes no explicit assumptions about the language being targeted and is likely to extend to precision grammars from typologically diverse language groups.

As is the case for any data-driven methodology, it is necessary to have some amount of data that supports development and evaluation. The type of data required for the development of this kind of methodology for syntactic phenomenon discovery is text that is annotated with occurrences of syntactic phenomena. In this thesis, I argue that existing syntactically annotated corpora are not appropriate for this purpose, as they typically invoke formal analysis rather than descriptive, or do not employ exhaustive annotation, consisting of exemplar sentences that are selected to illustrate specific phenomena, omitting other phenomena that occur incidentally within the sentence and which are targeted by the corpus. This thesis therefore also presents the construction of an annotated corpus designed to support the development of the phenomenon discovery methodology, in addition to an investigation into strategies for assessing the quality of the annotations collected in the process of its development.
1.1 Research Scope and Contributions

Having described the context this research is situated within and the outcomes that motivate it, in this section I more concretely identify the focus of study. In order to do this, I firstly outline the scope of this research, followed by the identification of the specific contributions that this research makes.

1.1.1 Research Scope

The research in this thesis aims to develop techniques and resources that support the challenging task of navigating the alignment of precision grammars and corresponding descriptive grammars of the same language. This alignment enables the two applications of 1) augmenting descriptive grammars with precision grammar-enabled affordances and 2) improving the accessibility of precision grammars, both of which I described in the previous section. In the following section, I outline concretely what the contributions of this research are in pursuit of developing these supporting techniques and resources, and in Chapter 2, I unpack the two motivating applications and provide arguments as to why the antecedent techniques and resources developed in this thesis are required for these applications.

While this research focuses on the development of supporting techniques and resources, rather than on the applications themselves, in order to assess the efficacy of the grammar alignment techniques developed in this research, in Section 5.7 I use the application of improved precision grammar accessibility as a case study, demonstrating that a tool for navigating the syntactic analyses produced by a precision grammar can be used to gain a rapid understanding of the analyses found within the precision grammar, as well as gain an understanding of how they relate to the set of analyses found in a descriptive grammar of the same language. This successful demonstration shows that these techniques would likely have utility for the application of augmenting descriptive grammars with precision grammars, however I do not investigate this particular application within this research.

1.1.2 Research Contributions

The research contributions of this research can be divided into two groups. The first group of contributions, which is described in Chapter 4, involves the construction of a corpus annotated with syntactic phenomena, created to support the development of syntactic phenomenon discovery techniques. The second group of contributions, which is described in Chapter 5, relates to the development of a methodology for performing syntactic phenomenon discovery with and within precision grammars.
Syntactic Phenomenon Annotation

1. In order to drive the development of a syntactic phenomenon discovery methodology, I constructed a small corpus of English text annotated with occurrences of relative clauses, complement clauses, and passive clauses. This was composed of 250 lines from a Sherlock Holmes novel and 477 lines of Wall Street Journal text from the Penn Treebank (Marcus et al. 1993). This corpus, entitled the Phenomenal Corpus, was developed on the basis that no existing resources provided the necessary characteristics to support the development and evaluation of syntactic phenomenon discovery methodologies. These properties—which the newly derived corpus possesses—are as follows:

   - The text is sourced from naturally occurring corpus data, as opposed to being manually curated.
   - Across all sentences, every instance of each target phenomenon is annotated.
   - The phenomenon instances are annotated at the character level, delimiting the textual span involved in each construction.
   - Annotations are based on descriptive characterisations of syntactic phenomena, as opposed to being extracted by a computational model.

2. I investigated strategies for performing quality assurance over annotations produced by annotators both segmenting and labelling text. This constitutes a novel annotation task, as techniques typically used for assessing inter-annotator agreement within computational linguistics, such as Fleiss’ kappa (Fleiss 1971) and other kappa-like measures, require that annotators label pre-segmented spans. I assessed the use of $\alpha_U$ (Krippendorff 1995), an inter-annotator agreement measure that does support annotations involving both segmentation and labelling, but which has not seen adoption within the field of computational linguistics (Artstein and Poesio 2008). The results of my investigations do not lead me to recommend the use of $\alpha_U$ in the context of annotation tasks such as that used to develop the Phenomenal Corpus. Instead, I provide concrete suggestions for an alternative approach for performing quality assurance, which takes the form of a two-tiered methodology. The first tier involves inter-annotator agreement being measured using a kappa-like measure over tokenised annotations labelled with IOBE tags and provides immediate feedback that might indicate issues with the annotation guidelines. The second tier involves performing error analysis using automatically detectable error categories against a disagreement-resolved set of gold standard annotations, which is more expensive to perform, but provides a more informative picture of the quality of the annotations.
Syntactic Phenomenon Discovery

1. I developed an interactive methodology that facilitates the alignment of a given characterisation of a syntactic phenomenon with associated internal constraints of a precision grammar. This methodology, which I call the Typediff methodology, is intended to support tasks such as the construction of queries that retrieve syntactic phenomena from corpora, labelling exemplars with syntactic phenomena, and improving the navigation of precision grammar internals—all of which first require discovering the signature of the target phenomena within the precision grammar. The methodology takes the form of an interactive process, allowing users to navigate the complexity that is involved in this alignment problem. It adopts a query-by-example mode of interaction, in which users supply input exemplars of the phenomenon, in order to locate corresponding grammar components. Using exemplars as input means that users do not need to be familiar with a query language, as is common with treebank query tools, nor do they require an existing understanding of how the phenomenon is represented within analyses produced by the precision grammar. The use of a query-by-example mode of interaction also yields a methodology that supports the dynamic characterisation of diverse syntactic phenomena, as users only have to provide a group of exemplars that delimits the desired syntactic phenomenon.

2. I developed an implementation of the Typediff methodology as an open source library. This is implemented as a web application, which is accessed via an HTML client that communicates with a server implemented as a Python package. The web application realises all the components of the Typediff methodology, allowing users to submit groups of phenomenon exemplars for processing and providing a range of affordances for users to interact with the extracted features, in order to locate grammar components associated with target phenomena. In addition to serving as a reference implementation of the Typediff methodology, this played an integral role in the development of the methodology itself and was also used to support the experimental application of the methodology to the alignment of phenomena from the Phenomenal Corpus to internal components of the ERG.

1.2 Thesis Outline

Chapter 2 provides an overview of background material relevant to the research described in this thesis. Section 2.2 describes the academic landscape this work is situated within: the fields of computational linguistics, descriptive linguistics, formal linguistics, linguistic typology, and corpus linguistics. Section 2.3 then defines the notion

---

4https://github.com/ned2/typediff
of a grammatical phenomenon as it is employed within linguistics, before delimiting the focus of the thesis to syntactic phenomena, a subset of grammatical phenomena. Section 2.4 introduces the linguistic resource that is the descriptive grammar, describing its broad characteristics so as to establish how it can be enhanced with precision grammars. Section 2.5 provides a brief overview of treebanks, notably outlining traditional approaches for querying their contents to retrieve syntactic phenomena, as well as typical approaches to their construction. Section 2.6 provides an overview of grammar engineering—the practice of developing precision grammars—and unpacks some of the aspects of their development, including how they are used to create high quality treebanks containing analyses. Section 2.7 describes the emerging field of electronic grammaticography, identifying, in particular, the enhancements that precision grammars have to offer electronic descriptive grammars. Section 2.8 presents a selection of works from the computational linguistics literature that involve the operationalisation of the abstract notion of grammatical phenomena. Lastly, Section 2.9 describes a range of approaches that are used for performing syntactic phenomenon discovery, along with some prototypical works illustrating their application. These approaches include querying pre-annotated databases of phenomena, traditional treebank query tools, and tools that adopt a query-by-example methodology.

Chapter 3 describes the resources that were used to support this research. Section 3.2 provides an overview of the ERG, the precision grammar that was used to develop the syntactic phenomenon discovery methods that are described in Chapter 5. Section 3.3 then outlines DeepBank, a dynamically annotated treebank containing text from the WSJ section of the Penn Treebank that is paired with disambiguated syntactic and semantic analyses from the ERG. These analyses were combined with the phenomenon annotations from the constructed phenomenon corpus that is described in Chapter 4, enabling the exploration of syntactic phenomenon discovery methods against the best available analyses produced by the ERG. Section 3.4 then describes The Cambridge Grammar of the English Language, the descriptive grammar that was used to source characterisations of syntactic phenomena and accompanying exemplars, which could be used as input into the implemented syntactic phenomenon discovery tool described in Chapter 5.

Chapter 4 describes the construction of a corpus that is annotated with syntactic phenomena and which I created to support the development of syntactic phenomenon discovery methodologies. Section 4.2 establishes the necessary properties this corpus should have in order to support its raison d’être. Section 4.3 runs through candidate resources that already exist, concluding that none meet all of the established requirements. Section 4.4 presents the syntactic phenomena selected for annotation, outlining their characteristic properties. Section 4.5 describes the two sources from which text was selected to constitute the corpus, and then Section 4.6 describes the methodology that was used to collect annotations of this text, as well as the process used to convert these into the final corpus. The last section of this chapter, Section 4.7, involves an in-depth investigation into suitable approaches to assessing
the reliability of the collected annotations, which differ from annotations typically performed within computational linguistics, arriving at a suggested methodology for annotation tasks mirroring this one.

Chapter 5 presents the results of my research into the development of techniques for syntactic phenomenon discovery with precision grammars, which involves the development of TypediFF, an interactive methodology for navigating the alignment between syntactic phenomena and precision grammar components associated with their analyses. This begins with Section 5.2 outlining the general characteristics the methodology takes, which were established on the basis of the requirements of the motivating contexts for this research, combined with observations from some of the background works described in Chapter 2. In order to properly unpack the discovery methodology, Section 5.3 then presents a brief overview of the architecture of delph-in grammars, focusing on aspects relevant to the methodology. The TypediFF discovery methodology itself is then presented in Section 5.4, describing how grammar components of analysed sentences are extracted, labelled, weighted, and exposed for manipulation by users, in order to improve their discoverability. Section 5.5 then describes the implementation of the TypediFF methodology that I developed in the course of this research. Section 5.6 presents a small exploration into the application of the TypediFF methodology against the Phenomenal Corpus from Chapter 4. Section 5.7 then presents an exploration that applies the TypediFF methodology to a small corpus of relative constructions, which is composed of exemplars extracted from Chapter 12 of Huddleston and Pullum (2002) and labelled with more fine-grained syntactic phenomenon categories than those found in the Phenomenal Corpus. Finally, a discussion of the TypediFF methodology is given in Section 5.8, covering some potential limitations of the methodology and its potential for application to other precision grammar frameworks.

Chapter 6 concludes the thesis by firstly presenting a summary of this research’s two areas of contribution in Section 6.1 and Section 6.2. This is followed by Section 6.3, in which I identify and discusses aspects of this research which could be explored in future work. I then offer some concluding observations in Section 6.4, in which I draw out the wider implications of this research.
Chapter 2

Background

2.1 Introduction

In this thesis, I investigate strategies for improving the discoverability of syntactic phenomena within precision grammars and their accompanying treebanks. This investigation is motivated by the need for techniques enabling both resources to be more readily used to enhance electronic descriptive grammars. The two enhancements being targeted in particular are the dynamic retrieval of additional examples of syntactic phenomena from within treebanks and the navigation of descriptive grammars by syntactic phenomena. The need for improved discoverability within precision grammars lies in the twofold difficulties of 1) identifying how syntactic phenomena are represented within the grammar and 2) the need to navigate the alignment of two different resources which potentially contain divergent analyses of the same syntactic phenomena of the language.

In order to effectively document the research that this thesis describes, this rather dense problem statement first needs to be unpacked. In the remainder of this chapter, I proceed to do this firstly by situating the problem within its relevant academic traditions and outlining necessary concepts from these fields, and then by more fully illuminating the motivations for this research—which was already briefly outlined in Chapter 1—in addition to characterising the core challenges involved in tackling the problems my research aims to address. During this process, I also identify and describe relevant research from the literature that my research is informed by and builds on.

The research in this thesis largely falls into the area of computational linguistics, but also draws heavily on various other subfields of linguistics. Since I only assume that the reader has a background in computer science, I dedicate the first parts of this chapter to outlining the core concepts which this research is premised upon, and which some familiarity with will be assumed. Section 2.2 begins this process by outlining the salient research goals of computational linguistics and of the relevant subfields of linguistics. In Section 2.3, I define the notion of a grammatical phenomenon, a
catchall term that is often used in linguistics to describe an arbitrary or underspecified feature of language in the context of grammatical analysis. The characterisation of this term is necessary partly for terminological clarity, but also because strategies for the discovery of grammatical phenomena will need to be informed by how people describe them. Section 2.4 discusses descriptive grammars—reference publications produced by linguists describing the workings of particular languages. I outline challenges in the development and construction of these documents, focusing in particular upon limitations of descriptive grammars which precision grammar-based syntactic phenomenon discovery methods could improve. Section 2.5 describes another type of linguistic resource—the treebank—a corpus of text that has been annotated with some form of linguistic information, either by hand, semi-automatically, or completely automatically. The value of treebanks to this research is not just that they are stores of concrete linguistic analyses of actual language usage, but also that there is a large amount of existing work developing methods and corresponding tools for querying the contents of treebanks based on linguistically-grounded constraints. The treebanks that are particularly relevant for this research are those containing syntactic analyses produced by precision grammars. In Section 2.6, I then describe the practice of grammar engineering, the craft of implementing and maintaining precision grammars—rich linguistically-grounded computational models of language that are characterised by being capable of providing linguistically accurate syntactic (and often semantic) analyses for a for given language, in some cases, over a wide range of grammatical phenomena. These properties, along with their ability to produce high quality treebanks, which I also outline, makes them particularly well suited for the task of syntactic phenomenon discovery. The precision grammar used in this thesis is the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011) and is discussed in Section 3.2.

Having outlined the required background concepts from the fields of linguistics and computational linguistics, Section 2.7 turns to look at electronic grammaticography, the emerging practice of developing descriptive grammars that embrace natively digital formats and provide features and functionality not possible within the confines of the linear document of a traditional descriptive grammar. I identify key enhancements that embracing natively digital formats can have for the development of descriptive linguistic resources, focusing in particular on those which overcome limitations of traditional descriptive grammars discussed in Section 2.4 and which the use of precision grammar-based syntactic discovery techniques can facilitate.

In the next two sections, I identify research from the literature that either contributes towards the framing of the research in this thesis or which this work directly builds upon. Section 2.8 presents a range of projects which either explicitly or implicitly invoke the notion of grammatical phenomena, involving the operationalisation of different types of grammatical phenomena within various computational frameworks for performing grammatical analysis. Section 2.9 identifies three overarching kinds of methodologies for performing syntactic phenomenon discovery that have been de-
scribed in the literature. For each methodology, I identify and outline representative examples and also assess its applicability within the context of providing the targeted enhancements for descriptive grammars. Finally, in Section 2.10, in addition to briefly summarising the contents of this chapter, I return to the goals of this research, restating them in the context of their departure points from existing work in the literature.

2.2 Setting the Scene

In this section, I describe the academic landscape this work is situated within, which notably includes the field of computational linguistics as well as three other major subfields of linguistics. This then sets the scene for a discussion of the notion of a grammatical phenomenon, a concept which is central to the work in this thesis.

2.2.1 Computational Linguistics

This thesis lies largely within the field of computational linguistics, the field of study which concerns itself with developing and implementing computational models of human languages, along with algorithms and software for evaluating these models over real-world language use. The goals of this field could be described as developing computational methods for answering scientific questions regarding the nature of human language—which is to say, questions that are posed by the field of linguistics. Computational linguistics bears many similarities to another field, natural language processing (NLP), which also involves the application of computational methods to human language, however, is less concerned with evaluating linguistic hypotheses and more with finding practical solutions to problems requiring the automatic analysis or generation of natural language, such as automated information extraction and summarisation of texts.\footnote{For a good overview of the field of NLP, see Jurafsky and Martin (2009).} There is a strong overlap between the two fields of practice, with shared methodologies, data, and communities. While opinions on the subject matter of the two fields inevitably differ from one practitioner to the next, there is perhaps a general consensus that the characteristic difference between the two fields involves the object of study—language itself in the case of computational linguistics, and practical ways of extracting information from language use in the case of NLP.\footnote{See, for example, Martin Kay’s ACL Lifetime Achievement Award speech (Kay 2005), where he argues for this distinction.} Since this thesis involves the use of deep, linguistically grounded models of language being put to the task of enriching descriptive linguistic resources, it lies clearly on the computational linguistics side. For simplicity, I will then lump NLP in with computational linguistics, using the latter as the more general term, with the understanding
that this does not represent an implicit claim regarding the status of either field.\footnote{The term human language technology is also sometimes used as a superordinate term for both fields.}

Within the various approaches to computational modelling of human language, a distinction is often made between symbolic and statistical models. While this distinction does not represent a strict dichotomy, the two approaches can be broadly characterised as pertaining to models which explicitly encode hand-crafted linguistic constraints that are informed by linguistic theory on the one hand, and statistical models on the other hand, in which probabilistic constraints are learnt through the application of machine learning techniques over large amounts of data. The focus on the use of hand-crafted computational grammars of language puts the work in this thesis predominantly within the symbolic approach to language modelling.\footnote{For a discussion of the history of the symbolic-statistical divide within the field of computational linguistics see Church (2011).} This approach to language modelling is outlined in more detail in Section 2.6, which describes the practice of grammar engineering. In the next section I outline some of the core methodological approaches within the field of linguistics that have informed the development of both computational and descriptive grammars.

### 2.2.2 Describing Language

There are numerous subfields within the field of linguistics, each with its own characteristic questions pertaining to the phenomenon of language and characteristic methodologies for discovering answers to these questions. In this research, of particular interest are those areas of linguistic study that aim to understand and describe the grammar of individual languages—the system of rules internalised by speakers of a language that determine the acceptable utterances of each language. In other words, in each case, what is being sought is a grammatical theory of a language. There are a variety of approaches within the linguistic tradition of tackling this problem, and in this section I outline those salient to the research in this thesis. It is first worth indicating some broad dimensions by which investigation into the structure of language has traditionally proceeded, and in doing so, immediately exclude some as not being relevant to the present investigation.

First and foremost, it should be noted that, while the term grammar is often associated with prescriptive notions of how one ought to use language, modern linguistics is instead concerned with impartial descriptions of how people actually use language (Wasow 2001). Whether an utterance is grammatical or not within a language is an empirical matter, to be determined through a combination of grammaticality judgements by native speakers and attested instances of similar constructions within corpora of real-world language use (Schütze 1996).

Secondly, within the field of linguistics, approaches to the study of language are often characterised as being either synchronic or diachronic, with the former pertaining...
ing to the analysis of a language at a specific point in time, and the latter approaches, often characterised as falling within the domain of historical linguistics, pertaining to the analysis of how language—or specific languages—change over time.\(^5\) The kind of linguistic description relevant to this work is synchronic.

Another relevant dimension is whether a theory of grammar purports to be only a descriptive theory, accounting for the facts of language, or whether the theory is also explanatory, accounting for why the facts of language are the way they are. While much work in theoretical linguistics has focused on developing explanatory theories of grammar, a grammatical theory of a language is not necessarily deficient if it does not address questions of explanation. As argued by Bender (2008b), accounting for the descriptive facts of a language is antecedent to arriving at a parsimonious explanatory theory for their existence. Furthermore, explanatory theories of languages are often functional in nature, invoking forces whose effects can be seen diachronically, such as cognitive processing constraints of the human mind and the recurrent communicative needs that arise in the context of social interaction. Such forces and accompanying theories can be expressed independently of the grammatical descriptions of a language at a given time, and it has proven to be methodologically fruitful to treat the grammar of a language as an object of study which can be described and studied independently of the forces that led to the grammar having the particular shape that it does (Dryer 2006). It is largely under this paradigm that the linguistic inquiry invoked in this thesis operates.

The methods used by linguists to understand and analyse linguistic systems are often characterised as being either descriptive or formal in nature, corresponding to the characteristic methods used in the subfields of descriptive linguistics and formal linguistics. A third subfield of linguistics, linguistic typology, focuses on the comparative task of categorising the different languages of the world according to their structural properties. Each of these distinct, yet complementary, approaches to analysing human language are relevant to the work in this thesis and in the remainder of this section I briefly outline them.

**Descriptive Linguistics**

Descriptive linguistics is concerned with carefully and accurately describing the nature of the individual languages of the world. Himmelmann (1998) identifies three salient characteristics that distinguish the practice from other approaches to the study of language. Firstly, it is synchronic, contrasting with historical or comparative linguistics; secondly, it is impartial rather than prescriptive; and lastly, the yield of the endeavour can be thought of as an informal statement of the facts of a language, contrasting it with formal and explanatory elements of other approaches. While not formalised, descriptive linguistics still constitutes an analytic exercise: in addition to

\(^5\)The distinction between synchronic and diachronic approaches goes back to Saussure’s *Course in General Linguistics* (1986), first published in 1931.
simply accounting for the available data of a language, the presentation of the facts of a language must also form an internally coherent theory of the language. Descriptive linguistics is also increasingly seen as a distinct exercise from documentary linguistics (Himmelmann 1998), which is concerned with the collection of naturalistic linguistic data, often performed by a field linguist visiting a region where the language. Descriptive linguistics is, instead, an analytic exercise premised on the existence of such data.

The canonical artefacts produced by a descriptive linguist when describing a language are a dictionary, texts, and a descriptive grammar—sometimes referred to as the Boasian trilogy after the notable American anthropologist, Franz Boas (Evans and Dench 2006). The dictionary, or lexicon, contains an inventory of the documented words of the language. The collection of texts serve to show the language in use, as well as record the data upon which the analysis of the language was based. The descriptive grammar, which I describe in more detail in Section 2.4, describes the workings of the language: the structures—or forms—of the language, how these forms combine together to form grammatical utterances, and how these forms contribute to the meaning of an utterance. This analysis is generally broken down into different levels of description which mirror divisions within the field of linguistics: phonology—the analysis of the sounds of a language; morphology—the analysis of how words of a language are formed, and syntax—the analysis of how words and phrases combine together to form grammatical sentences of a languages. Semantics, the analysis of how meaning arises out of the forms of a language also plays a crucial role, but often in a way that cuts across these levels of analysis.

Formal Linguistics

The characteristic activity of formal linguistics is the analysis of language using precisely defined formal models which have their roots in formal logic and formal language theory. Of particular interest to a formal linguist is the question of what properties this abstract machinery should possess in order to capture the structure of language (Evans and Dench 2006). Once a candidate formalism is developed, it can be used to develop a model of a particular language—an internally coherent collection of analyses of different phenomena within the language, which together represents a theory regarding the constraints of the language, or at least some portion of it. Often such models target a specific structural level from those identified in the previous section (Bender 2008b)—phonological, morphological, syntactic or semantic—with much work in particular being dedicated to the areas of formal syntax and formal semantics.

The development of a formal model involves both the construction of a formalism to express the model and an effective procedure, or algorithm, for determining whether a given model accepts some input as valid or not. The existence of these verification procedures means that theories expressed in terms of formal models are
readily falsifiable—problems in the theory can be highlighted through the discovery that some input results in a linguistic absurdity. A second methodological advantage of the use of verification procedures is the ability to evaluate the predictive power of a theory—its ability to generalise to unseen data by analysing interactions between phenomena that it was not explicitly designed to support (Chomsky 1957). These two properties—the ability to readily falsify and also evaluate the predictive power of linguistic theories—have resulted in developments from formal linguistics playing an important role in improving the analyses arrived at in descriptive linguistics (Rice 2006).

Theories of grammar developed in the formal tradition have tended to prioritise explanatory adequacy and descriptive adequacy to different extents. Grammatical theories developed within the Chomskyan transformational paradigm have tended to prioritise explanatory adequacy, with an explicit goal of being able to reveal properties of a hypothesised innate linguistic capacity. Other theories of grammar, such as Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994)—the grammatical framework underpinning the computational grammars used in this thesis—Lexical Functional Grammar (LFG; Bresnan and Kaplan 1982) and Combinatory Categorial Grammar (CCG; Steedman 2000) have instead prioritised descriptive accuracy, with their formalisms being refined to capture a wide range of grammatical phenomena with high fidelity. It is these latter approaches that are of more direct relevance to this work.

As our understanding of the world’s languages has increased, so too has the complexity of many of the formal models being developed. This is especially true of formal grammars that model the syntax and semantics of a language and that set their sights on providing analyses for a large proportion of the grammatical sentences in a language. These models face the problem of not only having to ensure accuracy across an ever expanding range of grammatical phenomena, but also having to ensure internal consistency across different sub-modules of analysis, a challenge made all the more difficult by the highly interrelated nature of language, with distinct grammatical phenomena often interacting with each other in various and sometimes surprising ways (Bender et al. 2011a). The use of computational approaches for implementing these models and developing automatic verification procedures in the form of parsers represents an opportunity to manage this complexity and also opens up data-driven approaches to evaluating linguistic hypotheses over large-scale corpora that would be impossible with manual approaches (Bender et al. 2011a). I discuss grammar engineering—the practice of building wide-coverage implemented grammars—in Section 2.6 and some of the specifics of the particular computational grammars used in this thesis are described in Section 5.3.

Formal linguistics and descriptive linguistics both seek to analyse the structure of languages but using different methodologies: one formal and the other informal. These approaches are complementary, offering different trade-offs. Formal approaches yield theories which are more readily falsifiable, offer superior predictive power, and
can be implemented to perform data-driven experimental linguistics (Abney 2011). However, their inherent complexity means they often only target single levels of linguistic abstraction, and the content of their analyses are expressed in formalisms that may appear arcane to the uninitiated and run the risk of becoming irrelevant as developments and trends in formal approaches move in different directions. Informal descriptions using natural language, on the other hand, are accessible to a wider audience and have proven to remain in currency for longer periods of time (Evans and Dench 2006). In the context of extending a non-trivial theory of a language to include an analysis of a new phenomenon, the benefits of formalisation mean that the inevitable introduction of undesirable side-effects upon the analyses of other grammatical phenomena are more likely to be detected within a formal model than an informal model, where they may go unnoticed. The merits of descriptive analysis over formal analysis are discussed further in Section 2.4.3.

Linguistic Typology

Linguistic typology is the subfield of linguistics which concerns itself with the comparative analysis of the world’s languages. Croft (2002) identifies three distinct aspects of typological analysis. The first aims to classify the languages of the world according to their structural properties as well as by strategies used to encode meaning. Secondly, typologists seek to explore the extent to which languages of the world vary structurally, and whether there are commonalities that generalise across all languages—so called language universals. Lastly, typologists also seek to develop theories of language using functional explanations that are consistent with the results of typological classification and typological generalisation. While these modes of analysis are not independent, it the outcome of the first two modes that are of more relevance to the work in this research.

Typological analysis presupposes a conceptual framework that permits classification and comparison along various dimensions of linguistic description that can account for the linguistic diversity that is seen across the languages of the world. In order to handle the large amounts of cross-linguistic variation, the nature of this framework is typically approached as being informal. Within the methodological approach of formal linguistics, there lies an implicit assumption that the different languages of the world must possess sufficient uniformity in the way they are assembled from their component pieces in order for the project of creating a formalism with cross-linguistic validity to be a sensible one. From a typologist’s perspective, however, such an assumption would represent a methodological straight-jacket that would run the risk of obscuring as of yet unnoticed differences (Polinsky 2010).
2.2.3 Corpus Linguistics

Corpus linguistics is an area of linguistics characterised by methodologies that employ large-scale corpora in order to study linguistic phenomena (McEnery and Hardie 2011). The supporting corpora are frequently composed of textual documents—and to match the scope of this research, I will limit my discussion to these kinds—however they can also be composed of other media such as video and audio. The value of corpus linguistics lies in the scale of the target corpora. The use of computational tools to process machine-readable documents enables the consultation of corpora that are of a size that would not be feasible to analyse manually. This use of large-scale datasets of naturally occurring language use facilitates the discovery of facts about languages, which, due to the limitations of human imagination and the relative infrequency of a large number of linguistic phenomena, could remain hidden from analyses driven by introspection and ad-hoc sampling of larger corpora (Fillmore 1992). This application of linguistic corpora is premised on the existence of computational tools that allow practitioners of corpus linguistics to search for target phenomena across large-scale datasets and collect statistics of their occurrence. Of particular importance when using corpus data to attempt to answer a specific research question is that the chosen dataset be an appropriate match for the linguistic phenomenon being investigated. For instance, for a given target phenomenon, the corpus being consulted must contain text from a domain and language register that has a propensity to feature that phenomenon at frequencies that allow a sufficient amount of data to be collected.

A further property of a textual corpus that can impact its suitability for supporting the investigation of a linguistic phenomenon is whether its text is accompanied by linguistic annotations and additionally, the specific nature of these annotations. Corpus linguistics tools can operate over raw text, returning frequencies and concordance results for simple character-based queries over orthographic forms. While this can be effective, the range of phenomena that can be queried for in this way is limited and it is therefore common to enrich corpora with structured linguistic annotations such as part of speech tags, lemmas, and morphological forms, all of which can be queried through the use of appropriate tools. Due to the increased complexity of the linguistic phenomenon that forms the focus of this research—syntactic phenomena—these types of linguistic annotations are often insufficient for capturing such phenomena. It is here that the more syntactically oriented annotations found in treebanks, along with tools explicitly designed to query these annotations, can be valuable. I discuss annotated treebanks and their accompanying querying tools in Section 2.5. The use of syntactically annotated treebanks for performing corpus linguistics style data-driven exploration and hypothesis-testing introduces its own challenges however. Of relevance to this research is the lack of discoverability with respect to how specific syntactic phenomena are represented within a treebank, combined with the requirement of having to become familiar with complex treebank querying languages. I discuss these issues further in Section 2.9.2.
2.2.4 Situating this Research

Evans and Dench (2006) describe the triadic and complementary relationship between the three subfields of descriptive linguistics, formal linguistics, and linguistic typology. Both typologists and practitioners of formal approaches require preexisting descriptions of languages to drive their research. On the flip-side, descriptive linguists must ensure that they pay attention to developments within linguistic typology to ensure that the linguistic concepts and terminology they invoke remain relevant to the linguistics community for as long as possible. Typological investigation also continually poses new questions and challenges for descriptive linguists and formal linguists to investigate, as new languages and linguistic phenomena are made subject to scrutiny (Cristofaro 2006, Dryer 2006). From a historical perspective, focused investigations into specific grammatical phenomena within formal linguistics have also played an important role in improving the analyses arrived at in descriptive linguistics (Rice 2006). Similarly, the focus on specific constructions such as relative clauses and passive clauses found in formal linguistics led to early work in typology involving fruitful in-depth cross-linguistic investigations of these constructions, which attempted to define cross-linguistic criteria for the identification of these phenomena (Dryer 2006).

The research in this thesis taps into this synergistic relationship between these three fields. Firstly, it involves the use of implemented computational grammars of languages (precision grammars), which are grounded in formal linguistic theory and contain analyses that are invariably informed by descriptive analyses of their respective languages. The reverse relationship, however, of already-established implemented grammars being used to enhance descriptive grammars, is one that has received much less attention, and forms a focal point of this thesis. In particular, I explore how precision grammars—discussed in Section 2.6—with their capacity to automatically generate high-fidelity syntactic and semantic analyses of language, can be exploited to enrich the utility of the linguistic resource that is the descriptive grammar. The subfield of linguistic typology is also implicated by this research, in part through its relevance to the challenge of organising descriptive grammars, but also because it involves tackling the challenge of capturing—and making use of—the notion of grammatical phenomena, an inherently cross-linguistic concept, which I establish a working definition of in the following section.

Finally, the relevance of corpus linguistics to this research is found in the challenges associated with data-driven exploration of syntactic phenomena. Queries defined in terms of simple linguistic annotations, such as part of speech tags and morphological forms—which characterise the annotations of many corpora—struggle to capture a large range of syntactic phenomena. On the other hand, the richer, syntactic annotations of treebanked corpora can be difficult to query due to accessibility issues, with practitioners having to know how a syntactic construction is represented within a treebank before an appropriate query can be constructed, in addition to the practical constraint of having to become familiar with the potentially complex query language
associated with the target treebank. The discovery techniques developed in this research aim to make the syntactic phenomena within treebanks produced by precision grammars more accessible, which has the potential to make syntactic phenomena more readily amenable to analysis using the empirically grounded and quantitative methods that are associated with corpus linguistics.

2.3 Grammatical Phenomena

The term *grammatical phenomenon* evokes a somewhat nebulous concept, often used in the context of discussing linguistic analyses in order to refer to a non-specific feature of language.\(^6\) Potentially referring to a broad and heterogeneous collection of linguistic concepts, it is a tricky term to pin down. Indeed, its utility lies in its underspecificity—that it can be used as a catch-all for an arbitrary grammatical feature of language. As can be seen in this thesis, it is common for this term to be used in meta-discussions of linguistic analysis, where it facilitates the discussion of strategies for performing or presenting analyses abstracted away particular linguistic features (e.g. Evans and Dench 2006, Mosel 2006, Cristofaro 2006). In such discussions, the term tends not to be defined explicitly, since the audience is assumed to be familiar with the typical objects of linguistic analysis, further contributing to the vagueness of the term. I will now explicitly define the term as it will be used in this thesis, which will hopefully be consistent with most usage of the term in the literature.

I will take the term *grammatical phenomenon* to refer to a feature of language—from either a specific language or potentially found across multiple languages—that can be expressed in terms of characteristic structural properties or in terms of characteristic contributions to the meaning of a sentence.\(^7\) This broad definition invokes a distinction that is commonly made in grammatical analysis: the difference between the *forms* of a language—the structural building blocks of the language—and the *function* of these forms. Typically, an analysis of the grammar of a language—whether formal or informal—involves first identifying the basic forms of a language, before identifying the constraints on how these forms combine with other forms to create complex, well-formed structures and ultimately complete grammatical sentences of the language. Since a grammatical analysis would be incomplete without address-
ing how sentences of a language have the specific meanings that they do, the analysis will also involve identifying how the different structures of a language contribute to the meaning of grammatical sentences.

More precisely then, I will define a grammatical phenomenon to be a feature of language that can be described in terms of one or more linguistic forms, constraints over these forms, one or more linguistic functions, or some combination of these. The distinction between form and function plays an important role in the characterisation of grammatical phenomena, but before exploring this further, I will first briefly outline the two individual concepts.

2.3.1 Linguistic Forms

When performing a grammatical analysis of a language, the identification of the forms of the language is a crucial process. This analysis proceeds on the working assumption that within all languages, forms are built up compositionally out of other forms, starting with smaller forms and working up, ultimately yielding complete sentences. Rather than assuming this process operates over a uniform collection of form types, linguistic forms are traditionally divided up into different levels of linguistic abstraction. Broadly speaking, these levels are sounds and gestures, morphemes, words, and phrases, corresponding approximately to the different linguistic fields of phonology, morphology and syntax. In this section I briefly describe the nature of these forms, focusing in particular on those more relevant to the research in this thesis, describing the kind of analysis involved in identifying syntactic—and to some extent—morphological forms.

The study of phonology involves the analysis of the sound system of a language. This involves the identification of the sounds, or phones, that make up the language’s phonetic inventory. From a grammatical perspective, the forms that most phonological frameworks are concerned with identifying are distinct groupings of sounds referred to as phonemes (Major 2012). Since I do not investigate phonological phenomena in this work, I will not describe the process of phonological analysis here. For an overview of the principles involved in phonological analysis, see Major (2012).

Moving slightly higher in the level of linguistic abstraction, morphology involves the study of how the words of a language are formed. This involves the identification and analysis of the morphemes of a language, which can be approximately described as the smallest linguistic forms that bear meaning or grammatical function (Bauer 2003). A morphological analysis of the word misinformation, for example, might see it as being broken up into three morphemes consisting of the root inform, the prefix mis- and the suffix -ation.9

---

8Phonological analysis can also be performed on gestures within a signed language (Sandler 2012).
9For simplicity, I use orthographic forms, whereas a proper morphological analysis of a language should generally be performed on a phonological basis.
Table 2.1: The paradigm for English lexical verbs, from Huddleston and Pullum (2002:1596).

<table>
<thead>
<tr>
<th>Plain Form</th>
<th>Plain Present</th>
<th>Preterite</th>
<th>Past Participle</th>
<th>3rd sg Present</th>
<th>Gerund-participle</th>
</tr>
</thead>
<tbody>
<tr>
<td>take</td>
<td>took</td>
<td>taken</td>
<td>takes</td>
<td>taking</td>
<td></td>
</tr>
<tr>
<td>love</td>
<td>loved</td>
<td>loves</td>
<td>loves</td>
<td>loving</td>
<td></td>
</tr>
<tr>
<td>cut</td>
<td>cuts</td>
<td>cutting</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In addition to forming new words with distinct meanings, morphological affixes can also yield grammatical functions, such as number, tense, and case, which contribute important information pertaining to the meaning of the sentence, and play a critical role in the syntax of a language. For example, in the sentence *The cats take everything*, the noun *cat* is marked as being plural due to the plural morpheme *-s*. This also means that the verb is found in the plain present form rather than the third person singular form, since English has subject-verb agreement, which is typically modelled as a syntactic constraint. The co-implication of morphology and syntax means that the analysis of both levels of form often needs to be pursued simultaneously, with the interface between the two commonly being referred to as *morphosyntax*.

The systematic variation seen across the word forms of a lexeme is often referred to as an *inflectional paradigm* or just *paradigm* and the identification of the verbal and nominal paradigms of a language are a notable component of the morphological analysis of a language. Table 2.1 shows the three paradigms that English lexical verbs can take, with *love* illustrating the paradigm for regular verbs. Each column in the table indicates the particular grammatical function that the form encodes. A single form can realise multiple grammatical functions—such as in the case of *loved* being the form for both the preterite and the past participle. This is referred to as *syncretism* and can result in syntactic ambiguity.

Where the study of morphology involves determining how the words of a language are formed, the study of syntax, broadly speaking, involves determining how words combine with other words to form grammatical sentences (Carnie 2013). This involves the identification of the system of rules which constrain word sequence combination within a language. While these rules must ultimately account for the linear order in which words can appear in a sentence, this requires the illumination of sentential structure beyond linear constraints. A part of the process of discovering these structural constraints is first working out which sequences of words can combine together to form *constituents*, groupings of words that function grammatically as a single unit.

---

10Pertaining to nouns or noun phrases.
11This analysis is from *The Cambridge Grammar of the English Language* (Huddleston and Pullum 2002), a descriptive grammar of English that I use in this research, and which is discussed further in Section 3.4.
In English, one of the tests employed to determine constituency is whether the candidate sequence of words can be topicalised, which involves moving it to the front of the sentence, making it the sentence topic (Culicover 2009:84). (1) illustrates the use of this test, with (1b) showing that the noun phrase fried eggs on toast can be topicalised, whereas (1c) fried eggs on toast for cannot,\(^{12}\) suggesting that the former is a constituent and the latter is not.

\begin{enumerate}
\item a. I ate fried eggs on toast for breakfast.
\item b. Fried eggs on toast, I ate for breakfast.
\item c. *Fried eggs on toast for, I ate breakfast.
\end{enumerate}

According to many analyses (e.g. Pollard and Sag 1994), phrases are constituents that are headed by a particular part of speech, which determines both the internal structure of the phrase and also how it combines with other constituents in the sentence. For example, the constituent fried eggs on toast from (1) is a noun phrase headed by the noun eggs. The internal structure of this noun phrase is that the head noun is modified by the adjective fried as well as by the prepositional phrase on toast, which itself is headed by the preposition of. This can be seen in the tree diagram found in Figure 2.1, which contains a complete phrase-structure based analysis for the sentence found in (1a). In terms of combining with other constituents in the sentence, a characteristic property of noun phrases is that they frequently serve as the arguments of predicates, as can be seen in Figure 2.1, where fried eggs on toast is the complement of the verb ate. The combination of this noun phrase with the verb means that the verb is no longer looking for the optional single complement that it can take, resulting in a complete verb phrase. The combination of particular constituents in a characteristic way is often referred to as a construction. Individual constructions are often known by one or more names by members of the linguistics community. In this case this construction is the grammatical phenomenon that might be referred to as a transitive verb phrase.

The description of the syntax of a language can be performed informally using prose—as I have just done—which is typical of research in descriptive linguistics, or formally, as in formal syntax, where the syntactic rules of the language are captured precisely using well-defined formal models and the grammaticality of a sentence is defined as whether the formal grammar that represents a theory of the language accepts it. In both cases, the rules draw upon similar underlying linguistic concepts, however in the case of formal models, these rules must be fully expressed in terms of the adopted formalism, such that a verification procedure could be employed, resulting in the kind of analysis seen in Figure 2.1. This analysis is based on the underlying concept of a phrase structure grammar, which is characterised by the use of nodes that correspond to constituents (Jurafsky and Martin 2009:387). An alternative type of grammar, dependency grammar, takes the form of a graph where

\(^{12}\)It is a convention within the field of linguistics to mark ungrammatical data with an initial *.
Figure 2.1: A possible phrase structure analysis for the sentence *I ate fried eggs on toast for breakfast.*
every node corresponds to exactly one word (Jurafsky and Martin 2009:414). It is used somewhat less frequently in linguistic theory but has proven popular in the context of natural language processing due to head-dependant relationships being able to be used as approximations of predicate-argument relationships meaning that no additional argument-mapping step is required (Covington 2001), in addition to the existence of a range of efficient parsing algorithms.\textsuperscript{13} The formal grammatical framework used in this thesis, Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag 1994), is a type of phrase structure grammar and is outlined briefly in Section 5.3.

A grammatical analysis that aspires to account for the meaning of different sentences must also identify the contribution that different constituents have, identifying core elements such as that which provides the main predicate, as well as the constituents that contain the entities forming arguments to the predicate. I discuss this further in the next section.

2.3.2 The Function of Forms

Having addressed the types of linguistic forms that are implicated within grammatical analysis, I now turn to the function of forms. Within this context, function is used in at least two distinct ways. One sense of the term refers directly to the semantic contribution—or communicative function—of the form within an utterance (Kohler 2017). It could be said, for example, that the function of a comparative construction, such as in (2) is to compare one referent in the sentence to another referent. The function of an imperative clause, such as that found in (3) is to communicate an instruction or plea the audience, and the function of a relative clause, such as found in the underlined component of (4) is to either restrict or elaborate upon a referent in the sentence—in this case the seedling.

(2) This coffee is fruitier than the previous one.
(3) Eat the food!
(4) The seedling that I planted has flowered.

The second way in which the function of forms might be discussed, pertains to the morphosyntactic properties of the arguments within a clause, which identify constituents as taking roles such as subject, object, and adjunct. These are often referred to as grammatical functions or grammatical relations (Bickel 2010). Languages use a range of strategies to encode grammatical relations, such as syntactic form through word order and case marking, through different lexical or morphological forms. In English, grammatical relations are determined predominantly via word order, with the standard order of constituents being subject, verb, then object. A traditional analysis of the sentence found in (5), for example, might see the noun phrase the chinchilla as having the grammatical function of subject, and the noun phrase the

\textsuperscript{13}See Choi et al. (2015) for a comparative analysis of dependency parsing algorithms.
meal as having the grammatical function of direct object, each of which being an argument of the main verb. The constituent in record time is not an argument of the verb in this case, and is instead considered an adjunct, as it only modifies the predicate contributed by the main verb, rather than filling a mandatory slot.

(5) The chinchilla ate the meal in record time.

As it stands, this simple framework for analysing the grammatical function of forms is insufficient for determining the underlying predicate-argument structure of a sentence. While it is often the case that the referent in the subject position corresponds to the agent—the entity causing the event to occur—as it is in (5), this is not necessarily so. Both (6) and (7) illustrate grammatical phenomena in which the grammatical subject expresses the semantic patient of the underlying predicate—the entity experiencing the event. In (6), this is caused by the use of the passive construction, for which this feature is characteristic and which is also discussed further in Section 4.4.3. In the case of (7) this property is inherent to the verb bake, which, for this reason is referred to as an unaccusative verb (Levin 1993). It is therefore common for grammatical frameworks to invoke the notion of thematic roles—such as agent and patient—as being distinct from grammatical functions. Grammatical functions can then be thought of as syntactic slots, which the framework must map thematic roles onto based on various constraints applicable to the language, such as word order, morphological marking, lexical selection of the verb, and specific syntactic constructions (Harley 2011).

(6) The cake was consumed.
(7) The cake baked in the oven.

Since the meaning of a sentence ultimately arises out of grammatical structure, these two senses of the word, while having different denotations, are inherently interrelated. A working definition of the function of a form that encompasses these two senses might be what a form is doing in a sentence, both with respect to its grammatical and semantic contribution.

A property of linguistic forms that should also be noted, is that they can realise different functions depending on the environment in which they occur. As discussed in Section 2.3.1, when the English suffix -ed is applied to a verb, it can form the past tense inflected form of the verb, or it can also form a participle form of the verb. Since these forms are homophonous, in order to determine which form is derived, the surrounding environment must be considered. In (8a), the past tense form of the verb is formed, whereas in (8b) and (8c) it forms the past participle, and in (8d), it forms the passive participle. The multifunctional nature of this form

---

14 Sometimes the past participle form of the verb differs from the past tense, as in the case of grow, where the past tense is grew and the past participle is grown.
15 Under some analyses, the participle form is referred to as the past participle even within a passive clause (e.g. Huddleston and Pullum 2002, 1429).
is highlighted by the fact that each of these examples in (8) represents a distinct syntactic construction—the name of which is indicated alongside each example—in which -ed is a characteristic component.

(8)  
   a. I considered the shark. \hspace{1cm} \textit{past tense}
   b. He has considered the shark. \hspace{1cm} \textit{present perfect}
   c. He had considered the shark. \hspace{1cm} \textit{past perfect}
   d. The shark was considered. \hspace{1cm} \textit{passive}

2.3.3 Characterising Grammatical Phenomena

Grammatical phenomena can be characterised in terms of formal properties, functional properties, or a combination of the two. The importance of this distinction can be drawn out by considering the following basic facts about language:

1. The same form within a language can yield different functions in different contexts (Mosel 2006).

2. Languages tend to have multiple strategies for encoding similar functions using their formal inventory (Mosel 2006).

3. Different languages have different formal inventories (Cristofaro 2006).

The first fact can be re-stated by saying that languages involve decoding ambiguity, meaning that the person performing a grammatical analysis of a language must consult the environment a form occurs in to determine its function for a given occurrence, as was seen in the previous section, with the English suffix -ed. This can also be seen with the English suffix -s, which, if it attaches to a noun, has the effect of marking it as plural, or if it attaches to a verb, has the effect of inflecting the verb to take the third-person present tense singular form. Another example is the English verb have, which can be the main verb in a clause, where it contributes a predicate that denotes the sense of possession, as in (9a), or it can take on the role of an auxiliary verb, where it assists in forming a syntactic construction, such as the perfect aspect, as illustrated by (9b). A purely form-based characterisation of grammatical phenomena, then, can often result in a set of multiple distinct functions, for a given phenomenon specification.

(9)  
   a. I have all their albums.
   b. I have seen them play live.

The second of these facts results in language having encoding ambiguity: a speaker of a language wishing to encode a proposition into an utterance will often have to decide—consciously or unconsciously—between multiple forms that realise the same
function. This means that, for a given language, querying its formal inventory with a functional characterisation of a phenomenon will often yield multiple forms or constructions. For example, the grammatical phenomenon of negation might be defined as a construction which involves reversing the denotation of a linguistic expression. In English, this functionally characterised grammatical phenomenon can be realised through both morphological and syntactic means. (10) demonstrates this through two sentences that are paraphrases of each other and each use a different form of negation. In (10a), the negation is realised through an affix, whereas in (10b) it is realised through a word, a distinction that might be described as morphological versus syntactic negation.

(10) a. I found a non-conforming sentence.
   b. I found a sentence that did not conform.

The distinction between syntactic and morphological negation is one simple way of distinguishing types of the phenomenon in English. Huddleston and Pullum (2002) provides a further framework for characterising the different types of negation in English, which constitutes a good example of how the variation within a particular grammatical phenomenon can be characterised in terms of both form and function. The analysis involves four different contrasts—which I briefly outline below using examples from Huddleston and Pullum (2002:787). The first two contrasts involve formal distinctions, and also provide further examples of encoding ambiguity. The verbal vs non-verbal contrast is characterised respectively by whether the negation marker is linked to the verb, as in (11a)—in which case the auxiliary do is required—or is a dependent of the verb, as in (11b). In the case of the analytic vs synthetic contrast, the former involves forms whose syntactic function only ever indicates negation, as in (12a), whereas the latter involves forms that have additional functions, as in (12b). The second two contrasts involve functional distinctions that pertain to the semantic contribution of negation. The clausal vs subclausal contrast is characterised by whether or not the polarity of the containing clause changes. Clausal negation results in a clause with negative polarity, as in (13a), whereas with subclausal negation, the polarity of the containing clause remains positive, as in (13b). In the case of the ordinary vs metalinguistic contrast, metalinguistic negation is characterised not by negating the underlying truth claim of a proposition, as in (14a), but rather by a rejection of how a proposition has been expressed, as in (14b).

(11) a. He doesn’t dine out.  
   b. He never dines out.

(12) a. Not many people liked it.  
   b. Nobody liked it.

(13) a. She didn’t have a large income.
b. She had a not inconsiderable income. 

(14) a. She didn’t have lunch with my old man: he couldn’t make it. 
b. She didn’t have lunch with your old man: she had lunch with your father. 

decoding and encoding: as I have illustrated, these are indicative of purely formal and purely functional characterisations of grammatical phenomena, respectively, and they mirror the distinction between language construction and language comprehension (Mosel 2006). Both of these perspectives are critical when performing grammatical analysis, as linguists must consider the cognitive plausibility of theories with respect to both producers and consumers of the language. These perspectives are also important for the analysis of the language itself as an artefact, with respect to its accessibility. That characterising grammatical phenomena by either form or function results in different groupings of grammatical phenomena, then, has implications for the design of descriptive grammars, which must ultimately present the analyses arrived at by the linguist for the grammatical phenomena of a language in a coherent manner. The role of form-based grammatical phenomenon characterisation and function-based grammatical phenomenon characterisation in identifying different strategies for organising descriptive grammars is discussed further in Section 2.4.4. Since the work in this thesis is ultimately concerned with developing additional means of accessing and discovering analyses of different grammatical phenomena, this distinction is also relevant to the syntactic phenomenon discovery techniques developed in this research. In Section 2.8.3, I discuss how the form-function distinction relates to phenomenon querying techniques.

Finally, the accessibility of grammatical analysis must also be considered from a cross-linguistic perspective. The third fact about language—that different languages have different repertoires of forms—means that it cannot be assumed each language will have the same forms to be compared, and it may even be that cross-linguistic comparison of forms is not a sensible notion. From a typological perspective, primarily form-based comparisons are of limited utility, and functional characterisations of grammatical phenomena are particularly useful, as they do not require the same form to be present across languages, thus more readily facilitating cross-linguistic comparison (Cristofaro 2006, Zaefferer 2006).16 A typologist is therefore likely to pose questions like: how is this specific meaning or functionality realised in this language? A descriptive linguist, on the other hand, is potentially more likely to ask the question: how do the specific forms of a given language contribute to the meaning

16This mode of cross-linguistic comparison relies, to some extent, on the assumption that languages possess the ability to encode—in some fashion—the same functions. For a discussion of this issue, see Haspelmath (2010a).
of a sentence? Again, these considerations bear relevance to both the organisation of descriptive grammars and importantly, for the work in this research, the design of methodologies for grammatical phenomenon discovery.

### 2.3.4 Syntactic Phenomena

I have been describing grammatical phenomena in a general sense in this section, however this research focuses on syntactic phenomena. It should be clear by now that the category of syntactic phenomena is subsumed by the more general concept of grammatical phenomena. Attempting to concretely delimit the boundary between syntactic and other grammatical phenomena cross-linguistically would go beyond the scope of this research. One area that has received considerable attention in the literature is around the morphology-syntax boundary (e.g. Mereu 1999). Teasing apart syntactic and morphological phenomena represents a challenging prospect due to the complex interaction between these two different levels, as discussed in Section 2.3.1, with many phenomena needing to be described in terms of both. A thorough investigation into this problem would require gathering and analysing a large amount of evidence from across typologically diverse languages. However, with the preceding discussions of form and function and their use in characterising grammatical phenomena in place, it is possible to provide some general heuristics that will be sufficient for identifying candidate phenomena for the objects of study in this research.

It might at first appear tempting to treat syntax as pertaining to phenomena described exclusively in terms of linguistic form due to its characteristic concern with words and phrases and how they combine to form clauses and sentences. However, after looking at examples of grammatical phenomena typically considered to be syntactic in nature, it is readily apparent that they frequently invoke function, both in the sense of grammatical function and communicative contribution. The phenomenon of raising to subject, for example, is illustrated by (15a). This phenomenon involves a clause in which the grammatical subject (in this case, the pronoun *she*) functions not as an argument of the predicate it is headed by (in this case, the verb *appear*), but rather as an argument of a predicate in a subordinate clause (in this case, *win*) (Huddleston and Pullum 2002:226). The predicates that license raising to subject vary across languages; in English, other predicates that license raising to subject include *seem, might,* and *happen*. Another illustrative syntactic phenomenon is the *it*-cleft in English, which is shown in (15b). In this construction, the clause is divided into two parts: the foregrounded component—in this case, *yesterday*—and the backgrounded component—in this case, *she won the race*. The pronoun *it* forms the subject of the main clause, which is headed by the verb *be*. The foregrounded component is the complement of *be*, and the backgrounded component is subordinated, appearing as a relative clause. In addition to placing focus on the foregrounded component, the effect of this construction is to mark the information contained in the backgrounded component as being presupposed, which is to say assumed as being known by dis-
course participants, or generally uncontroversial (Huddleston and Pullum 2002:1414). In both of these examples, a description of the syntactic phenomenon would be incomplete without mention of the function of the combination of forms that represents the construction. In the case of raising to subject, the function pertains to the impact the construction has on the mapping of grammatical relations to semantic arguments across the main and subordinate clauses. In the case of it-clefts, the function is expressed in terms of communicative contribution, in particular, by marking the information in the backgrounded component as being presupposed.

(15) a. She appears to be winning the race. \(\text{raising to subject}\)  
b. It was yesterday that she won the race. \(\text{it-cleft}\)

In order to capture the necessary concern with the combination of words and phrases into sentences, the heuristic I will apply for this investigation is that a syntactic phenomenon must, at a minimum, require some form of characterisation by means of constituency or dependency relationships.

This heuristic excludes English grammatical phenomena such as the plural formation (e.g. *cat* and *cats* (Huddleston and Pullum 2002:1585)), and spelling alternations like consonant doubling (e.g. *bat* and *batting* (Huddleston and Pullum 2002:1575)) and *e*-deletion (e.g. *edge* and *edging* (Huddleston and Pullum 2002:1576)), all of which invoke only morphological and phonological description within their analysis. This would also exclude phenomena such as the distinction between distributive and joint coordination in English, which is illustrated by (16) from Huddleston and Pullum (2002:1281). Distributive coordination, the most frequently occurring of the two, is shown in (16a), where the property of *knows Greek* applies to both Pat and Kim independently. In the case of joint coordination, as shown in (16b), where it is not the case that either Kim or Pat are a happy couple independently of each other, the predication applies only to both entities in the coordination together, not to each individually. While there are some restrictions on the form of this construction—such as joint coordination almost always requiring the coordinator *and* (Huddleston and Pullum 2002:1282)—the salient distinction between these two types of coordination only manifests at the level of semantic representation, thus excluding it from consideration in this investigation.

(16) a. Kim and Pat know Greek. \(\text{distributive}\)  
b. Kim and Pat are a happy couple. \(\text{joint}\)

### 2.3.5 Complexity

Grammatical phenomena often involve a high degree of complexity, making the task of mapping out and analysing even a single grammatical phenomenon a potentially challenging endeavour. This is especially true of syntactic constructions, where complexity can manifest in multiple ways. One dimension of complexity is
found within the characteristic forms that comprise a construction, which may be both morphological and syntactic. Another dimension is that a given construction will often include a number of variants, all deemed to fall within the same family of constructions but structurally or functionally distinct from each other. Additionally, constructions inevitably co-occur with other grammatical phenomena, often yielding interactions between the different phenomena, which should not only be accounted for by a comprehensive analysis, but can also serve to confound the analysis.

A grammatical phenomenon which exhibits all of these flavours of complexity is the predicative complement in English. In this section I provide a brief sketch of this construction in order to demonstrate the potential complexity a syntactic phenomenon can exhibit, while also providing a more in-depth example illustrating how form and function can be used to characterise a grammatical phenomenon. The analysis of predicative complements described here is a brief sketch, taken largely from Huddleston and Pullum (2002:251), which provides a thorough overview of the construction, including a range of different variants.

Predicative complements include constructions where a constituent is both a complement and also a predicate, applying a property to a referent within the sentence—namely, the predicand. The syntactic properties of the construction are that the predicative complement itself may either be an adjective phrase, as in (17a), or a bare noun, as in (17b). There are a number of ways in which Huddleston and Pullum divide up variants of the construction. One is structurally—which is to say, by form—noting that it occurs as both intransitive and transitive constructions. In the case of the intransitive variant, the predicand of the predicative complement tends to be the subject, and in the case of the transitive variant, the predicand tends to be the object. This can be seen in (17c) and (17d), which show an intransitive verb and transitive verb respectively, with the predicative complements being underlined and the corresponding predicands being double-underlined. Another structural classification that Huddleston and Pullum employ, is whether the predicative complement is obligatory, as it is in (17e), or whether it is optional, as it is in (17f).

(17)  
|   a. He seemed nice.    |
|   b. She remained treasurer. |
|   c. Kim became angry.   |
|   d. He made Kim happy. |
|   e. Kim became ill.     |
|   f. They departed content. |

Huddleston and Pullum also make a functional distinction, noting that predicative complements can be either depictive, where the predicative complement provides a static property of the predicand, or resultative, where the predicative complement
occurs with a verb that indicates a change of state. For example, (17a), (17b), and (17f) are all depictive and (17c), (17d), and (17e) are all resultative.

In order to convey a sense of the complexity involved, I will briefly describe one further form of the construction Huddleston and Pullum identify. This variant is characterised by the predicative complement being linked to the subject through the verb be—referred to as a copula when used to perform this linking function. As illustrated by (18) (Huddleston and Pullum 2002:266), Huddleston and Pullum divide this construction into two categories, which are differentiated through functional properties. The first category, which is illustrated by (18a), involves an ascriptive function, where the predicative complement represents a property which is ascribed to the referent—typically the subject. The specifying predicative complement, on the other hand, involves the introduction of a referent, which is then specified or identified, as seen in (18b).

(18)  
(a) His daughter is very bright / a highly intelligent woman.  
(b) The chief culprit was Kim.

When a predicative complement is formed using a copula and a deverbal adjective—such as excited or exciting—the clause can be superficially ambiguous between the passive and progressive constructions due to the homophonous nature of some adjectives with past and present verb forms, as illustrated by (19). These can be teased apart using various analytic tests (Huddleston and Pullum 2002:540), however the point of interest here is simply that overlapping morphological forms introduce further complexity for the analysis of this syntactic construction.

(19)  
(a) They were seen.  
(b) She was sleeping.

Finally, an example of interaction between predicative complements and another syntactic phenomenon is shown in (20), which illustrates how the adjective in the predicative complement can take various modifiers. In these examples, the predicative complement is shown in square brackets and the constituent modifying the adjective is underlined. In the case of relative clauses—where the modifier takes the form of a subordinate clause—the adjective is required to take the form of a superlative (Huddleston and Pullum 2002:547).

(20)  
(a) She is [quite incredibly generous].  
(b) It surely isn’t [that important].  
(c) The nail was [two inches long].  
(d) The view was [beautiful beyond description].  
(e) He is now [the fattest he’s ever been].
The purpose of unpacking this syntactic construction was to provide a sense of their potential complexity, which therefore extends also to their analysis. It also served to more concretely illustrate how both form and function can be used to characterise a syntactic construction and its variants, which supports the observation that there is not one single way to delimit a grammatical phenomenon. For example, rather than focusing on all predicative complements, the characterisation could have placed more emphasis on form, restricting the classification to only adjectival predicates. For a given language or language fragment, there is not necessarily a single correct solution to this process, but rather competing theories, which are evaluated by the extent to which they account for the data of a given language as well as their overall degree of parsimony.

The act of characterising grammatical phenomena and the accompanying circumscription of the phenomenon space are integral parts of the grammatical analysis process, but these activities also occur in a range of other applications, and with the distinct needs of each use case will come a different approach to dividing up the phenomenon space. In Section 2.8, I identify and discuss a range of computational linguistics projects which involve the characterisation of grammatical phenomena.

### 2.3.6 Frequency

In the wild, which is to say, real world language use, grammatical phenomena do not occur with a uniform distribution. Instead, the distribution has something of a long tail, with a relatively small number of phenomena accounting for a large proportion of phenomena seen in actual usage. This property of language is often invoked by the descriptor Zipfian, a term which refers to Zipf’s law. This law states that in a given corpus of text, the relative frequency of a word is inversely proportional to its rank in the list generated by ordering all distinct words found in the corpus by descending frequency (Zipf 1949). This power law is shown in (2.1), with \( r \) being the \( r^{th} \) most frequent word, and \( f(r) \) being the frequency of that word, and \( \alpha \) being a constant that varies across languages and corpora. It can be read as saying that the second most frequent word in a corpus will occur \( \frac{1}{2^\alpha} \) as many times as the most frequent word, the third most frequent word will occur \( \frac{1}{3^\alpha} \) as many times as the most frequent word, and so on.\(^{17}\)

\[
f(r) \propto \frac{1}{r^\alpha} \tag{2.1}
\]

A practical consequence of Zipf’s law is that corpora used for training statistical

---

\(^{17}\)Zipf’s law was subsequently modified by Mandelbrot to better fit the distribution of words found in language usage and there has since been some debate in the literature over how accurately this formulation fits the distribution (Piantadosi 2014). The distribution of word occurrences within language use can still, however, be broadly characterised by an inverse power law, and it is in this sense which is generally intended when describing language use as being Zipfian.
language models must be large enough to supply a sufficient range of environments for each word-form in order for the model to be capable of making meaningful predictions. This issue of data sparsity becomes particularly problematic when models are trained over longer sequences of words, which have distributions with even longer long tails, requiring even larger corpora for training.

It has been well-established that word-forms have a Zipfian distribution, however it is less certain if this holds for syntactic phenomena. A large part of the challenge in exploring this proposition is that in order to measure their frequency of occurrence, the notion of a syntactic phenomenon must be operationalised within the context of a well-defined and scalable annotation scheme. It is certainly uncontroversial that there is considerable variety in the frequency of syntactic constructions, with some constructions occurring with a dramatically smaller frequency than others. For instance, within the Redwoods Treebank (Oepen et al. 2004), the verbal right-node raising construction occurs in less that 1% of the nearly 40 thousand trees, whereas subordinate clauses can be found in about 55% of trees.\footnote{These coverage numbers come from my investigations using the Typediff tool described in Chapter 4 and are based on the treebank accompanying the 1214 releases of the ERG.} In observing this fact, Flickinger (2011) speculates that given a large enough corpus, syntactic constructions may well be Zipfian, however there has been little investigation into whether this holds. One helpful source is Piantadosi (2014), who investigates the applicability of Zipf’s law to various linguistic phenomena, discovering that syntactic categories from the Penn Treebank—the part of speech tags from word tokens—do follow a Zipfian distribution. Restricting attention to syntactic frameworks that make use of constituency relationships—as opposed to dependencies—the majority of constituents (which could be extracted from phrase structure trees) will likely also be Zipfian. However, while syntactic categories and headed constituents are associated with syntactic constructions, they are not the same thing. Take for example, the predicative complement discussed in Section 2.3.5. As characterised by Huddleston and Pullum, the complement in this construction is a constituent which can be headed by either a noun or an adjective. In other words, the distribution of syntactic constructions is dependant upon how they are defined, which, as I have established, will vary across different analyses and with the needs of different downstream users and applications. Furthermore, implicit in defining a collection of syntactic phenomena for counting is also selecting those which will be counted. To empirically investigate this question properly would require selecting a set of syntactic constructions and anchoring their definition within a linguistically rich annotation scheme that could be automatically generated across large corpora. The challenges involved in performing such an ambitious exercise are to a significant extent what

\footnote{It is possible for constituents to not have a lexical head. This can be seen in some types of free relative constructions (Bresnan and Grimshaw 1978), which are also referred to as fused relatives by Huddleston and Pullum (2002:1068).}
Chapter 2: Background

this current research is premised upon.

That syntactic constructions follow a true Zipfian distribution is therefore a hypothesis that requires empirical testing. This would require a concrete operationalisation of a syntactic construction in order to gather the necessary corpus statistics. Such a project is beyond the scope of this work, however it seems clear that word forms and syntactic categories do follow a Zipfian distribution, and these are at least correlated with the notion of a syntactic construction. I will henceforth refer to the Zipfian long tail of syntactic phenomena with the understanding that it at least indicates a long tailed-distribution of some kind, with there being a distinct possibility that it is a true Zipfian distribution, but that this depends upon the particular definition of syntactic phenomenon. The Zipfian long tail of syntactic phenomena has important consequences for the selection of data used to perform grammatical analysis. This is discussed in Section 2.6.4 and is also relevant for the choice of data used to construct an annotated corpus of syntactic phenomena in Chapter 4.

2.3.7 Summary

In this section, I observed that the term grammatical phenomenon is typically used as a catchall for referring to an arbitrary feature of language, especially in the context of discussions regarding the process of grammatical analysis. I provided a more concrete working definition of the term, namely that it is a feature of language that can be described in terms of one or more linguistic forms, constraints over these forms, one or more linguistic functions, or some combination of these. I also provided a brief outline of the two different concepts of linguistic form and linguistic function, describing their roles in the process of grammatical analysis. I subsequently discussed the impact of characterising grammatical phenomena in terms of either form or function, both from the perspective of individual languages and cross-linguistically. A property of grammatical phenomena, especially seen within syntactic phenomena, is the scope for significant complexity within their analysis. I illustrated this through a brief outline of the syntactic construction of predicative complements, which also provided a more in-depth example of the role that form and function play in the process of characterising grammatical phenomena. Finally, I discussed the importance of the frequency of grammatical constructions, with the problem of data sparsity having notable implications for the data upon which grammatical analyses are based. I also observed that syntactic phenomena most likely possess a Zipfian distribution.

2.4 Descriptive Grammars

Given that this research aims to develop syntactic discovery techniques intended for enhancing descriptive grammars, it is necessary to delve into the nature of this linguistic resource. In this section I discuss descriptive grammars, outlining their
salient properties and challenges involved in their construction. In addition to more generally providing the background context upon which to ground discussion of descriptive grammars throughout the rest of this thesis, this section serves two specific purposes. The first is to establish the limitations of traditional descriptive grammars which motivate the need for syntactic phenomenon discovery techniques. The second is to identify the target audience and their requirements, so that the techniques for syntactic phenomenon discovery that are developed in this thesis are informed by the needs of potential users of tools that would employ these techniques.

What, then, is a descriptive grammar? It is a resource which represents a significant outcome of the practice of descriptive linguistics, with each descriptive grammar aiming to describe how grammatical sentences in its target language are formed and how they have the meaning that speakers of the language understand them to have. As I stated in Section 2.2.2, the result is more than a simple inventory of grammatical phenomena, as each analysis must be consistent with all other analyses found in the grammar. Instead, the grammar must form an internally coherent theory of the language. In other words, “a descriptive grammar of a language is a theory of what it is possible to say, with specified meaning(s), in that language” (Evans and Dench 2006:7).

If the author of the work is successful in compiling a coherent and thorough analysis, it has the potential to become the primary reference for the language, especially for previously undocumented languages—of which there are still many. Of the 7,943 languages of the world listed on Glottolog, 20 2447 of them are known to have published descriptive grammars ranging from medium length descriptions to longer, more extensive grammars over 300 pages, with a further 1959 languages having shorter sketches of a descriptive grammar, under 150 pages in length (Hammarström et al. 2016).

2.4.1 Audience

The potential audience of a descriptive grammar is a diverse range of people that depends, in part, on the nature of the grammar. For well-resourced languages, it is common to find a variety of descriptive grammars aimed at different audiences. Some notable examples of specialised grammars include those targeted towards language learners, and those targeted towards native speakers of the language—in which case the language will be described using the native language itself.21 Such works, however, generally follow at least one descriptive grammar written for an academic audience engaged in linguistic research of one stripe or another. For under-resourced languages, it may often be the case that a descriptive grammar aimed at an academic audience—if one indeed exists—is the only available documentation of the language. The relevance

---

20Glottolog (http://www.glottolog.org/) is an online resource containing a catalogue of the world’s languages and language families, as well as a bibliographic database of linguistic references for these languages.

21See Mosel (2006:42) for a breakdown of ways of classifying different descriptive grammars.
of this is that different kinds of users have different needs, each of which will impact how information in descriptive grammars will be optimally presented. It cannot be assumed, for example, that readers of non-academic grammars will be familiar with terminology used in the linguistics community. The kind of descriptive grammars that are of concern for the current work, however, are those intended for consumption by an academic audience engaged in linguistics research.

Linguists require access to descriptive grammars in order to perform effective research for the same reason that particle physicists, biochemists, and chemical engineers require access to the results of basic physics and other sciences. In the same way that it would be impractical for particle physicists to have to derive the gravitational constant themselves rather than consult reference material, it would be impractical for syntacticians to have to perform an analysis of, say, subordinate interrogatives in French each time an understanding of this phenomenon was needed. Linguists therefore require access to preexisting descriptions of languages in order to drive their own research. Mere documentation of a language is also insufficient for reducing the analytic burden of the linguistics researcher. A repository of French texts and recordings—while an important resource for performing linguistic investigation—does not itself save the syntactician from having to perform an analysis of subordinate interrogatives themselves, and would be akin to providing the particle physicist with the raw output of particle accelerator experiments. It is for this reason that good linguistic theory requires good description (Comrie et al. 1993).

The audience of academic linguists is a heterogeneous group, with its members using descriptive grammars in wide-ranging ways. Typologists need access to descriptions of languages in order to compare, contrast, and classify a diverse range of languages with respect to grammatical phenomena. Formal linguists need access to descriptions in order to establish the range and nature of phenomena that their frameworks must be able to account for, as well as develop models of specific languages. Similarly, for grammar engineers implementing a computational model of a language, a descriptive grammar is a vital reference, as I will discuss in Section 2.6. Descriptive linguists will desire to look at existing analyses of a language, for inspiration in developing an analysis of a language which has similar grammatical phenomena, or for verifying that an analysis is consistent with the facts about a language, potentially finding deficiencies with the existing analysis—as per the practice of challenging existing scientific hypotheses.

2.4.2 Accessibility and Basic Linguistic Theory

In order to maximise the accessibility of descriptive grammars pitched at a linguistic audience, it is important that, to the extent possible, they use established
terminology within the linguistics community to describe grammatical phenomena (Comrie et al. 1993, Noonan 2007, Rice 2007). In the same vein, theory-specific terminology or notation should also be avoided where possible, since familiarity cannot be assumed across the entire linguistics community, and neither can the continued ubiquity of grammatical theories or frameworks currently in vogue. Rice (2006) critiques her own grammar of the Slave language (Rice 1989), which used terms such as subjacency and trace to describe constraints on the location of question words within sentences functioning as questions. These terms invoke concepts associated with the then-current incarnation of Chomskyan transformation-based theories of grammar, but which have been superseded by newer Chomskyan theories of grammar with different conceptual apparatus.

It has been argued that the most accessible and enduring descriptive grammars are those that make use of an informal descriptive metalanguage which draws from the shared body of linguistic knowledge referred to as Basic Linguistic Theory\(^{23}\) (Dryer 2006, Evans and Dench 2006, Rice 2007), with Dryer (2006) claiming that most descriptive grammars written this century make use of it as their underlying theoretical framework. Basic Linguistic Theory is characterised by an emphasis on the use of traditional linguistic theory and a conservative approach towards theoretical and terminological innovation. It is continually subject to reassessment and refinement with respect to descriptive parsimony and cross-linguistic applicability through ongoing developments in formal linguistics and linguistic typology (Dryer 2006). The kinds of analyses presented when describing different linguistic forms in Section 2.3.1, could, for example, be said to be couched in terms of Basic Linguistic Theory.

A further characteristic of Basic Linguistic Theory identified by both Dixon (1997) and Dryer (2006) is that it attempts to describe each language in its own terms. Haspelmath (2010b), however, argues that if Basic Linguistic Theory represents a true theoretical framework, as Dixon and Dryer claim, then this goal cannot be achieved, since a theoretical framework implies a pre-existing inventory of concepts from which to select, preventing real conceptual innovation. I will not explore the debate over whether Basic Linguistic Theory should be treated as a true theoretical framework any further. However, the general notion of a shared inventory of concepts used in the process of performing linguistic description and which is continually open to revision is clearly both a coherent and useful abstraction, and this is how I will continue to use the term. Haspelmath’s objection does, however, draw out a relevant challenge involved in the construction of descriptive grammars. The maxim of describing each language in its own terms is a valuable heuristic for the construction of descriptive grammars (Cristofaro 2006, Dryer 2006, Evans 2008, Rice 2007) but there is an underlying tension between this ideal and the exhortation to use familiar—and thus accessible—analytic tools and terminology. This tension is one that every descriptive grammar

---

\(^{23}\)The term Basic Linguistic Theory was coined by Dixon (1997), and has also been referred to as general comparative grammar (Lehmann 1989). Dryer (2006) contains a good overview of the concept.
Chapter 2: Background

writer must navigate, with there being no single way to solve the problem.

2.4.3 The Anatomy of a Descriptive Grammar

Lehmann (1989) states that a complete description of a language includes the following components:

1. Description of the language system
2. Lexicon
3. Text collection
4. Description of the historical situation of the language

Within a descriptive grammar, the description of the language system is required as a minimum, although a comprehensive descriptive grammar publication could include all four. The description of the language system is the characteristic component that the term descriptive grammar evokes. This component along with the text collection are the components which this thesis is concerned with. In this section I look at common features found within the description of the language system, returning to the use of text collections in Section 2.4.5.

Good (2004) performs a survey of four descriptive grammars, identifying and discussing five features pertaining to information presentation that are shared by all the surveyed grammars. The survey is notable for the research in this thesis since its motivation was to catalyse the development of a suitable model for representing electronic descriptive grammars—descriptive grammars that provide enhanced functionality through computational means, and which are discussed in Section 2.7. For the current purposes though, the inventory that emerges from this survey represents a convenient overview of the salient structural properties of a typical descriptive grammar. In order to frame further discussion of descriptive grammars and their limitations in particular—which to a significant extent catalyses the work in this thesis—I now briefly outline the five structural features identified by Good. I illustrate each with an example from The Cambridge Grammar of the English Language (Huddleston and Pullum 2002), a descriptive grammar of English used in this research, and which is outlined in Section 3.4.

Use of Ontologies

All descriptive grammars invariably make use of some sort of general linguistic ontology of concepts, which can often be characterised as being drawn from Basic Linguistic Theory. This can be seen in Figure 2.2, which shows an excerpt from Huddleston and Pullum describing the verb word class. The excerpt pertains to the subclass of verbs referred to as auxiliary verbs, which are then further subdivided into
the categories of modals and non-modals, revealing the use of an implicit ontology. Such ontologies are generally informal, however there are existing efforts to produce a shared formal ontology of linguistic concepts, such as GOLD (General Ontology for Linguistic Description; Farrar and Lewis (2005)) and ISOcat (Kemps-Snijders et al. 2009), for use in electronic language resources. The large amount of variability seen across languages means that it is often necessary to look beyond the conceptual inventory of Basic Linguistic Theory. Good notes that in addition to general linguistic ontologies, descriptive grammars also invoke ontologies used by subcommunities of linguists—such as those associated with specific language families—as well as local ontologies, which are associated with a specific language.

In terms of developing discovery methodologies, this property of descriptive grammars means that different discovery contexts will use different sets of terminology, often to refer to analogous concepts. Syntactic phenomenon discovery methodologies should therefore take this into account, by supporting dynamic characterisations of syntactic phenomena.

Organisation into Nested Sections

In the interest of accessibility, the contents of descriptive grammars are divided into sections and nested subsections. These divisions tend to mirror those found within a general linguistic ontology, such as Basic Linguistic Theory. For example, Figure 2.3 shows the outline of Chapter 2 of Huddleston and Pullum, showing how the treatment of the verb is divided up into sections and subsections, corresponding to commonly used concepts in descriptive linguistics. However, as Good notes, strict adherence to an ontologically-driven organisation is not always possible when striving to produce a coherent and accessible presentation of the analysis within the confines of a linear document. Further to this, the object of analysis itself—language systems—are resistant to being modelled hierarchically. The inherently interrelated nature of grammatical phenomena and their predisposition to be characterised by multiple dimensions, as discussed in Section 2.3.5 and Section 2.3.3, respectively, means that descriptive grammars may be better modelled as networks or related concepts, with the use of hierarchies stemming from the fundamental linearity of documents (Good
In Section 2.4.4, after discussing strategies for organising grammatical phenomena into a linear presentation, I conclude that additional means of getting around this constraint are required and that techniques for syntactic phenomenon discovery could assist with this.

**Descriptive Prose**

As already discussed, the use of prose text to perform linguistic analysis is a distinctive feature of descriptive analysis. An advantage of prose description is that, unlike formal analysis, it does not presuppose any specialised knowledge of a particular formalism. This makes for a resource that is accessible to a wider audience, and one that is also more likely to stand the test of time, remaining relevant even as formalisms come in and out of fashion. Good illustrates this last consideration through Williamson’s 1965 grammar of Ijo (Williamson 1965), describing it as a “legacy” grammar due to its being based on early transformational-based grammatical theories, which are now outdated. This is reflected in the grammar’s terminology, notation, and even organisation, all of which make it less accessible to a contemporary linguistic audience.

Another feature of descriptive grammar prose, also noted by Good, is the extensive use of inline references, including references to lexical items, terms drawn from ontologies, linguistic examples, and cross-references to other sections. Figure 2.4 shows an excerpt from Huddleston and Pullum illustrating the latter two types of references. The first paragraph contains a reference to linguistic examples and the second contains a cross-reference to another section of the grammar. The use of cross-references provides a means to work around the limitations of the hierarchical ordering required by a linear document, while reference to linguistic examples forms an integral component of grammatical description, and is a feature that can be augmented through the use of precision grammars. I discuss the use of examples further in the next section.

**Exemplars**

An indispensable feature of a document discussing the workings of a language is the use of examples of the language in action. Good makes the terminological distinction between an exemplar of a grammatical phenomenon and an example of a grammatical phenomenon. He defines an exemplar as a piece of language data selected by an author in order to illustrate and aid in the description of a specific grammatical phenomenon. This is contrasted with an example, which Good takes to refer to a positive instance of a phenomenon that was not explicitly selected. Exemplars, then, imply an aspect of curation, whereas examples do not. Using Good’s terminology, the results of a syntactic discovery procedure being applied to a corpus to retrieve

---

24See Evans and Dench (2006) for a discussion of the tension between the accessibility of informal prose and the increased rigour and efficiency of formal models.
Figure 2.3: A portion of the table of contents for Chapter 2 of *Huddleston and Pullum (2002:71)*, showing how the analysis of the syntactic category of *verb* is divided into sections and subsections.
instances of a phenomenon are *examples* of the phenomenon in question, whereas those originally chosen for inclusion in the descriptive grammar are *exemplars*. I will adopt this terminology from here on.

Good identifies two distinct types of exemplars: *lexical exemplars* and *textual exemplars*. The former pertain to words or morphemes accompanied by their grammatical function, while the latter pertain to sentences or fragments. Figure 2.5 shows the use of lexical exemplars to illustrate the different categories of English determiners identified by Huddleston and Pullum. Figure 2.6 shows the use of textual exemplars to illustrate the contextual restrictions on genitive pronouns from Huddleston and Pullum. Since the research in this thesis is concerned with syntactic phenomena, which manifest at the level of sentences or fragments, I will take the term *exemplar* to refer to a textual exemplar from here on.

There is, however, only a limited number of exemplars that can be associated with the description of a grammatical phenomenon due to practical considerations, such as the amount of space available in a physical document, readability, and also the limited time of the descriptive grammar author (Schultze-Berndt 2006). The author may ultimately have to include fewer exemplars than would be preferable for a comprehensive treatment of the phenomenon in question. From the reader’s

---

25See Weber (2007a) for a discussion of good practice in presenting linguistic exemplars in descriptive grammars.
Chapter 2: Background

### Figure 2.5
An excerpt from *Huddleston and Pullum (2002:356)* showing the use of lexical exemplars order to illustrate the different kinds of determiners.

| 5 | i the, a | articles |
|   | ii this, that | demonstrative determinatives |
|   | iii we, you | personal determinatives |
|   | iv all, both | universal determinatives |
|   | v each, every | distributive determinatives |
|   | vi some, any | existential determinatives |
|   | vii one, two, three, . . . | cardinal numerals |
|   | viii either, neither | disjunctive determinatives |
|   | ix no | negative determinative |
|   | x another | alternative–additive determinative |
|   | xi a few, a little, several, . . . | positive paucal determinatives |
|   | xii many, much, few, little, . . . | degree determinatives |
|   | xiii enough, sufficient | sufficiency determinatives |
|   | xiv which, what, whichever, whatever | interrogative and relative determinatives

### Figure 2.6
An excerpt from *Huddleston and Pullum (2002:471)* showing the use of textual exemplars in order to illustrate the environment in which genitive pronouns can occur.

| 48 | i The Guardian seems to respect its readers more than the Sun respects its. |
|    | ii *The Bank is being sued by a rich client of its.* |
|    | iii The council appears to be guilty of the illegal sale of houses that were not its to sell in the first place. |
perspective though, after reading the included exemplars, the reader might still have unanswered questions regarding a phenomenon, such as whether it can co-occur with other phenomena, and if so, whether there are any interactions. Given the important role linguistic exemplars play in conveying to a reader the nature of a grammatical phenomenon, it would be desirable for the reader to have access beyond those included by the author (Bender et al. 2012). In Section 2.7.2, I argue that query tools driven by syntactic phenomenon discovery techniques could be used to provide users with the ability to dynamically retrieve further examples of a given syntactic phenomenon.

**Structured Description**

The last feature of descriptive grammars Good identifies within the survey is structured description. Good describes this as being typically characterised by tabular data that is offset visually from the surrounding prose and that covers a particularly coherent domain of a language’s grammar, making it amenable to systematisation. This structural feature of descriptive grammars is also often referred to as a *linguistic paradigm*, which Bird (1999:33) describes as “any kind of rational tabulation of linguistic forms, such as phrases, words, or phonemes, intended to illustrate contrasts and systematic variation.” This definition captures many types of structured description, but perhaps one of the most familiar examples of a linguistic paradigm is the set of possible conjugations of a verbal lexemes, as illustrated by Table 2.1 found in Section 2.3.1. This feature of descriptive grammars is less relevant to the work in this thesis.

**2.4.4 Organisational Strategies**

In the previous section, I established that a feature of descriptive grammars is the organisation of the analysis into nested sections typically corresponding to linguistic concepts found in Basic Linguistic Theory. There are many ways this organisation can be performed, with the final choice by the grammar author impacting upon the quality of the resulting grammar. A number of factors contribute towards the particular organisation chosen for a descriptive grammar. Firstly, the analysis of the language performed by the author—which could be informed by a specific grammatical framework or draw divergent conclusions about the nature of the language’s grammatical phenomena from other analyses of the language—plays a significant role in determining the structure of the grammar. The target audience of the grammar also plays an important role, with different audiences, even within the linguistics community, having different accessibility needs, as discussed in Section 2.4.1. Even given an analysis of the language and a target audience, the grammar writer must also navigate the tension between maximising accessibility by using familiar concepts from Basic Linguistic Theory and describing the language in its own terms, a process that can impact the organisational structure in addition to the terminology used. Ad-
ditionally, the very nature of language systems contributes towards the multitudinous possibilities for organising a descriptive grammar. As discussed in Section 2.4.3, the inherent inter-relatedness of grammatical phenomena means that language systems and their analyses are perhaps better modelled as networks rather than hierarchies. Ultimately, however, the constraints of a static document mean that a single linear structure must be chosen for the presentation of the analysis, with there being many possible ways for this to be performed, each with their own resulting compromises.

The relevance of this problem to my research is that regardless of the ultimate organisation arrived upon, there will be users, or use-cases, for which the chosen arrangement will suboptimal. This demonstrates the need for alternative means of navigation beyond the chosen hierarchical one, which syntactic discovery techniques have the potential to facilitate. Additionally, the organisational strategies employed within the construction of descriptive grammars are shaped by the same user-driven needs that alternative navigation tools will also need to be informed by, and should therefore be consulted as a part of the process of developing syntactic discovery techniques that are used for navigating descriptive grammars. I will now briefly describe three notable kinds of organisational strategies that have been identified as providing templates or heuristics to follow when deciding how to organise a descriptive grammar in order to maximise the accessibility of the analysis.

**Form and Function**

Descriptive grammar authors must decide how to divide up the analysis of grammatical phenomena into chapters and sections. When describing strategies for categorising and organising grammatical phenomena within a descriptive grammar the distinction between linguistic forms and their function is often invoked (Comrie et al. 1993, Cristofaro 2006, Lehmann 2004, Mosel 2006, Bouda and Helmbrecht 2012), yielding two distinct perspectives from which to arrange grammatical phenomena. When discussing the characterisation of grammatical phenomena in Section 2.3.3, I made the distinction between encoding ambiguity and decoding ambiguity. This distinction of perspective is relevant again here for form and function-based organisational strategies. The form-oriented, or *semasiological*, approach takes the vantage point of a hearer decoding language, analysing a sentence by its component pieces in order to arrive at its meaning. This form-to-function approach groups grammatical phenomena by related forms, and then describes the function of each of the respective forms in the group, which will often have distinct functions. The function-oriented, or *onomasiological*, approach on the other hand, takes the vantage point of a speaker encoding language: starting with a target meaning, they must synthesise a sentence in order to convey the meaning. This function-to-form approach groups grammatical phenomena by their function, then outlines and describes the range of individual forms and constructions that can realise each function. To make this more concrete, the function-to-form approach to grammar organisation could yield a chapter
on the grammatical phenomenon of negation, covering the various forms of a language that serve to negate a portion of the propositional content of the sentence. Under this approach, negation strategies which are structurally distinct—such as both morphological and syntactic negation—would be included within the same category. A form-to-function approach, on the other hand, would place these different kinds of forms into different chapters.

Both these organisational strategies have different advantages depending on the present needs of the reader of the descriptive grammar. The form-to-function organisational strategy has the advantage of providing the reader with a more immediate sense of how the entire grammatical system works (Cristofaro 2006, Mosel 2006). Once the reader has a scaffolding by means of the important forms of a language, they can use this to hang more nuanced discussions of function and complex variation in forms. Beginning with the forms of a language also enables description to better follow the maxim of describing a language in its own terms, since functional categories used to describe languages are largely defined independently of any one language (Cristofaro 2006). Arrangement by form also has the potential to illuminate similarities and relationships between different forms, which can reveal the grammatical function of forms, whereas arrangement by function will inevitably split up constructions similar in form across different functionally characterised sections, potentially obscuring the shared grammatical function contributed by a form (Cristofaro 2006, Mosel 2006).

As discussed in Section 2.3.3, the functional characterisation of grammatical phenomena, on the other hand, more readily permits cross-linguistic comparison due to the fact that different languages have different inventories of forms. Form-based queries across multiple languages are limited in their utility as they will yield null results in the case of languages which simply do not possess the form. Under the assumption that all languages have the capacity to encode the same range of functional content, presenting a typological analysis in functional terms, then, is more likely to result in a unified analysis, and is the preferred mode in linguistic typology. The function-to-form arrangement of phenomena within a descriptive grammar would therefore more readily permit typologists to obtain answers needed to drive their research (Cristofaro 2006, Good 2012, Zaefferer 2006).

Ideally, a descriptive grammar should therefore provide access to the grammatical phenomena of a language using both form-to-function and function-to-form lenses. This, however, would result in a significant duplication of work, with the analyses of the grammatical phenomena of the language being presented twice in parallel. Practical constraints mean that only one is chosen, and the pragmatism of improved accessibility from a non-typological perspective has proven to have won out, with most descriptive grammars taking a broadly form-to-function approach (Cristofaro 2006, Nordhoff 2012b, Mosel 2006, Payne 2007, Zaefferer 2006).26 This does not mean a

26Two exceptions that have been noted to this trend are Leech and Svartvik’s (1975) A Communicative Grammar of English (Evans and Dench 2006, Mosel 2006, Evans 2008) and Willett’s (1991)
strict adherence to this guiding principle, as some portions of a language system will be more naturally suited to organisation by function. Determining when such deviations from a form-to-function layout offer increased utility to the reader is another challenge authors of descriptive grammars are faced with. One simple heuristic is to organise sections on phonology and morphology by form and, for sections pertaining to syntax to use more function-to-form organisation (Rice 2007\textsuperscript{27}).

These competing organisational strategies of form and function are also relevant from the perspective of grammatical phenomenon querying systems, since, similar to a strategy for organising a descriptive grammar, they also serve as a means of navigating the analyses of a particular grammar, albeit a computational one. Depending on the underlying annotations being queried, they have the potential to be constrained with either formal constraints or functional constraints. In Section 2.8.3, I discuss the ESD project (Flickinger et al. 2014b), which, as a part of the documentation of the ERG, involves the development of a querying tool that is able to query treebanks for semantically characterised phenomena, and in Section 2.9 I also discuss the use of a range of querying approaches which largely constrain their target phenomena using phrase-structural constraints. As I discuss in Section 2.8.3, while the distinction between phrase-structural annotation and meaning representation is not the same thing as the form-function distinction, there is sufficient overlap that these querying methods can be used as proxies for form and function-based querying.

Order

From the perspective of how to order the macrostructure of a descriptive grammar—the chapters and larger sections—one recommendation that is often made is to proceed in the direction of increased complexity of grammatical phenomena (Mosel 2006, Rice 2007). In accordance with this, a typical progression that is seen across many grammars is phonology $>$ morphology $>$ syntax. Moving from simpler forms to more complex forms provides a more accessible user experience, gradually introducing more basic distinctions made by the language which more complex phenomena implicitly invoke. Another recommendation pertaining to section ordering is to put sections that are cross-referenced by other sections first (Mosel 2006, Rice 2007). This recommendation is presumably premised on the fact that backward references to already seen material incur less cognitive friction that forward references to unseen and consequentially unfamiliar material. Both of these strategies should be considered subject to the descriptive needs of the language in question, and can perhaps be thought of as useful heuristics to apply when all other factors remain equal. There will inevitably be occasions where the decision to organise a descriptive grammar on the basis of one of these heuristics will result in a suboptimal arrangement for a particular reader or particular use case. This underscores the need for alternative navigation strategies,

\textsuperscript{27}Rice (2007) cites personal correspondence with Martin Haspelmath.
such as phenomenon-based navigation, that frees the user up from any one particular linear ordering.

Cross-Linguistic Comparability

Comrie et al. (1993) argue that in order for a descriptive grammar to be accessible to the widest possible audience, not only should it use standardised terminology where possible, but it should also aim to maximise cross-linguistic comparability by following a standardised framework for performing descriptive analysis. This represents not just the use of a shared conceptual framework to draw from when performing the analysis, as found in Basic Linguistic Theory, but also the use of a shared framework for arriving at the organisational structure of the analysis, thereby dictating the structure of the resultant descriptive grammar. The challenge for such a framework is that it must be sufficiently constrained so as to facilitate cross-linguistic comparability but also sufficiently flexible to accommodate the cross-linguistic variation that exists. One attempt at creating such a cross-linguistic framework for descriptive grammar-writing is the Lingua Descriptive Studies questionnaire (Comrie and Smith 1977), which constitutes a hierarchical series of questions pertaining to the structure and composition of a language, which, when answered, aimed to yield a comprehensive description of the language in question. Grammars of different languages modelled off this questionnaire share the same structural layout and can thus be more readily compared. This represents a different organisational strategy towards the arrangement of the structure of a descriptive grammar. Rather than attempting to come up with a dynamic solution based on the needs of each language, the questionnaire forms a template, which grammar authors can choose to subscribe to. The questionnaire has proven to be quite popular for this purpose, with over 30 grammars being based on it (Mosel 2006).

The high-level organisation of the Lingua Descriptive Studies Questionnaire is syntax > morphology > phonology, a sequence that runs counter to that typically seen in descriptive grammars. It has been suggested that this counter-intuitive structural feature has contributed towards the framework’s decreased use (Mosel 2006). With respect to form and function, the structure of the questionnaire takes a hybrid approach, with questions pertaining to functional domains being interleaved within the aforementioned overarching formal sections. Comrie et al. (1993) note this as a deficiency, citing the example of §1.1.2 within the questionnaire, which deals with subordination and contains a subsection dealing with time clauses. This arrangement fails to generalise cross-linguistically, however, since time clauses are not always subordinate across different languages. The Comrie and Smith questionnaire could be changed to improve these deficiencies, however these improvements would come at the cost of comparability with all existing grammars based on the Comrie and Smith questionnaire, whose table of contents and organisational structure would no longer match that of the new framework (Comrie et al. 1993). More generally, it is important
that new advances in linguistic theory and understanding of cross-linguistic variation can be incorporated into the framework, making this update problem one that would persist indefinitely (Comrie et al. 1993, Zaefferer 2006).

There are clearly advantages to be had from prioritising cross-linguistic comparability, however the decision for a descriptive grammar to target a shared—and static—organisational structure incurs a significant cost. Firstly, the use of a rigid analytic framework for performing linguistic description conflicts with the goal of describing a language in its own terms. Secondly, in order to preserve cross-linguistic comparability, the framework itself cannot take advantage of analytic innovations within the linguistics community. For many descriptive grammar authors, these costs are unacceptable and will lead the author to instead choose to balance the demands of cross-linguistic comparison with the presentation of their language’s characteristics on a case by case basis.

Implications

The use of these strategies and heuristics can dramatically improve the accessibility of a descriptive grammar, however they do not change the fact that the linear document of a traditional descriptive grammar cannot capture the interrelatedness of grammatical phenomena. Heavy use of cross-references—as discussed in Section 2.4.3—and the inclusion of an index can be used to work around the constraint of linearity and expose, to some extent, the interconnected nature of grammatical phenomena. These strategies, however, inevitably involve trade-offs in terms of accessibility and cannot truly capture the network-like structure of grammatical phenomena. This has negative implications for the discoverability of grammatical phenomena within traditional descriptive grammars.

A strategy for mitigating these shortfalls in the discoverability of descriptive grammars, as suggested by Bender et al. (2012), is to use techniques for grammatical phenomenon discovery to automatically insert cross-references between exemplars and all sections of the grammar describing phenomena found within the exemplars. This opens up the possibility for non-linear navigation through the grammatical phenomena found within exemplars used in the grammar. This would then ameliorate the effects of the inevitable compromises incurred by selecting a single hierarchical organisation. In Section 2.7, I discuss how the use of precision grammars for performing syntactic phenomenon discovery has the potential to provide syntactic phenomenon-based navigation of descriptive grammars, thereby enhancing the discoverability of syntactic phenomena within descriptive grammars.

2.4.5 Text Collections

A comprehensive descriptive grammar will be accompanied by a collection of texts of the language being described, preferably presented with morpheme glosses and
translations (Evans and Dench 2006, Lehmann 1989). If included within the publication, this is typically found as an appendix following the description of the language system (Mosel 2006), or otherwise as a companion publication (e.g. Heath 1980, Mosel 1977). In this section, I briefly describe the relationship between text collections and descriptive grammars. This will serve to further motivate the use case of syntactic phenomenon discovery for the retrieval of examples and, in doing so, will also help illuminate some of the characteristics that techniques developed in this thesis will need to possess.

Text collections present the opportunity for the grammar author to make available to the reader the data upon which the analysis of the language was performed. Broadly speaking, the data used to develop an analysis of a language can be divided into two kinds: naturally occurring corpus text, and examples elicited from native speakers by linguists. Attitudes towards the use of these respective sources for developing linguistic analyses vary considerably within the linguistics community, and I discuss this debate further in Section 4.2.1, in the context of selecting a source of data for the annotated syntactic phenomenon corpus that is constructed in this research. In the context of developing descriptive grammars, Weber (2007b) provides some pragmatic advice for constructing text collections advocating for careful use of elicited examples.28

Within the description of the language system, textual exemplars of grammatical phenomena are typically presented devoid of any context. When exemplars are sourced from a corpus, the inclusion of the corpus in the publication provides the opportunity for the reader to see them in their original context of use. This enables the investigation of supra-sentential grammatical phenomena—typically treated as outside the scope of descriptive grammars—as well as providing a window on the norms of a language—such as the relative frequency of different grammatical phenomena and their characteristic context of utterance, neither of which are available from decontextualised exemplars (Lehmann 2004).

The inclusion of text collections also serves to improve the scientific accountability of the descriptive grammar. In the same way that a scientific theory must be based on empirical evidence, so too should the theory of the language system that the descriptive grammar represents (Mosel 2006). The inclusion of a collection of texts presents the author with the opportunity to make available the complete primary data upon which their analysis was based. In an appropriately cross-referenced text collection, readers would be able to follow exemplars to their context of use in the included corpus, thus more readily facilitating the verification or falsification of analyses in the descriptive grammar (Mosel 2006, Thieberger 2008). Since the grammar writer cannot be expected to identify and analyse every grammatical phenomenon within a language, this practice also allows readers to extend the theory to account for phe-

---

28See also Evans and Dench (2006) for an overview of attitudes towards the use corpora in descriptive grammars.
nomena not analysed within the grammar (Cristofaro 2006, Rice 2007, Thieberger 2008). Mosel (2006), for example, discusses an instance where he did not provide an analysis for the Tolai particle *iat* within his book on Tolai syntax (Mosel 1984b) nor notice its equivalence with the Tok Pisin word *yet* in his comparison of the two languages (Mosel 1984a). Based on Mosel’s Tolai text edition (Mosel 1977), Sankoff (1993) was subsequently able to identify *iat* as a focus marker and also link it to Tok Pisin *yet*. Similarly, as linguistic theories change, the inclusion of a text collection also permits the application of updated analytical approaches to the original data (Mosel 2006). The inclusion of texts beyond those used to support the analysis also allows others to test the predictive power of the theory, determining whether the analysis presented in the descriptive grammar extends to data that has not yet been analysed (Good 2012, Thieberger 2008).

Despite the clear value of including text collections in descriptive grammars, they are all too often omitted entirely (Thieberger 2008), and when they are included, the amount of text they contain is often limited and usually relegated to an appendix—a status incongruous with their importance to the resource (Mosel 2006). In terms of trying to explain this discrepancy, first and foremost, it should be noted that the preparation of a high quality text collection, which involves transcribing, translating, and providing morpheme glosses, represents a particularly large amount of work. Mosel (2006) also identifies some cultural factors which have contributed towards this situation, including the observation that descriptive grammars themselves have low-prestige status, often not being considered suitable for the contribution of Ph.D. theses, and also that linguistic typology tends to focus on sentence-level grammatical phenomena, which do not require the surrounding context that a corpus provides. There are two more pragmatic considerations that stand out as contributing towards this situation. The first is that, within a printed publication, space is at a premium; the practical constraints of a printed resource mean that the amount of text that can be feasibly included in a descriptive grammar is limited. The adoption of electronic descriptive grammars, which I discuss in Section 2.7, can assist with the first problem, by removing the physical constraints of the printed publication, allowing for the inclusion of larger text collections.

The other factor contributing towards the paucity of text collections is the lack of facilities for integrating grammatical descriptions with their corresponding text collection. Ideally, users should be able to readily navigate between grammatical descriptions and relevant data from the text collection. However, even when included within the one publication, the two components are still constrained by the linear nature of a text publication (Musgrave and Thieberger 2012). Just as within the organisation of grammatical descriptions, cross-references can be used to work around these constraints, by linking exemplars in grammatical descriptions to their context of use within the text collection and also linking instances of grammatical phenomena with

---

other usages within the text collections. However, the extensive use of cross-references that would be required to take full advantage of the potential improvements in scientific accountability represents an onerous undertaking, meaning cross-references are rarely used to this extent. A notable exception to this is Jeffrey Heath’s description of the Australian language Nunggubuyu, which is spread across three works: a text collection (Heath 1980), dictionary (Heath 1982), and descriptive grammar (Heath 1984). These works feature a high degree of interlinking, with links from grammatical phenomenon description in the grammar and also lexical entries in the dictionary both pointing to locations within the text edition where corresponding examples can be found. While it could be argued that Heath’s grammar, which he refers to as a corpus-based grammar, represents the paragon of descriptive accountability, users of these resources, however, found them hard to use, as time consuming navigation between the different physical volumes was required (Musgrave and Thieberger 2012). This issue can be solved through the use of hyperlinks, which, as discussed in Section 2.7.2, can be used to improve the accessibility of descriptive grammars in a range of ways. Perhaps an even bigger concern, however, is the creation of such links in the first place. Heath’s extensively cross-referenced works represent an exceptional undertaking, requiring a considerable amount of time for preparation (Heath 1984); it would be unrealistic to encourage this level of effort from all descriptive grammar authors. Additionally, hyperlinks such as these would be limited to those included by the grammar author, who will not necessarily be able to anticipate the ways in which users of the descriptive grammar may want to explore grammatical phenomena.

There is clearly then a need for techniques that facilitate the integration of grammatical descriptions with their supporting text collections. In Section 2.7.2, I discuss how syntactic phenomenon discovery techniques could be used to reduce the challenges involved in integrating these resources, which, if successful, would lead to the improved scientific accountability of descriptive grammars. The importance of this integration also feeds into my decision in Section 4.2.1 to use corpus data for the construction of a collection of syntactic phenomenon annotations.

### 2.4.6 Summary

In this section, I described the resource that is the descriptive grammar, focusing in particular on descriptive grammars that are targeted at an academic audience engaged in linguistic research. In the context of performing and presenting the analysis of a language in a descriptive grammar, I discussed the use of the shared conceptual inventory often referred to as Basic Linguistic Theory, and how a particular challenge for descriptive grammar authors is balancing the use of concepts familiar to a linguistic audience in order to promote accessibility, while also presenting the language in

---

30Musgrave and Thieberger (2012) describe initial work converting portions of Heath’s Nunggubuyu descriptive trilogy into as hyperlinked resource. I described this work briefly in Section 2.7.3.
terms of its own particular characteristics, which may not always align with concepts
found in Basic Linguistic Theory. I also outlined the anatomy of descriptive gram-
mars, describing five typical structural features identified by Good (2004). Through
discussion of these features, two observations salient to this research emerged.

The first observation is that the contents of descriptive grammars are organised
into nested sections corresponding to descriptive concepts from Basic Linguistic The-
ory. This organisation into a hierarchical structure is necessary for presenting the
analysis of a language system within the confines of a linear document, however
grammatical phenomena are not themselves organised hierarchically, instead being
interconnected. I proceeded to argue that the use of syntactic phenomenon discovery
techniques could be used to drive phenomenon-based navigation, allowing naviga-
through automatically inserted links within exemplars, pointing to other parts
of the grammar, forming a kind of automated phenomenon-based cross-references.
The second observation was regarding the importance of linguistic exemplars within
descriptive grammars, with descriptions of grammatical phenomena making frequent
use of them for illustrative purposes. Traditional descriptive grammars, however, are
limited by the number of exemplars that can be included alongside descriptions. The
use of syntactic phenomenon discovery techniques for the retrieval of additional ex-
amples could allow users access to a much wider range of examples without taking
up any additional space. I also discussed the importance of text collections, and how,
despite the potential for increasing the scientific credibility of the theory, they are fre-
cently omitted or relegated to the status of an appendix. I argued that, when used
to query accompanying text collections, syntactic phenomenon discovery techniques
which return positive instances of the grammatical phenomena being described would
tighten the integration of the two descriptive resources, providing direct access from
the descriptions to relevant data found in the text collection.

In the following section, I move onto describing another type of linguistic resource
relevant to the work in this thesis, this one rooted in computational linguistic research.

2.5 Treebanks

Treebanks are corpora of text that are annotated with linguistic structure, most
typically syntactic or semantic. While treebanks are not necessarily restricted to con-
taining tree-like structures, the most common types of treebanks contain constituency
or dependency trees. The value of treebanks for this research lies in their potential for
being used as a source of examples of syntactic phenomena. They are prime candi-
dates for this not just because of the kind of linguistic annotations they contain, but
also because of the considerable size associated with many treebanks, as well as the
effort that is put into maximising the quality of these annotations. Furthermore, the
problem of how to retrieve specific kinds of trees from within a treebank is one that
has received a significant amount of attention, with there existing a range of differ-
ent methodologies and implemented tools for querying a treebank for tree fragments meeting a target set of grammatical constraints.

In this section, I outline the some of the applications treebanks are used for and briefly describe typical approaches used for querying their contents. I also discuss traditional approaches to their construction—such as that exemplified by the Penn Treebank (PTB: Marcus et al. 1993)—in order to illustrate the resource-intensive nature of their construction.

Treebanks come in both constituency and dependency forms\(^3\) and while the majority of this overview could be applied to both, I focus on constituency treebanks, as this is the type of treebank used in this thesis. It is worth noting however that, while dependency parsing has a strong history of use in the context of applications oriented more towards NLP, there is increasing interest in the use of typological information in the development of cross-linguistically applicable resources (O’Horan et al. 2016). The emergence of the Universal Dependency project\(^4\) has played a notable role here, developing a set of annotation guidelines for producing cross-linguistically consistent dependencies as well as a growing number of treebanks across different languages that are based on this annotation scheme (Nivre et al. 2016). As noted by Croft et al. (2017), the annotation scheme employed by the Universal Dependency project has a considerable degree of compatibility with recent typological theory, and as such, there is potential for exploring the use of these resources in the context of enriching descriptive grammars.

### 2.5.1 Applications

Treebanks have two broad types of applications. The first involves the use of treebanks of gold-standard syntactic annotations being used to train statistical language models. A notable example of this is the methodology of inducing probabilistic context free grammars, resulting in statistical parsers such as those of Collins (1997) and Charniak (2000). Statistical parsers such as these are typically more robust than hand-crafted rule-based parsers, and with a large enough amount of treebanked data to train over, can attain wide levels of coverage without the need for time-consuming manual rule construction. Of course, these types of statistical models are premised on the existence of treebanks containing gold-standard syntactic analyses in the first place, the construction of which is discussed in Section 2.5.3. Treebanks can also be used for the training of parse selection models, which are used to rank, in order of likelihood, the many competing analyses that a computational grammar can yield for a single sentence. The ERG, for example, has a parse selection model which is trained on the Redwoods treebank (Oepen et al. 2004). The importance of parse selection models is discussed further in Section 2.6.2, in the context of grammar engineering

\(^3\)The distinction between these two types of syntactic annotations was discussed in Section 2.3.1.

\(^4\)http://universaldependencies.org
The other broad type of application of treebanks—which is of more relevance for this research—is for driving linguistic hypothesis formation and testing, in order to advance linguistic theory. Researchers can exploit large treebanks to empirically ground a particular theory by looking for counter-examples that refute a hypothesis, or ascertaining frequencies of positive instances to increase the support for a particular hypothesis (Meurers and Müller 2009). Treebanks support this kind of research by facilitating the querying of certain types of grammatical phenomena. The simplest kinds of linguistically annotated corpora are sequentially tokenised and tagged with linguistic categories such as parts of speech and grammatical categories like number and tense. These types of annotated corpora are commonly associated with corpus linguistics investigations, as was described in Section 2.2.3. A syntactically annotated treebank offers further querying power, as it is not limited to simple sequential queries, such as tag sequences and regular expressions over surface text, but can support queries with complex structural constraints.

The use of treebanks for linguistic hypothesis testing is exemplified by Meurers and Müller (2009), who perform three case studies using the TIGER German corpus (Brants et al. 2004) to test three distinct hypotheses pertaining to German. The first involves the syntactic construction of extraposition from within complex noun phrases, the second involves fronted particle verbs, and the third pertains to the use of fronting as a constituency test. In each case, Meurers and Müller (2009) demonstrate the need for syntactic annotations in order to craft queries which can effectively test the hypothesis. In the case of the third case-study, the number of trees returned by the query constructed to investigate the hypothesis was insufficient for making concrete claims, which, as observed by Meurers and Müller (2009), serves to illustrate the importance of the size of the treebank for this application. In accordance with Zipf’s law—which was discussed in Section 2.3.6—some syntactic constructions occur with such low frequency that meaningful investigation would require corpora that would be too large for manual annotation. This forms an argument for the use of treebanks which are constructed from the output of linguistically precise computational grammars, which can also be subjected to a significant degree of automation. I discuss such treebanks in Section 2.6.5. These considerations of corpus-size are pertinent to the current research, since additional examples of phenomena can only be retrieved if the chosen corpus contains some instances in the first place.

Some other examples of treebanks being used for linguistic hypothesis testing include Weber and Müller (2004), who investigate German word order variation using a newswire corpus, and van Bergen and de Swart (2009), who investigate scrambling—a specific type of word order variation—in a spoken Dutch corpus. Also related is the Linguist’s Search Engine (Resnik and Elkiss 2005), discussed further in Section 2.9.3, which represents another type of approach for performing linguistic hypothesis testing. It enabled portions of the Web to be queried for syntactic patterns, but did not make use of an underlying treebank.
2.5.2 Querying Treebanks

A common paradigm for finding examples of syntactic phenomena from within treebanks is for a user to construct an expression that captures the desired structural constraints using a query language and then to execute the query against a treebank using an appropriate tool. There exists a number of such tools for querying treebanks; two popular tools include TGrep233 and TIGERSearch (König et al. 2003), each which use their own query language. These query languages must balance the needs of linguistic expressivity, ease of use, and efficiency (Lai and Bird 2010, Clematide 2015). Satisfying all of these simultaneously is difficult. How these needs translate into concrete requirements varies according to the nature of different treebanks and associated research goals. Consequently, there exists a range of different query languages offering different advantages. Clematide (2015) identifies two main types of query languages. Path-based languages invoke queries that select nodes by constraining paths through the queries, a notable example being LPath (Bird et al. 2006), an extension to XPath—a general purpose XML querying language—and which was motivated by the need for improved linguistic expressivity within XPath. Logic-based languages use first-order logic to apply constraints to tree structures. The query language used by TIGERSearch belongs in this category. The query language that the popular TGrep2 tool uses falls into neither of these two categories, being defined around a number of node relationships, such as immediate precedence.

Table 2.2 shows, for each of the three query languages identified above, the query that corresponds to the command find noun phrases whose rightmost child is a noun. This command and corresponding queries are taken from Lai and Bird (2004). This illustrates the mapping between target linguistic constraints and precise query expressions that a researcher must perform in order to use a treebank query tool,34 and which requires familiarity with the syntax of the relevant query language. The sometimes arcane syntax used by these tools, in addition to the lack of standardisation across the many tools and accompanying query languages, make this requirement a barrier to the use of treebank query tools. Alternative approaches for searching for instances of grammatical phenomena within treebanks that don’t require knowledge of a specific query language are discussed in Section 2.9.3.

2.5.3 Treebanking Methodology

In order to support linguistic hypothesis testing and training of accurate statistical models, not only must treebanks be of a considerable size, but their annotations must also achieve a high level of reliability, making the construction of a treebank a laborious endeavour. The process begins with the design of an annotation scheme for

---

33 https://tedlab.mit.edu/~dr/Tgrep2
34 For a more comprehensive survey of different query languages, see Lai and Bird (2004) and Clematide (2015).
Capturing the desired linguistic structure. Treebank designers must take into account
the downstream applications the treebank will aim to support, what kind of gram-
matical phenomena will be captured by the annotations, and how the annotations
will be represented. This process typically involves the use of formal linguistic theory
for designing the formalism used to represent linguistic structure (Frank et al.
2012).

The annotation itself can either be performed completely manually, or semi-
automatically. In the latter case, a common approach is to parse the underlying
corpus with an existing parser and then use its annotations as a starting point for
human annotators to manually correct and extend, as required by the annotation
guidelines. This is the approach taken in the PTB (Marcus et al. 1993), a corpus of
4.5 million words annotated with part of speech tags, over half of which have been
annotated with phrase-structure trees in the form of labelled brackets. These trees
are notable for being relatively theory-neutral, however decisions made during the de-
velopment of the annotation scheme have nevertheless had a limiting effect upon the
kind of grammatical phenomena which can be recovered from them. This is discussed
further in Section 4.3. Despite this, the PTB has proven to be of considerable value
for training wide coverage statistical parsers and for performing linguistic enquiry.

The annotation schemes used by large-scale treebanks such as the PTB and those
produced within the Universal Dependencies project are typically of a much reduced
complexity when compared to analyses produced by formal grammars that aim to
accurately model our linguistic understanding of a language. This is shaped in part
by the need for the task of producing consistent and reliable annotations at scale to be
one that is tractable—if an annotation scheme becomes too complex, then it is unlikely
that two annotators will produce the same annotation for a given input. The other
factor shaping the reduced linguistic fidelity of many treebanks is the downstream
tasks they are used for. In the context of using a treebank for inducing a statistical
parser from a treebank, the target syntactic structures must be both learnable and
also result in models that are not of pathological complexity. In the context of
using a treebank to explore a linguistic phenomenon, an important consideration
then is whether the annotation scheme used to produce the treebank captures the target phenomenon. This is especially relevant in the context of this research, where the phenomena of interest are syntactic constructions that have the potential for a high degree of complexity. I discuss the deficiencies of PTB analyses for linguistic exploration in both Section 2.8.5 and Section 4.3.

In order to support the aforementioned applications, the annotations in a treebank must capture the target phenomena with sufficient reliability, meaning that quality control is a critical aspect of treebank construction. Comparing the results of multiple annotators annotating the same subset of the corpus is a common strategy to assess the reliability of the annotations (Artstein and Poesio 2008). If inter-annotator agreement is low, then this suggests that the annotation guidelines do not provide sufficient instruction to reliably capture the target phenomena, or the task itself is not feasible. Inter-annotator reliability is discussed further in Section 4.7, in the context of determining annotation quality for an annotated corpus of syntactic phenomena developed for this research. Even with such measures in place, it is inevitable that errors will be found within the final annotations; the analysis of the impact of such errors and approaches for remedying them has been addressed by a range of authors (e.g. Dickinson and Meurers 2005, Boyd et al. 2008).

The manual labour involved in the construction of treebanks means that their construction is usually resource-intensive. The creators of the PTB, for example, estimate that a team of five experienced annotators working three hours a day could annotate 2.5 million words of text a year with the syntactic annotations used in the initial version of the PTB (Marcus et al. 1993). It is for this reason that only a three million word subset of the approximately seven million words in the complete PTB were annotated with syntactic structure. A further issue with the use of treebanks constructed in the manner described here is that it can be difficult to update the annotations. The manual nature of the original annotations means that annotation updates often must also be performed manually, with each round of updates potentially requiring a manual pass of the entire treebank. In Section 2.6.5, I discuss the Redwoods dynamic treebanking methodology (Oepen et al. 2004), a semi-automatic approach to treebanking, which uses the output of precision grammars to form the annotations found in the treebank. This methodology reduces the amount of time involved in creating and especially in updating treebanks.

2.5.4 Summary

Treebanks are valuable linguistic resources for training statistical language models, for performing linguistic hypothesis testing, and—for the purposes of this research—represent a compelling resource for discovering examples of syntactic phenomena. This value is particularly tied to their capacity to support the retrieval of trees containing positive instances of syntactic phenomena through the use of query tools. A significant limitation of these query tools, however, is that users must be familiar with
the particular query language of tool, in addition to being required to work out how the grammatical phenomenon is represented in the treebank. Alternative approaches to querying for grammatical phenomena which attempt to overcome these barriers are discussed in Section 2.9.3. Two other limitation of treebanks I identified is the large amount of resources required for their construction and modification, in addition to their simplified nature when compared with the types of analyses produced by formal linguistic grammars. In Section 2.6.5, I discuss an alternative methodology for performing treebanking that uses precision grammars, which addresses both these limitations.

2.6 Grammar Engineering

Grammar engineering is the practice of developing and maintaining elaborate hand-crafted linguistic models. These models take the form of implemented computational grammars and are typically informed by a particular grammatical framework developed within formal linguistics. Like models developed by formal linguists, they aim to explicitly encode our understanding of different languages using systems of rules. They differ from models typically developed within formal linguistics in two notable ways. The first is that grammar engineering involves the implementation of computational models with machine readable formalisms that allow parsing software to automatically apply the constraints encoded by the model in order to parse a given input. Formal linguists, in comparison, typically derive the results of applying a model to a given input manually. The second difference is that grammar engineering involves the development of holistic language models that map input sentences all the way through to representations of their syntactic, and frequently, semantic structure. In order to do this, such models must account for the morphological, syntactic, and semantic phenomena within an input sentence, all of which interact together. Models developed by formal linguists, on the other hand, tend to target individual phenomena—or a limited range—and often at a particular level of linguistic abstraction (Bender 2008b).

Grammar engineering approaches are often referred to as deep linguistic processing methods by virtue of the fact that they produce rich analyses capable of modelling complex grammatical phenomena and also because they frequently target the generation of the underlying predicate-argument structure of input sentences. This is contrasted with shallow linguistic processing techniques which produce simpler representations of syntactic structure and are limited in the scope of grammatical phenomena they can capture. Shallow parsing approaches may make use of simple rule-based techniques, such as regular expression patterns over part of speech tags, but also frequently involve the use of stochastic models trained over corpora of pre-analysed text. At an intermediate level of linguistic modelling lie treebank parsers trained over
large amounts of syntactically annotated text, such as the Penn Treebank\textsuperscript{35} (Marcus \textit{et al.} 1993). Implemented models produced through the practice of grammar engineering are sometimes referred to as \textit{precision grammars}—the term I use to refer to them in this document—since they prioritise linguistic fidelity over robustness and focus on developing comprehensive and accurate coverage over a small fragment of a language before expanding to cover further grammatical phenomena—as contrasted with a breadth-first style strategy of development.

From the perspective of this research, precision grammars are appropriate language models for enhancing descriptive grammars with features involving syntactic discovery for a number of reasons. Firstly, as implemented computational grammars, they are capable of parsing and annotating naturally occurring language with candidate syntactic analyses. Compared with other approaches to automatically parsing language for syntactic structure, such as treebank parsers, precision grammars stand out due to their capacity to successfully model more complex syntactic phenomena. Precision grammars are also much closer to descriptive grammars in terms of their linguistic underpinnings, as they both represent explicit linguistic hypotheses of the workings of the language they model. I discuss these two important properties of precision grammars in Section 2.6.1. Another pertinent property of precision grammars is that they can be used to produce high quality treebanks. In Section 2.5, I outlined the value of treebanks for syntactic phenomenon discovery. Treebanks produced by precision grammars have the advantage of offering more consistent annotations than those annotated manually (Bender \textit{et al.} 2015). Furthermore, through the use of dynamic treebanking methodology, which I discuss in Section 2.6.5, the annotations produced by precision grammars are much less costly to update than manually constructed treebanks.

There exist a range of different projects producing hand-crafted precision grammars, each of which invokes a specific syntactic framework as part of the design of their linguistic models. Some notable examples include the CoreGram project (Müller 2015), which develops precision grammars that are based on Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag 1994), the ParGram Project (Butt \textit{et al.} 2002), which develops precision grammars based on Lexical Functional Grammar (LFG; Bresnan and Kaplan 1982), and the XTAG Project (Doran \textit{et al.} 1994), which develops lexicalised tree adjoining grammars (Joshi and Schabes 1997).

The precision grammars used in this research originate from the \textsc{delph-in} consortium,\textsuperscript{36} a distributed collaboration involving multiple research sites from around the world, dedicated to developing open source deep linguistic processing resources for mapping input strings into their corresponding semantic representations. The grammars produced within this collaboration are based on the grammatical framework of HPSG and target a joint reference formalism for implementation, which enables the

\textsuperscript{35}A frequently used syntactically annotated corpus within computational linguistics, discussed further in Section 2.5.3.

\textsuperscript{36}http://www.delph-in.net
use of a shared toolchain for text processing and grammar development. The output semantic representations produced by the grammars target the Minimal Recursion Semantics formalism (MRS; Copestake 2005). The main DELPH-IN grammar used in this research is the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011), a wide-coverage grammar that has been under development for over two decades, and which I outline further in Section 3.2. I identify some of the grammar engineering tools used in the DELPH-IN consortium in Section 2.6.2, and a more in-depth discussion of the structure of DELPH-IN grammars is reserved for Section 5.3.

In the remainder of this section I outline some of the features of the practice of grammar engineering in order to unpack the advantages precision grammars have to offer for the task of syntactic phenomenon discovery and also to provide a basis for understanding the challenges involved in using them for this application.

### 2.6.1 Applications

Grammar engineering has both practical and theoretical value. In this section I briefly outline two broad types of applications for precision grammars: engineering tasks that involve natural language processing and the testing of hypotheses formed in theoretical linguistics. Outlining the motivations for the development of these resources serves to highlight the characteristic capacities of precision grammars that attract people to their development, and which are therefore reflected in the artefacts of the grammars themselves—artefacts which this research aspires to improve the discoverability of.

**Natural Language Processing**

Treebank parsers have proven effective for a range of natural language processing applications, however, there are characteristic applications that they can struggle with. These include tasks which involve a strong reliance on accurately extracting the propositional content of a sentence, such as information extraction and question and answering, in addition to natural language generation tasks, such as found in machine translation and automated response systems (Baldwin et al. 2007). Common to these types of application is the need for precise identification of the underlying predicate-argument structure—i.e. \textit{who did what to whom?} (Oepen et al. 2002a). Parsers induced from treebanks can extract the semantic information for simple sentences but often struggle with accurately analysing sentences containing more complex grammatical phenomena (Rimell et al. 2009, Bender et al. 2011b).

One type of grammatical phenomenon that can be a challenge for predicate-argument structure extraction is the family of syntactic constructions known as \textit{long-distance dependencies}. These constructions are broadly characterised by the occurrence of a dependency between constituents which crosses one or more clausal boundaries (Huddleston and Pullum 2002:1080). (21) shows a range of sentences that
feature this property. Common to all of them is a characteristic gap which is filled by a constituent elsewhere in the sentence. The gaps and their corresponding filler constituents are indicated with a subscripted \( i \) and the scope of embedded clauses is indicated by square brackets. In (21a), for example, the gap is located within the embedded clause, where the complement of the verb \( be \) would be found in the equivalent independent clause, and the constituent that fills this gap is located at the start of the sentence outside of the embedded clause. These constructions are challenging to parse due to the dependency they introduce between the filler constituent and another clause containing the gap, with there being the potential for an unbounded number of clause boundaries to occur between.\(^{37}\) This is highlighted by (21c) and (21e), where the gap that is being filled is located within an embedded clause that is also embedded within another clause. In order to determine the correct predicate-argument structure, it is necessary to identify and link the filler with the gap, which the involvement of nested clause boundaries makes more challenging.

\((21)\)  
\(\begin{align*}
\text{a. A great singer}_i [\text{you will never be } _i]. \\
\text{b. This example sentence}_i [\text{the audience might enjoy } _i]. \\
\text{c. Who}_i [\text{does she think } [_i \text{ ate the chicken}]]? \\
\text{d. This is the song}_i [\text{I've been talking about } _i]. \\
\text{e. This is the song}_i [\text{you might recall } [\text{I've been talking about } _i]].
\end{align*}\)

Constructions such as these are typically too complex to capture reliably using shallow parsing approaches and, in the case of statistical parsers that are capable of capturing this complexity, they have often been explicitly ignored due to the increased complexity of the statistical model their inclusion would incur (Rimell et al. 2009). This can be demonstrated with (21b), which is an example of topicalisation, in which a constituent is located at the front of the sentence, in order to place focus upon it, rather than in its unmarked position further to the right. This results in a long-distance dependency, with, in this case, the noun phrase \textit{this example sentence} forming the constituent that fills the gap found in the incomplete clause \textit{the audience might enjoy}. A precondition for a complete and accurate analysis of this sentence is the successful identification of the clausal boundary found in this less frequently occurring sentence structure. Figure 2.7 presents two different analyses for this sentence, with Figure 2.7a showing the analysis produced by the Stanford PCFG Parser (Klein and Manning 2003),\(^{38}\) which uses a probabilistic context free grammar, and Figure 2.7b, showing the analyses produced by the 1214 release of the ERG. The Stanford Parser’s analysis has failed to identify the clausal boundary between the filler noun phrase and the clause containing the gap, instead combining the two noun phrases as an appositive construction. Under this syntactic analysis, the recovery of the correct predicate-argument structure is not possible. The ERG, on the other

\(^{37}\)It is for this reason long distance dependencies are also referred to as \textit{unbounded dependencies}.

\(^{38}\)This analysis is produced using version 3.7.0. of the Stanford Parser.
hand, does analyse this construction appropriately, making use of a rule specifically
designed to license head-filler constructions. Characteristic of this analysis is the use
of a slash feature, which allows the grammar to construct phrases which are still
“looking for” a constituent that can be satisfied from outside of the containing clause
(Pollard and Sag 1994). In the context of discovering instances of topicalisation within
a corpus, it is clear that the ERG would be a more appropriate resource than the
Stanford parser for locating examples of this construction.

While the enhanced linguistic fidelity of precision grammars allows them to more
accurately capture the propositional content of language, the hand-crafted constraints
which give rise to this improved precision also incur some costs. A general problem
associated with deep linguistic processing methodologies employing rule-based ap-
proaches is a lack of robustness, where noisy input and slight grammatical deviations
from the constraints explicitly encoded by the grammar writer often result in total
parse failure and no resulting analysis (Uszkoreit 2002).

Stochastic models, on the other hand, are much more resilient in the face of
noisy input, generally yielding some analysis—even if inaccurate—rather than none
at all. This is an important consideration in the case of natural language processing
applications involving a parser embedded in a pipeline processing user input, where a
parse failure causes the entire pipeline to fail. In the context of syntactic phenomenon
discovery, issues of robustness are a concern insofar as the coverage of a grammar
and noise found in particular corpus will impact the amount and type of candidate
sentences for retrieval. Provided such factors are appropriately managed, robustness
is less of a concern as it is in the case of embedding a precision grammar in a user-
facing application, where a failure of the text processing pipeline would be considered
a poor user experience.

Another drawback inherent to precision grammars is their resource-intensive na-
ture. The development of the hand-crafted linguistic constraints found within pre-
cision grammars requires grammar engineers to be familiar with the formalism the
grammar is implemented in and have access to sufficiently detailed descriptions of the
language being modelled. Extending the grammar to cover new grammatical phenom-
ena or adapting it to a new domain requires hands-on development to add support
for the requisite additional constraints, which can be a time-consuming process. Sta-
tistical parsers, on the other hand, can be readily trained over further annotated
data in order to extend their coverage over additional grammatical phenomena and
text characteristic of different domains—provided, of course, that such labelled data
exists. In general, for a given application, the selection of the underlying language
model depends upon the requirements of the task itself, the resources available to the
project, and also the amount of training data that can be leveraged.
Chapter 2: Background

Figure 2.7: Two analyses produced for the sentence *This example sentence, the audience might enjoy.*. The Stanford Parser’s syntactic analysis does not support the extraction of the long-distance dependency, whereas the ERG’s does.
Linguistic Hypothesis Testing

Grammar engineering is also valuable for the purpose of performing linguistic research (Bender 2008b, Bender et al. 2011a). The construction of implemented computational grammars—each instance of which corresponds to a specific theory of an individual language—can assist with the exploration and validation of hypotheses formed within the field of linguistics. Bender (2008b) identifies two ways in which the use of implemented computational models facilitates linguistic hypothesis testing. Firstly, the models constructed by linguists tend to focus on a single level of abstraction, such as morphology, syntax, or semantics. The systems produced by grammar engineers on the other hand involve an end-to-end analysis from an input string to at least a full syntactic analysis if not also a semantic one. Grammar engineering therefore presents the opportunity for the holistic testing of implementations of these subsystems together, potentially revealing unexpected interactions that could go unnoticed when tested in isolation. Secondly, the use of a computational model means that a parser can automatically test the theory the grammar represents over large corpora containing potentially many thousands of examples of real-world language usage, which would be impossible to achieve through manual verification. This facilitates a form of data-driven experimental linguistics (Abney 2011) that is capable of verifying—or falsifying—a specific hypothesis over a much more comprehensive range of grammatical phenomena and phenomenon-variants, notably including those which occur with a much lower frequency and might therefore go otherwise untested (Bender 2008b, Abney 2011).

Grammar engineering can facilitate the testing of a number of different kinds of linguistic hypotheses. The most intuitive form involves testing the validity of an analysis of a specific grammatical phenomenon. Given a precision grammar containing the implementation of grammatical phenomenon to be investigated, this can be performed by parsing a range of sentences known to contain the phenomenon with the grammar and determining, firstly, whether it yields an analysis for each sentence, and if so, then assessing the correctness of each analysis. Bender et al. (2011a) describe an example of this process during the development of the Jacy Japanese grammar (Siegel et al. 2016), a precision grammar produced within the DELPH-IN consortium. An analysis of number names in Japanese was developed based on an existing analysis in English, which sees number names like hundred function as heads, selecting for complements and specifiers, corresponding to lower and higher orders of magnitude respectively. Jacy was extended to include an implementation of this analysis, and after testing the grammar over examples of the phenomenon, it was discovered that the analysis was lacking a rule to combine heads with complements in the head-initial order required for number names.

Another form of linguistic hypothesis testing involves the exploration of the comparative impact that competing analyses of a grammatical phenomenon have upon other parts of the grammar. Bender (2010), for example, explores the downstream
ramifications of two distinct approaches towards analysing the auxiliary verb system of the Australian language Wambaya, each of which involved considerable re-working of the architecture of the grammar. This style of hypothesis testing involves evaluating the impact that the selection of alternative analyses of auxiliary verbs has as analyses of further phenomena are added, each of which has the potential to interact with the selected analysis of auxiliary verbs in different ways. The grammar engineering methodology makes this type of investigation possible through the exploration of divergent paths in the development of a holistic implemented language model.39

Finally, grammar engineering can also enable cross-linguistic hypothesis testing—the exploration of the validity of linguistic theories across multiple languages (Butt et al. 2002, Bender 2008b, Bender et al. 2010). I discuss grammar engineering projects that aim to facilitate cross-linguistic hypothesis testing in Section 2.6.6.

That precision grammars are well suited to the application of linguistic hypothesis testing is indicative of their capacity to model complex grammatical phenomena in the context of implemented holistic language models. This capacity contributes towards their suitability for the application of syntactic phenomenon discovery.

2.6.2 Engineering Requirements

In this section I outline the engineering requirements for effective development of a grammar engineering project. This serves to provide a foundation upon which to base further discussion of precision grammars, in particular, providing the context for some of the methodological decisions made in Chapter 4 and Chapter 5. The engineering requirements are adapted from Bender (2008b) and Bender et al. (2011a). For each requirement, I identify the corresponding tools used within the delph-in consortium, which places a strong emphasis on the use of shared formalisms and a shared toolchain.

A Well-defined and Stable Descriptive Formalism

As per Bender (2008b), it is helpful to distinguish between the formalism—the language in which analyses are implemented; the theory—a collection of analyses; and the grammatical framework—an overarching approach to developing analyses of grammatical phenomena. The formalism used for implementing grammars should be well-defined and stable, so that it can be safely targeted by multiple software processors. Given this need to fix the formalism, it should also be expressive enough to support a diverse range of potential theories. DELPH-IN grammars target a joint reference formalism (Copestake 2002a) which employs typed feature structures and

39See also Fokkens (2011)—which is discussed in Section 2.6.6—for a framework designed to support a grammar engineering methodology capable of supporting parallel analyses of grammatical phenomena.
Chapter 2: Background

multiple inheritance from a type hierarchy in order to support the capturing of linguistic
generalisations in accordance with certain conventions specified by the formalism.
The formalism is encoded using Type Description Language (TDL; Copestake 2002b),
which supports the definition of typed feature structures through the specification of
constraints over types and type inheritance. The use of a declarative formalism de-
dsigned to capture linguistic knowledge makes grammar development more accessible
to linguists who may not have programming experience. The delph-in joint reference
formalism is discussed further in Section 5.3.

Parsing and Generation Software

In order to interact with text, a precision grammar requires the existence of pro-
cessing software capable of parsing. During parsing, the processor takes as input a
string and a grammar defined with the formalism, and returns the linguistic struc-
ture of the string assigned by the grammar if an analysis is found. Depending on
the grammar, this may yield only a syntactic analysis or both a syntactic and se-
matic analysis. In the case of delph-in grammars, a single monostratal structure
containing the complete analysis is built up during parsing, from which both syntac-
tic and semantic representations can be extracted. Parsers also require support for
a number of preprocessing operations, including tokenisation, the separation of an
input string into discrete tokens, and part of speech tagging—the labelling of each
token with one or more candidate parts of speech. Part of speech tagging is necessary
for the grammar and parser to provide support for unknown word handling, which
enables the analysis of input containing tokens that do not correspond to any entry in
the grammar’s lexicon (Adolphs et al. 2008). DELPH-IN parsers provide configurable
tokenisation support and require the use of an external tool for part of speech tag-
gging. Some precision grammars and processors also support the generation of text.
In this case, the processor takes as input a semantic representation and a grammar,
generating from these the set of strings for which the input grammar yields the in-
put semantic representation. This enables the use of precision grammars for tasks
involving natural language generation.

Within the delph-in consortium a range of grammar processors are available. The Linguistic Knowledge Builder (LKB; Copestake 2002b) provides support for in-
teractive parsing and generation. More efficient batch parsing is facilitated by the
PET parser (Callmeier 2000), and efficient batch parsing and generation are both
provided by ACE (Packard 2011). A further delph-in processor, agree (Slayden
2012), provides parsing and generation for the Windows operating system. The exis-
tence of a range of different processors within the delph-in ecosystem would not be
possible without the stability of TDL, the joint reference formalism.
Chapter 2: Background

Grammar Development Tools

In the same way that software engineers make use of interactive development environments, effective grammar engineering requires a grammar development environment. This should include facilities for stepping through the application of derivation rules during parsing and generation in order to identify the causes of parse failure, as well as tools for visualising the structures a grammar assigns to an input string. In the DELPH-IN ecosystem, these facilities are provided through the LKB, ACE, and agree.

Testing Software

In order to manage the complexity inherent to precision grammars, diagnostic tools are needed for gauging the overall health of the grammar, detecting regressions in the coverage or accuracy of the grammar, and assisting in locating the cause of the regression. This can be achieved through the use of tools which manage test suites and facilitate their automatic parsing and aggregation of results. The nature of these test suites is discussed further in Section 2.6.4. The primary tool used within the DELPH-IN ecosystem for test suite management and regression testing is [incr tsdb()]\(^{40}\) (Oepen and Flickinger 1998, Oepen and Carroll 2000). This tool exposes summary statistics, such as the number of test items that parsed (the coverage), and the average number of analyses produced per test item (the ambiguity), as well as more specific item-based information. Since this information is generalisable to any parsing system that assigns one or more analyses to an input string, other grammar engineering projects outside of DELPH-IN have also made use of [incr tsdb()]\(^{41}\) for managing test suites.

Tools for Managing Ambiguity

Precision grammars are typically capable of assigning multiple structures to a given input string, each corresponding to a different analysis. For a given sentence and a grammar with interesting coverage, it is possible for the set of analyses generated by the parser—referred to as the parse forest—to be numbered in the thousands or even millions. Some of these analyses are spurious, arising from the grammar being underconstrained, and can therefore be pruned from the parse forest through the addition of constraints to the grammar. Many of the analyses within the parse forest, however, arise from the inherent ambiguity of language, which was discussed in Section 2.3.3, and therefore cannot be removed from the grammar. Sometimes this type of ambiguity cannot be resolved without knowledge of the context of utterance,\(^ {42}\)

\(^{40}\)Pronounced ‘tee ess dee bee plus plus’.

\(^{41}\)For example, the CoreGram Project (Müller 2015).

\(^{42}\)For example, sentence I bought the medpack with a kimble is ambiguous between at least two readings until it is known whether, in the context of utterance, the term kimble refers to a form of currency or a piece of equipment expected to be found amongst medical supplies.
however many of these analyses can be ruled out through sentence-internal context combined with an understanding of how the words involved are frequently used. Given that the most appropriate analysis may be buried somewhere within the sizeable parse forest, it is vital that there exist strategies for managing this ambiguity in order for precision grammars to remain true to their moniker.

An important strategy that is often used to manage this ambiguity is the use of a parse selection model, a stochastic model which is trained over gold standard analyses and used to rank analyses in the parse forest according to their likelihood of being the preferred analysis (Toutanova et al. 2002). Selecting the best analysis according to the parse selection model doesn’t guarantee the selection of the intended analysis, but makes it much more likely and, furthermore, can be performed efficiently as part of the parsing process, making it suitable for use in real-time applications. The parse selection model used within DELPH-IN parsers makes use of a maximum entropy model (Zhang et al. 2007). Within DELPH-IN processors, the memory footprint of the parse forest is decreased through the use of ambiguity packing, which allows tree structures that are shared across analyses in the parse forest to re-use the same data-structure. The incorporation of the parse selection model into the parsing process also allows for selective unpacking of this packed representation, thus improving both memory and CPU usage (Zhang et al. 2007).

In order to train this kind of parse selection model, it is necessary to have a disambiguated treebank of gold standard analyses which contains the grammar’s best analysis from the parse forest of each sentence. The use of traditional treebanking methodologies (discussed in Section 2.5), would be prohibitively expensive for this task, given that the parse selection model must be retrained when the grammar is updated. The use of a dynamic treebanking methodology (Oepen et al. 2004), however, which is discussed further in Section 2.6.5, and which has been developed for use in the context of grammar engineering methodologies, facilitates the synchronisation of the gold standard analyses with the current state of the grammar, making ongoing maintenance of such a resource feasible. In addition to supporting parse-selection models, this process of disambiguating grammar output also provides a valuable means of tracking the quality of the analyses being produced by the grammar.

2.6.3 Workflow

Bender et al. (2011a) provide a comprehensive overview of the workflow of a grammar engineering project. In this section I summarise this overview, providing a condensed picture of a prototypical methodology for grammar engineering development. This will serve to highlight the salient role that grammatical phenomena play in the development of precision grammars, while also underscoring the significant complexity involved in their construction.

Once a target language is chosen, a grammar engineering project typically begins with the selection of some initial linguistic test data, which might include natu-
rally occurring text taken from one or more domains, as well as a selection of hand-
constructed examples that illustrate core grammatical phenomena, such as word or-
der. This test data will be used to track the coverage of the grammar and also for identifying candidate grammatical phenomena to extend the grammar to cover. A minimally-functioning grammar that is capable of assigning structures to short, grammatical sentences of the target language must first be established, after which, development proceeds in an iterative fashion, on a phenomenon-by-phenomenon basis.

An iteration begins with the identification of a grammatical phenomenon to tar-
get, followed by the development of an analysis of the phenomenon using linguistic references and other available resources. During this process, examples pertinent to the phenomenon are collated into a phenomenon-specific test suite. The analysis is then implemented and incorporated into the grammar, a process which often results in fine-tuning of the analysis and which may also require some iterations of debugging until the grammar is accepted as valid by the parser. The grammar engineer then parses the phenomenon-specific test suite and inspects the results, checking that all grammatical test items successfully parsed, ungrammatical ones did not, and that the parser output for grammatical items corresponds to the intended analysis. This likely results in modifications to the implementation and possibly also the underlying analysis, followed by further inspection of the parser output, with this process being repeated until the grammar engineer is satisfied.

Given that the overall goal is to develop a linguistically precise model of a poten-
tially wide portion of the language, the accuracy of the current phenomenon being im-
plemented cannot come at the expense of the accuracy of phenomena already covered by the grammar. As discussed in Section 2.3.5, grammatical phenomena interact, and it is unsurprising then, that so too do their implementations. A characteristic compo-
nent of the grammar engineering development cycle is establishing that a change to a grammar has not resulted in deleterious effects on the overall health of the grammar.

As a first step in quality control, the grammar engineer can parse the full set of test data—which may include previously constructed suites and larger test corpora of naturally occurring text—using the modified grammar. These results are then compared with a baseline that was recorded before the change was made. Of particular importance are changes found within signals pertaining to the analyses produced by the grammar, such as increases or decreases in numbers of items receiving an analysis, the average number of analyses per sentence, and the number of items whose assigned structure (syntactic or semantic) has changed. Also important are signals of a more pragmatic nature, such as processing time, amount of memory used, and average number of operations required per sentence. Dramatic or unexpected changes to any of these signals may be indicative of problems associated with the grammar modifications. This comprehensive approach to the monitoring of precision grammar diagnostics has been referred to as competence and performance profiling (Oepen and Flickinger 1998).

While this diagnostic approach provides coarse-grained insight into changes in
parsability, it does not provide any illumination into the nature of these changes. The highly interconnected nature of grammatical phenomena means that it cannot be assumed that any changes in the analysis of sentences found in the larger test data will mirror the desirability of changes seen in the manually inspected items. One strategy to verify that changes constitute enhancements rather than regressions is to compare modified analyses with previously vetted analyses from before the change on an item-by-item basis. When an issue is discovered, it is corrected, if feasible, and the diagnostic process is started afresh. If no issues are discovered, the stored analyses are updated with the newly vetted analyses. This strategy is enabled by a dynamic treebanking methodology—described in Section 2.6.5—which is dependent on treebanking tools that reduce the number of comparisons required of the grammar engineer. Even with such tools, the potentially large number of decisions required of the grammar engineer means that the treebanking stage is typically reserved for last, so that more readily identifiable problems are identified first.

When compared with traditional approaches to syntactic modelling within linguistics, the grammar engineering methodology stands out as having a strong empirical basis, as evidenced by its data-driven approach to development, which involves the extensive use of curated test suites and large corpora of naturally occurring text. This is made possible through the use of computational modelling, and furthermore, can be performed at scale due to improvements in parsing algorithms and hardware performance, combined with the fact that sentence-based parsing is readily amenable to parallelisation, with each test item being able to be processed independently. An important aspect of this empirical approach to hypothesis testing, which is characteristic of the grammar engineering approach, is the verification of a candidate analysis of a grammatical phenomenon, not in isolation, but in conjunction with the existing analyses of all grammatical phenomena covered by the grammar so far.

This outline of the grammar engineering process, with its holistic approach to grammatical analysis and its firm empirical grounding, seen especially in its data-driven approaches to quality assurance, provides a sense of the kind of linguistic accuracy precision grammars aspire to attain, and indeed, is required for their use in linguistic hypothesis testing. This accuracy makes them particularly good candidates for their use in the discovery of syntactic phenomena.

### 2.6.4 Linguistic Test Suites

When developing a descriptive or formal analysis, a requirement is the use of test data for performing the analysis and testing its validity. In this section I discuss the test data used in grammar engineering, focusing in particular on linguistic test suites, which are often oriented around grammatical phenomena. In addition to further highlighting the prominent role of grammatical phenomena within the development of precision grammars, linguistic test suites offer a potential source of phenomenon-labelled sentences for evaluation of syntactic discovery methods, as is discussed in
Section 4.3, and so it is necessary to briefly describe them here.

As already established in Section 2.6.3, linguistic test data used in grammar engineering projects can come in the form of test suites and test corpora. Test suites are generally smaller collections of sentences that have been curated for monitoring a specific dimension of grammar coverage or grammar health, such as coverage over a specific grammatical phenomenon. Once created and incorporated into the project, test suites become indispensable for regression testing, as discussed in Section 2.6.3.

Test corpora, on the other hand, are larger collections of naturally occurring text, often pertaining to a specific domain. They play an important role in determining the coverage of a grammar over a particular textual domain, as well as potentially uncovering specific issues associated with real-world language use from anywhere in the text-processing pipeline—such as missing lexical entries, incorrect tokenisation, and infrequently occurring syntactic constructions—that might have otherwise gone unnoticed from the inspection of constructed test suites alone. Baldwin et al. (2005), for example, parse the British National Corpus with the ERG, classifying each parse failure and determining that the most frequent causes of failure were predominately missing lexical entries and missing constructions. Dost and King (2009) also describe a methodology for using corpus data to improve the coverage of a precision grammar, using a tool to search for syntactic patterns in parser output that were identified as likely having been misanalysed by the grammar. In both cases the resultant insights could not have been gleaned from the use of test suites alone.

Test suites and test corpora mirror a distinction in the type of data used to develop linguistic theories, namely, between the use of native speaker judgements and corpus data, with there being an established debate over the appropriateness of these two sources of data for performing grammatical analysis.\(^{43}\) A notable criticism of native speaker judgements—which can be extended to test suites due to their frequent inclusion of grammaticality judgements—is that they are limited by the imagination and creativity of the linguist soliciting or constructing the examples. Corpus data, on the other hand, is criticised for being deficient in the capacity to fully illuminate the boundaries of the grammatical phenomena under investigation, since it is limited to whatever instances of the phenomena happen to occur within the initially selected data. This contrasts with collections of native speaker judgements, which can be intentionally constructed to systematically cover a slice of the phenomenon space through the use of positive and negative examples—with the latter notably being absent from corpus data. In the context of precision grammar development, Baldwin et al. (2005) argue that these are complementary sources of evidence, with grammaticality judgements and test corpora both playing an important role in driving the development of a grammar: grammaticality judgements—in the form of test suites—offer immediate and targeted feedback during the implementation of an analysis of a

\(^{43}\)This is discussed further in Section 4.2.1. See also Fillmore (1992) for an overview of this debate.
grammatical phenomenon, whereas test corpora provide an important diagnostic resource for ascertaining the coverage of the grammar, as well as directing development towards phenomena not yet covered, which might include subtle variants of phenomena unanticipated by those constructing the test suites. From the perspective of evaluation, test corpora also provide an opportunity to evaluate precision grammars over real-world data that is more likely to resemble input seen in practical applications. On the other hand, Cohen et al. (2008) found that within NLP systems, the testing of code coverage—the degree to which the source code of a program is used during its execution—is optimised through the use of curated test suites, rather than test corpora.

Both test suites and test corpora are also valuable from the perspective of regression testing, which is a critical component in the maintenance of precision grammars, whose inherent complexity means that minor changes can have unforeseen consequences to overall performance. The use of test suites, in particular, presents the opportunity for targeted regression testing, through the use of careful curation. A notable way this is performed in the context of grammar engineering is through the isolation of distinct grammatical phenomena across test items in order to assist grammar engineers to more rapidly identify the origin of an introduced parse failure or misanalysis. One of the most well-known phenomenon-based linguistic test suites is the Hewlett-Packard English test suite (Nerbonne et al. 1988), which contains a variety of syntactic and semantic phenomena, divided into core and peripheral phenomena. The TSNLP project (Lehmann et al. 1996, Oepen et al. 1997), which is discussed in more detail in Section 2.8.1, involved the development of a methodology for rigorously isolating target phenomena and used it to construct test suites for English, French and German. The MRS test suites developed for delphin grammars target a range of core semantic phenomena, whose resulting semantic representations are considered stable, with any changes therefore likely indicating a regression. Another project, the PARC Bridge question-answering system, which employed a precision grammar, made use of a range of syntactic and semantic phenomenon-based test suites (de Paiva and King 2008).

A number of best-practices have emerged through the development of these resources, in particular, from the TSNLP project (Lehmann et al. 1996, Oepen et al. 1997): individual test items should be kept to the minimum length required to illustrate the target phenomenon in order to avoid introducing sources of noise via other phenomena. The use of a controlled vocabulary—and, if appropriate, a controlled set of sentence structures—also helps to increase the likelihood that problems encountered with the items pertain to the target phenomenon. It is also important to avoid genuine ambiguity where possible, so as to not introduce additional complexity in the parser output. Finally, the use of negative items, in the form of ungrammatical input, is an indispensable strategy for ensuring that the grammar does not

---

44http://moin.delph-in.net/MatrixMrsTestSuite
overgenerate—producing analyses for input it should not. (22) shows a fragment of a test suite designed to map out grammatical and ungrammatical permutations of English interrogatives.\footnote{This is taken from a small English grammar fragment released with the LKB, which is intended to illustrate a set of analyses of interrogative clauses.}

\begin{enumerate}
\item Who did Sandy see?
\item *Who Sandy saw?
\item Who did Sandy think she saw?
\item I wonder who Sandy saw.
\item *I wonder who did Sandy see.
\item Who saw Kim?
\item *Who did see Kim?
\item Who can see Kim?
\end{enumerate}

Regression testing can be performed at the level of item parsability or also at the level of representations produced by the parser. As discussed in Section 2.6.3, the latter offers a superior degree of quality control, as it allows regressions within the generated analyses to be identified. A prerequisite for this level of regression testing is that for each test item, one of the many potential analyses returned by the parser must be selected and stored as the current best analysis yielded by the grammar whenever a change to the grammar impacts the analysis. A methodology for efficient disambiguation and treebanking of gold standard analyses produced by precision grammars is described in the next section.

\subsection*{2.6.5 Dynamic Treebanking}

Treebanks produced by precision grammars which contain gold standard analyses are valuable resources for a range of applications. As already discussed, disambiguated treebanks constitute an important diagnostic tool for the grammar engineer, enabling the detection of regressions within the analyses produced by the grammar. They also allow individual test items to target specific phenomena, as without the use of a disambiguated treebank, the ambiguity within parse results would frequently mask the intention behind each test item (Toutanova \textit{et al.} 2002, Bender \textit{et al.} 2011a). Precision grammar generated treebanks are also valuable in their own right as linguistic resources, due to their having the potential to capture precise representations of complex grammatical phenomena, as well as their increased consistency when compared to hand-annotated treebanks (Bender \textit{et al.} 2015). As precision grammars are commonly subject to continuous improvements, however, for their treebanks to remain usable for these purposes, their contents must be periodically synchronised with the analyses produced by the current state of the grammar. In this context, the treebanking task
becomes the selection of the best analysis produced by the grammar from within the parse forest of each sentence. Manual selection from the potentially large number of analyses within each parse forest would, however, be impractical. This problem is overcome by the dynamic treebanking methodology used to construct the Redwoods Treebank (Oepen et al. 2004). In this section, I briefly describe this methodology, as it differs substantially from that involved in traditional treebanks.

Under the Redwoods dynamic treebanking methodology, rather than directly selecting the gold tree, annotators are presented with a list of syntactic discriminants (Carter 1997, Oepen et al. 2002b) for each sentence that requires disambiguation. Each discriminant corresponds to local ambiguity within the parse forest and is composed of a syntactic derivation node label46 along with the surface string of the corresponding subtree. Annotators must mark discriminants as either good or bad, depending on whether the relevant subtree is identified as being ruled in or out of the desired analysis. After each discriminant is marked, the treebanking software automatically infers and filters out the remaining discriminants that are incompatible with the currently marked discriminants. This has the effect of implicitly pruning the parse forest, potentially discarding a large number of unwanted analyses based on a single decision. This syntactic discriminant approach to treebanking greatly reduces the time to locate the best tree from the parse forest, with the number of decisions required to disambiguate the parse forest of a sentence being proportional to the logarithm of the number of analyses it contains (Tanaka et al. 2005).

The dynamic aspect of this methodology comes into play when a grammar is updated and used to re-parse the contents of the treebank. Rather than having to repeat the treebanking process from scratch for each sentence, the local disambiguation decisions that an annotator made during the previous round of annotation for a given sentence are automatically applied where the same ambiguity exists in the parse forest. This results in many sentences being fully disambiguated and some requiring only a small number of additional decisions, such as when a new construction is added to the grammar or previously unseen ambiguity is created. This means that even for relatively large treebanks—the current version of the Redwoods Treebank consists of around 45 thousand sentences47—synchronisation of the annotations to the current version of a grammar becomes a manageable update task and the treebank is not locked to a particular version of the grammar. This represents a distinct advantage over traditional treebanks (Flickinger et al. 2017), which, as discussed in Section 2.5.3, typically rely on a component of manual annotation, making them difficult to update, as changes to the annotation scheme must be applied across all trees, requiring an additional round of potentially laborious manual annotation for each set of changes.

In addition to the Redwoods Treebank, another notable treebank produced using this dynamic methodology is the DeepBank (Flickinger et al. 2012) treebank. I use

---

46It is also possible to use semantic discriminants, as described in Oepen and Lønning (2006).
Chapter 2: Background

2.6.6 Multilingual Grammar Engineering

The treatment of grammatical phenomena is clearly a central concern to all grammar engineering projects, however there are some projects where an even greater emphasis on modelling grammatical phenomena may be found. In this section, I discuss one such candidate type of grammar engineering endeavour: multilingual grammar engineering projects. These projects involve the construction of grammars across different languages and use this task to engage in a form of cross-linguistic hypothesis testing. Of particular interest to these projects is how the analyses of grammatical phenomena are represented and constrained across grammars of different languages and the extent to which analyses can be shared across languages—especially across diverse language groups.

All grammar engineering projects must clearly be concerned with how grammatical phenomena are modelled within the grammar architecture to some extent, however it is likely that the aspiration of multilingual grammar engineering projects to model grammatical phenomena cross-linguistically will be reflected in some way within design principles and patterns of the implemented grammars. I hypothesise that, within multilingual grammar engineering projects, grammatical phenomena may become further reified than they might in a monolingual grammar engineering project. This is the first reason why such projects are of potential relevance to this research: if the hypothesis that such projects are more likely to have tell-tale traces of grammatical phenomena within the structure of their grammars is true, these projects could potentially be leveraged directly for the purpose of syntactic phenomenon discovery. The second reason such multilingual grammar engineering projects are relevant to the present research is that in order to successfully explore cross-linguistic hypotheses pertaining to grammatical phenomena, these projects must inevitably provide some form of operationalisation of the abstract concept of a grammatical phenomenon in accordance with the theoretical aims of each project. Ascertaining how cross-linguistic abstractions manifest within these projects therefore represents an opportunity to contrast the definition of grammatical phenomena that I put forth in Section 2.3 with the definition implied by each project’s operationalisation of grammatical phenomena.

In the remainder of this section, I briefly outline four projects, each of which involves a form of multilingual grammar engineering and also enforces some kind of analytic consistency across grammars developed within the project. In doing so, I assess the hypothesis that these projects result in the increased reification of grammatical phenomena within their architecture, in addition to identifying any notable differences between their cross-linguistic operationalisation of the notion grammatical phenomena and my definition.
ParGram

The ParGram project (Butt et al. 2002) involves the creation of a range of grammars across different languages, which conform to a consistent standard of representation for grammatical phenomena. The grammars produced are based on Lexical Functional Grammar (LFG; Bresnan and Kaplan 1982). One characteristic of LFG is the use of two levels of representation: constituent structure (c-structure) and functional structure (f-structure). The c-structure encodes word order and the phrasal structure of individual languages. The f-structure takes the form of a feature structure which encodes grammatical relations, syntactic features, and predicate-argument relations. The cross-linguistic hypothesis the ParGram project makes is that f-structure is a language universal and thus a particular feature geometry for representing f-structures should be discoverable which supports the analysis of all grammatical phenomena across every language. Development of the grammars occurs at different sites independently of each other, with f-structures being harmonised through the use of a shared feature space. This is controlled by a feature committee which approves the addition of new features and will only do so with sufficient evidence that the feature has universal applicability or is essential for the language motivating the addition (King et al. 2005). Compliance with the shared feature space is enforced by incorporating checks into the grammar compilation stage. This means that the cross-linguistic abstractions found within the ParGram project are found primarily within f-structure representation produced as parser output, rather than within the constraints—or rules—of the grammar. An artefact of the shared feature space is a feature table with each supported language as columns and all features as rows. Each cell then has a value with the possible values of the feature in the corresponding language. As argued by King et al. (2005), this provides a convenient means for determining which grammars cover specific grammatical phenomena as well as the type of analyses each grammar ascribes.

The Grammar Matrix

Another project that facilitates cross-linguistic hypothesis testing is the LinGO Grammar Matrix (Bender et al. 2002, Bender et al. 2010), which bootstraps new grammar engineering projects by generating small but functional DELPH-IN HPSG grammars of target languages (Bender 2008b). It does this by feeding the results of a typological questionnaire—filled out by a user—into its customisation system, which uses libraries of pre-analysed grammatical phenomena to generate a bespoke grammar of the target language. This makes the Grammar Matrix a metagrammar. The Grammar Matrix’s analyses are divided into a core grammar covering

---

48ParGram grammars do have the capacity for the sharing of cross-linguistic abstractions at the constraint description level, however this is predominately for pragmatic code-reuse purposes and is not enforced.
grammatical phenomena considered to be common to all languages and a set of phenomenon-specific libraries, which are selectively imported on the basis of the responses to the typological questionnaire. The contents of the core grammar represent hypotheses about shared constraints that are language universal—such as compositionality. Each phenomenon-specific library aims to account for the variation of that phenomenon across all languages consulted, with its semantic representation being harmonised across these languages. The aim is also for the analyses found within each phenomenon-specific library to be cross-compatible with analyses invoked by other phenomenon-specific libraries. This ability to explore the validity of predicted analyses for interactions between phenomena cross-linguistically amounts to another form of cross-linguistic hypothesis testing.

Unlike ParGram, where the primary form of cross-linguistic generalisation is found at the level of representation, the Grammar Matrix’s cross-linguistic abstractions are focused around the level of grammar constraints, with shared abstractions being captured by the type hierarchy of DELPH-IN grammars. Additionally, a further level of abstraction is found within the phenomenon-specific libraries, which sees grammar constraints from each library being grouped contiguously. This is in contrast with output grammars generated by the Grammar Matrix— and DELPH-IN grammars more generally—where the location of constraints is required to conform to a particular layout, more closely resembling divisions in analytic levels, such as syntax, morphology, and the lexicon, meaning that the group of constraints that pertain to a single phenomenon is often located non-contiguously. The grammars produced by the Grammar Matrix all initially share the same representation, insofar as the output feature structures representing both the syntactic and semantic analysis of a given input all share the same feature geometry. Unlike ParGram, where grammar development continues in parallel to some extent, development of downstream grammars produced by the Grammar Matrix continues independently, with adherence to this initial feature geometry not being enforced.

**CLIMB**

The CLIMB methodology for precision grammar development (Fokkens 2011) extends the Grammar Matrix methodology, with ongoing grammar development continuing at the metagrammar level. This contrasts with the Grammar Matrix, where ongoing development is performed directly on the generated grammars. Just as with the Grammar Matrix, CLIMB provides a level of abstraction that facilitates the organisation of grammar constraints by phenomenon, meaning that the problem of constraints often being located non-contiguously within the grammar can be circumvented. It is at this level of abstraction that the grammar engineer works, building the current version of the grammar from the metagrammar after each change is made in a fashion analogous to code compilation. A notable feature of CLIMB is that it supports the maintenance of multiple competing analyses, with the metagrammar
system providing the capacity to choose between collections of analyses. This facilitates the exploration of alternate analyses and how these interact with different analyses of other grammatical phenomena, which would otherwise be difficult to assess.\(^{49}\) Notably, this allows for the systematic exploration of how previously identified and subsequently dispreferred analyses for phenomena behave in light of additional evidence in the form of analyses of further grammatical phenomena (Fokkens and Bender 2013, Fokkens 2014).

CoreGram

The CoreGram project (Müller 2015) is a multilingual grammar engineering project that involves the development of HPSG grammars of various languages. Grammars are developed independently in a bottom up fashion, according to the available data of different languages, with linguistic abstractions being re-used heavily across the collection of grammars. When the data from the relevant language provides equal support for competing analyses, external data and analyses from grammars of other languages in the project can be used to select one. CoreGram is similar to the Grammar Matrix in that it has a core component of shared analyses, however the requirements for inclusion are different from the Grammar Matrix’s core: if more than one language has a shared analysis, then its implementation goes into the core. This bottom-up approach towards harmonisation of analyses also contrasts with the top-down approach found in ParGram’s feature committee. Since CoreGram does not use a metagrammar approach, the core library is directly shared by every CoreGram grammar. This means that each constraint found in the core component must be compatible with the current version of every grammar which invokes it. This makes development harder, as changes to a grammar of one language may often impact upon the grammar of another language, however, it also makes the resultant cross-linguistic generalisations stronger.

Discussion

Looking at the different multilingual grammar engineering projects discussed, in terms of how grammatical phenomena are represented, an initial observation regarding differences in the nature of cross-linguistic abstractions within in each project can be made. Within ParGram, the cross-linguistic abstractions are located within parser output, which is harmonised across different grammars. In CoreGram, on the other hand, the cross-linguistic abstractions are located at the level of grammar constraints, which are shared across all grammars in the project. The Grammar Matrix

\(^{49}\)Fokkens et al. (2012) discuss the application of the CLIMB methodology for exploring competing analyses in the context of Germanic languages, Slavic languages and Mandarin Chinese. This approach can be compared with that of Bender (2010), which explores the comparative impact of alternative analyses of a relatively core phenomenon within a grammar of Wambaya, however without the aid of the CLIMB methodology.
Table 2.3: Sample feature table from the ParGram project (King et al. 2005) that illustrates feature names and the different values the feature can take in grammars of different languages. TNS-ASP, which is a complex feature, is encoded instead by the feature names of the embedded feature structure.

and CLIMB differ again, both representing cross-linguistic abstractions at the metagrammar level. These distinctions can also be couched in terms of the difference between abstractions at the level of representation and description (King et al. 2005), with ParGram’s harmonised f-structures capturing abstractions at the representation level, and CoreGram, the Grammar Matrix, and CLIMB, on the other hand, all capturing constraints at the description level in addition to the representation level.

Turning now to the goal of comparing the cross-linguistic operationalisation of grammatical phenomena across the different projects with my proposed definition of a grammatical phenomenon, I firstly restate the definition I put forth in Section 2.3:

A grammatical phenomenon is a feature of language that can be described in terms of one or more linguistic forms, constraints over these forms, one or more linguistic functions, or some combination of these.

Of the cross-linguistic abstractions that the different projects outlined in this section employ, the ones that ParGram invoke stands out in particular as being difficult to reconcile with my definition of grammatical phenomena. Looking at the sample feature table in Table 2.3, which provides examples of how cross-linguistic abstractions within f-structure are captured across different grammars, entries in the table cannot be used to capture a group of different forms nor a group of different functions. This is due to the aforementioned characteristic of ParGram, that it aspires to capture cross-linguistic abstractions at the level of representation rather than description. ParGram aims to provide a language universal feature geometry that has the capacity to capture all grammatical phenomena, but it and of itself, does not actually capture any abstractions that describe classes of grammatical phenomena, as I have defined the term. The abstractions present in CoreGram, the Grammar Matrix, and CLIMB, on the other hand, are at least broadly compatible with my definition of grammatical phenomena.
Moving on to the other goal of investigating multilingual grammar engineering projects, this was to assess the hypothesis that, in modelling cross-linguistic abstractions, these projects would be more likely to feature reified structures within their architecture corresponding to grammatical phenomena. The motivation here being that, if this hypothesis held true, these grammars would then potentially offer additional affordances for use in the detection of grammatical phenomena. As already identified, the ParGram project’s cross-linguistic abstractions are found at the representation level, and do not extend to the description of grammatical phenomena per se, meaning that, at least for the notion of grammatical phenomena as I have described, this hypothesis does not hold for ParGram. In the case of CoreGram, its cross-linguistic abstractions are found at the level of grammar constraints, with the identification of shared constraints across grammars of different languages triggering the inclusion of these constraints within the core library. This bottom-up approach to the collection of linguistic abstractions results in the organisation of grammar constraints by groups of languages which make use of them. Under this approach, grammatical phenomena are modelled the same as they would be in a monolingual grammar, but their associated constraints are reused across languages where possible. The Grammar Matrix, on the other hand, employs a top-down arrangement of constraints by course-grained grammatical phenomenon families, with the association between individual constraints within these libraries being handled by the Grammar Matrix Customization System. The Grammar Matrix clearly confers an additional level of abstraction, with respect to grammatical phenomena, that is not found within DELPH-IN monolingual grammars, suggesting that, for this project, the grammatical phenomenon reification hypothesis being tested does hold true. However, since the Grammar Matrix generates new grammars, but does not support the development of ongoing grammars, it is ruled out as being a candidate for use in the detection of “interesting” grammatical phenomena, as it is limited to producing immature grammars. Turning to the remaining project, CLIMB, grammatical phenomena are indeed able to be rendered more concretely within its metagrammar architecture, since it facilitates the development of grammars that are organised by grammatical phenomena. This property, however, is contingent on the grammar engineer choosing to embrace this style of development, which CLIMB happens to be well-suited for. Furthermore, CLIMB does support the ongoing development of grammars, enabling the construction of wide-coverage grammars that include more “interesting” grammatical phenomena.

Of the multilingual grammar engineering projects identified in this section, CLIMB is then clearly the most likely candidate for being used to support grammatical phenomenon discovery by virtue of leveraging a phenomenon-rich design. In Section 5.2, I discuss the possibility of this approach, however, conclude that it would not be suitable for the motivating application of this research, namely, the augmentation of descriptive linguistic resources.
2.6.7 Summary

In this section, I have provided an overview of the practice of grammar engineering, which included an outline of: its applications (both applied and theoretical), engineering requirements, a prototypical development workflow, phenomenon-based test suites commonly used during the development of precision grammars, the dynamic treebanking methodology used to produce disambiguated treebanks annotated by precision grammars, and a brief survey of projects with the goal of producing grammars across different languages in support of developing cross-linguistic hypotheses.

In the course of this, I identified a number of properties of precision grammars that make them strong candidates for being used in the discovery of grammatical phenomena. Notably, precision grammars enable the automatic annotation of large amounts of text with representations of syntactic and semantic analyses, which would not be possible to do by hand. Importantly, the analyses they generate capture more complex grammatical phenomena than other language models, making them especially suited to the task of phenomenon discovery. Furthermore, the use of a dynamic treebanking methodology enables the construction of high quality disambiguated treebanks, which contain, for each sentence the grammar can parse, the most appropriate analysis produced by the grammar. The combination of these linguistically rich treebanks with expressive treebank query tools represents a potentially compelling strategy for retrieving candidate examples of grammatical phenomena.

Finally, the importance of grammatical phenomena to precision grammars is underscored by a number of aspects of grammar engineering that I identified in this section. In particular, this includes the application of precision grammars for supporting linguistic hypothesis testing, where they can be used to either falsify or add evidence in support of an analysis of a particular grammatical phenomenon. It is also seen in the phenomenon-driven development cycle of grammars, a typical component of which is the construction of test suites designed to systematically illuminate grammatical and ungrammatical examples of phenomena. Multilingual grammar engineering projects extend the grammar construction and hypothesis-testing activities across multiple languages, which has the effect of requiring even more attention to be given to how grammatical phenomena are analysed and represented.

In Chapter 5, proceeding under the premise that existing precision grammars developed within grammar engineering projects do represent suitable resources for supporting grammatical phenomenon discovery, I develop techniques for performing syntactic phenomenon discovery using grammars produced within the DELPH-IN consortium.

2.7 Electronic Grammaticography

In recent years, there has emerged an increasing body of literature dedicated to discussing the creation of descriptive grammars that leverage digital technologies
in order to increase their utility as linguistic resources. This emerging field, often referred to as electronic grammaticography, encompasses a variety of approaches towards the enhancement of traditional descriptive grammars. A characteristic feature of these strategies is that they do not simply constitute the release of digital editions of descriptive grammars—a strategy that an increasing number of publishers are adopting—but explicitly aim to take advantage of enhancements only made possible through the digital medium. I will refer to such creative works as electronic descriptive grammars, although they are also referred to as electronic reference grammars in the literature.

In this section, I first briefly outline the nascent field of electronic grammaticography. I then describe potential enhancements of descriptive grammars that have been identified in the literature, focusing on three types in particular that motivate the work in this research: syntactic phenomenon-based navigation, the retrieval of additional examples of syntactic phenomena, and the application of phenomenon retrieval to text collections accompanying the descriptive grammar. This will serve to better illuminate the motivations for this research and, in the process, identify the requirements for these enhancements, as well as potential challenges involved in their development.

2.7.1 Overview

The first comprehensive vision for what an electronic descriptive grammar could be was put forth by Johnsen (1996), who noted that electronic information sources used for educational purposes were increasingly making use of hypertext—text that uses hyperlinks for navigation and organisation—due to the flexibility it affords both authors and consumers and that descriptive grammars are well-suited to exploiting this document architecture. Since then, arguments for developing electronic descriptive grammars have also been put forth by Good (2004), Bender et al. (2004), Nordhoff (2008), Thieberger (2008), as well as the various contributors to Nordhoff (2012a).

Good (2004) and Nordhoff (2012b) address the problem of identifying the properties of traditional descriptive grammars that an electronic format would need to support, with Nordhoff proposing a candidate XML schema for structuring electronic descriptive grammars that is capable of supporting the structural properties identified by Good (2004). Thieberger (2008), noting most extant descriptive grammars’ paucity of accountability with respect to the accessibility of their source data, puts forth a vision and proposed methodology for embedding a descriptive grammar within the source data that was used to drive its analysis. The contributions of Musgrave and Thieberger (2012) represent a tangible first step towards realising this goal, involving the manual conversion of a portion of Heath’s description of the Australian language Nunggubuyu—which was discussed in Section 2.4.5—into an electronic descriptive grammar fragment.

Premised on the observation that users of electronic descriptive grammars will en-
compass a diverse range of contexts of use, Nordhoff (2008) puts forth a concrete set of expectations and values that different authors and users of electronic descriptive grammars might have, which are intended to be selectively prioritised by different projects according to their respective needs. These values are divided into three domains—data quality, layout assistance and templates, and accountability—with a series of maxims being presented for each, which would all be agreed upon by users holding the overarching value. These values and maxims have served to focus the discussion surrounding the development of electronic descriptive grammars. This can be seen in Nordhoff (2012a), a collection of papers on topics in the area of electronic grammaticography, which contains four papers focusing on broader issues surrounding the development of electronic descriptive grammars and six focusing on concrete enhancements and their potential applications. All but one of the latter group make use of Nordhoff’s (2008) values and maxims for framing their contributions.

Finally, Drude (2012b) presents an overview of existing research into electronic descriptive grammars, identifying three directions of ongoing research: the use of hypertext, the use of treebanks produced by computational grammars, and the use of literate programming for interleaving computational and descriptive grammars.

### 2.7.2 Enhancements

The adoption of a natively electronic platform for descriptive grammars opens up a range of potential enhancements, many of which have been discussed in the literature. Electronic descriptions can be augmented with linguistic resources containing media that require a digital device to view, such as the collections provided by PARADISEC,\(^50\) which, in addition to text, also include audio, video, and images that document less-resourced languages of the world. Linguistic ontologies, such as GOLD and ISOcat, could be leveraged to provide external glossaries of linguistic concepts (Nordhoff and Hammarström 2014). Similarly, online typological databases such as WALS Online (The World Atlas of Language Structures Online; Dryer and Haspelmath 2013) and The Database of Syntactic Structures of the Worlds Languages (Collins and Kayne 2009) could also be leveraged, as well as tools intended to facilitate querying typological databases like the Typological Database System (Dimitriadis et al. 2009) and the TYTO “Typology Tool” (Schalley 2012), which, given a description in a descriptive grammar, could be used to explore related grammatical phenomena cross-linguistically.\(^51\)

Electronic descriptive grammars also enable the separation of the information that constitutes the descriptive grammar’s analysis from the way it is displayed to the reader, meaning that same underlying data could be used to generate different views. This would remove the need for a single organisational strategy, with, for example,\(^52\)

\(^{50}\)http://paradisec.org.au

\(^{51}\)For more on typological databases, see Everaert et al. (2009).

\(^{52}\)http://paradisec.org.au
an electronic grammar being able to produce both form and function-oriented layouts from the same underlying data (Cristofaro 2006), or being designed to produce views that cater for different audiences (Baraby 2012, Schmidt and Dik 2007).\footnote{In support of this separation of data and presentation, Nordhoff (2012b) develops a framework for modelling the underlying structure of electronic descriptive grammars as a collection of self-contained form-meaning-pairs.}

Electronic descriptive grammars also open up the possibility of permitting significant revision after the time of publication. As the analysis of a language is improved, linguistic frameworks evolve, and new typological discoveries are made, the content of descriptive grammars inevitably becomes dated. Electronic descriptive grammars can be much more readily updated than their traditional static counterparts (Schmidt and Dik 2007, Comrie \textit{et al.} 1993). This is especially true of the modular architecture argued for by Good (2012), Nordhoff (2012b), and Nordhoff and Hammarström (2014), in which the grammar is no longer a single monolithic publication authored by a single author but is composed of a collection of micropublications which are authored in a distributed fashion and can be incrementally upgraded through targeted revisions.

The aforementioned enhancements motivate the adoption of electronic descriptive grammars, but they do not directly pertain to the research in this thesis. In the remainder of this section I outline the enhancements that have been identified in the literature as being enabled by electronic descriptive grammars and which directly motivate the development of syntactic discovery techniques developed in this thesis. 

**Hypertext and Non-linear Navigation**

The use of hypertext in descriptive grammars offers the potential for increased flexibility from the perspective of both grammar authors and consumers, overcoming the limitations of the linear format of traditional descriptive grammars (Johnsen 1996, Nordhoff 2008, Baraby 2012, Drude 2012a, Drude 2012b, Musgrave and Thieberger 2012). From the perspective of the reader, hyperlinked grammatical descriptions facilitate non-linear navigation, allowing the reader to easily traverse relationships between phenomena that cut across the high-level categories the grammar has been organised into. This non-linear navigation also corresponds more closely to how descriptive grammars are actually read (Johnsen 1996). From the perspective of grammar authors, the use of hypertext presents an opportunity for authors to explicitly target a non-linear architecture. Nordhoff (2008) makes the distinction between electronic resources whose internal structure is linear and those which are non-linear, noting that even traditional descriptive grammars presented in an electronic format with hyperlinked cross-references are still fundamentally linear in nature, with the link target always being a numbered page within a single linear sequence. A non-linear descriptive grammar, on the other hand, takes on a web-like structure of interconnected documents—a structure which is a more natural fit for the inherently interconnected
and web-like nature of language systems (Mosel 2006). Electronic descriptive grammars that fully embrace non-linear structure and navigation free the author from the need to coerce the contents of the analysis into a hierarchical organisation that is demanded by the linear constraints of the traditional descriptive grammar, thereby obviating the inevitable compromises this selection of a single linear organisational strategy incurs, and which was discussed in Section 2.4.4.

The type of non-linear navigation relevant to the work in this thesis involves the creation of links between exemplars and the sections of the descriptive grammar that describe the grammatical phenomena that are exhibited by the exemplars. This feature is suggested by Bender et al. (2012), who argue that exemplars which have been parsed and treebanked with precision grammars can be exploited to have cross-references to relevant sections of the descriptive grammar automatically inserted. Notably, this would allow users to navigate through various grammatical phenomena found within exemplars without being restricted to those the exemplars are intended to illustrate, resulting in less-restrictive and more organic exploration of grammatical phenomena within descriptive grammars. The challenge involved in developing this feature within electronic descriptive grammars is that in order to automate the insertion of such cross-references, preexisting associations between precision grammar output and descriptive grammar sections are required.

Query Facilities

In addition to the use of hyperlinks, query facilities provide another means for more flexible interaction with descriptive grammars (Johnsen 1996, Good 2004, Bender et al. 2012, Bouda and Helmbrecht 2012, Maxwell 2012, Mosel 2012, Bouma et al. 2015). From the perspective of usability, the inclusion of textual search facilities is essential for any hyperlinked text collection of non-trivial size and such facilities can already be seen in the digitised versions of traditional descriptive grammars. Documents with structural annotations could further enhance these queries by enabling the restriction of query terms to occur only in specific types of content, such as prose descriptions and, notably, linguistic exemplars, permitting users to search directly for examples matching their own needs.

Focusing on queries targeting terms occurring in linguistic exemplars, these could be further developed in two ways. Firstly, the queries could be extended to cover text in additional corpora beyond the exemplars included in the descriptive grammar. This would enable the retrieval of further examples of a phenomenon beyond those already included with its description, potentially allowing users to obtain answers to residual questions pertaining to a phenomenon after reading through the description and accompanying exemplars (Bender et al. 2012, Maxwell 2012, Bouma et al. 2015), the number of which are limited within a descriptive grammar, as discussed in Section 2.4.3. Enriching a descriptive grammar with the capacity to query corpora for grammatical phenomena also presents the user with the opportunity to
better understand the variation of constructions as well as their relative frequency across real-world language use (Bouma et al. 2015).

Simple text-based matching only enables the retrieval of basic grammatical phenomena or a limited range of instances of more complex phenomena. Querying facilities could improve upon this by leveraging linguistic abstractions beyond the level of simple text matching in order to compactly specify constraints which capture complete grammatical phenomena. In addition to augmenting descriptive grammars with the capability to explore and retrieve additional examples of potentially complex phenomena from alongside their descriptions, this enhanced querying would also provide users an additional dynamic method of navigating the grammar, helping to circumvent the linear constraints of traditional descriptive grammars (Bender et al. 2012).

Mosel (2012), for example, demonstrates a simple form of this kind of querying being used to augment grammatical descriptions. This involves the use of regular expressions for performing queries over unannotated text from a language documentation corpus in order to retrieve certain types of morphemes and explore the noun-verb distinction within the language of Austronesian Oceanic Teop. This approach, while valuable in the context of developing an analysis of an under-resourced language—where unannotated corpora may be the only resource available—is limited in the range of phenomena that can be retrieved, as it relies on the presence of regular morphological marking in the language, or constructions which can be detected through the appearance of closed word classes, such as determiners and auxiliary verbs. In order to capture the full range of grammatical phenomena found in descriptive grammars it is clear that linguistically annotated corpora should be incorporated with query facilities.

van der Wouden et al. (2017) describe enriching a descriptive grammar of Dutch with different types of querying facilities. This enhanced resource forms a part of the Taalportaal online language portal (Landsbergen et al. 2014) and enables users to query a range of linguistic analyses targeting phonological, morphological, and, notably, syntactic annotations. The syntactic querying component, described in Bouma et al. (2015), is realised through the use of two treebank querying tools, PaQu (Odijk et al. 2017) and GrETEL (Augustinus et al. 2017), which are configured to query treebanks produced using the Alpino Dutch precision grammar (van Noord 2006). The queries are manually created offline using the treebank querying tools in conjunction with the Alpino grammar so as to match target phenomena in the descriptive grammar. They are then attached to exemplars and textual grammatical descriptions, forming links into the corpora that retrieve further examples of the phenomena being described. Users are also able to dynamically modify the queries and construct new ones. I discuss Taalportaal and the syntactic querying methodology of Bouma et al. (2015) further in Section 2.7.3, while the treebanking tools PaQu and GrETEL are discussed in Section 2.9.2 and Section 2.9.3, respectively.

The use of treebanks produced by computational grammars for querying and retrieving grammatical phenomena within descriptive grammars is also argued for by
Bender et al. (2012) in the context of a more general argument for augmenting descriptive grammars with computational grammars. Bender et al. (2012) argue that descriptive grammars enriched with treebanks offer more flexibility for how examples can be retrieved, identifying the two modes of use that Bouma et al. (2015) employ: curated queries presented inline with descriptions of grammatical phenomena for retrieval of further examples, and custom queries created by the user of the resource for unconstrained exploration. Both Bender et al. (2012) and Bouma et al. (2015) also argue that, in order for curated queries to achieve maximal utility, the underlying query must be made available to the user. For realising the querying facility, Bender et al. (2012) propose using treebanks produced by DELPH-IN grammars, along with the treebank querying tool Fangorn (Ghodke and Bird 2012).

Bender et al. (2012) also discuss how enriching descriptive grammars with the capacity to query treebanks generated by precision grammars could realise the different values identified by Nordhoff (2008). From the perspective of accountability, the ability to query corpora annotated with rich linguistic annotations enables the retrieval of a broad range of phenomena retain their context of utterance. Additionally, the use of precision grammars also means that the results of the full analytic pipeline are potentially available to the user, including morphological, syntactic and semantic representations. From the perspective of actuality, the ability for treebanks to be dynamically updated to the current state of the grammar, as discussed in Section 2.6.5, helps prevent the resource from becoming outdated. Lastly, from the perspective of history, query results from treebanks produced by different versions of the precision grammar can be compared, providing a window into how the analyses in the computational grammar has changed over time.

These discussions of querying facilities have centred around their augmentation—or potential augmentation—with electronic descriptive grammars. In Section 2.9, I discuss underlying techniques for corpus querying that have been used to perform syntactic phenomenon retrieval within the field of computational linguistics.

Integration with Text Collections

As discussed in Section 2.4.5, a limitation of traditional descriptive grammars is the lack of accessibility associated with accompanying text collections containing data used to support the analysis put forth in the grammar. In particular, there is not much to be found in the way of integration between the descriptions of grammatical phenomena and accompanying text collection. This situation can be improved through the use of electronic descriptive grammars and their capabilities unavailable to traditional descriptive grammars (Thieberger 2008, Mosel 2012, Musgrave and Thieberger

---

53 Maxwell (2012) also makes this distinction, referring to queries included by grammar authors as write-time links or queries, as they are created during the constructing of the resource, and those which the user generates as read-time queries, since they are created during the consumption of the resource.
For example, Musgrave and Thieberger (2012), describe the process used to convert a portion of Heath’s print-based description of Nunggubuyu—described in Section 2.4.5—into an electronic format, using the extensive cross-references between the descriptive grammar, dictionary, and text collection to create a hyperlinked resource that constitutes a unification of the Boasian trilogy (Musgrave and Thieberger 2012). While the conversion into a hyperlinked resource was performed manually, this work forms the initial steps towards a methodology for the automatic conversion of legacy descriptive grammars into electronic descriptive grammars, while also helping identify the potential challenges involved in the creation of more advanced, natively-electronic descriptive grammars.\footnote{See also Nordhoff and Hammarström (2014), who argue for the need to digitally archive legacy descriptive grammars and describe a process for automatically extracting structural information from a collection of 7500 descriptive grammars.}

The resource produced by Musgrave and Thieberger (2012) demonstrates the potential for the construction of an electronic descriptive grammar which links grammatical description with the text upon which the analysis of the language is based. The process used to derive this integrated resource is, however, premised on the existence of a descriptive grammar and text collection pair that is already annotated with cross-references. As noted in Section 2.4.5, the annotation of a descriptive grammar with an extensive collection of such cross-references is a costly exercise, with Heath’s richly annotated description of Nunggubuyu being an exception rather than the norm. The grammatical phenomenon query facilities discussed in the previous section offer a potential strategy for realising the integration between description and corpus, with links between descriptions and text collection being realised as queries which are executed when a user follows them. Authors would then be spared the burden of having to create cross-references for every instance of each phenomenon in the corpus. Instead, a single query could be created for each phenomenon, which retrieves all instances from the text collection. This more dynamic approach has the advantage of automatically scaling to retrieve additional examples as more corpora are added. Furthermore, if the interface were to allow for the construction of new queries or the adaption of queries associated with existing links, then the user would not be limited to cross-references that were predicted as being valuable by the grammar author.

While the automatic generation of links between simple phenomena such as the occurrence of lexical entries and morphological forms is readily achievable—as illustrated by Mosel (2012)—coverage over a more comprehensive range of phenomena described within a descriptive grammars is necessary to fully embrace the potential advantages that the integration with a text collection can bestow. In this research I focus, in particular, on syntactic phenomena. If successfully applied, the integration of grammatical descriptions and accompanying text collections through syntactic discovery techniques would have the effect of raising the primary data found in the text collection from the status of a supplementary appendix to being directly accessible from within the descriptive grammar (Weber 2007a, Thieberger 2008). As discussed
in Section 2.4.5, this allows readers to verify the analyses found within the descriptive grammar, analyse phenomena which was not addressed by the grammar, and, if text beyond the primary data is included in the collection, would also allow readers to determine if analyses extend to new data, all of which improves the scientific accountability of the theory of the language put forth by the descriptive grammar (Nordhoff 2008, Bender et al. 2012, Maxwell 2012, Mosel 2012, Bouma et al. 2015).

These enhancements of descriptive grammars—non-linear navigation, query facilities, and integration with text collections—have in common that they can be supported through precision grammar enabled methods for syntactic phenomenon detection. In order for precision grammars to be used effectively for phenomenon detection, however, improved methods for phenomenon discovery within precision grammars need to be developed. This is supported by Bouma et al. (2015), who observe the challenges in aligning syntactic phenomena from a descriptive grammar with the analyses produced by a precision grammar. These alignments occur in the context of the development of a set of treebank queries that are used to annotate an electronic descriptive grammar. I discuss Bouma et al.’s (2015) work in Section 2.9.2, and I also discuss Taalportaal, the electronic descriptive grammar, for which these queries were created, in the following section.

2.7.3 Taalportaal

Taalportaal (Landsbergen et al. 2014) is an online language portal providing a comprehensive reference for the phonology, morphology, and syntax of Dutch, Frisian, and, more recently, Afrikaans. Written in English, it is intended to be a reference for an international audience of linguists and other researchers. Its descriptive content was initially seeded with three volumes from the The Syntax of Dutch descriptive grammar series (Broekhuis and Keizer 2012, Broekhuis and Dikken 2012, Broekhuis 2013), which was converted from Microsoft Word documents into XML. New content is authored directly in XML using an editing environment which gives authors access to a large collection of bibliographical references, the ability to annotate phonological, morphological, and syntactic exemplars with appropriate markup, and also enforces the use of a topic-based schema, dividing articles into three distinct views: the main content of the article, a brief overview, and supplementary material. The web interface allows users to navigate through the core grammatical descriptions—which are initially presented in a layout similar to traditional descriptive grammars—as well as navigate through included texts, external corpora, and a bibliography of related linguistic literature. Also available to the user is a linguistic terminology database, which can be queried directly, accessed from within topics as a glossary and is also leveraged for query expansion, with, for example, a query for the term article being extended to include the synonym determiner. All these components feature extensive

55http://taalportaal.org
use of cross-references, created by the authors of topics during an editorial phase, and also through an automatic process.

Since its release, the Dutch component of Taalportaal has been augmented with various tools that facilitate structured queries over linguistically annotated corpora (van der Wouden et al. 2017). Users of the enriched resource are able to query a range of different linguistically annotated collections, including the SoNar corpus (Oostdijk et al. 2013), which contains over 500 million tokens, annotated with part of speech tags and lemmas, and the CELEX lexical database (Baayen et al. 1995), which provides morphological and phonological analyses for over 100 thousand Dutch lexical items. Of particular relevance to this research is the ability for users to also query three syntactically annotated corpora for examples of syntactic phenomena. These corpora include two small treebanks with manually verified syntactic analyses—the LASSY Small Corpus (van Noord et al. 2013) and the syntactically annotated portion of the Spoken Dutch Corpus (Oostdijk 2002), each containing one million words of annotated text—and the LASSY Large treebank, containing 700 million words of syntactically annotated text, which was created automatically using the Alpino Dutch grammar (van Noord 2006). In addition, a selection of syntactic phenomena within Taalportaal are annotated with curated treebank queries designed to retrieve additional examples of syntactic phenomena from the syntactically annotated corpora. Queries are presented as links, which are attached to both prose descriptions pertaining to a syntactic phenomenon and exemplars and which direct users to a treebank querying tool pre-loaded with the relevant query and the option to select the target treebank for querying.

The methodology used to develop the treebank queries that syntactic phenomena within Taalportaal are annotated with is described in Bouma et al. (2015). This involved the selection of a subset of grammatical phenomena from Taalportaal which were to be annotated. The selection included adjectival phrases, adpositions, and a range of phenomena subsumed by these categories. Within the corresponding sections of Taalportaal, a team of annotators then constructed queries to attach to prose descriptions and exemplars of these syntactic phenomena, a process that resulted in 1200 distinct queries being generated and embedded within Taalportaal. The queries were developed with the aid of two XPath-based treebank querying tools, PaQu (Odijk et al. 2017) and GrETEL (Augustinus et al. 2017), both of which were also used to present queries and their results to the user. These tools allow users to modify the stored query and change the target treebank, with PaQu additionally providing frequency distributions of node attributes. Of the 1200 queries, Bouma et al. (2015) report that 77% of queries were lexical, targeting a specific lexical item that occurs in some syntactic context, with the remainder only specifying a syntactic context as a constraint. They also report that 73% of queries constrain word order, with the remaining queries applying no constraints to word order. Bouma et al. (2015) also observe that annotators often found construction of the queries challenging, partly due to the difficulty of having to establish how a desired constraint should be expressed.
within XPath, but also because it is not necessarily transparent to an annotator how a phenomenon is represented within the treebank. The difficulty of establishing how a target construction is represented within the treebank is further compounded by the fact that the analysis found within the Alpino-derived treebank may be different to that of Taalportaal’s, and sometimes the two may even be incompatible. These challenges identified by Bouma et al. (2015) are outlined further in Section 2.9.2, in the context of discussing the limitations of treebank querying tools that require users to understand how their target phenomena are represented in advance.

Taalportaal’s design and feature set, which includes non-linear navigation through extensive use of hyperlinks, global search, and grammatical phenomenon-based querying facilities over a range of corpora, coupled with its breadth and depth of coverage over grammatical phenomena, likely make it—at this point in time—the most fully realised example of what an electronic descriptive grammar is capable of being. At the same time, Taalportaal does not represent too dramatic a departure from a traditional descriptive grammar. The resource still contains the familiar features of prose grammatical descriptions with accompanying exemplars, all of which are organised into nested sections and can be navigated via a table of contents. This continuity with interface components associated with traditional descriptive grammars is important from the perspective of the accessibility of innovative electronic descriptive grammars (Drude 2012a). In the case of Taalportaal, which was converted and extended from an existing descriptive grammar and enhanced with features made possible by the digital medium, this also illustrates that a radical re-imagining of the nature of descriptive grammars is not required in order to start reaping the considerable rewards electronic gramaticography has to offer.

Taalportaal represents an example of how traditional descriptive grammars can be augmented with the two enhancements that motivate the development of syntactic phenomenon discovery techniques in this research: syntactic phenomenon retrieval from a corpus and non-linear phenomenon-based navigation. This makes Taalportaal and, in particular, the syntactic phenomenon querying work of Bouma et al. (2015) the most similar work that I have identified to the research presented in this thesis. My work differs in a couple of respects from Bouma et al. (2015). Bouma et al. (2015) describe a methodology for descriptive grammar augmentation that involves the selection of phenomena to be annotated, the identification of their representation within precision grammar output and associated treebanks, and then the development of queries that retrieve instances of these phenomena from a treebank. My work is motivated by the challenges Bouma et al. (2015) observe as being involved in identifying the way target grammatical phenomena are represented within the analyses produced by the precision grammar, which arise from the alignment of two different theories of a language which may potentially involve incompatibilities. My research focuses specifically on developing improved techniques for the task of syntactic phenomenon discovery, which, in the context of descriptive grammar augmentation with precision grammars, is antecedent to the task of phenomenon detection. This involves
the development of a methodology for making the analyses of syntactic phenomena within precision grammars more discoverable, which means the work in this thesis also represents a response to Bender et al.’s (2012) exhortation to make precision grammars and their treebanks more accessible.

This research also differs from Bouma et al. (2015) in that in order to perform syntactic discovery, I exploit structural properties of precision grammars beyond that of the derivation tree. DELPH-IN grammars, like Alpino, are inspired by HPSG, meaning that they make extensive use of a type hierarchy to capture linguistic abstractions involved in the analysis of the grammatical phenomena of a language. Since the complete analyses produced by DELPH-IN grammars include the types from the hierarchy used to constrain analysis, it is therefore possible to extract types from analyses produced for exemplar sentences in order to associate types with target syntactic phenomena. This is the underlying approach that I adopt in Chapter 5, in developing the Typediff methodology.

2.7.4 Summary

In this section I outlined the emerging field of electronic grammaticography, which involves the enhancement of traditional descriptive grammars with features made possible through computational means. I identified and outlined three specific enhancements that the syntactic discovery methods investigated in this thesis can potentially enable. Firstly, the use of syntactic discovery techniques applied to exemplars found within descriptive grammars could enable users to navigate between sections of the grammar by phenomenon, thereby enabling a form of non-linear navigation, and improving the discoverability of the grammar. Secondly, the use of querying facilities to retrieve instances of grammatical phenomena from within corpora provides users with the opportunity to consult additional examples beyond those provided by the grammar author, as illustrated by Bouma et al. (2015). Finally, using such querying methods to retrieve instances of phenomena being described from text collections that accompany the descriptive grammar would better integrate grammatical descriptions with the primary data upon which they were based, thereby increasing the scientific accountability of the descriptive grammar.

I also outlined the work of Musgrave and Thieberger (2012), who describe a proof of concept which demonstrates how an existing descriptive grammar with extensive use of cross-references can be converted to an electronic format to better exploit existing internal links between description and text. This approach, however, is premised on the existence of a resource already annotated with internal links, and therefore does not represent a sustainable methodology for generating new descriptive resources of this nature. Finally, I described the Taalportaal online language portal, another example of a descriptive grammar converted to an electronic version with integrated

---

56 The type hierarchy used by DELPH-IN grammars is outlined in more detail in Section 5.3.
grammatical descriptions and corpora. However, as described in Bouma et al. (2015), the links between these two components are realised not through individual cross-references to examples of phenomena but by curated queries into treebanks produced by a precision grammar, reducing the burden on grammar authors. A notable outcome of this work was the identification of the dual challenges of isolating the signature of a given syntactic phenomenon within the analyses produced by the grammar and also aligning the potentially divergent analyses of the precision grammar and the descriptive grammar. This shows that in addition to treebank querying tools, such as PaQu and GrETEL, which were used by Bouma et al. (2015), there is a need for tools and interfaces that enhance the discoverability of syntactic phenomena within precision grammars, in order to support the sustainable integration of computational and descriptive grammars. It is this space that the work in this thesis targets.

### 2.8 Projects Involving Grammatical Phenomena

I have argued that precision grammars are strong candidates for performing syntactic phenomenon discovery due to their capacity for providing accurate analyses of complex grammatical phenomena. While this is made possible through phenomenon-centric evaluation and development workflow, their underlying architecture, however, does not necessarily ascribe grammatical phenomena ontological status. This is not to say that clusters of related linguistic abstractions cannot be located within grammar constructs, but rather that the practicalities of implementing a theory of a language within a formalism that can be consumed by both grammar engineer and machine mean that constraints associated with one phenomenon will frequently be located non-contiguously within the grammar source code. Even when related constraints are adjacent to each other, they may not be explicitly identified as belonging to the same phenomenon. This follows partly from the interconnected structure of language systems and from the dynamic nature of grammatical phenomena, which was established in Section 2.3. The range of phenomena that a given grammatical description picks out varies greatly depending upon the characterisation used, the degree and manner to which it has been constrained, as well as background influences such as particular approaches to grammatical analysis favoured by the developer of the theory or imparted from an assumed grammatical framework. There are, in other words, no Platonic ideals of grammatical phenomena.

A necessary step, then, in the use of precision grammars for discovering syntactic phenomena is the development of a methodology for associating grammatical phenomena with structural properties of the grammar. Since there are no canonical characterisations of grammatical phenomena and the particular characterisation used to describe any given phenomenon will vary across queries submitted to a phenomenon discovery system this methodology must be sufficiently flexible to support the discovery of a wide range of phenomenon characterisations. The problem of dis-
coverability of grammatical phenomena within precision grammars can be unpacked into two related, but potentially distinct, methodological components. One is the need to operationalise the notion of a grammatical phenomenon in terms of grammar constraints or the formalism used by the grammar to represent analyses of sentences and the other is the need to support a flexible approach to classifying grammatical phenomena. There are many existing computational linguistics projects that employ a notion of grammatical phenomena and for which these issues are either explicitly or implicitly addressed. It is worth posing the question then, whether any of these projects can be leveraged for inspiration with respect to strategies for, firstly, operationalising grammatical phenomena in terms of an existing formalism and, secondly, developing an appropriate framework for classifying grammatical phenomena.

In this section I identify and describe a range of computational linguistics projects which either explicitly or implicitly involve one or both of the challenges of syntactic phenomenon operationalisation and syntactic phenomenon classification. The first two projects develop classification systems for syntactic phenomena, largely without operationalising phenomena into a particular formalism. The remaining projects all involve an inventory of concrete grammatical phenomena, which are based on the motivation and requirements of each project, with these phenomena being operationalised into a specific representational formalism. Some of these projects also develop strategies for the retrieval of instances of the phenomena from a corpus, a task I discuss explicitly in Section 2.9. For each project, I outline its motivation, goals, and methodology, focusing on its approach to classifying or selecting the grammatical phenomena, and, where relevant, looking at the strategies used to associate descriptive characterisations of grammatical phenomena with formal correlates.

2.8.1 The TSNLP Project

The TSNLP project (Test Suites for Natural Language Processing: Lehmann et al. 1996, Oepen et al. 1997) was a long running project across four partner institutions, motivated by the need for improved approaches for the evaluation of NLP resources. As discussed in Section 2.6.4, the importance of linguistic test suites for grammar engineering lies in the fact that they are curated to focus on specific grammatical phenomena through careful selection or construction of test items. Based on a survey of existing test suites (Estival et al. 1994) the project established that existing test suites tended to be under-documented, were largely devoid of systematic annotation schemes, did not attempt to record the depth and breadth of their coverage over included phenomena, and as a result, had limited reusability. In response to this, the TSNLP project developed guidelines, methodologies and tools for the construction of general purpose and multilingual reusable test suites for use as diagnostic and evaluation tools in the development of NLP resources, with an emphasis on the development of computational grammars in particular. The survey of existing test suites identified three key properties that the TSNLP methodology incorporated:
control over test data, in order to ensure items could be constructed to isolate and test specific phenomena; progressivity, the ability for test suites to be built and tested sequentially over the phenomenon space; and systematicity, the capacity for the test suite to ensure systematic coverage over all identified variants of the target phenomena as well as ungrammatical variants through negative test data. In addition to the development of methodologies for test suite development, the project also used these methodologies to produce three extensive test suites for English, French and German.

The TSNLP methodology for test suite construction included a set of guidelines for constructing test items, an annotation scheme, and a data model for representing test data and annotations. The guidelines focused on supporting the isolation of phenomena through stipulations such as: the use of a limited vocabulary for each language, no ambiguity, no modifiers or adjuncts, unmarked word order, declarative sentences, and the use of the 3rd person singular present tense—subject to the requirements of each phenomenon. The data model for storing test data and annotations consisted of four levels of representation: core data, phenomenon-related data, test sets, and parameters specific to each project. Of these, the first two representation levels contain the annotations relevant to the operationalising of syntactic phenomena, and so I now elaborate on these, focusing on the associated portions of the annotation scheme.

The core data representation level included: the test items that the test suite was made up of, with each item including the text string itself, the item ID, and a range of annotations, including an indicator of the grammaticality of the item, the length of the text, and two different levels of phenomenon-independent structural annotation of the text. The first level of linguistic structural annotation involved identification of the syntactic categories found in the text fragment along with their positions within the string. Aiming to be theory-neutral, these corresponded to relatively uncontroversial categories found in traditional phrase structure analysis (e.g. S, C, DET, NP, N, and PRON), totalling 20 broad syntactic categories. These category labels were further augmented by features as appropriate for each language and category, such as agreement features, like case, person, number and gender, as well as other features such as clause type and verb form. Some examples of these categories are presented in Figure 2.8. The second level of linguistic structural annotation involved the identification of predicate-argument structure (or functor-argument in the TSNLP project). The particular classification scheme for functional relations varied across the three languages, however they all contained the following distinctions: functors (func), arguments/complements (e.g. subj, obj, s-comp), modifiers (mod), or specifiers (spec), and the text span each element depends upon. Annotation of a simple dependency graph was chosen over that of a phrase structure tree as this was deemed to yield analyses that would be less controversial and more stable.

Instances of the phenomenon-related representation level corresponded to occurrences of specific grammatical phenomena from an inventory of grammatical phenom-
V_3.sg
NP-pl-fem
S
S_imp
S_int
C_main-decl
C_main-int-wh-inv
VP_ing
VP_perf

Figure 2.8: Example syntactic category labels used in the annotation of TSNLP test items (Estival et al. 1995).

ena, which were selected by the following process (Estival et al. 1995). As part of the initial survey, an inventory of syntactic, semantic and extra-grammatical phenomena was compiled for establishing the coverage rates of existing test suites. This list was presented to the four partner institutions involved in the project, with each being asked to select phenomena they would consider indispensable for evaluation of NLP systems targeting the language they had been assigned. Similarities across phenomena in the responses were then used to collapse the results into a final list of 10 broad categories that represented core grammatical phenomena in English, French, and German. As described in Oepen et al. (1997), the phenomena selected were:

- Complementation
- Agreement
- Modification
- Diathesis
- Modality, tense and aspect
- Sentence and clause types
- Word order
- Coordination
- Negation
- Extragrammatical (parenthetical, temporal expressions)

With these being fairly coarse-grained categories, two further subdivisions in the classification scheme were created in order to increase the specificity of the target
Figure 2.9: TSNLP phenomenon subcategories capturing distinct kinds of NP agreement (Estival et al. 1995).

- NP_Agreement_DET_N
- NP_Agreement_DET_N_Adj
- NP_Agreement_DET_AP

Figure 2.10: TSNLP phenomenon subcategories capturing distinct kinds of clausal complementation, with respect to valence patterns (Estival et al. 1995).

- C.Complementation_zerovalent_subj(PRON_impers)_V
- C.Complementation_monovalent_subj(PRON)_V
- C.Complementation_monovalent_subj(NP)_V
- C.Complementation_zerovalent_subj(PRON_impers)_V_obj(NP)

The phenomenon classification scheme used in the TSNLP project is based around increasing levels of granularity, beginning with the broad syntactic category associated
Figure 2.11: A sample instance of the TSNLP annotation schema for an item from the German test suite (Oepen et al. 1997:82).

with the phenomenon and expanding to additional levels of structural distinctions as deemed necessary for each phenomenon. For example, the broad category of noun phrase agreement is divided up into subphenomena corresponding to the different elements of the noun phrase that exhibit agreement, as illustrated in Figure 2.9. The broad category of clausal complementation, on the other hand, is divided up according to the identified verbal subcategorisation frames, as illustrated in Figure 2.10, which are also annotated with limited grammatical functions. Overall, this can be characterised as a hierarchical and granular classification system, based on formal properties, including phrase structure and morphological features, as well as grammatical function. The advantage of such a granular coding scheme for classification and annotation of phenomena is that it facilitates the goals of systematicity and progressivity. Systematic coverage over known phenomenon variants is facilitated by checking that the various parameters identified as part of the coding scheme and their possible permutations have been exhaustively instantiated. Progressivity is realised through the constraint that each item can only contain one (sub)phenomenon that distinguishes it from other test items. This allows test suites to be constructed such that failure on one item indicates that the failure was associated with the phenomenon the item introduces.\(^57\)

\(^57\)Since sentences usually contain multiple phenomena, a further requirement was identified which
While the phenomenon-classification developed by TSNLP is intended to be applied to a range of NLP tasks without relying on any particular formalism, it does in fact invoke a degree of formal analysis, as can be seen from the phenomenon labels found in Section 2.10. An issue that is associated with the TSNLP methodology is that were the syntactic analysis used for the original annotation of the test item to be changed, its corresponding phenomenon category might need to be updated, resulting in increased overhead in the maintenance of test suites.

A more general difficulty associated with the methodology developed by the project, however, is the resource-intensive nature of the resultant test suites. In order to realise the properties of systematicity and progressivity, developers of the test suite must fully comply with the annotation scheme and methodology, requiring comprehensive coverage over a large number of distinct phenomena, generated from granular divisions in the selected phenomenon space. If any of the phenomenon space is insufficiently covered by a set of test items or if a test item has failed to be identified as containing a target phenomenon, then the properties of systematicity and progressivity are jeopardised. This significant burden placed upon those responsible for constructing and maintaining these linguistic test suites is likely a factor which contributed towards the lack of adoption of the TSNLP methodology.

From the perspective of the research described in this thesis, it is clearly desirable for a phenomenon discovery system to be able to support granular divisions within the phenomenon space, however this cannot come at the cost of an onerously manual methodology that might compromise adoption. Rather than the granular divisions in the phenomenon space being generated through a top-down process, as Figure 2.10 is indicative of, it would be preferable for the operationalisation of syntactic phenomena to be more dynamic, arising due to the needs of the user making a query pertaining to a specific phenomenon.

### 2.8.2 A Theory of Syntactic Phenomena

Lehmann (2000) develops a framework for classifying syntactic phenomena, largely motivated by the desire to support phenomenon-oriented grammar engineering. Within the development of precision grammars, Lehmann observes, there is a discrepancy between their development and evaluation, on the one hand—which, as discussed in Section 2.6.3, are both phenomenon-driven—and the way in which grammars are constructed on the other, with phenomena not being localised, but rather with associated constraints being dispersed over various locations. This, Lehmann says, is a consequence of the underlying formalisms of precision grammars being primarily focused around the notion of constituency, for which there is no convenient mapping to the phenomena precision grammars are designed to model. These competing involved annotating these incidental phenomena in a presuppositions field, indicating the phenomena which are required to have already been tested before this test item.
dimensions of description result in difficulties for the grammar engineer, most notably, when trying to determine the location of a problem despite knowing which phenomenon the issue is associated with. Lehmann argues that precision grammars with a phenomenon-oriented architecture would not only see a reduction in the complexity involved in development and maintenance, but would also be more reusable, with phenomenon modules potentially being able to be reused across different languages (c.f. the Grammar Matrix (Bender et al. 2002, Bender et al. 2010), which is discussed in Section 2.6.6). In order to ground future work on phenomenon-oriented grammars, Lehmann develops an operational definition for syntactic phenomena and maps out a framework for classifying syntactic phenomena.

Under this framework, a syntactic phenomenon is characterised as one or more relations between properties of linguistic objects. Here, a linguistic object is approximately equivalent to a constituent. Properties are used to specify which characteristics of the linguistic object are relevant for a given phenomenon, and are analogous to grammatical features, being represented as attribute-value pairs, such as \texttt{NUMBER=plural}. Relations are divided into three types, selection, order, and agreement, with the selection relation being subdivided into the dependency relation and coordination relation. Syntactic phenomena are primarily characterised by the number of relations involved, with phenomena being divided into basic and complex phenomena. Basic phenomena are composed of just one relation, while complex relations are composed of multiple relations. The complete inventory of basic phenomena is shown in Table 2.4, which also shows which relation each basic phenomenon is characterised by. All other phenomena are complex, being composed of multiple basic phenomena.

A concrete example of a complex phenomenon is found in (23), which provides an example of an English interrogative sentence. Of the basic phenomena, the salient phenomenon that characterises this complex phenomenon is word order, more specifically, the inversion of the linguistic objects of the auxiliary verb and the subject,
with respect to declarative word order. Also present, however, is the subcategorisa-
tion phenomenon, as the verb selects the noun phrase Alex as its subject and the
auxiliary verb has selects for the verb phrase left already. Since these two phenom-
ena will necessarily be present in any interrogative sentence, in this framework, these
are considered obligatory phenomena, but with word order considered a foreground
phenomenon—as it is salient to the complex phenomenon—and subcategorisation
considered a background phenomenon. Other phenomena can also be found in the
form of agreement between the subject and verb, as well as modification of the verb by
the adverb already. However, since the example sentence could have been constructed
to exclude these phenomena—for example, by omitting the adverb, and changing the
tense of the auxiliary verb—while still constituting an interrogative sentence, these
phenomena are considered optional.

(23) Has Alex left already?

Lehmann does not put the classification system to work for its motivating ap-
plication of phenomenon-oriented grammar design, but does describe ongoing work
into the development of such an approach (Oliva et al. 1999), which the syntactic
phenomenon framework could be applied within. Under this approach grammars are
organised into modules for each basic phenomenon, in a similar fashion to how the
Grammar Matrix—discussed in Section 2.6.6—is organised into distinct phenomenon
libraries. Lehmann (2000) does, however, describe using the classification system in
the context of a phenomenon retrieval task—which is discussed in Section 2.9.2—for
the purpose of performing phenomenon-centric error analysis.

Focusing on the framework more generally as a means of operationalising syntac-
tic phenomena, a few observations can be made. Like the phenomenon classification
system employed within the TSNLP project, this framework is not based on a par-
ticular formalism for performing grammatical analyses and is therefore independent
of any implemented language model. Where TSNLP used a top-down hierarchical
classification system, this takes the form of a framework in which syntactic phenom-
ena are described compositionally through the combination of constituent phenom-
ena. This alleviates the problems associated with TSNLP’s prescriptive approach
to phenomenon characterisation. However, Lehmann’s framework introduces its own
difficulties when considering its use in the context of syntactic phenomenon discovery
within descriptive grammars. From the perspective of a user of a descriptive grammar
it isn’t clear that the framework’s reductive divisions in the phenomenon space would
 correspond to categories that are of practical use. Looking at Table 2.4, for both the
relation-types and their corresponding basic phenomena, if taken as phenomenon cat-
egories, these would cut across the typical divisions found in a descriptive grammar
in a kaleidoscopic manner, with these categories likely being associated with almost
every chapter and section of any plausible descriptive grammar layout. It appears as
though the use of this framework for the discovery of the kind of syntactic phenomena
discussed in descriptive grammars would still require an additional translation stage,
Chapter 2: Background

whereby accessible labels of phenomena likely to be seen in a descriptive grammar are mapped onto their constituent phenomena from the framework. Such an approach would then require the maintenance of some kind of separate phenomenon inventory that records these mappings, adding additional overheads to development. This mismatch arises from the fact that Lehmann’s framework was primarily intended as a means of organising and structuring precision grammars by phenomena that would be flexible enough to support arbitrary syntactic phenomena, rather than aiming to support user-oriented navigation of grammars by familiar phenomenon categories. It seems likely that, even if Lehmann’s framework was successfully used to develop precision grammars organised by grammatical phenomena, the challenges motivating this thesis would likely remain.

2.8.3 The ERG Semantic Documentation Project

Flickinger et al. (2014b) describe the development of an encyclopedia of mutually compatible semantic analyses containing a broad range of semantic phenomena covered by the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011). This resource, which is referred to as the ERG Semantic Documentation (ESD; Flickinger et al. 2014a), is motivated by the need for documentation of the semantic interface to the ERG—how meaning is represented in its semantic analyses. This kind of documentation is desirable for non-expert users of the grammar, who are unfamiliar with the semantic analyses, as well from the perspective of developers of the grammar, who, through the development of this resource, can deepen their understanding of interactions between phenomena and potentially identify aspects of the grammar that can be refined.

The main component of the encyclopedia is a catalogue of semantic phenomena identified as being covered by the ERG. The identification of the phenomena for inclusion in this inventory was performed through a data-driven discovery procedure, involving the classification of syntactic grammar components by their semantic effects (Flickinger et al. 2014b). The first step in this procedure involved identifying grammar entities—phrase structure rules, lexical rules and lexical entries—which were indicative of semantic analyses that “go beyond the basics,” (Flickinger et al. 2014b:878) meaning they are sufficiently noteworthy, in terms of the semantics they introduced, to be documented.

As with most other grammars produced within the DELPH-IN consortium, the semantic analyses generated by the ERG are represented using the Minimal Recursion Semantics framework (MRS; Copestake 2005). From a high level perspective, MRS represents the propositional content of a sentence as a bag of elementary predications, each of which is parameterised by a series of arguments. Each argument encodes a relationship with either the logical variables of events or instances (prototypically verbal and nominal expressions respectively) or representation-internal variables used to capture scope underspecification. Within the discovery procedure, semantically
noteworthy phrase-structure rules and lexical rules were both taken to be those that
introduced a predicate to the semantic analyses, and for lexical types, candidates
were required to either contribute more than one predicate or take at least one scopal
argument. Each candidate grammar entity was associated with the set of predications
it introduces, and then, for each class of grammar entities, a deterministic clustering
process was performed over the predication signatures. These clusters were then
manually inspected to ensure within-cluster consistency and collapse clusters where
appropriate. This resulted in a set of around two dozen phenomena generated from
phrase structure and lexical rules, including such phenomena as measure phrases,
partitives, conditionals, and coordination. The criteria used to identify semantically
interesting lexical types, however, was insufficiently discriminatory, resulting in too
large a set of candidate lexical types to undergo the clustering procedure.

This resulting list of phenomena was used to begin the initial version of the ESD.
Each entry in this resource contains a linguistic description of the phenomenon, some
exemplars, a discussion of any known interactions with other phenomena, any open
questions regarding treatment of the phenomenon in the documentation or within
the grammar itself, and notably, the semantic fingerprint of the phenomenon which
uniquely identifies it. This fingerprint is represented through a compact templating
language which permits the expression of underspecified MRS fragments. Flickinger
et al. note that, building off the semantic querying work of Kouylekov and Oepen
(2014), this fingerprint can be used for performing queries over ERG treebanks (or
other corpora parsed with the ERG) in order to retrieve additional examples of the
phenomenon, gather frequency data, or gauge the degree of interaction with other
phenomena.

Alongside the documentation formed by the catalogue of phenomena, Flickinger
et al. also describe the construction of a semantic test suite of 65 sentences, containing
at least one exemplar for each phenomenon from the catalogue. These exemplars were
created from sentences found during the discovery procedure and simplified so as to
remove unnecessary complexity and minimise the target vocabulary. By pairing the
semantic fingerprint with each sentence—at least for phenomena whose analyses have
stabilised—this semantic test suite allows developers of the ERG to perform regression
testing through automatic detection of changes to the analyses of these phenomena
which were intended to be unaffected after an update to the grammar.

A notable feature of the classification of grammatical phenomena found within
the ESD project is that it concerns itself with semantic phenomena, as opposed to
the broadly phrase-structure based classification found in the TSNLP project and
Lehmann’s (2000) framework for syntactic classification. In Section 2.3.3, I estab-
lished that grammatical phenomena can be characterised in terms of form and func-
tion and that this is true of syntactic phenomena, the focus of this research. While a

58Within delph-in grammars, every lexical entry has one lexical type, which can be seen as
elaborate part of speech tags. I discuss lexical types in Section 3.2.1 and Section 5.3.
strict separation of form and function is difficult to delineate, it is pretty clear that the semantic phenomena of the ESD corresponds closely to the notion of functionally characterised phenomena, insofar as the phenomena are characterised by the way contributions to the final meaning of a sentence are represented. The significance of this lies in language’s propensity for realising the same function through multiple and often diverse forms, which, as previously discussed in Section 2.3, means that a purely functional characterisation of a grammatical phenomenon would result in a dramatically different division in the phenomenon space than when primarily taking form into account.

A concrete example of a functional characterisation leading to different divisions in the phenomenon space can be found in Chapter 14 of Huddleston and Pullum (2002). This chapter is titled *Non-finite and verbless clauses*, delimiting a single—albeit rather broad—category of phenomena on the basis of formal properties, namely: clauses that have either a non-finite form of the verb or do not contain a verb but still have a subject + predicate structure. This chapter is then further divided up according to both functional and formal properties, with, for example, Section 9 covering *Non-finite clauses as modifiers and supplements*—a characterisation of a phenomenon that is both formal and functional. Turning to the current version of the ERG Semantic Documentation, it identifies the category of *non-adverbial clausal modifier* as a distinct phenomenon, which captures constructions that are described as “sentence-level modifiers lacking a finite verbal head.” At first sight, this appears as though it could align with the category of non-finite clauses as modifiers and supplements from Huddleston and Pullum. The semantic fingerprint that represents the phenomenon in the ESD is presented in Figure 2.12, and is characterised by a two-place subord relation. This fingerprint, however, turns out to insufficiently constrain the definition of the phenomenon in order to align with the category from Huddleston and Pullum.

The lack of alignment is illustrated by the sentences in (24), which are all sourced from Huddleston and Pullum. Of the four sentences, (24a), (24b) and (24c) receive analyses from the ESD which match the ESD semantic fingerprint in Figure 2.12 but do not, however, all lie within the same section within Huddleston and Pullum.

---

60 http://moin.delph-in.net/ErgSemantics/NonAdverbialClausalModifiers accessed on 11/05/2017
(24a) and (24b) are both found in the non-finite clauses as modifiers and supplements section, but (24c) is identified as a predicative adjunct, hailing instead from Chapter 6, which deals with adjectives and adverbs. The overarching organisation by form has resulted in the adjectival adjunct being treated in the chapter concerned with adjectival forms. Furthermore, (24d) provides an example of a sentence that the Huddleston and Pullum category includes but the ESD fingerprint does not capture. While (24d) is not yet correctly handled by the ERG, were it to do so, it would yield a semantic analysis that does not match the semantic fingerprint. 61

(24) a. His hands gripping the door, he let out a volley of curses. (p1265)
b. Born in Aberdeen, Sue had never been further south than Edinburgh. (p1265)
c. Upset, the children had daubed paint on the walls. (p530)
d. To put it bluntly, they’re utterly incompetent. (p1266)

In the context of a consumer of Huddleston and Pullum who is reading Chapter 14 and desires to retrieve further examples of clausal supplements, the fingerprint found in Figure 2.12 would be an inappropriate query as it will also return adjectival supplements, which do not meet the formal criteria of the overarching chapter—clauses either having a non-finite verb form or not featuring a verb but still having a subject + predicate structure—and are found in a different section of the reference grammar. It would, of course, be unreasonable to expect two different resources which model grammatical phenomena to align perfectly, since, as already discussed in Chapter 1, differences in theoretical presuppositions, source data, and methodological approaches will inevitably yield different divisions in the phenomenon space. Indeed, the alignment of two different theories of the same language is one of the central challenges that is being addressed in this research. This example does serve, however, to illustrate concretely that purely functional characterisations, as used by the ESD, make distinctions within the phenomenon-space that would be inappropriate for the purpose of retrieving instances of syntactic phenomena.

Of course, a user may in fact desire to retrieve all examples of the functionally characterised phenomenon of non-adverbial clausal modifier from the ESD. As discussed in Section 2.4.3, there are advantages to the navigation—and hence organisation—of descriptive grammars both by form and function. The same arguments that motivate both types of organisation also motivate the need for both types of phenomenon querying approaches, since the proposed querying mechanism also constitutes a kind of navigation. While I do not investigate the discovery of semantic phenomena in this research, it is clear that functionally characterised phenomenon querying is complementary to predominantly form-based phenomenon querying. Fully realised electronic descriptive grammars should ideally support both types of phenomenon querying within the same interface.

61 Personal correspondence with Dan Flickinger.
While the precise method of grammatical phenomenon characterisation used within the ESD is not appropriate for use in this research, there are a number of useful observations that can be made regarding its methodology. The resultant inventory of phenomena within the ESD was arrived at in a highly data-driven manner, as opposed to an a priori listing of course-grained phenomena, as found in the TSNLP project. This makes the process of establishing the phenomenon-categories a more bottom-up and dynamic methodology than the one found within TSNLP. The ESD methodology does still require a manual stage of verification in order to arrive at a final phenomenon inventory. Furthermore, the data-driven discovery procedure was firmly grounded in the MRS formalism used by Delph-In grammars to represent semantic analyses. This means that, unlike the classification systems of Lehmann (2000) and the TSNLP project, which are intended to be applied to various implemented language models, this classification system is operationalised in terms of a specific formalism used to implement language models. The use of a representational formalism enables generation of the ESD’s semantic fingerprints, which are associated with each phenomenon category and can be used for querying treebanks for further instances of the phenomenon (Kouylekov and Oepen 2014). The use of an underlying grammatical formalism in characterising phenomena is also present in the remaining projects considered in this section.

2.8.4 DepBank

The PARC DepBank (King et al. 2003) is a freely available collection of 700 sentences randomly selected from section 23 of the PTB and semi-automatically annotated with the help of a deep LFG grammar. Construction of the resource was motivated by the need for treebanks which better supported the evaluation of predicate-argument structure. Like other dependency-based treebanks, the sentences are annotated with both predicate-argument and adjunct dependency relations. At this time if its construction, this resource stood out from other dependency-based treebanks due to its inclusion of a range of features that pertain to syntactic and morphosyntactic structure, in addition to the standard predicate-argument style dependencies. Not only does this imply a process of operationalising syntactic phenomena, but is also means that this resource is a potential candidate for use in syntactic phenomenon discovery.

The methodology for the construction of the resource involved parsing the 700 sentences, followed by manual selection and correction of the best parse for each sentence. The constituent-based e-structure for each sentence was then discarded, with the f-structure being converted into a dependency format through an automatic process of flattening and re-writing of attributes associated with the target dependencies selected for inclusion in the resource, followed by a final manual validation and correction stage, which was performed with the help of visualisation and checking tools.
structure(
    mood(jump::0, imperative),
    adjunct(jump::0, not::5),
    adjunct(jump::0, yet::4),
    stmt_type(jump::0, imperative),
    subj(jump::0, pro::1),
    vtype(jump::0, main),
    pers(pro::1, 2),
    pron_type(pro::1, null),
    adegree(yet::4, positive),
    adv_type(yet::4, sadv),
    adjunct_type(not::5, negative)
)

Figure 2.13: DepBank annotation for the sentence Don’t jump yet (King et al. 2003).

DepBank includes standard dependency relationships such as the subcategorised grammatical functions of subject, object, and secondary object, as well as non-subcategorised grammatical functions like adjuncts and conjunctions. It also includes syntactic and morphosyntactic features that do not pertain to predicate-argument structure, such as sentence type, passivisation, tense, mood and aspect, as well as case, person, number and gender. In the DepBank format, all features are represented as two-valued predicates, with the first argument being the index of the item the predicate applies to, and the second the value of the feature. In the case of predicate-argument features, this value takes the form of an index, and in the case of the syntactic and morphosyntactic features, the value is simply an identifier. Users of the resource who are only concerned with the predicate-argument features can use a structure pruning tool to remove all predicates whose second argument is not an index. A DepBank annotation for the sentence Don’t jump yet is presented in Figure 2.13. A notable limitation of this format is that character spans from the original surface string are not associated with predicates. Additionally, since a sentence is represented as a bag of predicates, word order is not retained.

DepBank was intended primarily to be a resource for evaluating systems performing deep parsing, by assessing their ability to detect predicate-argument structure, which is difficult to extract from constituency trees (King et al. 2003). The addition of extra features pertaining to syntactic and morphosyntactic structure—for which considerable attention was given during the design of the resource—set it apart from other dependency banks at the time of its construction. In practice, however, while the dependencies from DepBank have proven useful for performing evaluation across different parsers, the additional syntactic and morphosyntactic annotations have not been put to much use (By 2007). From the perspective of syntactic phenomenon discovery, while DepBank facilitates the convenient discovery of instances of the syn-
tactic and morphosyntactic phenomena that were included within the annotation scheme—as well as phenomena which can be potentially analysed as complexes of these simpler phenomena—the loss of phrase structural annotations places significant limitations upon the syntactic phenomena which can be recovered from the resource. That word-order cannot be recovered from the resource is especially limiting.

### 2.8.5 CCGBank

CCGBank (Hockenmaier and Steedman 2007) is a corpus of sentences from the PTB annotated with both syntactic derivations and lexical dependencies. The syntactic derivations take the form of Combinatory Category Grammar (CCG; Steedman 2000) derivations, a highly lexicalised grammar formalism that facilitates the extraction of non-local phenomena, such as long-distance dependencies, which were discussed in Section 2.6.1. The motivations surrounding the construction of CCGBank did not pertain directly to the treatment or handling of grammatical phenomena, but instead were geared towards construction of a reusable resource which facilitated the extraction of predicate-argument structure. More specifically, The CCGBank project aimed to create a treebank that could be used to train parsers whose output supported predicate-argument extraction better than existing statistical parsers, notably those trained with PTB annotations, for which this type of extraction was difficult. In the process of developing a translation procedure for automatically converting PTB analyses to CCG analyses, however, a number of different constructions required individual attention. As a result of this, an artefact of the construction of CCGBank is a well-documented inventory of syntactic phenomena, which have been operationalised in terms of a precision grammar formalism.

In addition to standard phrase-structural information, PTB annotations include function tags on non-terminal nodes, which encode the grammatical function of phrases—such as subject and direct object—as well as null elements and coindexation of nodes used to capture various non-local phenomena including long-distance dependencies (Marcus et al. 1993). In practice, however, statistical models that have been induced from the PTB have largely used reduced representations that omit function-tags and null elements (Gabbard et al. 2006, Hockenmaier and Steedman 2007, Rimell et al. 2009). Such decisions were made as cost-benefit trade-offs, especially in the case of long-distance dependencies—whose analysis are associated with the occurrence of null elements and coindexation—with the marginal increases in global accuracy (on a per-dependency basis) due to these infrequently occurring constructions seen as not being worth the increased complexity of incorporating null elements and coindexation into the models (Hockenmaier and Steedman 2007, Rimell et al. 2009). However, from the perspective of the successful recovery of predicate-argument structure for use in real-world applications, as discussed in Section 2.6.1, the successful handling of long-distance dependencies is critical. Hockenmaier and Steedman argue that while some approaches have been developed to re-introduce function tags and null elements
into the parser output, they typically involve an additional processing step, which can be a source of noise and errors.

CCGBank tackled these problems by constructing a corpus of syntactic annotations with the expressivity to capture non-local constructions and which could readily facilitate the extraction of predicate-argument structure, resulting in richer and more consistent dependencies which could be used to train improved statistical parsers. Both of these features are made possible through the use of the highly lexicalised and linguistically expressive formalism of CCG. Non-local constructions are analysed through the use of complex syntactic types and CCG’s surface-compositional syntax-semantics interface allows the predicate-argument structure of a sentence to be directly extracted from a syntactic derivation. The methodology Hockenmaier and Steedman used to create CCGBank was a translation procedure that mapped PTB annotations onto CCG syntactic derivations. The core translation algorithm involved the following four distinct stages: using head-percolation heuristics to label nodes as heads, complements, or adjuncts; binarizing the tree; assigning CCG categories to the nodes of the tree; and, finally, generating the lexical dependencies by retracing the derivation and determining the lexical heads associated with each category in the derivation.

The analyses of the constructions in CCGBank largely target the CCG analyses found in Steedman (1996) and Steedman (2000). While the generic phrase-structure analyses found within the PTB are, for the most part, sufficiently rich to capture the information required to generate the target CCG analyses, the divergent nature of the formalisms combined with noise in the original PTB annotations meant that a number of additional steps were required to augment the core translation algorithm in order to match some of the target CCG analyses. This involved systematically working through the PTB’s analysis of problematic constructions and modifying the translation algorithm in order to arrive at the desired CCG analysis. Included in these changes were insertions into the input trees, such as for coordination, which is not explicitly represented in the PTB, and also for noun phrases, which have a relatively flat structure within the PTB. While some of the missing structure could be fully recovered, some constructions, such as quantifier phrases, required the use of heuristics, and in the case of compound nouns, correct recovery was deemed impossible without re-annotation and a simple default right-branching structure was assumed. Modifications were also required for constructions which are analysed in the PTB using null elements, which are not used in CCG. These included constructions such as passives, infinitival and participial verb phrases, inversion, and ellipsis. These also included various long-distance dependencies, including constructions such as wh-questions, tough movement, topicalisation, pied-piping, subject extraction, subject

\footnote{This assumes the semantic interpretation of lexical entries are known. Within CCGBank, semantic interpretations are approximated using simple bilexical dependencies.}

\footnote{A more comprehensive account of the modifications required for each construction can be found in the CCGBank manual (Hockenmaier and Steedman 2005).}
extraction from embedded sentences, and clefts. In addition to careful massaging, 
the heuristics for these long-distance dependencies also made use of coindexation 
templates, which encoded how dependencies should be resolved across constituents 
for each construction.

Although the motivation for developing CCGBank did not focus on the treatment 
of grammatical phenomena—other than a perceived need for improved parsing 
performance over specific phenomena—the need to extend the translation algorithm 
to handle individual constructions on a case-by-case basis meant that an artefact of 
the development of this resource was a document that records the steps taken to 
convert PTB analyses of distinct syntactic constructions into CCG analyses, along 
with the final CCG signature of the construction (Hockenmaier and Steedman 2005). 
This resource then contains enough information to retrieve all instances of these con-
STRUCTIONS FROM CCGBank and potentially also for further corpora, through the use 
of parsers trained over CCGBank, such as Clark and Curran (2007) and Clark and 

The syntactic phenomena within this inventory can be can be essentially charac-
terised as constructions for which the core translation algorithm was insufficient to ar-
rive at the appropriate CCG analysis without intervention. Within this set, individual 
constructions can also be seen as being classified—or perhaps simply demarcated—
by the extent to which a separate procedure was required to arrive at the necessary 
analyses. When construed as an approach for classifying syntactic phenomena, it is 
clearly a much more emergent system than compared with others identified in this 
section. More generally, while the constructions identified through this procedure are 
more likely to be syntactically interesting, their inclusion is contingent on the details 
of the core translation algorithm, the nature of the analyses found within the PTB, 
and the target CCG analyses. One possible application for CCGBank in the context 
of syntactic phenomenon discovery is for the retrieval of instances of syntactic phe-
nomena found in this inventory from within CCGBank or treebanks derived from 
parsers trained over the CCGBank annotations—an application that I discuss, along 
with other phenomenon-annotated resources, in Section 4.3. A notable limitation of 
using CCGBank for this, however, is that this inventory is limited to those which 
happened to be explicitly identified and targeted within the translation procedure. A 
phenomenon discovery tool would be of limited value if restricted to this particular 
inventory. It would be desirable to extend this range of constructions, which would 
require additional work identifying the corresponding CCG signature.

In terms of operationalisation, each syntactic phenomenon within this inventory 
yielded a characteristic CCG analysis that can be picked out though a syntactic 
category, or larger derivation pattern. Each construction then has a concrete repre-
sentation that is couched in the framework of CCG. As a part of the development 
of the translation procedure that yielded this operationalisation, Hockenmaier and 
Steedman (2007) describe a number of cases where the PTB analyses could not be 
deterministically translated into CCG analyses due to missing information in the
PTB analyses. These included the lack of a distinction between complements and adjuncts, as well as missing information pertaining to the internal structure of noun compounds, which I discuss further in Section 4.3. Furthermore, Hockenmaier and Steedman (2007) also identify a number of constructions, including small clauses, pied-piping, subject extraction from embedded sentences and argument cluster coordination, where the analysis found in the PTB is incompatible with CCG analyses and are therefore modified accordingly. This challenge relates to the more general problem that is found when attempting to align two different language models which attempt to provide analyses for the same language. This alignment problem arises again in Section 2.9.2 in the context of Bouma et al.’s (2015) work developing treebank queries designed to retrieve examples of syntactic constructions under construction within a descriptive grammar.

2.8.6 Construction-focused Parser Evaluation

Another area of work where syntactic phenomenon classification has played a prominent role is in construction-based parser evaluation. This involves evaluating the performance of parsers over distinct phenomena, as opposed to traditional approaches to parser evaluation, such as the PARSEVAL metrics (Black et al. 1991), which use a single aggregate score, combining constituents or dependencies from across the entire test corpus. A consequence of using a single aggregate score is that constructions which are found further down the Zipfian long tail—accounting for a small number of total dependencies—contribute very little to the final score. A parser might appear to perform well according to such measures, while having poor performance over relatively infrequent constructions (Hockenmaier and Steedman 2007, Rimell et al. 2009). This is potentially problematic for constructions that play an important role in determining the underlying predicate-argument structure of language. As discussed in Section 2.6.1, a notable family of constructions which falls into this category is that of long-distance dependencies—which was also targeted specifically by CCGBank, discussed in Section 2.8.5. When aggregated across all head-dependency relations in a corpus, these constructions only account for a very small proportion of dependencies, however on a per-sentence basis, this number becomes somewhat larger.\textsuperscript{64} Correct recovery of the predicate argument structure of these sentences is dependent on the recovery of the long-distance dependencies, meaning that parser performance over these types of constructions is potentially quite important for real-world applications.

In order to gain more insight into the ability of state-of-the-art parsers to recover long-distance dependencies, Rimell et al. (2009) constructed a corpus of around 700 sentences annotated with long-distance dependencies. The seven target constructions included in this corpus are: object extraction from a relative clause, object

\textsuperscript{64}Rimell et al. (2009) report seeing up to 10% of sentences within a subset of the PTB containing at least one of the six long-distance dependencies they investigate.
extraction from a reduced relative clause, subject extraction from a relative clause, free relatives, object *wh*-questions, right node raising, and subject extraction from an embedded clause. The corpus was constructed by selecting approximately 100 sentences for each construction, with the text being taken from the Brown and WSJ sections of the PTB, excluding the *wh*-questions, which used text sourced from the question data used by Rimell and Clark (2008). The selection of construction-positive sentences was performed by firstly extracting candidate sentences through the use of regular expressions constructed to match the PTB analysis of the respective constructions. These were then manually reviewed and the vetted sentences then annotated with the relevant dependencies. The annotations were performed using grammatical relations: head-based dependencies that have been argued to be better suited for parser evaluation than phrase structure trees (Carroll et al. 1998).

Five parsers deemed representative of then-current approaches for performing wide-coverage robust parsing were selected and evaluated with respect to their ability to recover the long-distance dependencies from the corpus. The results of these experiments showed that across all parsers and target constructions, accuracy was lower than would be achieved when aggregating all construction types, and dramatically lower in the case of some constructions such as subject extraction from an embedded clause and object *wh*-questions.

Bender et al. (2011b) extend the work of Rimell et al. (2009), also constructing a corpus of grammatical phenomena for parser evaluation, but investigating a wider range of phenomena considered to be difficult for state-of-the-art parsers, including local dependencies in addition to long-distance dependencies. Ten phenomena were selected in total, with the motivating criteria being that the involved dependencies should be subtle, with their correct recovery potentially requiring a richer linguistic model than might be found in statistical parsers. These were selected from across the four broad phenomenon categories of long-distance dependencies, non-dependencies, phrasal modifiers, and subtle arguments, with two of the three long-distance dependencies chosen to overlap with constructions targeted by Rimell et al. (2009). For details of the phenomena, see Bender et al. (2011b), which also includes a description of each phenomenon, including exemplars. Bender et al. also investigate the representation of these phenomena within the PTB by determining their intended representation from the PTB annotation guidelines and then use a treebank query tool to locate correctly annotated examples and also look for potentially incorrectly annotated examples. This showed that these phenomena had varying degrees of noise in their representation in the PTB, with four constructions in particular showing considerable amounts of inconsistency. Bender et al. observe that not only does this highlight the inherent difficulty involved in producing consistent manual annotations for complex constructions, but that it also has potentially negative ramifications for the performance of PTB-induced parsers over these constructions.

In order to construct the evaluation corpus, Bender et al. used a treebank containing 900 million tokens of Wikipedia text parsed with the LinGO English Resource
Chapter 2: Background

Grammar (ERG; Flickinger 2000, Flickinger 2011) (Flickinger et al. 2010), using the best derivation trees according to the parse ranking model. A set of candidate sentences was randomly selected from these derivations such that they included specific ERG identifiers that had been manually associated with the phenomena. These candidate sentences were then vetted, and 100 confirmed examples were selected for each phenomenon and annotated with the appropriate dependencies. Each example was doubly annotated with a reconciliation process used to resolve disagreements. If an example featured multiple instances of the phenomenon it was representing, all instances were annotated. If an example featured any phenomena from categories other than the one the example was representing, these were not annotated.

Seven parsers—again, chosen to be representative of state of the art parsing of English at the time—were used to parse the 1000 example sentences and the recall of different parsers was compared across the different constructions. As in the case with Rimell et al. (2009), performance across the parsers varied considerably, however even the best parsers failed to recover a large number of dependencies involved with phenomena that occur with some frequency within Wikipedia text.

The work of Rimell et al. (2009) and Bender et al. (2011b) differ from the other projects discussed in this section in that they both involve the explicit choice of a set of target grammatical phenomena based upon a set of criteria, which was informed by the application of parser evaluation. Broadly speaking, the target phenomena were characterised as being candidates for testing the performance of English parsers over complex constructions that might prove difficult for the correct identification of predicate-argument structure. In both cases, the methodology used for constructing the corpus involved taking the chosen grammatical phenomena, which were captured through prose descriptions, and deriving corresponding heuristics for the retrieval of instances of the phenomena. In the case of Rimell et al. (2009), this involved the use of regular expressions that were identified as being indicative of the PTB analysis of the target phenomena.\footnote{Rimell et al. (2009) note that a treebank query tool, such as TGrep2, could potentially have been used instead of regular expressions.} In the case of Bender et al. (2011b), this involved determining identifiers found in ERG parser output that are indicative of the target phenomena.

Both of these strategies for operationalising prose descriptions of syntactic phenomena into signatures within the respective representational formalisms resulted in sufficient noise within their results that a manual vetting stage was required. In the case of Bender et al. (2011b), the noise was the result of the parse selection model, since the candidate analysis for each sentence was taken to be the top-ranked derivation from the parse forest. In the case of Rimell et al. (2009), the noise was from the PTB annotations themselves, which is consistent with the findings of Bender et al. (2011b), who also performed an investigation into the representation of their target phenomena within the PTB, finding that four out of their 10 target phenomena were difficult to detect within the PTB due either to inconsistent application of the
This process of converting prose grammatical phenomenon descriptions into methods for retrieving instances of the phenomena being described—which featured in both the work of Rimell et al. (2009) and Bender et al. (2011b)—can also be seen as modelling part of the motivating context for this current research: embedding the retrieval of examples of syntactic phenomena within an descriptive grammar. Since these projects were not focused on the real-time retrieval of grammatical phenomena instead being directed towards the creation of annotated phenomenon corpora, within this context, the respective methodologies of Rimell et al. (2009) and Bender et al. (2011b) are limited by their resource-intensive natures. In both cases, manual oversight was required to both identify the signatures corresponding to each phenomenon as well as vet the output sentences in order to remove false positives. While these methodologies are unsuitable for driving queries which are executed online for a user of a descriptive grammar, they do serve to provide examples for how this task can be tackled in an offline manner as well as underscore the need for a more dynamic approach. I discuss research that directly tackles the problem of syntactic phenomenon retrieval in Section 2.9.

2.8.7 Summary

In this section, I identified and outlined a number of different computational linguistics projects that involve the notion of grammatical phenomena, and which involved the development of strategies for characterising or classifying grammatical phenomena. The relevance of each project from the perspective of the current research into syntactic phenomenon discovery is summarised below.

The approaches found in the TSNLP project (Lehmann et al. 1996, Oepen et al. 1997) and the work of Lehmann (2000) both involved the development of frameworks for classifying syntactic phenomena not tied to a particular formalism, but intended for use within the development of various computational linguistics projects, especially precision grammars. The top-down TSNLP methodology demonstrated the desirable property of supporting granular divisions in the phenomenon-space, however the methodology requires an onerous level of compliance in order to work. Lehmann’s framework also appears to be sufficiently flexible to support the classification of arbitrary phenomena, however the reductionist approach used to achieve this resulted in a framework that is at odds with the kinds of categories found in a descriptive grammar. More generally, there are significant overheads required to maintain such taxonomies of phenomena for use with precision grammars, and it is likely that the resource-intensive nature of these frameworks played a significant role in their lack of adoption.

The ESD project, which provides documentation for the representation of semantic phenomena within the ERG, stands out due to its use of a bottom-up data-driven approach to deriving phenomenon categories. Like the remaining projects outlined in
this section, the ESD associates distinct phenomena with fragments from the representational formalism found in parser output. The resulting signatures can be used to retrieve phenomenon-positive instances from a corpus parsed with the ERG. Since the ESD project involves the identification of semantic phenomena, this results in largely functionally characterised divisions in the grammatical phenomenon-space, which, as was illustrated by comparing phenomenon categories from the ESD and Huddleston and Pullum, cuts across divisions required for syntactic phenomenon discovery. The divergent query results also showed how queries based on function are complementary to those based on form, dividing up the phenomenon space into different groupings, which, when embedded in an electronic descriptive grammar, would enable different modes of enquiry. While I do not investigate semantic phenomenon discovery in this research, fully realised electronic descriptive grammars would ideally integrate both syntactic and semantic phenomenon discovery. Querying approaches similar to that found in the ESD, but which target constituent structure produced by parsers, are discussed in Section 2.9.

DepBank is interesting for its creators’ decision to include syntactic and morphosyntactic features along with the primary dependency annotations. While the input LFG f-structures that were used to derive the DepBank annotations are already a formalised representation, the entire pipeline from input sentences through to f-structures and then to DepBank annotations, which include syntactic and morphosyntactic features, can be construed as an operationalisation of syntactic phenomena. The loss of both phrase-structure and word order within the DepBank annotations, however, make this methodology for operationalisation unsuitable for supporting the discovery of a flexible range of syntactic phenomena.

CCGBank, similar to DepBank, did not explicitly set out to model or locate specific grammatical phenomena. In this case, a set of syntactic phenomena emerged as an epiphenomenon of the translation process used to convert PTB analyses into CCG analyses, with this set being characterised as constructions which needed special attention. The potential utility of the resulting phenomenon inventory is that each phenomenon is paired with a corresponding CCG syntactic category, which could be used as its signature. This inventory, however, as an artefact of the translation process, is essentially fixed to the current list of phenomena. The more general strategy of identifying phenomenon signatures within CCG analyses produced by CCG parsers could be adopted, however, in this research, I focus on the use of DELPH-IN HPSG precision grammars. In Section 5.4, I discuss the properties of DELPH-IN grammars that make them well-suited for this application.

The construction-focused parser evaluation of Rimell et al. (2009) and Bender et al. (2011b) both involved a process of taking a grammatical description of a phenomenon and using it to develop a procedure for retrieving instances of the phenomenon. Of all the work discussed in this section, this is most similar to the motivating context of syntactic phenomenon discovery within a descriptive grammar. The prose descriptions of each phenomenon used in these two projects could
be substituted with descriptions sourced from descriptive grammars. Freed from the original context of fine-grained parser evaluation, this methodology could be repeated with any desired syntactic construction, provided, of course, it is supported by the underlying representational formalism and found in the chosen corpus. These approaches, however, suffer from the limitation of requiring human supervision in order to vet candidate examples, meaning that they are less suited to contexts involving live queries. They are, however, informative in that they reveal challenges involved in this kind of pipeline. In particular, Rimell et al. (2009) shows that there is too much noise associated with PTB annotations in order for simple regular expression-based patterns to yield accurate results, which is also supported by Bender et al.’s (2011b) investigation into the representation of target constructions in the PTB. The methodology adopted by Bender et al. (2011b) reveals that the noise introduced by the ERG’s parse selection model also requires a manual vetting process of the candidate construction-positive sentences. I discuss existing work that tackles the problem of online phenomenon-based queries in Section 2.9.

2.9 Syntactic Phenomenon Discovery

In this section, I provide an overview of different methodologies which have been used to solve the problem of retrieving examples of a given syntactic phenomenon from a corpus. Similar to the selection of projects involving grammatical phenomena in Section 2.8, this list is not exhaustive, but serves as a representative sample of methodologies that have been used for the retrieval of syntactic constructions. I divide these into three broad approaches. The first involves the use of databases containing text that is pre-annotated with syntactic phenomena. This approach is premised on the existence of such databases and is limited to only retrieving existing annotations. In the context of the present research, the relevance of these lies not in the re-use of existing databases of syntactic phenomena, but rather in the nature of the underlying annotations within the database and the manner in which they are made available to the user. The second approach involves the querying of databases that are annotated with some form of linguistic structure. This requires the manual construction of queries that isolate the signature of the syntactic phenomenon within the representational formalism used to annotate the corpus. The last type of approach involves the use of a characteristic query-by-example methodology, in which the corpus is queried not using a phenomenon signature, but using an example already known to be a positive instance of the target phenomenon. In these approaches, the work of deriving a query which captures the signature of the target phenomenon is transferred from the user to the query tool.


2.9.1 Querying Pre-annotated Databases

The first approach towards syntactic phenomenon retrieval involves the use of preexisting databases that contain text annotated with instances of syntactic phenomena. Like treebanks, such databases typically involve manual annotation, making them resource-intensive to produce. It is therefore not the methodology for their construction that makes these databases interesting for the current purposes. Nor would the use of the annotations found in the databases be overly valuable for performing syntactic phenomenon discovery directly, as they would be limited to the retrieval of existing annotations, with the extension of the database to additional phenomena requiring further time-consuming annotation. The relevance of these approaches for the current research lies in the nature of the annotations themselves, including how they are modelled, and also how they are exposed to the user for discovery.

In this section I describe two tools from the DELPH-IN grammar engineering ecosystem which provide support for querying databases containing examples of text annotated with syntactic phenomena. I outline both how they model syntactic phenomena and also how they expose the annotations to users.

As discussed in Section 2.6.2, [incr tsdb()] (Oepen and Flickinger 1998, Oepen and Carroll 2000) is a tool that facilitates test suite management and competence and performance profiling over these test suites. It is predominantly used within the DELPH-IN consortium, however it has also been used by other computational linguistics projects for test suite management. Within [incr tsdb()], test suites are maintained through the use of profiles, which contain sets of items and information pertaining to how the grammar performed over the items. Each item is associated with a string from the target language, a flag indicating whether the string is grammatical or ungrammatical, as well as other metadata. Once the items in a profile have been parsed with a particular version of a grammar, the profile is populated with a range of information associated with the parse results. In addition to capturing information pertaining to parser operational performance, [incr tsdb()] is capable of storing all analyses returned by the parser for each item, ranked by the parse selection model.

[incr tsdb()] also includes support for the annotation of test items with grammatical phenomenon labels, a feature which has its origin in [incr tsdb()’s use within the TSNLP project described in Section 2.8.1. This means that it facilitates the development of phenomenon-based test suites which provide systematicity and progressivity over a portion of the phenomenon space. Phenomenon annotations are performed at the level of test items, meaning that annotation of sub-spans within the item text

---

66Subject to storage limitations. Since some input-grammar combinations yield pathological numbers of analyses, it is common to provide a cutoff for each item when storing parse results.
is not supported. Each test item can be annotated with the occurrence of multiple phenomenon instances, provided the phenomenon types themselves are already populated within [incr tsdb()]’s database.

[incr tsdb()] stores all information using a relational database and makes use of an SQL-like query language, tsq1, which is also exposed to users, enabling them to perform manual queries over test items using arbitrary constraints, such as whether an item was successfully parsed, its input string length, its rank given by the parse selection model, and, notably, whether it includes a given phenomenon. From the perspective of syntactic phenomenon discovery, [incr tsdb()] represents a convenient means of retrieving phenomenon-positive test items from a profile that has been already annotated with target phenomena. Its real value, however, is unlocked by leveraging the tsq1 query language to combine constraints pertaining to both parse results and the presence of phenomenon annotations. Table 2.5 shows a selection of tsq1 queries along with a description of what they would return, given a profile parsed with a particular grammar, starting with the simple retrieval of test items annotated with a specific phenomenon, and progressing to more complex constraints. Notably, this includes the retrieval of successfully parsed phenomenon-positive items, along with the highest ranked analysis according to the parse selection model (query 3), or—of even more utility—in the case where the profile has been disambiguated, the retrieval of successfully parsed phenomenon-positive items, along with the analysis marked as the gold tree, if one has been identified (query 4). This makes [incr tsdb()] a useful tool for retrieving the disambiguated analyses of phenomenon-positive items found within an already phenomenon-annotated treebank produced by a precision grammar.

Unfortunately, the phenomenon-annotation capacity of [incr tsdb()] has not seen much use since the TSNLP project, with the only extant phenomenon-annotated test suites being the Hewlett Packard test suites (Nerbonne et al. 1988) and those produced by the TSNLP project. This is likely a consequence of the time-consuming nature of developing and maintaining a high quality collection of such annotations, as I argued was also a problem with the TSNLP methodology discussed in Section 2.8.1. While a valuable tool for exploring phenomenon-annotated collections, with the very small number of such collections in existence, [incr tsdb()] is of limited value as a means of facilitating syntactic phenomenon discovery.

LTDB

The Linguistic Type Database (LTDB; Hashimoto et al. 2007) is a tool that assists with the documentation of the internal components of delph-in grammars. As a precision grammar’s coverage over grammatical phenomena increases, the inevitable corresponding increase in complexity means that it becomes harder for grammar engineers to keep track of how different phenomena are both analysed and implemented within the grammar. This challenge is compounded by the fact that much grammar
<table>
<thead>
<tr>
<th>tsql Query</th>
<th>Query Result</th>
</tr>
</thead>
</table>
| 1. `select i-input
where p-id=1`                                         | Returns the text of all test items recorded as containing instances of the phenomenon with ID 1.                                           |
| 2. `select i-input
where i-length>10 and p-id=1`                         | Returns the text of all test items which are greater than 10 words in length and are also recorded as containing instances of the phenomenon with ID 1. |
| 3. `select i-input derivation
where p-id=1 and result-id=0`            | For each test item which is recorded as containing an instance of the phenomenon with ID 1 and which the grammar was able to parse, returns the text of the test item and the derivation corresponding to the best analysis according to the parse selection model. |
| 4. `select i-input derivation
where p-id=1 and t-active>0`            | For each test item which is recorded as containing an instance of the phenomenon with ID 1 and has an analysis marked as the gold tree, this query returns the example text along with the derivation corresponding to the gold tree. Note that this query must be performed on a profile that has been both tree-banked and also had each item pruned to include only gold trees, a process referred to as thinning within the DELPH-IN consortium. |

Table 2.5: Sample tsql queries and descriptions of what they would return when provided to `[incr tsdb()]` to run over a profile that has been annotated with phenomena and parsed with a grammar.
Chapter 2: Background

Engineering is performed through distributed collaboration, with contributors often being located in disparate locations. The LTDB aims to improve this situation by providing grammar engineers with high-level summaries of salient grammar internal components, which document the grammatical phenomena that components are associated with. These summaries are made available in an online interface, enabling a distributed team of grammar engineers to rapidly gain a sense of how grammatical phenomena are analysed and implemented within the grammar.

The grammar components that the LTDB indexes are *types*, which are the primary abstraction that grammar engineers work with when constructing DELPH-IN grammars. Each type introduces zero or more constraints and inherits the constraints from one or more parent types. The combined set of constraints from these two sources associated with a single type can be used to capture a linguistic abstraction. The use of inheritance allows these abstractions to be re-used across the implementation of distinct grammatical phenomena that have similar properties. For example, the definition of the type `n_.pn-pl-def_le` from the 1214 release of the ERG is presented in Figure 2.14. This lexical type captures the constraints that apply to lexical entries that are proper nouns by inheriting from `basic_n_proper_lexent`, while also adding additional constraints, which includes one that makes the number of the resultant lexical entry plural. Figure 2.14 also illustrates how grammar engineers use inline annotation of type definitions to provide an entry within the LTDB’s database. This results in a form of literate programming, where source code and documentation are maintained alongside each other. A more detailed discussion of DELPH-IN grammar internals relevant to this research is presented in Section 5.3.

Within the browser-based interface of the LTDB, users are presented with a distinct view for each grammar type. Each type’s view includes one or more of the following entries—depending on whether they were recorded alongside the definition of the type: the name of the phenomenon the type is associated with, a brief prose description of the phenomenon, and a canonical example of the phenomenon. Additionally, the user is presented with the fragment of TDL that defines the type, and, if any treebanks have been configured for use, a sample of sentences whose stored analysis includes the type is also presented.

From the perspective of assisting grammar engineers with more readily determining the role of grammar components and establishing a more concrete understanding of which grammatical phenomena each type is constraining within the grammar, the LTDB is most valuable when used with disambiguated treebanks containing gold standard analyses, as the retrieved examples are then more likely to be indicative of how the grammar engineer intended the type to constrain the analysis.

If grammar engineers provide a high-level description of the phenomenon a type is associated with, as well as concrete examples of how the type is used within a

---

67In a previous incarnation, when known as LEXTYPE DB, the tool only indexed lexical types, the subset of types that capture reusable abstractions within the lexicon.
Chapter 2: Background

Figure 2.14: The definition of a single type from the 1214 release of the ERG, with the lines beginning with a semicolon forming the inline documentation of the type for use in the LTDB.

treębark, users of the LTDB can very quickly get a sense of how the analysis of different grammatical phenomena are implemented within the grammar. The inclusion of a name for the phenomenon in the database also provides a means of establishing the canonical name used in the project, encouraging terminological consistency, a problem that must be addressed in any electronic grammaticography project (Good 2012) and one which is especially pronounced within a distributed project with developers collaborating from across different sites. The phenomenon-centric nature of this documentation means that the LTDB can help grammar engineers view and navigate the grammar by phenomena—to the extent that the phenomena have been documented—thereby ameliorating the problem of precision grammars largely not being phenomenon-oriented, which was identified in Section 2.6.3.

Using the LTDB for grammar documentation requires grammar engineers to incorporate the creation and maintenance of documentation—in the form of inline type annotations—into their development cycle. The value that the resultant LTDB database provides will depend on the amount of information that is provided in the annotations and therefore on the amount of time dedicated to this part of the development cycle. While the rewards for such efforts are likely to be worth the investment, grammar engineers must be persuaded of this before adoption will occur. The LTDB is currently actively used by the ERG68 and the Jacy69 Japanese grammar (Hashimoto et al. 2008).

68See the ERG documentation: http://moin.delph-in.net/ErgTop.
69See the Jacy documentation: http://moin.delph-in.net/JacyTop.
Discussion

Comparing the way in which these two tools represent phenomenon annotations, \[\text{incr tsdb()}\] supports item-level annotation of phenomena, whereas the LTDB’s manual annotation is performed at the level of grammar internal types, with items whose parser output includes these types being implicitly annotated as having this phenomenon. From the perspective of producing candidate sentences for use in syntactic phenomenon retrieval, the LTDB’s annotation methodology is a more scalable approach than that of \[\text{incr tsdb()}\], since the LTDB’s item-level annotations can be derived automatically from any pre-parsed treebanks once the initial type-to-phenomenon mapping has been performed.

From the perspective of how phenomenon annotations are exposed to the user, by indexing precision grammar components and documenting the grammatical phenomena they are associated with, the LTDB represents a valuable contribution towards answering Bender et al.’s (2012) call for improved means of exposing precision grammars’ analyses of grammatical phenomena, which was discussed in Section 2.7.3. However, the LTDB does not enable discovery of the analyses for those unfamiliar with the grammar internals, as its interface presumes the user knows the types of interest in advance. Phenomenon discovery is also limited to those types that have been annotated by grammar engineers. In order to better answer Bender et al.’s (2012) call, an improved means of discovering the phenomenon annotations by those unfamiliar with the grammar is required, as are improved strategies for grammar engineers to use in applying phenomenon annotations to grammar components, in order to remove some of the burden of populating this type of documentation. This forms the topic of investigation for Chapter 5, which describes the development of a syntactic phenomenon discovery tool that is augmented by the LTDB.

Both the LTDB and \[\text{incr tsdb()}\] provide support for the retrieval of items pre-annotated with syntactic phenomena. In the case of the LTDB, these annotations take the form of implicit annotations consisting of the presence of the target type and are presented to the user within the view corresponding to the entry of that type. \[\text{incr tsdb()}\], on the other hand, enables users to execute queries for specific phenomena within a collection of items pre-annotated with a set of phenomena. It represents a useful tool for querying treebanks that have already been manually annotated with phenomena, especially when combined with additional constraints pertaining to parse results. However, the resource intensive nature of such annotation—especially when performed at the level of granularity stipulated by the TSNLP methodology—has resulted in few such resources being developed. In spite of this, there is still scope for the phenomenon annotation capacity of \[\text{incr tsdb()}\] to be useful. As discussed in Section 2.8.6, there has more recently been a resurgence in construction-focused parser evaluation, such as Rimell et al. (2009) and Bender et al. (2011b). In contrast to the comprehensive and fine-grained coverage over slices of the phenomenon space that the TSNLP project aspired to, these approaches involve the annotation of text collec-
tions which target a much smaller number of constructions and construction variants, focusing on a specific dimension of parser coverage. For such collections which are much more tractable from the perspective of manual annotation, \texttt{[incr tsdb()]} is then a convenient tool for storing syntactic phenomenon annotations such that they can be readily queried along with constraints pertaining to the output of a precision grammar. This is precisely the purpose it is used for in Chapter 4.

2.9.2 Signature-based Querying

The second broad category of approaches for locating examples of syntactic phenomena within a corpus involves the identification of patterns indicative of the presence of the target phenomenon within a representational formalism that captures syntactic structure. Typically these indicative patterns are matched against annotations found within an annotated corpus, such as a treebank, however it is also possible for annotations to be generated dynamically, for example, by parsing candidate sentences on the fly. I refer to this type of approach—which encompasses a broad range of more specific methodologies—as signature-based approaches for syntactic phenomenon retrieval, due to the common feature of the identification of a characteristic signature for each target phenomenon.

This general methodology has arisen in a number of different places within this chapter already. In order to provide an evaluation of the framework for classifying syntactic phenomena which was discussed in Section 2.8.2, Lehmann (2000) performs a text profiling exercise, attempting to count instances of syntactic phenomena within a corpus by means of queries defined as regular expressions over POS-tags. This approach is limited by the nature of POS-tag annotations, with many syntactic constructions requiring target annotations that capture richer linguistic information in order to be captured. Rimell \textit{et al.} (2009), as discussed in Section 2.8.6, also make use of manually constructed regular expressions to retrieve syntactic constructions, but target these queries at PTB trees, enabling more complex constructions to be captured than by simple POS-tag annotations. Rimell \textit{et al.} (2009) note that this approach yields many false positives, and consequentially requires a vetting stage. They note that a significant component of this noise arises from the fact that regular expressions are a general purpose querying language, rather than being intended to query linguistic annotations. Also described in this section, Bender \textit{et al.} (2011b) use identifiers from the output of the ERG that are deemed to be associated with target syntactic constructions, with candidate sentences that these queries produce being manually vetted. Related to this approach of using precision grammar components, the LTDB (Hashimoto \textit{et al.} 2007), discussed in Section 2.9.1 uses single grammar types as the signatures for distinct phenomena, which are used to query derivation trees stored in treebanks to locate phenomenon-positive instances. Another use of structures produced by precision grammars for sourcing phenomenon-signatures is found in the semantic encyclopedia developed in the ESD project, which is discussed
in Section 2.8.3. Within the catalogue, each phenomenon entry is associated with a semantic fingerprint, representing an underspecified MRS structure, which can be used to query a treebank annotated with MRS analyses. This is based off the semantic querying work of Kouylekov and Oepen (2014).

Perhaps the most prototypical form of signature-based approaches for retrieving instances of syntactic phenomena is the use of treebank query languages and corresponding tools, which are designed to allow users to specify constraints that target corpora annotated with representational formalisms capturing syntactic structure. As outlined in Section 2.5.2, this approach involves the use of a query language designed to capture syntactic constraints, software which is capable of executing the query over a treebank, and of course, a treebank annotated with an appropriate representational formalism supported by the querying language and tool. While there are numerous treebank querying languages and accompanying tools available, I present here just two tools that are representative of this general paradigm, along with, in each case, an example of their use for retrieving target syntactic constructions in the context of linguistic investigation.

**Tündra**

The Tündra query tool (Martens 2012) is a web application that provides support for browsing, querying, and visualising the contents of large-scale treebanks containing either constituency or dependency trees. Its querying language is an extended version of the popular TIGERSearch query language (Lezius 2002) mentioned briefly in Section 2.5.2. Hinrichs et al. (2015) demonstrate the use of Tündra for exploring the frequency of the Zwischenstellung phenomenon, a German construction that involves the position of finite auxiliary verbs in subordinate clauses, and which has seen some debate concerning the extent to which it is actually used by speakers.

In terms of the phenomenon itself, within the verbal cluster of a German subordinate clause, the finite auxiliary is usually located in the right-most position (25a) or in the left most position if the auxiliary governs a modal verb (25b). There are, however, some conditions under which it is attested that finite auxiliaries can occur in the intermediate position—to the right of the main verb and to the left of the non-finite auxiliary verb (25c). This is the Zwischenstellung construction. Hinrichs et al. (2015) investigate the occurrence of this construction within the Tübingen Partially Parsed Corpus of Written German (TüPP-D/Z; Müller 2004), a corpus containing 11.5 million tokens of German text taken from a German newspaper and parsed with a finite-state chunk parser.

(25)  a. dass Eike gesungen hat.

that Eike sung has.

“that Eike has sung.” [deu]
b.  *dass Eike hat singen können.*

“that Eike was able to sing.” [deu]

c.  *dass er arbeiten hat können.*

“that he has been able to work.” [deu]

Hinrichs *et al.* (2015) identify two forms of this construction and construct two corresponding queries capturing them, which are shown in (26). Within the TüPP-D/Z corpus, main and subordinate clauses are annotated with topological field labels that indicate the shape of the clause in terms of verbs, auxiliary verbs and subordinators—a feature that is helpful when managing the relatively free word-order of German. Both queries leverage this feature, constraining the results to contain subordinate clauses by targeting the topological field label \texttt{VCRAFVI}, which indicates a right-bracket verbal complex with a finite auxiliary and infinitive verb. This subordinator is constrained using the $>$ operator to immediately dominate the infinitive verb of the subordinate clause \texttt{VVINF}, which is constrained using the immediate precedence operator (.) to occur to the left of the finite auxiliary verb, which is restricted to the lexemes *haben* and *werden*, and is also constrained to occur to the immediate left of the final verb in the verbal cluster. This is where the two queries differ, with (26a) constraining this right-most verb to be an infinitive modal auxiliary verb that can take six different lexemes, whereas, in (26b), the right-most verb is restricted to being the infinitive form of the verb *lassen*.

\begin{verbatim}
(26) a.  [cat="VCRAFVI"] > #1:[pos = "VVINF"] & #1 . #2: [pos="VAFIN" & lemma = /haben|werden/] & #2 . #3:[pos="VMINF" & lemma=/müssen|können|dürfen|wollen|sollen|mögen/]

b.  [cat="VCRAFVI"] > #1:[pos = "VVINF"] & #1 . #2: [pos="VAFIN" & lemma = /haben|werden/] & #2 . #3:[pos="VVINF" & lemma=/lassen/] \end{verbatim}

Hinrichs *et al.* (2015) discovered that these queries failed to retrieve some instances of the Zwischenstellung construction due to noise in the automatic annotation process of the treebank and a number of additional supporting queries were also constructed in order to retrieve the missing instances. While these queries improved recall, they also decreased precision, retrieving some false positives, leading the authors to observe that, as a result of the noisy annotations within TüPP-D/Z, this particular querying methodology requires a final hand-vetting stage. Once the querying methodology was refined to include supplemental queries to improve recall and the results hand-vetted to remove false positives, Hinrichs *et al.* (2015) report that 92 instances of the Zwischenstellung construction were discovered. They conclude that this result both confirms the existence of this attested phenomenon within a corpus of naturally
occurring text and also emphasises the more general need to consult particularly large
corpora in order to investigate some phenomena.

PaQu

The PaQu query tool (Odijk et al. 2017) is a web-based application for querying
Dutch treebanks produced by the Alpino Parser (van Noord 2006). Users can select
a pre-parsed corpus or upload their own text corpus which is then parsed using
Alpino. It features a queryable grammatical dependency database based on the
LASSY Word Relations Search Engine, which enables the retrieval of word-pair
dependencies by allowing users to specify the desired grammatical dependency and
each word’s lemma, word form, or part of speech. More complex queries can be made
through a separate XPath query feature, which is based on the functionality of the
Dact desktop querying tool (van Noord et al. 2013) and constitutes the functionality
that I focus on here. In addition to retrieving sentences matching a user-supplied
XPath query, this feature also provides a visualisation of the syntactic dependency
graph associated with each result, as well as information on the number of matches
across corpora, including statistics over word properties and dependencies within the
matched results.

Bouma et al. (2015) describe using PaQu to annotate grammatical descriptions
found in Taalportaal—the online descriptive grammar of Dutch discussed in Sec-
tion 2.7.3—with queries that retrieve examples of syntactic phenomena. In their
methodology, PaQu is used to assist annotators in constructing queries as well as
being used to execute the queries within the published resource when users follow
such links. Bouma et al. (2015) illustrate this use of PaQu through queries that
were developed to retrieve examples demonstrating the linear order of adjectives that
take prepositional complements in Dutch. The following excerpt from Taalportaal
contains a description of this phenomenon, which is also accompanied by the two ex-
emplars shown in (27) that illustrate how prepositional complements can occur both
before and after their head adjective. Within the online resource, both the prose
description and the pair of exemplars are annotated with illustrative queries such as:

Adjectives typically select a PP as their complement. Although this PP-
complement can often either precede or follow the adjective, it is normally
assumed that its base-position is the one following the adjective, whereas
the pre-adjectival position is derived by leftward movement.

---

70 http://zardoz.service.rug.nl:8067/info.html
71 http://portal.clarin.nl/node/1966
72 As briefly discussed in Section 2.5.2, XPath is a general purpose XML querying language.
(27) a. Jan is ⟨over die opmerking⟩ boos ⟨over die opmerking⟩.  
   Jan is about that remark angry  

   b. Jan is ⟨over zijn beloning⟩ boos ⟨over zijn beloning⟩.  
   Jan is about his reward satisfied

(28) shows the two XPath queries Bouma et al. (2015) use to illustrate this phenomenon. (28a) selects all adjectival heads which have a prepositional complement, retrieving examples in which the preposition occurs both before and after the adjectival, whereas (28b) restricts this query such that the preposition must precede the adjective. Bouma et al. (2015) note that when run over the Lassy Small corpus (van Noord et al. 2013), the query retrieving examples with pre-adjectival prepositions amounts to around 10% of the total occurrences of adjectives taking prepositional complements, supporting the claim in Taalportaal that the post-adjectival preposition is the more frequently occurring variant of the construction.

(28)  

a. //node[@cat="ap"]/  
    node[@rel="hd" and  
    @pt="adj" and  
    ../node[@rel="pc" and  
    @cat="pp"]  
    ]

b. //node[@cat="ap"]/  
    node[@rel="hd" and  
    @pt="adj" and  
    ../node[@rel="pc" and  
    @cat="pp"]/number(@end) = number(@begin)  
    ]

Bouma et al. (2015) created 1200 queries such as those presented in (28) for annotating grammatical descriptions and exemplars within Taalportaal. As already discussed in Section 2.7.3, Bouma et al. (2015) note two particular challenges that were encountered during development of the queries. Firstly, query developers found that it was not always clear how desired constraints are expressed in the XPath query language, an accessibility problem that, as discussed in Section 2.5.2, is often attributed to treebank querying tools and their unintuitive query languages. The second challenge annotators encountered lay in discovering how a target phenomenon is represented within the Alpino-derived treebanks, without initial access to a set of already treebanked examples of the phenomenon. This difficulty was compounded by the fact that the syntactic analyses found in the treebanks can differ substantially from the analyses found in Taalportaal, with different distinctions being made across the two resources, and sometimes even incompatibilities.

An example of an analytic mismatch between Alpino-compatible analyses and Taalportaal is provided by Bouma et al. (2015), who observe that Taalportaal anal-
yses nominal complements of adjectives as taking both genitive and dative forms, as exemplified by (29a) and (29b), respectively. However, as illustrated by their syntactic analyses—presented in Figure 2.15a and Figure 2.15b—these two constructions are analysed differently from each other within the Alpino-derived treebanks. Figure 2.15a shows that the Alpino analysis of (29a) is consistent with the Taalportaal analysis, with the noun phrase *dat probleem* being analysed as a complement of the adjective *bewust*, however Figure 2.15b shows the analysis to differ from Taalportaal’s, with *Peter* forming an indirect object of the verb, alongside the predicative complement of *niet duidelijk*. For an annotator trying to use the description found in Taalportaal in order to construct queries targeting both variants of nominal complements of adjectives that it identifies, the annotator will likely fail to retrieve examples corresponding to (29b). Furthermore, there is no way of ascertaining whether the query was unsuccessfully crafted to match the representation of this phenomenon or if the treebanks being targeted by the query simply don’t contain any instances of the phenomenon.

(29)  a. Jan is zich *dat probleem* bewust.
     Jan is REFL that probém_{ACC} aware.
     *John is aware of that problem.*

     b. Het probleem werd *Peter niet duidelijk*.
     the probém became Peter_{DAT} not clear.
     *The problem didn’t become clear to Peter.*

In order to overcome the difficulties associated with establishing the appropriate XPath constraints and to assist with the discovery of the representation of target phenomena within the treebanks annotators were encouraged to use the treebank querying tool GrETEL, which uses a query-by-example methodology allows annotators to provide unannotated phenomenon-positive examples as input. GrETEL is discussed is further in Section 2.9.3.

Discussion

The overarching challenge faced by all signature-based approaches towards the retrieval of syntactic phenomena is the need to generate the signature for each phenomenon being investigated. In the context of developing signatures for use with treebank querying tools, Bouma et al. (2015) observe there are two notable challenges associated with this activity. The first is that this process requires the user to be familiar with the query language being employed in order to successfully convert syntactic constraints into appropriate queries. The Tündra and PaQu queries in (26) and (28) show that, even for relatively simple phenomena, the corresponding queries can be arcane in appearance and difficult to read. Bouma et al. (2015) report that,
(a) The phrase *dat probleem* is analysed as a complement of the adjective *bewust*.

(b) *Peter* is analysed an indirect object of the verb *werd*.

Figure 2.15: The divergent Alpino analyses for (29a) and (29b), which, within Taalportaal are both analysed as having adjectives with nominal complements.

Even after some practice, query annotators often found it difficult to determine the needed XPath constraints for PaQu. This is reflective of the fact that such query languages are designed for consumption by machines rather than humans. In addition to reducing the efficiency of the query construction process, this lack of user-friendliness forms a barrier to the adoption of treebank querying tools, especially in the context of use by linguists and other candidate users of such tools, many of whom will have not had experience working with programming languages (Muralidharan and Hearst 2014). These barriers are further compounded by the lack of standardisation that is found in the fragmented range of treebank query languages and target treebank
representations (Augustinus et al. 2012).

The second challenge identified by Bouma et al. (2015) is that users need also to be familiar with how a target phenomenon is represented within the treebank in order to create queries which will retrieve positive instances. This creates something of a circularity, since the best way to ascertain how a phenomenon is represented is to look at already annotated examples, however the user of the tool may not yet have a range of such examples, with the query they are in the process of creating being the very thing that would provide this missing information. This challenge of discovering the representation of a phenomenon is made more challenging by the fact that the analyses found within the treebank may not make the same distinctions as assumed by the user constructing the query. This is notably the case when aligning the analyses found in a resource with the complexity of a precision grammar to another source of analyses, where mismatches and even incompatibilities are inevitable, as shown by Bouma et al. (2015). In the following section I identify and discuss tools for querying treebanks and corpora that try to alleviate the requirement of having to be familiar with a querying language as well as reduce the need for already having a sense of how grammatical phenomena are represented within the treebank being targeted.

Where the aforementioned challenges have been navigated and a query has been developed that targets a specific phenomenon, a further issue that must be considered is the accuracy of the query. A common element of all the projects I identified as adopting a signature-based querying approach is the involvement of a component of manual supervision, which often forms a resource-intensive component of the adopted methodology. This manual supervision can come at different stages of the discovery procedure. In the case of Rimell et al. (2009) and Bender et al. (2011b), due to their goals of developing a once-off methodology for constructing a high quality annotated corpus that is intended to be reused, they both adopt the approach of creating simple signatures that yield low-precision results, followed by a manual vetting stage. In the work of Bouma et al. (2015) on the other hand, since it is the signatures themselves that are stored for downstream use, the manual oversight is focused on improving the quality of the queries so that their subsequent use will yield higher precision results. However, as demonstrated by Hinrichs et al.’s (2015) investigation using Tündra, if the source of the annotations being queried is inherently noisy, then a manual vetting process over query results may be necessary in spite of significant effort being dedicated to query construction. More generally, it is clear from observing these different projects that the quality of results produced by signature-based approaches must be closely monitored.

2.9.3 Query-by-example

In this section, I discuss strategies for syntactic phenomenon retrieval that aim to overcome the limitations associated with treebank querying tools that were identified in the previous section. These limitations include the need to be familiar with often
One approach that has been used to make treebank querying tools more accessible is the incorporation of graphical interfaces which allow users to form queries in terms of familiar phrase-structure or dependency representations through the use of visual representations of target syntactic constraints, which are converted into the corresponding query in the query language. Augustinus et al. (2012) identify SPLICR (Rehm et al. 2008), TIGERin (Voormann and Lezius 2002), and SearchTree (Nyggaard and Johannessen 2004) as examples of treebank querying tools which support this kind of graphical querying. While such tools help overcome the accessibility issues of having to be familiar with a query language, they don’t help with the second problem of the user needing to have an understanding of how the target phenomenon is analysed and represented within a treebank in order to arrive at an appropriate query (Augustinus et al. 2012). Another approach, that aims to address precisely this difficulty, is often referred to as *query-by-example*,74 and is characterised by the basic pattern that the user first provides an example known to exhibit the target phenomenon and then, with the aid of the tool, identifies structural components of the example pertinent to the target phenomenon. A query engine then takes this input and converts it into a corresponding query which can be applied to a corpus. By using an example known to illustrate the phenomenon as a starting point, this methodology removes some of the burden of having to be familiar with how the target phenomenon is represented within a treebank.

In the remainder of this section, I identify and discuss four approaches for syntactic phenomenon retrieval that use a query-by-example methodology. As will be seen, query-by-example approaches can also include a graphical component to their interface, and have been used to develop tools which target treebanked corpora as well as unannotated corpora that contain only text.

**The Linguist’s Search Engine**

The Linguist’s Search Engine (LSE; Resnik and Elkiss 2005) was designed to assist linguists in locating instances of grammatical phenomena within text on the Web in order to gather data adding support for or against a linguistic hypothesis. It was argued that an increasing number of linguists were already turning to Web search engines to gather data, however such tools offer limited utility for this application, since search engines are unable to support even simple grammatical constraints. Users interact with the LSE by providing an example sentence containing the phenomenon they wish to investigate. The LSE automatically parses this sentence and then presents the user with a graphical representation of the syntactic analysis, which the user is able to edit. Users can remove portions of the tree that are irrelevant to the...

---

74The term *query-by-example* was first used by Zloof (1975) to describe a technique used to query relational databases through partially filled tables, protecting users from needing to use SQL.
target phenomenon and also apply additional constraints, resulting in a generalised structure, which the user instructs the LSE to convert into a query to run against corpora that have been automatically parsed and indexed by the LSE.

The LSE includes a built-in collection of documents randomly selected from the Internet Archive, but also has the capacity to generate custom collections by scraping the results of user-submitted search engine queries. This allows users to construct custom collections with documents known to contain target words, in order to facilitate the investigation of infrequently occurring phenomena. Resnik et al. (2005) describe two examples of linguistic research that make use of the LSE, one investigating the processing of backwards anaphora\textsuperscript{75} within text, and the other investigating the syntactic construction of comparative correlatives\textsuperscript{76}.

The LSE employs a relatively simple English parser, however, Resnik and Elkiss (2005) claim that the quality of the analyses it returns is not necessarily that important for its ability to locate similar sentences within a corpus. They argue that even if the user does not agree with the linguistic soundness of the structure assigned to the input string, the same analysis will be found across both user input and the indexed corpora, meaning that the LSE would be able to locate sentences within the corpus that are structurally similar to the input. While it is true that the consistency of analyses across user input and corpus sentences yielded by the use of the same parser will enable the retrieval of relevant examples in the face of incorrect analyses, this does not render the quality of the analyses irrelevant. In Section 3.2.1, I argue that in order to support a nuanced range of potentially complex syntactic phenomena—which is certainly necessary in the context of augmenting descriptive grammars—the granularity and complexity of the model being used to generate features for syntactic phenomenon discovery is an important consideration.

**Query by Annotation**

Bird and Lee (2007) present a methodology for treebank search that they describe as query-by-annotation, but which still falls within the query-by-example paradigm. Under this methodology, the user starts with a tree from an existing treebank which is presented to them in a visual interface. They then annotate this tree so as to indicate the desired relationships and constraints of the target phenomenon they wish to retrieve examples of. In terms of available annotations, users are able to specify different types of relationships between node pairs, such as the vertical ancestor and descendant relationships, the horizontal preceding and following relationships, and sibling relationships, all of which have both immediate and unbounded versions. Users can also filter nodes in or out based on label and attribute values. Lastly, users can apply scopal restrictions to portions of the query as well as being able to apply left

\textsuperscript{75}When the nominal that it refers to occurs before an anaphoric word.

\textsuperscript{76}An example of the comparative correlative found using the LSE: *The darker the coffee bean, then the less caffeine* (Resnik et al. 2005).
or right alignment restrictions of subtrees relative to an ancestor node. As the user
annotates the query, the tool automatically produces an LPath\textsuperscript{77} query, which is kept
synchronised with the state of the current annotations. When the user is happy with
their annotations and runs the query, the search tool translates the LPath query into
an SQL query, which is then run against a database of treebanked sentences. The
results of the query are presented to the user as trees, with the original annotations
that the query was generated from being overlaid over the top of each tree. The
Fangorn treebank querying interface (Ghodke and Bird 2012) also incorporates the
query-by-annotation methodology into aspects of its interface, providing the facility
to use trees returned from a previous query as a means of visualising the previous
query and then to create a new one by modifying the visual annotations. I use
Fangorn to augment the syntactic phenomenon discovery methodology developed in
this research, which is described in Chapter 5.

In addition to providing a high-level interface that removes the difficulty of hav-
ing to interact with a complex query language—just as other graphical querying
approaches do—Bird and Lee (2007) argue that this methodology confers a number
of additional advantages. The annotation of an existing tree differs from other graph-
ical querying approaches, which require users to construct queries from scratch. This
approach offers a more accessible query construction process, providing users with a
starting point from which to develop their queries, giving them a sense of how their
target phenomenon might be represented in the treebank and a convenient way of
constructing the query directly on top of this example. Since the results of the query
are trees annotated with the original query, it is also easy for the user to visualise
why the particular results were returned. The ability to modify the annotations of
results and then run the query produced by these modified annotations also makes
it easy for users to successively refine their query, which represents a more common
mode of treebank querying, as opposed to the once-off construction of a single query.

The query-by-annotation methodology developed by Bird and Lee (2007) is clearly
an improvement over graphical treebank querying approaches which require users to
construct visual queries from scratch. There is however the significant limitation that
the user must first locate a treebanked sentence in order to begin the querying process.
Again, this introduces a circularity, where the user must locate a treebanked sentence
that is an example of the phenomenon in order to construct a query that can locate
examples of the phenomenon. This methodology is then somewhat more constrained
when compared with the LSE, which was able to take unrestricted text through the
use of a parser. Bird and Lee’s (2007) methodology could be extended to incorporate
text-based input with the use of a parser that is capable of producing output in the
same representational formalism as used in the target treebank. This, however, could
introduce a potential concern regarding the quality of the analyses produced by the
parser, given that they will not have the advantage of going through a treebanking

\textsuperscript{77}A linguistically motivated query language, which was discussed briefly in Section 2.5.2.
process, and the users of the tool will not necessarily have had experience in vetting their appropriateness. When used with already treebanked sentences—which will have undergone a quality assurance process and thereby be more likely to have an analysis that corresponds to the intended representation within the treebank—the query-by-annotation methodology is more likely to result in a user developing queries that successfully retrieve examples of their target phenomena. This highlights that there is a potential trade-off between the flexibility of input methods to query-by-example approaches and the quality of input analyses used to derive query signatures.

**GrETEL**

Another tool that uses the query-by-example methodology is GrETEL (Augustinus et al. 2017), which was used by Bouma et al. (2015) in addition to the PaQu treebank querying tool to develop the treebank queries used to annotate the Dutch grammatical descriptions within Taalportaal, the online descriptive grammar described in Section 2.7.3.

A user of GrETEL provides as input an example Dutch sentence known to exhibit the target phenomenon. The sentence is parsed with the Alpino Dutch parser (van Noord 2006) and the user then has the opportunity to view the parser’s analysis and modify the input sentence if needed. The user is then presented with an interface which allows them to specify the aspects of the sentence pertinent to their target phenomenon. For each word in the sentence the user can select whether they wish to constrain retrieval to sentences with corresponding terminals that: exactly match the surface form of the word, match its lemma, just match its part of speech—or alternatively, words can be left unconstrained. GrETEL’s query engine takes these supplied constraints over terminal nodes and constructs a corresponding XPath query, adding all ancestor nodes within the implied subtree, along with constraints over both their dependency relationships—such as obj, mod, and hd—and their part of speech. Users also have two options to make the generated query more general. One is to lift the constraint that terminals appear in the same order as found in the example sentence, the other is to remove constraints applying to the properties of the dominating node of the subtree. The user then selects which treebank they wish to apply their query to. They have the option of searching the entire LASSY Small Treebank (van Noord et al. 2013), which contains 65 thousand sentences, or they may restrict their query to one or more of its sub-corpora. Finally, users can opt to be taken straight to their query results, or they can select an advanced mode, where they have the option to first modify the resultant XPath query. Augustinus et al. (2012) demonstrate GrETEL’s ability to retrieve sentences that display a target syntactic phenomenon by investigating the two phenomena of Dutch nominalisations and the position of separable verb particles in Dutch subclauses.

Comparing GrETEL with the other approaches I have identified as using the query-by-example methodology being for retrieving examples of syntactic phenomena,
a number of observations can be made. In accepting unrestricted text as input, GrETEL offers the same flexibility as the LSE, in that it is not restricted to using already treebanked examples as input, as Bird and Lee’s (2007) query-by-annotation approach is. By using the Alpino precision grammar, however, GrETEL is able to capture much more complex syntactic phenomena than would be available to the LSE, which uses a relatively naive parser (Resnik and Elkiss 2005). This capacity is also premised on the existence of the LASSY Small Treebank, which contains syntactic annotations produced by Alpino. GrETEL’s methodology is also similar to Bird and Lee’s (2007) approach insofar as they both enable the user to select the desired components of a given syntactic analysis, which a query engine converts into a corresponding treebank query. In adopting a rich graphical approach, Bird and Lee’s (2007) approach enables a much more expressive range of syntactic constraints to be specified than is made possible through the simple feature-based grid layout that GrETEL exposes. Users of GrETEL who wish to apply more complex restrictions must use the advanced mode and make manual modifications to the generated XPath query.

This arguably places GrETEL in between the LSE and Bird and Lee’s (2007) approach, in terms of accessibility and expressivity. GrETEL offers the accessibility of the LSE through text-based input supported by on-the-fly parsing, while also being able to capture more complex phenomena. Bird and Lee’s (2007) approach, while not being as immediately accessible in terms of input example discovery, allows users to much more quickly construct queries capturing more complex phenomena, assuming an appropriate treebanked example of the phenomenon has been located to use as a starting point.

The TIGER Corpus Navigator

The TIGER Corpus Navigator (Hellmann et al. 2010) is another treebank querying tool that uses a query-by-example approach. Two salient features distinguishing it from other query-by-example tools are that users provide both positive and negative examples of the target phenomenon and that the query engine uses an active learning approach to convert the input examples into a query.

The TIGER Corpus Navigator targets the TIGER Corpus (Brants et al. 2004), a corpus of German newswire text containing 50 thousand sentences that have been semi-automatically POS-tagged and syntactically annotated. Rather than targeting TigerXML—the XML-based representation format used by the TIGER Corpus, and which is queryable by the TIGERSearch tool (Lezius 2002) discussed in Section 2.5.2—the TIGER Corpus Navigator requires the TIGER corpus to be converted into an OWL DL$^{78}$ representation—a variant of the triple-based RDF data model,

$^{78}$OWL (Web Ontology Language; https://www.w3.org/TR/owl-ref) is a family of semantic markup languages designed for publishing and sharing ontologies on the Web and is an extension to the Resource Description Framework$^{79}$ (RDF) data model, which uses subject-predicate-object
which supports inference through the use of automated reasoning tools. The TIGER Corpus Navigator makes use of the DL-Learner tool (Lehmann 2009), which uses an active learning methodology, taking a sample of positive and negative instances of the target concept as input and inducing an OWL class intended to capture the underlying concept the input instances were chosen to delineate. Here, these concepts correspond to grammatical phenomena. The use of OWL DL also means that a range of linguistic ontologies can be incorporated through the OLiA project (Chiarcos and Sukhareva 2015), which has produced a range of interoperable ontologies for linguistic annotation using OWL DL models.

In order to retrieve example sentences of a phenomenon, the user first locates candidate sentences from a treebank by searching for specific words or lemmas. They then select resulting sentences as either positive or negative examples of the phenomenon, and when satisfied, the user submits the input instances to the DL-Learner to induce an OWL class, which is displayed to the user. The user can then retrieve sentences from the corpus that match this class, which the TIGER Corpus Navigator does by converting the induced class into a SPARQL\(^{80}\) query. From these results, the user can then select additional positive or negative training instances, repeating this process until they are satisfied that the induced OWL class captures their target grammatical phenomenon. In order to facilitate the capturing of the target syntactic phenomenon, when an OWL class is generated, the user is also presented with the accuracy of the results returned by the class when used as a query over the training sentences, as well as the number of matching sentences from the corpus.

Hellmann et al. (2010) describe using the TIGER Corpus Navigator to develop queries corresponding to two types of German passive clauses, inducing an OWL class for each of the two passives. To simulate the use of the tool by a user, they developed a methodology which involved, for each target category, randomly selecting five positive instances and five negative instances from pre-labelled training data to be used as input into the DL-Learner, and then repeating the process for a total of four iterations. By looking at the precision, recall, and f-scores across the sentences returned when using the generated OWL classes to query the corpus, Hellmann et al. (2010) found that the TIGER Corpus Navigator was able to learn classes corresponding to the two grammatical phenomena. In general, recall for both variants of the phenomenon was high due to the nature of the DL-Learner’s algorithm—which discards classes that do not match all positive examples—however precision was lower, particularly so for one of passives, which had far fewer positive instances in the training data. Hellmann et al. (2010) argue that the high recall means that this approach would still be useful as it would still be feasible for the user to look through a sample of results, discarding

\[^{80}\]http://www.w3.org/TR/rdf-sparql-query

triples to model relationships between entities. The OWL DL sublanguage places restrictions on OWL expressions, making it a valid description logic—a family of formal languages used for representing knowledge and performing inference—meaning that automated reasoning tools can be used to perform inference on OWL DL expressions (Grau et al. 2008).
the false positives.

While Hellmann et al. (2010) demonstrate the ability of the TIGER Corpus Navigator to capture two types of the German passive construction, it’s not clear whether this would extend to more complex syntactic constructions given the available linguistic features within the linguistic data models from the OLiA project. Further empirical work is needed to assess the tool’s capacity in this respect. Additionally, Hellmann et al. (2010) observe that a notable drawback of the tool is that the iterative process of arriving at the final OWL class through selection and refinement of positive and negative examples can be somewhat time-consuming. They argue that the TIGER Corpus Navigator could complement existing treebank search tools in the context of a user investigating the suitability of existing corpora for use in a project, where it would help remove the difficulty of having to become familiar with the query language and theoretical underpinnings of the treebank before being able to assess the appropriateness of the treebank. Once the choice of corpus is finalised, the user could then invest time in becoming familiar with the relevant querying tools, which may be more efficient once mastered.

Discussion

Looking at the four different treebank search tools described in this section, which all use a form of query-by-example methodology, it is helpful to compare these over two dimensions: whether input text and corpus text is already assigned syntactic analyses or not and also how users manipulate input examples to assist the query engine in locating the desired instances in the corpus.

In terms of the text used, of the four approaches, the methodology employed by the LSE stands out as being distinct due to its capacity to query unannotated corpora, with the remaining three approaches all targeting pre-annotated treebanks. This property of the LSE affords users with a considerable degree of flexibility, opening up the possibility of querying diverse corpora that would be otherwise unavailable due to the significant overheads involved in constructing new annotated treebanks. This, however, comes at the cost of noisier results, as this feature is enabled through the use of a simple parser whose output is unvetted, meaning that the produced analyses have not had the benefit of a quality assurance process that annotated treebanks are typically subject to. The use of such a simple parser also means that it would likely struggle to capture more complex syntactic constructions. The approach used by GrETEL retains a component of the flexibility of the LSE by allowing users to supply arbitrary examples which are parsed dynamically, while matching examples are still retrieved from a treebank of higher quality analyses. In the case of both the LSE and GrETEL, users are presented with the analysis produced for their input sentence and can choose a different example if they are not satisfied with the parser’s analysis. GrETEL’s use of treebanks constructed from analyses produced by the Alpino precision grammar means also that it is likely to be able to capture more
complex syntactic constructions than the LSE. Bird and Lee’s (2007) methodology and the TIGER Corpus Navigator, on the other hand, both require input examples to be selected from the target treebank. This improves the quality of the input analyses, at the cost of restricting the input examples available to users, as well as necessitating a preliminary discovery phase, in which candidate examples must be first located. Provided that there exists a parser capable of producing analyses consistent with the target treebank, both these approaches could be extended to support user input which is then parsed online, in the same fashion as GrETEL.

The other dimension which these tools can be compared is how users manipulate input examples to assist the query engine in arriving an appropriate query for the target phenomenon. Both Bird and Lee’s (2007) approach and the LSE make use of a graphical interface for users to manipulate an input example, however, since Bird and Lee’s (2007) methodology supports the complete range of constraints found in LPath, it offers a much more expressive interface. This means that, when applied to a treebank containing rich analyses, Bird and Lee’s (2007) query-by-annotation methodology has the capacity to capture more complex and nuanced grammatical phenomena. The query-by-annotation methodology also enables the iterative refinement of queries by presenting the query results as trees annotated with the original query, which can be modified and subsequently queried again. GrETEL does present users with a graphical representation of the analysis Alpino generates for their input example, however this is only used to assess the appropriateness of the analysis and cannot be interacted with. Instead, users select from a grid of features, which correspond to tokens in the input example, at different levels of linguistic abstraction. The user interaction found within the TIGER Corpus Navigator, on the other hand, does not involve the addition or modification of constraints over individual input examples. Instead, users refine the quality of the generated query by adding or removing positive and negative examples. This indirect iterative process, where the user can only add or remove examples and not interact directly with extracted features from the examples, potentially results in a comparatively longer query development process, as was observed by Hellmann et al. (2010), but could reduce the cognitive burden on the user compared with the other approaches.

Common to all of these approaches is a process of input manipulation and refinement. This potentially time-consuming process is one disadvantage of the query-by-example approach to querying corpora for syntactic phenomena (Hellmann et al. 2010, Muralidharan and Hearst 2014). In effect, the query-by-example methodology involves exchanging the upfront burden of having to become familiar with a query language for a more involved querying process. Another trade-off is the potential loss of accuracy incurred by the user not having direct control over the generated query, with the exception Bird and Lee’s (2007) approach, whose graphical interface offers the full expressivity of LPath expressions. As noted by Hellmann et al. (2010), these trade-offs are more likely to be cost-effective for initial exploration of treebanks, however the longer an investigation proceeds, the more likely it is to make sense for a
user to commit to investing the time required to become familiar with the supported query language, as this will ultimately enable more rapid querying.

For some applications, however, it may be unreasonable to require users to be familiar with a treebank query language. For example, in the context of a linguist reading an electronic descriptive grammar with the aim of answering a specific question about a syntactic phenomenon, such a barrier to entry would likely result in the lack of adoption of the discovery features. Furthermore, even in the context where a user has had the opportunity to invest time in becoming well-versed in the query language, variation in how grammatical phenomena can be analysed and represented across different treebanks means that, in order to construct queries that successfully capture a phenomenon, the user must first be aware of how it is represented within the annotation scheme of the treebank being queried. For these reasons, there will then still be a role for query-by-example tools in assisting with the discovery of how syntactic phenomena are represented within a particular treebank. Additionally, the slow and often challenging process of query refinement found within these tools suggest that there is also a need for improving their efficacy. Bouma et al. (2015) report that even when annotators are equipped with the GrETEL query-by-example tool, they still found that navigating the alignment between Alpino treebank analyses and the analyses from the Syntax of Dutch descriptive grammar proved to be a difficult process, with different annotators often arriving at different queries for the same phenomenon.

There is, then, still room to improve query-by-example methodologies. In particular, a tool that is able to better assist with discovering the representation of syntactic phenomena within a treebank in both an accessible and rapid manner would be of considerable utility. Furthermore, the use of such a tool which targets treebanks produced by precision grammars is also consistent with Bender et al.’s (2012) call for improving the discoverability of analyses produced by precision grammars in order to make them more accessible to linguists. In Chapter 5, I describe the development of a query-by-example methodology and accompanying implementation that attempts to address these issues.

2.9.4 Summary

In this section, I outlined three broad categories of tools used to locate and retrieve instances of syntactic phenomena from a corpus. The two tools described in Section 2.9.1—[incr tsdb()] and the LTDB—both make use of a database of examples that have already been labelled with syntactic phenomena. Since they assume a set of pre-existing phenomenon annotations, their relevance for this research lies not so much in how these annotations were discovered and collected, but in how they are made available to the user. The two tools described in Section 2.9.2—Tündra and PaQu—are representative of traditional treebank search tools, which involve the construction of a query capturing the desired syntactic phenomenon within the target
Chapter 2: Background

treebank. As discussed, two challenges associated with the use of these tools are that users are required to be familiar with the relevant query language associated with the tool, which can suffer from accessibility issues, and that users also must have a sense of how syntactic phenomena are represented within the target treebank, which introduces a circularity in the discovery process. The query-by-example methodology, which was discussed in Section 2.9.3, allows users to retrieve instances of syntactic phenomena by providing examples of the target phenomenon along with additional feedback that a query engine converts into a precise treebank query, thereby removing the requirement of having to be familiar with a query language. By providing additional means of exploring an annotated corpus, these approaches also improve the discoverability of how analyses within a treebank are represented, however, as reported by Bouma et al. (2015), users of the GrETEL tool, which makes use of the query-by-example methodology, still found this discoverability issue to be a barrier to the construction of queries that targeted specific syntactic phenomena.

2.10 Chapter Summary

In this chapter, I outlined the background to this research in order to both motivate its goals as well as ensure familiarity with the requisite concepts. I also discussed a range of work within the literature that this research is informed by or builds upon. In Section 2.2, I established that, due to the focus on implemented computational grammars of language, this research falls largely within the field of computational linguistics. I also briefly outline three notable subfields of linguistics that are implicated—descriptive linguistics, formal linguistics, and linguistic typology. A central concern of these subfields, and also of the research of this thesis, is the notion of a grammatical phenomenon, a term frequently used within linguistics as a catch-all to refer to a nonspecific feature of language. In Section 2.3, I provided a more concrete definition of this term for the purposes of this research, and explore some of its salient properties, namely: grammatical phenomena can be characterised in terms of both form and function, and that different characterisations result in different divisions of the phenomenon space; grammatical phenomena can have a high degree of complexity, involving facets from across the different language subsystems, which typically interact with each other; and that the distribution of grammatical phenomena within language use has a long tail, which is often referred to as being Zipfian in nature. In this section, I also narrowed the scope of this research to focusing on syntactic phenomena.

I then discussed descriptive grammars in Section 2.4, reference publications that describe the workings of an individual language, identifying those produced for an academic linguistic audience as those of concern for this research. In addition to outlining Good’s (2004) five characteristic features of descriptive grammars—ontologies, organisation into nested sections, descriptive prose, linguistic exemplars, and struc-
tured description—I also identified two challenges involved in their construction: selecting an organisational strategy for laying out a descriptive grammar and well as the frequent lack of integration with text collections, despite their importance to language description. I argued that syntactic phenomenon discovery techniques could potentially alleviate these two difficulties, through phenomenon-based navigational affordances and the retrieval of phenomenon-positive instances from a text collection.

In Section 2.5, I discussed treebanks—corpora of text that are annotated with linguistic structure and which have the potential to be valuable sources of positive instances of syntactic phenomena due to the kinds of annotations they contain, their often considerable size, and the considerable amount of work that is typically dedicated to quality assurance during their construction. I also briefly outline approaches used to query the contents of treebanks, another feature of relevance to syntactic phenomenon discovery.

Section 2.6 covered grammar engineering, the practice of developing and maintaining precision grammars, which the syntactic phenomenon discovery techniques developed in this thesis are based on. Precision grammars represent valuable resources for driving syntactic phenomenon discovery methods because of their capacity to automatically produce high fidelity syntactic and semantic analyses for input sentences, often being able to handle more complex grammatical phenomena than other types of computational grammars and statistical parsers. Furthermore, as implementations of linguistic theories, they often closely model the analyses found within descriptive grammars. I also observed that precision grammars are capable of producing high quality treebanks, containing annotations corresponding to the aforementioned high-fidelity analyses, and which are more consistent than human-annotated treebanks. The practice of dynamic treebanking means that treebanks produced by precision grammars can be much more efficiently updated than those that require hand-annotation. I unpacked the nature of precision grammars further by discussing the applications of grammar engineering, the requirements for developing precision grammars, as well as a typical grammar engineering workflow. I also discussed linguistic test suites, curated collections of sentences illustrating specific grammatical phenomena, which are essential to the grammar engineering endeavour and could be of use for syntactic phenomenon discovery. Lastly, I discussed multilingual grammar engineering projects, which are relevant for syntactic phenomenon discovery because of their potentially increased focus on the representation of grammatical phenomena compared with monolingual grammar engineering projects.

Section 2.7 outlined the nascent field of electronic grammaticography, which involves the production of descriptive grammars that take advantage of affordances only possible within a digital medium and which has the potential to benefit from the development of syntactic phenomenon discovery techniques. There are numerous advantages to be had by adopting natively digital descriptive grammars, and in this section, I identified and unpacked three advantages that syntactic phenomenon discovery could help realise. The first is non-linear navigation, in the form of phenomenon-based
navigation of the exemplars found in the descriptive grammar; the second is enhanced query facilities, where users could be provided with the ability to retrieve, from a corpus, additional examples of syntactic phenomena under discussion; and lastly, the better integration of text collections using phenomenon-based query facilities. I also discussed Taalportaal, an online descriptive grammar that I identified as currently being the most fully developed example of what an electronic descriptive grammar can offer, notably featuring treebank queries to illustrate syntactic phenomena, which are enabled through the use of a precision grammar.

In Section 2.8, I discussed a range of computational linguistics research projects that involve either the operationalisation of grammatical phenomena in terms of a representational formalism or the development of a system for the classification of grammatical phenomena. An overarching theme that emerged across these was the necessary involvement of human supervision, from the painstaking construction of top-down phenomenon hierarchies within the TSNLP project and Lehmann’s (2000) theory of syntactic phenomena, to the manual vetting of results in the various discovery procedures employed by the ESD project, and the construction-focused parser evaluation work of both Rimell et al. (2009) and Bender et al. (2011b).

Finally, in Section 2.9, I discussed existing approaches for retrieving instances of syntactic phenomena from a corpus. The first category consists of tools which query pre-annotated databases of labelled phenomena. These are relevant from the perspective of how they make the contents of the phenomenon database accessible to the user. I use one of these tools, \[\text{incr tsdb()}\], in Chapter 4 for storing the results of an annotated corpus of syntactic phenomenon instances. The second category is characterised by traditional treebank search tools, which I also discussed in Section 2.5.2. These require users to construct queries that capture the treebank’s representation of a desired in a query language. While such tools will likely remain indispensable for accessing the contents of the a treebank based on linguistic constraints, they suffer from two notable problems of accessibility: users must become familiar with an often unintuitive query language and users must also have a sense of how the target syntactic phenomenon is represented in a treebank before a suitable query can be constructed. Lastly, the tools which use a query-by-example methodology allow users to query a corpus by supplying example sentences of the target phenomena as input. This approach to treebank search improves upon the accessibility of traditional tools by removing the need to be familiar with a treebank query language, however the difficulty of having to be familiar with the target syntactic phenomenon’s representation remains.

This residual difficulty of discovering the representation of a syntactic phenomenon within a treebank produced by a precision grammar provides the departure point for the research presented in this thesis, motivating investigation into the development of syntactic phenomenon discovery techniques for precision grammars. In Chapter 4, I describe the construction of a corpus annotated with instances which I use to support this investigation. Then, in Chapter 5, I present the syntactic phenomenon discovery
methodology which resulted from this investigation.
Chapter 3

Resources

3.1 Introduction

In this chapter, I describe the three primary resources that were used to support the research undertaken within this project and explain the motivation for their selection. The three resources are: the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011), a wide-coverage precision grammar of English that I used for developing syntactic phenomenon discovery methods; DeepBank (Flickinger et al. 2012), a dynamically annotated treebank of the Wall Street Journal component of the Penn Treebank (Marcus et al. 1993) containing syntactic and semantic analyses produced by the ERG, which I used to develop a corpus of syntactic phenomena for evaluation of syntactic discovery techniques; and Huddleston and Pullum’s (2002) *The Cambridge Grammar of the English Language*, a comprehensive descriptive grammar of English, which was used to simulate the task of aligning the analyses of similar phenomena across both a computational grammar and a descriptive grammar—a potential application of the syntactic phenomenon discovery techniques developed within this research.

3.2 The LinGO English Resource Grammar

The primary grammar used to develop the syntactic phenomenon discovery techniques proposed in this research was the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011), a wide-coverage grammar of English that prioritises linguistic accuracy and has been actively developed for over two decades, under the auspices of the Linguistic Grammars Online (LinGO) laboratory at the Center for the Study of Language (CSLI) within Stanford University. As with other grammars developed within the DELPH-IN research consortium, it is informed by the grammatical framework of Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag 1994) and uses Minimal Recursion Semantics (MRS; Copestake 2005) as the formal-
ism for the semantic representations it produces. In addition to parsing English, the ERG also supports the generation of English sentences that yield the same input semantics given a specific MRS structure. The ERG’s first application was in the context of the Verbmobil project (Wahlster 2000), a machine translation project for spoken language that made use of a semantic transfer-based approach, where it was involved in both German-to-English and Japanese-to-English translation. It has been under continual development since then, within the context of a range of different applications, including a further machine translation endeavour—the LOGON project\(^1\) (Lønning \textit{et al.} 2004)—a Norwegian-to-English machine translation system, as well as a grammar checking system in the context of a large-scale online language course for primary school students (Suppes \textit{et al.} 2012).

Within the fields of NLP and computational linguistics, the term \textit{language model} is typically used to refer to a statistical model that supports the computation of the probability of a sentence or a sequence of word tokens or the probability of token given a sequence of already seen tokens, using token frequencies from a text collection (Jurafsky and Martin 2009:83). The ERG, as a computational grammar of English, represents another type of language model. As discussed in Section 2.6, these models are often based on theories of language developed within formal linguistics and therefore encode much more linguistic information than statistical language models. The appropriate type of language model depends, of course, upon the application, and in Section 2.6, I argued that precision grammars are an appropriate choice for this research. I use the term \textit{language model} to refer to a model of a language more generally rather than specifically a statistical model.

The ERG stands out as an appropriate resource for use in the discovery of syntactic phenomena due to three salient properties: its capacity as a language model to produce high-fidelity analyses of a large number of English syntactic phenomena, its strong levels of robustness for a rule-based computational grammar, and its capacity to produce high-quality treebanks, which stems in part from the previous two properties. In the remainder of this section, I unpack these properties of the ERG and explain their relevance for syntactic phenomenon discovery.

### 3.2.1 Grammatical Phenomena

When selecting a language model it is important for it to be capable of supporting the necessary inter-class distinctions for the target application. Given the motivating application for the present research, a candidate language model must be capable of identifying the kind of syntactic phenomena that a descriptive grammar documents. This suggests a level of analytic fidelity greater than is often required by typical computational linguistics tasks. Looking at the different language models described in Section 2.8 and Section 2.9, which were used for facilitating syntactic phenomenon discovery.

\(^1\)http://www.emmtee.net
discovery, the simple language model used by the Linguist’s Search Engine (Resnik and Elkiss 2005), discussed in Section 2.9.3, stands out in opposition to this claim. In defending the language model used, Resnik and Elkiss argue that the correctness of the generated analyses is not particularly important; so long as the model produces consistent analyses—given similar contexts—across both user input and the contents of the corpus, it can be used to retrieve instances of the target syntactic phenomena. While it is true that in this context, the utility of a language model will ultimately be assessed by its capacity to support the retrieval of sentences a user would consider instances of the target phenomenon—rather than the accuracy of its generated analyses—this does not mean that the quality of the analyses can be ignored. Crucially, the granularity of analyses offered by the language model being leveraged must be commensurate with the granularity of analytic distinctions implied by the phenomena that are candidates for discovery.

The need for the language model to make the requisite analytic distinctions can be demonstrated through the translation procedure Hockenmaier and Steedman (2007) used to convert PTB analyses into CCG analyses used to construct CCGBank, which was described in Section 2.8.5. During this process, the authors found they were forced to assume a right-branching structure in the generated CCG analyses for nominal compounds, since the flat noun phrases within the PTB did not encode sufficient information to make a more principled recovery. The PTB—and any derivative language model—would therefore be incapable of distinguishing left-branching compound nouns from right-branching compound nouns. Consequentially, any syntactic phenomenon discovery system employing PTB analyses would not allow users to isolate and retrieve strictly left or right-branching compound nouns. In order to support the task of phenomenon discovery across a range of potentially nuanced syntactic phenomena, the language model used to support the present research will therefore need to be capable of accurately analysing sentences containing a potentially diverse range of phenomena.

Throughout its development, the ERG has prioritised linguistic accuracy, meaning that in addition to targeting output representations which have practical utility in the context of various applications, the analyses developed are also assessed by their viability with respect to contemporary understandings of syntax and semantics. This has been facilitated in part by the adoption of a grammar engineering methodology which enables and promotes the evaluation of analyses with respect to their linguistic plausibility, as well as through the ongoing consultation with experts on the evaluation and design of analyses (Flickinger 2011). Consistent with the description of the grammar engineering workflow from Section 2.6.3, the ERG’s development methodology monitors linguistic accuracy across both test corpora as well as domain and phenomenon-specific test suites. This is made possible through the use of the dy-

\footnote{This deficiency within the PTB noun phrase analyses was addressed in the work of Vadas and Curran (2007) and Vadas and Curran (2008), resulting in enriched PTB analyses and consequently improved accuracy within CCGBank analyses.}
namic treebanking methodology—described previously in Section 2.6.5—which allows grammar engineers to inspect and either validate or reject only those components of analyses produced by the grammar that have changed after a grammar modification.

The prioritisation of linguistic accuracy, combined with continuous efforts to improve the coverage of the grammar within the context of a variety of applications, has resulted in the ERG being capable of producing high-fidelity analyses of a large range of English grammatical phenomena. It is, however, difficult to quantify the ERG’s phenomenon coverage, in part due to the multifaceted nature of grammatical phenomena, with the grammatical phenomena of a language being able to be characterised in numerous ways, resulting in different divisions in the phenomenon space, as was discussed in Section 2.3. Compounding this challenge, it is often difficult to pinpoint specific phenomena that are associated with individual grammar components, as interactions between phenomena mean that the implementation of the analysis of one construction will often result in the addition of constraints at diverse locations within the grammar, meaning that it is not possible to directly use the artefact of the grammar itself and its component parts to directly enumerate and quantify the phenomena it covers. This difficulty in pinning down specific phenomena within the analytic edifice of a precision grammar such as the ERG speaks to the holistic nature of the grammar engineering endeavour that was identified in Section 2.6.3, in which a significant proportion of the effort involved in developing new analyses is taken up by its integration with the body of existing analyses within the grammar.

These difficulties mean that creation of a comprehensive resource documenting the grammatical phenomena covered by the ERG would likely be impractical, however, there does exist documentation of specific phenomenological aspects of the ERG. This includes documentation of the ERG’s types associated with derivation rules—phrasal rules, lexical rules, and inflectional rules—as well as the ERG’s inventory of lexical types, which are maintained using the LTDB (Hashimoto et al. 2007), described in Section 2.9.1. While a valuable resource, this database is more introspective, geared towards the internal documentation of grammar components, rather than how analyses of linguistic phenomena familiar to those in the descriptive linguistics community are implemented and represented. The other partial source of phenomenon documentation for the ERG is the catalogue of semantic phenomena being produced within the ESD project (Flickinger et al. 2014a), described in Section 2.8.3. While this does form a more externally geared resource, it is focused not on syntactic phenomena, but on documenting the semantic interface to the grammar, making the semantic representations produced by the ERG more accessible for other researchers.

One way of getting at least a relative sense of the scope of the ERG’s analytic complexity is to look at the number of various grammar internal components which are potentially involved in the development of analyses of grammatical phenomena.

3These statistics are taken from the 1214 release of the ERG, and for all other grammars, the latest available version as of 11/08/2017 was used. More information about these and other delph-in grammars can be found at http://moin.delph-in.net/GrammarCatalogue.
Table 3.1 provides a breakdown of such components, comparing the ERG’s inventory with a range of other DELPH-IN grammars. Of the grammar components included in Table 3.1, not all are necessarily appropriate proxies for gauging grammatical phenomenon coverage. Lexical entries, for instance, can be programmatically extracted in bulk from other lexical resources and their quality can vary greatly depending on how well they are mapped to the existing lexical types of the grammar. Lexical rules—which are unary rules generating word types from other word types—while clearly corresponding to the implementation of phenomena, represent a noisy signal, since languages which happen to have a greater degree of morphological complexity will likely have more lexical rules. Phrase rules are likely a better proxy, however this signal can be confounded by differences in architectural approaches taken by the grammar engineer. This same confound applies even more so to the total yield of all types in the grammar, as grammar engineers and different grammar engineering projects will adopt different practices in how abstractions are formed within the type hierarchy, which can lead to a large variation in the numbers of types used across different grammars. In the context of DELPH-IN grammars, features correspond to the attribute names found in the attribute value matrices (or feature structures) used as the underlying representational formalism which model the linguistic analyses that constitute the grammar. Their number is then also influenced greatly by the underlying properties of the language being modelled in addition to differences in architectural approaches to its modelling.

This leaves lexical types as the grammar internal component that might be the

---

4It should be noted that the total number of all types indicated in Table 3.1 includes GLB types (greatest lower bound) (Copestake 2000), which are generated by grammar processing tools and inserted into the compiled type graph for efficiency purposes.
best proxy for phenomenological coverage. Within DELPH-IN grammars, lexical types encode the selectional properties of head words, meaning that they can be thought of as fine grained POS-tags which capture different categories of valence frames. Given the strong ties between DELPH-IN grammars and HPSG, a heavily lexicalist framework characterised by richly structured lexica that encode selectional constraints, it is a reasonable hypothesis that much of the analytic complexity within DELPH-IN grammars would be found within the lexical types of a grammar. This effect would likely also be more pronounced once the basic phenomena of a language are covered, and the grammar engineer starts descending the Zipfian long tail of grammatical phenomena, which is certainly the status of the ERG given its maturity. Looking at Table 3.1, the relatively large number of lexical types found within the ERG is certainly consistent with the it having a large coverage over grammatical phenomena.

### 3.2.2 Robustness and Accuracy

An important consideration for the development of syntactic phenomenon discovery systems is that end-users are able to provide sentences or fragments needed to explore the target phenomena without being overly restricted by limitations on the range of available inputs. When developing computational grammars, there is a tension, however, between the prioritisation of linguistic accuracy and the prioritisation of robustness (Flickinger 2011). By emphasising the linguistic correctness of analyses yielded by the grammar, development of the grammar takes more of a depth-first approach, as opposed to breadth-first, meaning that for any reasonably-sized corpus, there will inevitably be a range of sentences containing as of yet unanalysed phenomena and for which the ERG will be unable to provide an analysis (Flickinger 2011). This challenge of robustness is one that any precision grammar which also aspires to have wide coverage must address. Development of the ERG has continually worked to improve coverage levels through the design and implementation of analyses for additional grammatical phenomena, steadily descending further down the Zipfian long tail.

In addition to the implementation of new analyses, there has also been a range of efforts focusing around the improvement of the ERG’s robustness. An important milestone for this was the work of Baldwin et al. (2005), which involved parsing a subset of the British National Corpus (BNC) with the ERG and performing error analysis over parse failures in order to identify areas which targeted development could yield significant returns on coverage levels. A notable result from this study was that many parse failures were associated with missing lexical entries for open class words. The ERG was then augmented with lexical entries for words occurring as verbs within the BNC with a frequency of greater than 100, resulting in the manual

---

5Some precision grammars—such as the DELPH-IN Wambaya (Bender 2010) and Hausa (Crysmann 2012) grammars—aim to analyse a specific selection of phenomena of theoretical interest, without aspiring to achieve broad coverage over the whole language.
### Table 3.2: Coverage rates over ERG treebanks (Flickinger 2011:36).

<table>
<thead>
<tr>
<th>Corpus type</th>
<th>No. items</th>
<th>Avg. item length</th>
<th>Observed coverage (%)</th>
<th>Verified coverage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting scheduling</td>
<td>11660</td>
<td>7.5</td>
<td>96.8</td>
<td>93.8</td>
</tr>
<tr>
<td>E-commerce</td>
<td>5392</td>
<td>8.0</td>
<td>96.1</td>
<td>93.0</td>
</tr>
<tr>
<td>Norwegian tourism</td>
<td>10834</td>
<td>15.0</td>
<td>94.2</td>
<td>90.1</td>
</tr>
<tr>
<td>SemCor (partial)</td>
<td>2501</td>
<td>18.0</td>
<td>91.8</td>
<td>82.0</td>
</tr>
<tr>
<td>Wikipedia (CmpLng)</td>
<td>11558</td>
<td>19.5</td>
<td>87.4</td>
<td>80.0</td>
</tr>
<tr>
<td>Online user forum</td>
<td>578</td>
<td>12.5</td>
<td>85.5</td>
<td>77.5</td>
</tr>
<tr>
<td>Dictionary defs.</td>
<td>10000</td>
<td>6.0</td>
<td>81.2</td>
<td>75.5</td>
</tr>
<tr>
<td>Essay</td>
<td>769</td>
<td>21.6</td>
<td>83.2</td>
<td>69.4</td>
</tr>
<tr>
<td>Chemistry papers</td>
<td>637</td>
<td>27.0</td>
<td>87.8</td>
<td>65.3</td>
</tr>
<tr>
<td>Technical manuals</td>
<td>4000</td>
<td>12.5</td>
<td>86.8</td>
<td>61.9</td>
</tr>
</tbody>
</table>

The result of all these efforts is that the ERG has high coverage levels over a range of different types of corpus text. Table 3.2 shows coverage for the ERG over the different collections within the Redwoods treebank (Oepen et al. 2004), which have been sourced from a range of different domains (Flickinger 2011). In this table, observed coverage refers to the proportion of items from each treebank that received at least one analysis, and verified coverage refers to the proportion of items within the treebank that an annotator identified as receiving a correct analysis. Observed coverage therefore pertains to robustness, while verified coverage pertains to accuracy. These relatively high coverage rates for a hand-crafted grammar mean that the ERG will suffer less from problems of robustness in the face of user input, thereby increasing the candidate pool of sentences which can be used as phenomenon exemplars for performing syntactic phenomenon discovery.

While observed and verified coverage are frequently used for evaluation of precision grammars within grammar engineering projects, these metrics are not typically employed in the context of evaluating statistical parsers. A common approach for
assessing the accuracy of dependency parsers is the use of labelled attachment score (LAS) and unlabelled attachment score (UAS). Both of these metrics are aggregate scores, representing the ratio of words receiving the correct local attachments within an analysis. UAS is defined as the percentage of words that are assigned the correct syntactic head within the parse tree, and LAS is defined as the ratio of words that receive the correct syntactic head and also the correct dependency label. Table 3.3 shows results from Ivanova et al.’s (2016) evaluation of three different approaches to the task of parsing to bi-lexical dependencies, which enables a comparison of the ERG’s performance with other statistical parsers. The results presented are LAS and UAS scores for the ERG and two statistical parsers of English: the transition-based dependency parser of Bohnet and Nivre (2012) (B&N) and the Berkely parser (Petrov et al. 2006), an unlexicalised phrase-structure parser that uses a PCFG. In the case of the ERG and the Berkely parser, which both produce phrase-structure analyses, these were converted into bi-lexical dependencies, so that LAS and UAS could be derived. The results in Table 3.3 are reported across a number of data sets, including the Wall Street Journal sections of the Penn Treebank (Marcus et al. 1993) (WSJ)—which is in-domain for all three parsers—and a selection of four out-of-domain datasets sampled from the Redwoods Treebank (Oepen et al. 2004), including The Cathedral and the Bazaar (CB), an essay on software advocacy; a subset of the SemCor portion of the Brown Corpus (SC; Francis and Kučera 1982); transcribed spoken dialogues from the Verbomobil project (VM; Wahlster 2000), and a sample of the WeScience corpus, a Wikipedia-derived collection of scientific topics (WS; Ytrestøl et al. 2009). These results show that the ERG has a consistently higher accuracy than the other parsers over both in and out-of-domain datasets. As Ivanova et al. (2016) argue, that the ERG’s performance over these out-of-domain datasets is closer to its results over the in-domain WSJ suggests that the ERG’s superior linguistic knowledge confers the advantage of being able to better accommodate domain variation.

3.2.3 High Quality Treebanks

In Section 2.6.5, I discussed how the dynamic treebanking methodology of Oepen et al. (2004) facilitates the creation and ongoing maintenance of treebanks that track the current state of the grammar by being periodically updated to contain, for each item, the best analyses produced by the grammar—or none if the grammar does not produce a correct analysis. The ERG’s developers have prioritised the development of a range of different treebanks, including those found within the Redwoods treebank, and which are listed in Table 3.2, and the DeepBank treebank, which I discuss in Section 3.3. These treebanks play an important role in the development of the ERG. They provide developers with a means of monitoring grammar coverage over time; they represent a valuable source of information regarding missing phenomena that the grammar does not yet cover; and they can be used as a baseline to assess the impact of changes to the grammar, in addition to being used to train the parse-selection model.
<table>
<thead>
<tr>
<th>Parser</th>
<th>LAS</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ</td>
<td>86.65</td>
<td>89.86</td>
</tr>
<tr>
<td>Berkely</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B&amp;N</td>
<td>86.76</td>
<td>89.65</td>
</tr>
<tr>
<td>ERG</td>
<td>92.87</td>
<td>93.95</td>
</tr>
<tr>
<td>CB</td>
<td>78.13</td>
<td>83.14</td>
</tr>
<tr>
<td>Berkely</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B&amp;N</td>
<td>77.70</td>
<td>82.96</td>
</tr>
<tr>
<td>ERG</td>
<td>90.77</td>
<td>92.47</td>
</tr>
<tr>
<td>SC</td>
<td>79.81</td>
<td>85.10</td>
</tr>
<tr>
<td>Berkely</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B&amp;N</td>
<td>78.08</td>
<td>83.21</td>
</tr>
<tr>
<td>ERG</td>
<td>90.84</td>
<td>92.65</td>
</tr>
<tr>
<td>VM</td>
<td>74.40</td>
<td>83.38</td>
</tr>
<tr>
<td>Berkely</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B&amp;N</td>
<td>75.28</td>
<td>82.86</td>
</tr>
<tr>
<td>ERG</td>
<td>90.44</td>
<td>92.27</td>
</tr>
<tr>
<td>WS</td>
<td>80.31</td>
<td>85.11</td>
</tr>
<tr>
<td>Berkely</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B&amp;N</td>
<td>80.63</td>
<td>85.24</td>
</tr>
<tr>
<td>ERG</td>
<td>91.33</td>
<td>92.48</td>
</tr>
</tbody>
</table>

Table 3.3: Labelled attachment scores (LAS) and unlabelled attachment scores (UAS) for three parsing systems across an in-domain dataset (WSJ) and four out-of-domain datasets.
Furthermore, by virtue of containing a large number of sentences annotated with gold standard linguistically accurate syntactic and semantic analyses, these treebanks also constitute valuable resources for other research projects and downstream applications. I discuss the specific importance of a disambiguated treebank for my research in the following section.

### 3.3 DeepBank

DeepBank (Flickinger et al. 2012) is a treebank composed of text from the 25 Wall Street Journal (WSJ) sections of the Penn Treebank (Marcus et al. 1993)—discussed in Section 2.5.3 and Section 4.5—and which is annotated with syntactic and semantic analyses from the ERG using the dynamic Redwoods treebanking methodology described in Section 2.6.5. The annotations recorded for each sentence in the treebank correspond to the manually disambiguated gold trees—the best available analysis selected from the parse forest by an annotator. The dynamic treebanking methodology enables periodic synchronisation of the analyses within the treebank with the current version the grammar. DeepBank covers sections 00-21 of the PTB—with the remaining three sections held out for evaluation purposes, yielding a total of 43746 sentences. In this project I used DeepBank 1.1, which targets the ERG 1214 release. In this version of DeepBank, the ERG achieved an observed coverage of 95.68%, and a verified coverage of 88.81%. Since its initial version 1.0 release, which targeted the 1212 version of the ERG, DeepBank has been updated to version 1.1, targeting the 1214 version of the ERG. Without the use of the dynamic treebanking approach it is unlikely that such large scale revisions of the treebank would be undertaken, due to the prohibitively expensive nature of the effort that would be required for a completely manual re-annotation. In the context of research into syntactic phenomenon discovery techniques, the use of the most current stable version of the ERG is important, as the ERG is being continually developed through refinement of existing analyses and the addition of analyses of new phenomena.

In Chapter 4, I describe the construction of a small corpus annotated with occurrences of syntactic phenomena. This corpus is based on a subset of the WSJ component of the PTB, so that DeepBank annotations could be leveraged. I selected DeepBank over the Redwoods treebank due to the strong history of use that the WSJ text and accompanying annotations from the PTB have seen within computational linguistics research. In producing this corpus, a side-goal was the construction of a reusable resource of phenomenon annotations, which could potentially be used for applications such as parser evaluation. By targeting WSJ text, the likelihood of the phenomenon annotations in this corpus being adopted for subsequent research is increased.

In the context of my research into techniques for performing syntactic phenomenon discovery using precision grammars, the use of a gold standard treebank synchronised
with the current state of the grammar is of notable value. Given that linguistically
precise grammars reflect the inherent ambiguity of language by typically returning
many analyses for a single input, any syntactic phenomenon discovery system that
uses precision grammars to generate analyses will have to address the issue of how to
select the appropriate analysis for a given input sentence. One option is to use the
best analysis returned by the parse ranking model. The drawback of this approach,
however, is that this analysis may not receive the appropriate treatment from the
grammar for the target phenomenon, which would require manual user intervention
to select an appropriate analysis, otherwise discovery of the relevant grammar compo-
nents would not be possible based off only this analysis. The use of a disambiguated
gold standard analysis for each input sentence, on the other hand, would greatly in-
crease the chance of an appropriate analysis being supplied as input, removing the
need for manual intervention and reducing noise introduced through misanalysis.

3.4 The Cambridge Grammar of the English Language

The descriptive grammar I used for this research was Huddleston and Pullum’s
(2002) The Cambridge Grammar of the English Language (Huddleston and Pullum,
from hereon). Huddleston and Pullum provides a comprehensive overview of the
grammar of international Standard English. Huddleston and Pullum (2002:4) define
Standard English to be the variety of English that is used in countries where English
is used for the majority of publications, media broadcasts, and government commu-
nications. While there is, of course, some variation amongst the English usage found
in these contexts, there is a high degree of consensus to be found within the decisions
made by editors of publications and broadcast authorities, which, on aggregate, form
the variety of Standard English (Huddleston and Pullum 2002:4).

Huddleston and Pullum follows many of the commonly adopted practices within
reference grammar writing that were identified in Section 2.4. It aims to provide
a descriptive (as opposed to prescriptive) account of the grammar of English in its
modern form—which is to say, a synchronic account. Huddleston and Pullum largely
targets the description of syntactic and morphological phenomena, covering some
aspects of English phonology and orthography that arise in the process, but it does
not focus explicitly on these dimensions of description. The descriptions found in
Huddleston and Pullum do not assume prior exposure to theoretical linguistics by
the reader, however, the book as a whole aims to incorporate into its analyses as
many as possible of the advances in our understanding of English grammar that are
found across numerous fragmentary descriptions of English within modern linguistics.

As a reference of English, Huddleston and Pullum includes only descriptions of
the aforementioned aspects of the language system, and does not include the other
two members of the Boasian Trilogy: the lexicon and text collections. It does, how-
ever, employ all of the different structural features commonly found in descriptive
grammars identified by Good (2004) and described in Section 2.4.3: use of ontologies,
organisation into nested sections, descriptive prose, the use of phenomenon exemplars,
and structured description using tabular data.

In order to tackle a work of such scope, the development of Huddleston and
Pullum was a collaborative effort between a number of different contributors and
the primary authors, Rodney Huddleston and Geoffrey K. Pullum. The book is
organised into twenty chapters, with each chapter being authored by one or more
different collaborators and with at least one of the primary authors as a co-author for
each chapter to ensure that the different chapters cohere together as an integrated
analysis. Regarding the form-function distinction, which I identified in Section 2.4.4
as a possible strategy for the organisation of descriptive grammars, the chapters are for
the most part divided up by means of formal distinctions, with distinctions pertaining
to function made within chapters.

Huddleston and Pullum represents an appropriate resource for the exploration of
syntactic phenomenon discovery techniques due to its comprehensive and also fine-
grained coverage of English grammatical phenomena coupled with its extensive use
of exemplars that augment the descriptions. Huddleston and Pullum therefore offers
a rich source of potentially complex and also less frequently occurring grammatical
phenomena, providing a range of candidate phenomena which can be used to road-test
syntactic phenomenon discovery techniques developed in this research. The granular
division of the phenomenon-space within Huddleston and Pullum also presents an op-
portunity to tackle the problem of aligning the analyses of different phenomena—and
their representation—across a computational grammar and a descriptive grammar, a
problem, that, as discussed in Section 2.7.3, was identified by Bouma et al. (2015) as
being a significant challenge in the context of using a precision grammar to enrich a
descriptive grammar.

3.5 Summary

In this chapter I outlined the three primary resources that were used to support
the research undertaken in this project. The first resource, the ERG, was selected
as the precision grammar used to develop the syntactic phenomenon discovery tech-
niques developed in this research, which are described in Chapter 5. The second
resource, DeepBank, was selected as the ERG’s treebank from which to develop a cor-
pus annotated with phenomenon instances to be used for evaluation of the syntactic
phenomenon discovery methods. I describe the construction of this annotated corpus
in Chapter 4. The last resource, The Cambridge Grammar of the English Language,
was selected as the descriptive grammar used to road-test the syntactic phenomenon
discovery techniques developed in this thesis in the context of a grammar alignment
task. This application, which I discuss in Section 5.7, involved using the Typediff
tool developed as a part of this research, and which is described in Chapter 5, for the purpose of identifying ERG components salient to specific syntactic phenomena that are identified within Huddleston and Pullum.
Chapter 4

The Phenomenal Corpus

4.1 Introduction

The development of any data-driven methodology requires sample data for implementation and evaluation. In this chapter, I identify the properties needed of data that can support the development of syntactic phenomenon discovery methodologies, and, after identifying existing resources that represent plausible candidates, argue that these do not meet the established criteria. I then describe the creation of a new corpus—dubbed the Phenomenal Corpus—annotated with grammatical phenomena, that is specifically designed for supporting the development of syntactic phenomenon discovery techniques.

The remainder of this chapter is structured as follows: Section 4.2 puts forth the desiderata for this kind of evaluation resource and then Section 4.3 proceeds to identify and compare existing resources against the established criteria, concluding that the creation of a new annotated resource is the best way to proceed. Section 4.4 outlines the selection of the target syntactic phenomena used for the resource and Section 4.5 details the selection of corpus data. Section 4.6 outlines the methodology used to create the corpus, including the procedure used for constructing the annotation guidelines and how the annotation task itself was performed. This section continues by presenting the annotation results and describing how this was used to create the final gold standard annotations found in the Phenomenal Corpus, before presenting an analysis of the errors made by each annotator when compared with the gold standard. This section on data collection ends with the description of an attempt to improve the quality of the annotations in the gold standard through the use of DeepBank annotations. Section 4.7 then details an investigation into how to best measure the reliability of the collected annotations, including the identification and evaluation of strategies for measuring inter-annotator agreement given the particular characteristics of this annotation task. Lastly, Section 4.8 ties the chapter together, firstly by presenting a discussion of the various strategies used to gauge the quality of the annotations found in the Phenomenal Corpus, followed by a critical discussion.
of the overall methodology for the construction of this resource.

4.2 Data Desiderata

The methodology I adopt in Chapter 5 for performing syntactic phenomenon discovery takes the form of an interactive process, in which users supply input exemplars of the target phenomenon in order to triangulate upon grammar components likely to be relevant to the phenomenon’s analysis. The development of such a methodology requires a variety of phenomenon exemplars which can be used as input, in order to verify the utility of the methodology during development, as well as to gauge the methodology’s performance over different types of user input. The other application that a syntactic phenomenon-annotated corpus could be used for is the evaluation of phenomenon detection and retrieval techniques, enabling the assessment of the precision and recall of queries designed to retrieve target phenomena from a corpus. I developed the Phenomenal Corpus with both applications in mind—supporting the development of syntactic phenomenon discovery techniques and evaluating the performance of syntactic phenomenon retrieval techniques—however as my research shifted to focus on the discovery aspect of syntactic phenomenon discovery-and-detection pipelines, I ultimately used the corpus only for the former application of supporting development of phenomenon discovery methodologies. As such, the desiderata put forth in this section takes into account both applications.

An intuitive format for a resource that supports both phenomenon discovery and retrieval evaluation is a collection of sentences or fragments that are annotated with attested instances of target syntactic phenomena. This also implies the existence of a set of chosen syntactic phenomena, along with a characterisation of each phenomenon. Linking this back to the motivating context of supporting the augmentation of descriptive grammars with syntactic phenomenon retrieval capacities, these characterisations correspond to the descriptions of the phenomena within the descriptive grammar. Alternatively, in a more ad hoc retrieval context, these characterisations could correspond to the working understanding of a construction that a researcher is trying to build upon by locating additional examples. The following sections outline the specific criteria that this syntactic phenomenon-annotated resource would need.

4.2.1 Corpus Data

The text in the resource should be sourced from corpus data rather than manually curated exemplars. This stipulation is made firstly on the basis that this better simulates one of the motivating applications for developing syntactic phenomenon discovery techniques. In Section 2.4.5, I discussed the importance of text collections in the development of the analyses found within descriptive grammars, and in Section 2.7.2, following Thieberger (2008), I argued that syntactic phenomenon retrieval
techniques could facilitate the integration of descriptive grammars with their text collections, thereby improving the reader’s capacity to verify or falsify hypotheses found in a descriptive grammar. Others who have argued for augmenting electronic descriptive grammars with corpus data—whether through explicit links or through dynamic queries—include Nordhoff (2008), Thieberger (2008), Bender et al. (2012), Bouda and Helmbrecht (2012), Drude (2012a), Maxwell (2012), Mosel (2012), and Musgrave and Thieberger (2012).

The other reason to use corpus data is to avoid the limitations associated with constructed test suites. Manual supervision of the contents might at first be tempting, as curation would be able to ensure that the resource includes a sufficient number of instances of each phenomenon. This process of curation could also involve the inclusion of known variants of each phenomenon, where relevant, in order to test discovery techniques over a range of phenomenon variation. This approach, however, runs afoul of problems that have been identified with the use of curated examples in the context of linguistic theory development. One argument in particular that has been levelled at the use of curated examples—whether elicited from a native speaker or constructed and then vetted by a native speaker—is that this data is ultimately limited by the combined knowledge and imagination of the researcher and any native speakers consulted (Schütze 1996). In the context of a syntactic phenomenon discovery system, the repercussions are that the system will remain untested on manifestations of the phenomenon that the compiler of the resource was either unaware of or inadvertently omitted. It would then be difficult to make any substantive conclusions regarding the ability of the system to generalise to unseen variants of the phenomenon. Just as in the context of developing linguistic theories, this problem can be alleviated through the use of corpus data. As more corpus data is used, coverage over the space of syntactic phenomena space will increase, as more of the Zipfian long tail of grammatical phenomena is seen. This results in the inclusion of less frequently occurring constructions and phenomenon variants, as well as instances of unexpected interactions between phenomena that might have otherwise gone unimagined.

A further point pertaining to background frequencies of phenomena is that a resource constructed out of items known to contain instances of the target phenomena will have higher rates of the phenomena than compared with corpus data. If used for evaluation, such a resource would present fewer opportunities for a system to

---

1For a nuanced presentation of the debate between the use of corpus data and native speaker judgements, see Fillmore (1992) and for an overview of the arguments against grammaticality judgements, see Schütze (1996). While perhaps less of a concern in the present context, a notable problem with grammaticality judgements that Schütze (1996) explores is that native speaker intuitions are better modelled as fuzzy or gradient judgements, as opposed to the binary grammatical/ungrammatical distinction traditionally invoked. This has been explored in more depth in Keller (2000) and Bresnan (2006). For a discussion of the use of judgement and corpus data in the context of precision grammar development, see Baldwin et al. (2005).

2See Section 2.3.6 for a discussion of the distribution of syntactic phenomena.
yield false positives; so it would be expected that precision scores would be positively biased compared to a resource constructed out of corpus data.

### 4.2.2 Exhaustive Annotation

Each sentence in the corpus should be exhaustively annotated with occurrences of all phenomena that the corpus is annotated with. Resources composed of curated exemplars often have the property that sentences chosen to illustrate a specific phenomenon will only be annotated with that phenomenon, but not others covered by the corpus found incidentally in the sentence. This is seen in the context of linguistic test suites, as discussed in Section 2.6.4, where examples are constructed to test the coverage of the grammar over specific phenomena. This is also seen in the phenomenon exemplars included in most descriptive grammars. As an example of the latter, (30) is an exemplar from *Huddleston and Pullum (2002:p960)* which illustrates content clauses occurring as an extraposed subject and that have *it* occurring in the subject position. The underlined portion of the example corresponds to the content clause in question, however this sentence also contains unannotated examples of various other phenomena, such as negation and the occurrence of a copula clause—both of which are also covered by the grammar in different chapters.

(30) It simply hadn’t occurred to us that the work might be a forgery.

This selective annotation is understandable in the context of a (non-electronic) descriptive grammar with a phenomenon inventory as large as that of *Huddleston and Pullum (2002)*, where inline annotation of incidental phenomena would be impractical.\(^3\) However, this is undesirable in the context of developing a resource for supporting phenomenon discovery methodologies. Firstly, leaving these phenomenon instances unannotated would amount to leaving a large amount of data “on the table”, potentially foregoing some of the coverage benefits of using corpus data that were identified in the previous section. Secondly, if used as an evaluation resource, a consequence of using non-exhaustively annotated data is that meaningful evaluation of recall would be impossible. More generally, since the exhaustive annotation of exemplars within descriptive grammars increases the utility of the resource (Good 2004) and given that the potential affordances of electronic descriptive grammars make this feasible, it would be desirable for the resource under construction in this chapter to reflect this property also.

\(^3\)Although Good (2004) highlights Haspelmath’s (1993) Lezgian grammar as an example of a descriptive grammar that does provide such exhaustive annotation, making use of an index to solve the cross-referencing problem.
4.2.3 Character-level Annotation

Each phenomenon occurrence should be annotated at the sub-sentence level, so that the portion of the sentence pertaining to the phenomenon can be isolated. In the context of linguistic test suites, curated exemplars can be constructed to be of minimal length while still exhibiting the desired phenomenon. Sentences sourced from naturally occurring corpus data, however, may be of substantial length, with only a small portion being relevant to a target phenomenon. For example, in the context of dynamically retrieving instances of parenthetical supplements to a user, returning the entire sentence found in (31) (Huddleston and Pullum 2002:1359) devoid of any further annotation leaves a lot to be desired, as only the relatively small underlined portion of the sentence is relevant to the phenomenon. It would be preferable then for phenomenon discovery techniques to mark the extent of the retrieved sentences that are pertinent to the phenomenon by following the already established convention within descriptive grammars of underlining the relevant spans of exemplars. From the perspective of evaluating a phenomenon retrieval tool, supporting sub-sentence level annotations within the Phenomenal Corpus leaves open the possibility of assessing the quality of returned instances at this more fine-grained level.

(31) Such behaviour runs the risk, wouldn’t you agree, of alienating our customers.

Sub-sentence level annotations would also be advantageous from the perspective of using the phenomenon annotations to emulate a user supplying input to the syntactic phenomenon discovery tool. More specifically, allowing users to mark the boundaries within input exemplars that they consider directly relevant to the target phenomenon would improve the discovery process by eliminating noise associated with extraneous phenomena from unrelated parts of the sentence.

These considerations motivate the need for at least token-level annotations, however not for character-level annotations. This is motivated by the desire to make use of annotations produced by the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011). The ERG employs a tokenisation scheme whereby punctuation characters bind tightly to word tokens, as opposed to forming distinct tokens, which differs from many other tokenisation schemes. Character-level annotations leaves open the possibility for aligning the phenomenon annotations with annotations produced by the ERG.

4.2.4 Grammatical Framework-independent

Both the annotations of the syntactic phenomena within the resource and their analyses should be independent of formalised grammatical frameworks. The primary motivation being that, when used as a source of phenomenon exemplars for the discovery process, the nature of the exemplars should be consistent with the motivating context of aligning syntactic phenomena from a descriptive grammar with precision
grammar internals. In order to model this scenario using annotated sentences from the corpus as exemplars, it is necessary that the phenomenon annotations originate from descriptive rather than formalised characterisations. It would be inappropriate, for example, to source the annotations from the output of an implemented formal model. While this would be convenient, it would no longer model the motivating context, instead involving the alignment of two formalised models of a language. A secondary reason for this stipulation is that making use of analyses that are either extracted from or based upon the analyses of a formalised model could potentially bias the discovery methodology towards implemented precision grammars that are based on that same grammatical framework that is invoked by the analyses used to generate annotations.

This has two distinct practical consequences for the resource. Firstly, the annotation format itself should avoid any conventions associated with specific parsers or grammar engineering frameworks. The safest way to achieve this outcome is to simply make use of flat character spans for the annotation format, rather than positing any internal structure. Secondly, the criteria used to identify each phenomenon covered by the resource should be specified in a descriptive manner, emulating the kind of analysis found within descriptive grammars.

4.3 Existing Phenomenon-annotated Resources

There are a number of existing resources which could be used to form a corpus annotated with syntactic phenomena to support the research in this thesis. In particular, the resources discussed in Section 2.8 could potentially be adapted to be used as a source of annotations or could provide a methodology from which to base the development of a new resource. In this section, I assess each of these resources based on the previously identified desiderata. The first type of candidate resource is linguistic test suites created to support the development and evaluation of computational linguistics resources, as discussed in Section 2.6.4, in the context of supporting grammar engineering. Of particular note for present purposes is the English test suite produced in the TSNLP project (Lehmann et al. (1996) and Oepen et al. (1997)) discussed in Section 2.8.1, which includes nearly 5000 examples annotated with granular grammatical phenomenon categories, each belonging to one of 10 core categories. As with the other TSNLP test suites, it features a strong focus on clearly defining and characterising the target phenomena as part of the annotation process. Furthermore, the core grammatical phenomena selected for annotation are of a syntactic persuasion and explicitly characterised in a grammatical-framework neutral way. Prima facie, these properties makes the English TSNLP test suite appear an intuitive candidate for use in this project. This resource, however, and linguistic test suites more generally, fail to meet the established criteria in two ways. Firstly, their constructed nature means that they fail the criterion of being naturally sourced text. The TSNLP data
is highly constrained—even for such a curated resource—due to the emphasis its creators placed on isolating phenomena through strategies such as the use of a controlled lexicon, preference for declarative sentences, unmarked word order, and the use of 3rd person singular present tense.\footnote{Scarlett and Szpakowicz (2000) describe using the English TSNLP test suite for evaluation of a broad-coverage English parser, concluding that while a useful diagnostic resource for this task, augmentation with a test suite that has the characteristics of a corpus is necessary.} Secondly, due to their primary motivation of tracking coverage over specific phenomena and identified variants, linguistic test suites do not meet the criterion of exhaustive annotation. Since a test suite item’s success or failure status is used by the grammar engineer as a proxy for the grammar’s ability to handle the construction that the example was chosen to exemplify, other phenomena that happen to be found within the example are of no particular consequence and need not be annotated for this application.\footnote{Furthermore, while the original TSNLP test suites supported character-level annotation, the only version of the English test suite I was able to access was distributed in the [incr tsdb()] format, which does not support sub-sentence level annotation, and thus this information was lost upon export.}

Another candidate is the ESD test suite (Flickinger et al. 2014b), which was discussed in Section 2.8.3, and which forms part of the documentation of the semantic interface to the ERG. The strengths of this resource are that it has a well-documented set of criteria for the characterisation and annotation of each phenomenon, as well as an operationally defined methodology for arriving at the set of target phenomena. However, as discussed in Section 2.8.3, the semantic phenomena it covers align poorly with the characterisation of phenomena being used in this research—namely syntactic phenomena. Furthermore, since the annotations in this resource are directly extracted using recorded semantic signatures from the MRS output of the ERG, this violates the grammatical framework-independent criterion. The same issue also applies to the DepBank collection (King et al. 2003) discussed in Section 2.8.4. While the annotations in this corpus experienced a substantial amount of massaging in order for them to arrive at their final form, they are ultimately based off the analyses of a deep LFG grammar (Riezler et al. 2002), which was developed in the context of the ParGram project (Butt et al. 2002). Additionally, recovery of character spans corresponding to phenomenon instances is not possible with this resource, as these are not recorded with the predicates that represent dependencies.\footnote{This is in contrast to the use of the ESD test suite, which also does not record the character spans, but are readily recoverable due to the fact that analyses produced by delph-in grammars include the character spans associated with each predicate.}

CCGBank (Hockenmaier and Steedman 2007), discussed in Section 2.8.5, differs from the resources mentioned above, in that its ancillary motivation was not the construction of a phenomenon-labelled corpus, but rather a resource which would readily facilitate the extraction of accurate dependencies from the PTB (Marcus et al. 1993), especially long distance dependencies. A side effect of the methodology used for creating this resource—which was the development of a complex algorithm that
translated PTB annotations into CCG annotations—is that it is possible to use the
CCG syntactic categories associated with distinct phenomena to project spanning an-
notations onto the corpus, thereby constructing a phenomenon-labelled corpus. This
possibility is underscored by the contents of the CCGBank manual (Hockenmaier and
Steedman 2005), which features a list of various phenomena (focusing specifically on
long distance dependencies), each of which is associated with a list of CCG syntac-
tic categories identified as being instances of the phenomenon. Again, however, this
would contravene the grammatical framework independence criterion.

Lastly, there are the two long-distance dependency resources of Rimell et al. (2009)
and Bender et al. (2011b), both described in Section 2.8.6. While the narrow scope of
these resources—with respect to target phenomena—means that they are unsuitable
as off-the-shelf resources for evaluation, it is possible that the methodology used in
their construction could be adapted here. The methodology found within Bender
et al. (2011b) is unsuitable once again due to the use of a deep grammar to extract
phenomenon annotations. It is for this same reason that, in the context of the tar-
get application of parser evaluation, the authors note that their use of the ERG to
construct the resource precludes it from being included in the array of parsers they
evaluate. The methodology found within Rimell et al. (2009) represents a more likely
candidate for adaptation. After identifying the specific constructions involving long
distance dependencies to be targeted, they used a set of regular expressions crafted
specifically to match each phenomenon’s representation within the PTB in order to
identify candidate sentences, followed by a manual review and annotation process for
each candidate sentence. This methodology of defining phenomena in terms of PTB
annotations could be adapted to target a set of grammatical phenomena as a means
of creating the required resource for evaluation. This approach is initially appealing,
since unlike many of the previously identified resources or methodologies, it does not
invoke the use of, nor appeal to the analyses underpinning, a formal grammar.

While it is certainly true that the analyses found within the PTB are flavoured
by certain trends within formal syntax,7 if these idiosyncratic shibboleths are over-
looked, the PTB annotations could arguably be described as generic phrase structure
analyses that are not wedded to any specific grammatical framework. The ability of
projects such as CCGBank to convert the PTB annotations into analyses of a specific
grammatical framework could even be construed as evidence that PTB annotations
contain sufficient information for capturing a broad range of interesting phenomena.
The reality, however, is that decisions made in the annotation scheme of the PTB have
been a limiting factor in the ability to convert PTB analyses into other target anal-
yses. For instance, as part of their translation algorithm used to create CCGBank,
Hockenmaier and Steedman (2007) describe having to contend with the relatively flat
noun phrase structure found in the PTB. In the case of quantifier phrases, they were

---

7 For instance, the use of trace elements used to encode movement analyses in constructions
like wh-questions, relative clauses, and topicalisation is associated with transformational theories of
grammar.
able to use heuristics to recover the missing structure required to match the appropriate CCG analyses. In the case of compound nouns, however, it was deemed infeasible to automate the recovery of the correct dependencies, so a right branching structure was assumed, resulting in incorrect bracketing in some cases. A further problem in the PTB’s annotation of nominals that is cited by Hockenmaier and Steedman (2007) was the representation of appositives, which are indistinguishable from noun phrase coordination and thus are also conflated within the final CCGBank analyses.

Outside of noun phrases, another limitation of the PTB annotation scheme is that it collapses the distinction between complements and adjuncts within the derivation structure. Additionally, grammatical function labels, which could be used to recover this distinction, are not used with sufficient consistency. CCG derivations on the other hand encode the valency of a verb within its category, meaning that the complement-adjunct distinction is represented explicitly and is required for a complete CCG derivation. To recover this missing information, Hockenmaier and Steedman (2007) again make use of heuristics to determine whether a constituent is a complement or adjunct in their translation algorithm.

All of these problems show that, while the methodology of defining syntactic phenomena in terms of PTB annotations might be plausible for a subset of phenomena, it does not represent a sound approach for the development of an extensible syntactic phenomenon corpus. While some of the above resources and methodologies came close to meeting the requirements for the evaluation resource, none ticked all the boxes. I therefore made the decision to create a resource annotated with phenomenon instances from scratch. This corpus would be: based on corpus data, exhaustively annotated, annotated at the character level, and manually annotated with a set of target phenomena defined using generic descriptions sourced from descriptive resources, as opposed to being associated with analyses extracted from an implemented grammar. The next two sections outline, respectively, the selection of the target syntactic phenomena and the selection of corpus data used to create this resource.

### 4.4 Selection of Phenomena

In order to select the syntactic phenomena targeted by the corpus, I established the following broad criteria. Firstly, the realisation of each phenomenon in English had to be achieved at least partly through syntax, as opposed to being realised

---

8This problem has since been tackled by Vadas and Curran (2008), who build on the work of Vadas and Curran (2007)—automatically adding gold standard noun phrase structure to the PTB—using enriched PTB data to correct errors in noun phrase structure in CCGBank, resulting in an improvement in parser performance.

9Honnibal and Curran (2007) improve upon this heuristic process by leveraging semantic labels from PropBank (Palmer et al. 2005), a corpus of sentences from the PTB annotated with predicate-argument structure.
solely through morphological marking, for instance. The chosen phenomena were also required to represent relatively broad categories, so as to encompass a range of within-phenomenon variation and subcategorisation. This is consistent with the way descriptive grammars tend to be organised and used. For example, a user of a descriptive grammar may be interested in retrieving instances of any type of relative clause, or they might specifically want reduced relative clauses or perhaps subject-extracted relative clauses as opposed to object-extracted relative clauses. While this approach doesn’t directly support granular phenomenon-variant discovery, since the subcategories are not identified and annotated, it does, however, avoid the problem of over-specifying each phenomenon and, as a result, potentially excluding variants. One way more fine-grained phenomenon-categories could still be explored is through the application of clustering techniques over the contents of the course-grained categories, in order to determine if subcategories of the phenomena can be discovered in a data-driven manner.

A further consideration I took into account was ensuring that the syntactic phenomena were cross-linguistically present over the languages of English, Japanese, and Spanish. Earlier on in this project, I had hoped to explore the efficacy of phenomenon discovery techniques cross-linguistically, with Japanese and Spanish selected in addition to English due to the existence of broad-coverage DELPH-IN grammars for Japanese (Siegel and Bender 2002) and Spanish (Marimon 2010). While this aspect of the research was discontinued, these cross-linguistic foundations can still be seen not only in the choice of syntactic phenomena, but also in the characterisation of the phenomena—most notably in the phenomenon reference document found in Appendix A. With a cross-linguistically aware methodology in place, the door is still left open in future work for investigation of the performance of phenomenon discovery across the same set of target syntactic phenomena for Spanish and Japanese in addition to English.

After consultation of the typological literature to ensure the above criteria were satisfied, I selected the following list of constructions:

1. Relative clauses
2. Interrogative clauses
3. Passive clauses
4. Imperative clauses
5. Complement clauses

In order to concretely identify the definition of these syntactic phenomena as used within this resource, I then created a reference document, which includes, for each phenomenon, a discussion of its function, a brief typological survey, and a cross-linguistic set of criteria for delimiting the bounds of the phenomenon. Rather than
trying to account for the full variability of each phenomenon across all the languages of the world, I attempted to identify a short set of criteria that was sufficiently broad to capture a range of different manifestations of the phenomenon across different languages. Once completed, the reference document was then used to inform the development of a more concise set of the annotation guidelines, which provided instructions on how to identify and annotate each phenomenon in English text. These guidelines are described in Section 4.6.2 and can be found in full in Appendix B.

I now present a brief overview of each phenomenon, describing its realisation in English and delimiting its boundaries as it is defined within the Phenomenal Corpus. For a more detailed treatment, including the cross-linguistic criteria for identification, see Appendix A, the phenomenon reference document.

4.4.1 Relative clauses

Relative clauses are subordinate clauses which serve to restrict or elaborate on a referent in the higher clause.\footnote{See Andrews (1985) for a typological survey of relative clauses.} For this resource, the scope of the phenomenon was restricted to relative clauses which modify nouns or noun phrases. This relativisation is achieved via one of the arguments of the subordinate clause sharing its reference with the head noun of the constituent on which the subordinate clause is dependant.

In English,\footnote{See Chapter 12 of Huddleston and Pullum (2002) for a thorough treatment of relative constructions in English.} relative clauses come in three broad formal categories: \textit{wh relatives} and two non-\textit{wh} relatives: \textit{that relatives} and \textit{bare relatives} (Huddleston and Pullum 2002:1034). \textit{Wh} relatives are formed through the use of a relativising pronoun such as \textit{who}, \textit{where}, \textit{which} and \textit{why}, which links the relative clause to the noun phrase being modified, as shown in (32), with the italicised text indicating the noun phrase being relativised and the underlined text indicating the relative clause. In a similar fashion, \textit{that} relatives can be formed through the use of \textit{that} to link the relative clause to the noun phrase being relativised, as shown in (33).

\begin{enumerate}
\item a. He chose the boy who eats worms.
\item b. He chose the boy whom I saw.
\item c. You know that band whose tunes I like.
\item d. The place where I live is awesome.
\item e. The reason why I left is silly.
\end{enumerate}

\begin{enumerate}
\item (32)  
\item (33)  
\end{enumerate}

If the relativised element is not the subject of the relative clause, then \textit{that} can be omitted to yield a bare relative (also referred to as a \textit{reduced relative}) as shown in (34a). Bare relatives can also be formed with the present and past participle form of
the verb, as in (34b) and (34c), in which case, the subject is the relativised element and is therefore omitted from the relative clause (Andrews 1985).

(34)  a. He chose the boy I saw.
   b. You know that band suing the little boy?
   c. Those people asked to stand will need to leave.

In English and also cross-linguistically, relative clauses are often divided up into two different kinds of relations: restrictive and non-restrictive.12 A restrictive relative clause serves to refine and further delimit the reference of the antecedent, whereas a non-restrictive relative clause elaborates upon the referent picked out by the head noun (Andrews 1985:207). In standard English orthography, the distinction between the two is made by placing commas around the non-restrictive relative clause, as illustrated by (35).

(35)  a. The dog with the white patch sat down.
   b. Their house, which has a green fence, sat on a hill.

During the refinement of the annotation guidelines two superficially similar phenomena were encountered which needed to be excluded. One is constituted by nouns which take a clausal complement headed by that, but where the head noun does not feature as an argument of the predicate in the subordinate clause and thus relativisation is absent. An example of this is presented in (36).

(36) The belief that the moon is made of cheese is fanciful.

The other phenomenon, noun phrases in apposition, is characterised by two noun phrases being placed side by side with the noun phrase on the right serving to modify or elaborate on the reference of the noun phrase on the left (Huddleston and Pullum 2002:447), as illustrated by the two examples in (37). This is similar in function to a relative clause, but syntactically distinct as the phrase modifying the head noun is nominal rather than clausal.

(37)  a. My friend Jessica also lived on the street.
   b. Jessica, the daughter of a gardener, also lived on the street.

4.4.2 Interrogative Clauses

Interrogative sentences are those uttered for the purposes of soliciting information from the audience.13 As shown by the sentences in (38), they can be broken up into

12 Rather than following the traditional distinction between restrictive and non-restrictive relative clauses, Huddleston and Pullum (2002) make use of a similar distinction between integrated and supplementary relative clauses, but which is based on differences in intonation contours. See Huddleston and Pullum (2002:1058) for their analysis.

13 See König (2007) for a typological survey of interrogatives.
different categories based upon the kind of response they solicit (König 2007). *Polar interrogatives*, or *yes-no interrogatives*, require a response which either confirms or denies the queried proposition. A felicitous response to an *alternative interrogative* requires the audience to select a response from one of the options presented in the question. *wh-interrogatives*, or *information questions*, on the other hand, are open ended questions which do not limit the response of the audience and are so-called because they make use of interrogative words mostly beginning with *wh*, such as *who, what, when, where* and *why*.

(38) a. Do you like coffee? \hspace{1cm} \textit{polar interrogative}
b. Do you like coffee, or tea? \hspace{1cm} \textit{alternative interrogative}
c. What kind of coffee do you like? \hspace{1cm} \textit{wh-interrogative}

In English main clauses, interrogatives are often indicated through the inversion of the subject and auxiliary verb, as shown in (39b), with *do* being inserted instead if no auxiliary verb is already present, as shown in (39d) (Huddleston and Pullum 2002:856). For *wh*-interrogatives with the interrogative word in the subject position, however, no inversion occurs, as shown in (39e). Polar and alternative interrogatives can also be formed out of declarative sentences through the use of rising intonation. The orthographic correlate of this is simply the use of a terminal question mark instead of a full stop, as shown in (39f) and (39g).

(39) a. They had left the building. \hspace{1cm} \textit{declarative}
b. Had they left the building? \hspace{1cm} \textit{polar interrogative with subj-aux inversion}
c. They left the building. \hspace{1cm} \textit{declarative}
d. Did they leave the building? \hspace{1cm} \textit{polar interrogative with do-insertion}
e. Who left the building? \hspace{1cm} \textit{wh-interrogative with no subj-aux inversion}
f. They left the building? \hspace{1cm} \textit{polar interrogative, rising intonation}
g. You want red, or blue? \hspace{1cm} \textit{alternative interrogative, rising intonation}

Another form of interrogative, the tag question, can be thought of as a construction which forms polar questions out of declarative sentences by adding an interrogative marker, as illustrated by (40) (Huddleston and Pullum 2002:891).

(40) Coffee is fantastic, isn’t it?

For this resource, I included all forms of the interrogative discussed here, but excluded indirect interrogatives which occur in subordinate clauses and do not exhibit subject-auxiliary inversion, as illustrated by (41). These were excluded on the basis that while they contain a question, the main clause which they belong to does not possess interrogative force (Huddleston and Pullum 2002:972).

(41) a. The researcher asked *whose coffee he had spilled.*
b. The technician wondered *how much the equipment was worth.*
4.4.3 Passive Clauses

The passive construction\(^{14}\) (or passive voice—from here on I just refer to the passive) is a valence-modifying construction that alters the way thematic roles, such as agent and patient, are mapped onto grammatical roles, such as subject, object and oblique object, so as to foreground a particular element—often the patient—and to background the element that fills the subject position in the non-passive form of the sentence (Keenan and Dryer 2007:325).

In the transitive sentence presented in (42a), the subject position is occupied by scientists and the direct object by the Higgs Boson. This sentence is an example of the active voice, in which the subject position corresponds to the agent of the predicate—in this case, the referent doing the discovering—and the direct object corresponds to the patient of the predicate—the referent being discovered. (42b) shows the corresponding passive sentence, where the patient is now found in the subject position and the agent now appears as the oblique prepositional phrase by scientists, which could also have been omitted.

(42) a. Scientists discovered the Higgs Boson.
   b. The Higgs Boson was discovered (by scientists).

The passive is often characterised as consisting of two distinct components: agent demotion, in which the agent is demoted from subject status to become an oblique object or is omitted entirely, and patient promotion, in which the patient is promoted from the direct object to become the subject (Siewierska 2011). The passive serves to foreground the patient role, placing emphasis upon it, or, if the identity of the agent is unknown, the passive can be used to avoid specifying it. In English, this often results in the patient being made the topic, since the subject position often corresponds to the topic (Huddleston and Pullum 2002:235). More generally, the following can be said of the form of the English passive (Huddleston and Pullum 2002:1428): the subject of the corresponding active form is optionally found as the complement of the preposition by, the object of the corresponding active form is found as the subject, the verb appears in the past participle form, and the auxiliary verb be is predominately used, however the verb get can also be used, and there are also variants of the passive where no auxiliary verb is used, such as in bare relatives formed with passive participles, as illustrated in (43).

Annotators were instructed to annotate the entire clause containing the passive, including the optional agent phrase when it was present. During the refinement of the annotation guidelines, a question arose of whether to include passivised relative clauses such as (43). These were chosen to be included so as to facilitate the exploration of interacting phenomena. Annotators were instructed to annotate these as instances of both relative and passive clauses.

---

\(^{14}\)For a typological survey of the passive construction, see Keenan and Dryer (2007).
(43) The Higgs Boson, recently discovered by a team of scientists, is referred to as the *God Particle*.

4.4.4 Imperative Clauses

Imperative clauses are those that convey a command, instruction or plea. There are different types of imperative or imperative-like sentence types found across the languages of the world,\(^{15}\) with some oft-identified imperative-like phenomena being presented in (44). For this study, the phenomenon was restricted to the more narrow definition of imperative constructions illustrated in (44b), which can be characterised as being uttered when appealing to an audience in order to bring about a desired state of affairs, and the audience is also deemed to be control of this state of affairs (Birjulin and Xrakovskij 2001:5). In English this is realised through the use of the infinitival form of the verb and a 2\(^{nd}\) person subject, which is often omitted, as shown in (44b) (Huddleston and Pullum 2002:857).

\[
\begin{align*}
(44) \quad &a. \text{ May you live long and prosper. } \quad \text{optative} \\
&b. \text{ (You) live long and prosper. } \quad \text{imperative} \\
&c. \text{ Let’s live long and prosper. } \quad \text{hortative} \\
&d. \text{ Let them live long and prosper. } \quad \text{hortative} \\
&e. \text{ Don’t (you) live long and prosper. } \quad \text{prohibitive}
\end{align*}
\]

4.4.5 Complement clauses

Complement clauses are clauses which function as the core argument to a predicate (Dixon *et al.* 2008:1). Since this is a somewhat nebulous definition, potentially covering a broad range of phenomena, I narrowed the scope of the phenomenon to constructions that meet the following criteria suggested by Dixon *et al.* (2008) in order to minimise ambiguity in the annotation task.\(^{16}\)

1. The constituent has the internal structure of a clause, with it possessing a predicate at a minimum, and with any arguments of this predicate being marked in the same way they would be in a main clause.

2. The constituent functions as an argument of a verbal predicate in a higher clause.

---

\(^{15}\)See König (2007) and also Birjulin and Xrakovskij (2001) for typological surveys of imperative clauses.

\(^{16}\)See Dixon *et al.* (2008), also, for a typological survey of complement clauses and complementation strategies.
In English, complement clauses can take the form of a full sentence, as in (45a), in which case they tend to represent a fact or proposition, or they can be reduced, as in (45b) and (45c). In the latter case, the verb will be non-finite and the subject is normally omitted, leaving the clause to appear as a verb phrase.\(^{17}\)

(45) a. I remembered (that) I don’t like durian. \(\text{that-clause}\)
b. I remembered to feed the trolls. \(\text{infinitival}\)
c. I remembered seeing her. \(\text{participial}\)
d. I remembered his singing the song. \(\text{gerundal}\)

A difficulty identified during the annotation task was in differentiating between clauses that function as complements of a verb, and those that function as clausal adjuncts. For example, in (46a), to let the cat out is a complement of the verb like, however in (46b), it is not. A heuristic that is helpful here is determining whether the phrase in question is required. If it can be omitted, as it can in (46b), then it is likely an adjunct.

(46) a. I like to let the cat out.
b. I opened the door to let the cat out.

After the annotation task was completed, it was brought to my attention that gerundal complements such as (45d) do not strictly meet the first criterion for inclusion as verbal complements since not all of their internal structure is characteristic of a clausal phrase. In particular, the appearance of a possessive determiner—rather than a noun phrase subject—is characteristic of a noun phrase-like constituent. Furthermore, in the final corpus, numerous gerundals were indeed identified as complement clauses. I treated this as a deficiency in the established criteria for the classification of verbal complements, as opposed to a failure in their interpretation, as the separation of gerundal complements from participle complements is not always possible. For example, in (45d), singing can easily be identified as a gerundal form by virtue of the possessive determiner his, however, in (45d), seeing is ambiguous between gerund and participle forms. Indeed, Huddleston and Pullum (2002:1220) argue that a useful distinction between the two verb forms cannot be maintained and argue for the use of a single gerund-participle form. Since asking annotators to distinguish between gerund and participle-formed complements in order to exclude the former is clearly not a practical option, I continued to include gerundal complements within this phenomenon. Were the annotation task to be performed again, the criteria and guidelines would be updated to explicitly include gerundals even though they partially violate the established criteria.

\(^{17}\)In Huddleston and Pullum (2002), complement clauses that take the form of full sentences are handled in Chapter 11 which deals with content clauses and reported speech. Non-finite complements are covered in Chapter 14, which covers non-finite and verbless clauses.
4.4.6 Summary of Phenomena

These characterisations of the five syntactic phenomena provide a concrete set of criteria for their identification. Their establishment was a crucial step in the development of the corpus, with it being required for the development of the annotation guidelines, which are described in Section 4.6.2. A summary of these syntactic phenomena is presented in Table 4.1, which includes a brief characterisation of each phenomenon, including notable decisions that were made in order to delimit the range of the phenomenon for use in this research.

4.5 Selection of Corpora

I selected two sources from which to draw text for the Phenomenal Corpus: The Adventure of the Speckled Band, a Sherlock Holmes novel by Arthur Conan Doyle and The Wall Street Journal section of the Penn Treebank (Marcus et al. 1993). Each of these sources formed a distinct section in the final corpus. Text from The Adventure of the Speckled Band became the first section, which was used for developing and trialling the data collection methodology. This text was selected in part for being public domain data, but also for its characteristic style, being syntactically rich and varied. The hope was that this syntactic complexity would offer a trial by fire of the annotation guidelines.

The Wall Street Journal section of the Penn Treebank (WSJ, from hereon) was selected to be the source of data for the second section, intended to be the primary resource for the development of the phenomenon discovery techniques developed in this thesis. This corpus was chosen for its strong history of being used for the development of annotated resources within the field of NLP, as discussed in Section 2.5.3. This is evidenced by many of the projects discussed in the previous section, such as CCGBank, DepBank and the long-distance dependency corpus developed in Rimell et al. (2009), which are all based on at least part of the WSJ. This popularity increases the likelihood of the corpus under development here being re-used for other applications.

A more concrete consideration for the selection of WSJ text is that it enables the use of the annotations found in DeepBank, the dynamic treebank containing disambiguated derivation trees produced by the ERG, which was outlined in Section 3.3. The relevance of this lies not in the annotation stage, as using annotations produced by the ERG at this stage would introduce an unacceptable bias towards the style of linguistic analyses found in the ERG, contravening the framework-independent desideratum stated in Section 4.2.4. Instead, a disambiguated treebank of ERG analyses is relevant for the syntactic phenomenon discovery methodology developed in Chapter 5. In the context of an interactive tool that parses user-supplied exemplars dynamically, one of the many analyses produced by the grammar must be selected to be used as input. The best analysis according to the parse selection model could
<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Description</th>
<th>Notable decisions</th>
</tr>
</thead>
</table>
| Relative clause | A subordinate clause which serves to restrict or elaborate on a referent in the higher clause. | • Both restrictive and non-restrictive relative clauses were included.  
• Bare relative clauses were included.  
• Noun phrases with a clausal complement that do not include a relativised referent were excluded.  
• Noun phrases in apposition were excluded. |
| Interrogative clause | Sentence uttered for the purposes of soliciting information from the audience. | • Polar, alternative and wh-interrogatives were all included.  
• Indirect interrogatives were excluded. |
| Passive voice | A valence-modifying construction which yields the patient role occupying the subject position and the agent manifesting as an optional prepositional phrase. | • Optional agent phrase was included.  
• Passivised relative clauses were counted as instances of both passive clauses and relative clauses. |
| Imperative | Sentence produced when appealing to an audience in order to bring about a desired state of affairs and the audience is also deemed to be control of this state of affairs. | • Optatives, hortatives, and prohibitives (negative imperatives) were excluded. |
| Complement clause | Subordinate clause which functions as the argument to a verbal predicate and itself possesses the constituent structure of a clause. | • Constituent needed to possess the internal structure of a clause.  
• Constituent was required to function as an argument of a verbal predicate in a higher clause.  
• Gerundal complement clauses were additionally included. |

Table 4.1: The syntactic phenomena chosen for use in the evaluation resource.
be used, however, if this does not include an appropriate treatment of the target phenomenon, for the discovery process to work, the user will need to select an appropriate analysis from amongst the others. By using a disambiguated treebank, for each exemplar provided as input to the phenomenon discovery methodology, the best analysis produced by the grammar can be used without requiring user-intervention, which will improve the efficiency of the discovery process and remove this potential source of noise.

4.6 Data Collection

In this section, I describe the methodology used to collect the data used to create the Phenomenal Corpus as well as presenting the results of the annotation. In addition to initially outlining the general methodology, this involves describing the annotation guidelines given to the annotators, presenting an overview of the annotations recorded by annotators, describing the creation of the final gold standard annotations used in the Phenomenal Corpus, presenting an error analysis of each annotator’s output, and, finally, describing the use of an external resource to improve the quality of the annotations.

4.6.1 Methodology

The format chosen for the annotations was simple character spans delimiting the range of the sentence pertinent to the phenomenon. This met the identified desideratum of supporting character-level annotations, while the unstructured nature of the format also served to satisfy the desideratum of the annotations being grammatical framework-independent. Annotation was performed using brat, a browser-based rapid annotation tool (Stenetorp et al. 2012). Both target corpora were already sentence-tokenised, so brat was configured to perform annotation on a line by line basis. Annotators were required to have a background in syntactic theory and were sourced from volunteers affiliated with the Natural Language Processing Group at The University of Melbourne and the Computational Linguistics group at the University of Washington. The procedure I adopted for collection of the data is outlined below.

1. A set of annotation guidelines was developed to provide annotators with instructions on how to perform the task, including specific instructions on how to identify each phenomenon. The contents of the final version of this document are discussed in further detail in the following section.

18http://brat.nlplab.org
2. A 250 line section of *The Adventure of the Speckled Band* was selected and then annotated by myself and one of my supervisors, Tim Baldwin, who had also been involved in preparation of the guidelines.

3. Based on feedback from the two annotators, the guidelines were refined to add missing instructions and to eliminate ambiguity by making the boundaries of each phenomenon clearer, and in some cases, changing the boundaries, updating the definitions of the phenomenon categories.

4. Four annotators were selected for annotation of DeepBank data. One of these annotators was involved in the preparation of the guidelines\textsuperscript{19} and the other three were previously unfamiliar with the task.

5. A collection of 20 lines of DeepBank text, manually curated so as to be target-phenomenon rich, was given to the three annotators unfamiliar with the task as a trial-run annotation round.

6. Annotators were provided with vetted annotations of the trial-run text to use as reference for the remainder of the task. A sample of these can be seen in Figure 4.1. The results of this trial run also led to some minor refinements of the annotation guidelines and revealed that one of the new annotators was inconsistently identifying target phenomena, leading to the decision to reduce the annotators for this task down to three.

7. Section 8 of DeepBank (corresponding also to section 8 of the PTB and consisting of 477 lines of text) was selected for annotation by the three annotators. The text was divided into two parts, with each being doubly annotated, and with the annotator more familiar with the guidelines (again, namely, myself) annotating both sections.

8. The final gold standard annotations for the WSJ text were then produced by proceeding through each line and resolving any differences between annotators. This resolution process was performed by referring to the annotation guidelines, and, where this was not able to resolve a discrepancy, I consulted with Tim Baldwin in order to make decisions on a case-by-case basis. This process is also discussed in Section 4.6.4.

### 4.6.2 Annotation Guidelines

The annotation guidelines included general instructions for annotation as well as phenomenon-specific instructions that provided both a set of criteria for identification

\textsuperscript{19}This annotator was me. In the presentation of annotator output later in this chapter, I appear as Annotator 1.
He also is a consensus manager, insiders say.

Associates say Mr. Hahn picked up that careful approach to management as president of Virginia Polytechnic Institute. But he also showed a willingness to take a strong stand.

In 1970, Mr. Hahn called in state police to arrest student protesters who were occupying a university building.

The son of a physicist, Mr. Hahn skipped first grade because his reading ability was so far above his classmates.

Mr. Hahn agrees that he has a "retentive" memory, but friends say that's an understatement.

"The record of companies that have diversified isn't all that impressive," he says.

It would be a good match, Mr. Hahn and many analysts say, of two healthy companies with high-quality assets and strong cash flows.

To accommodate the additional cash assistance, the House Appropriations Committee last week was required to reallocate an estimated $140 million from the Pentagon.

Figure 4.1: Screenshot of the brat annotation tool, showing some of the vetted annotations from the manually curated DeepBank text used for the trial-run annotation round.
of each phenomenon and instructions for its annotation. The criteria for identification were based on the phenomenon delimitation criteria in the phenomenon reference document, which were reduced to target only English constructions and presented so as to provide a simple guide which would hopefully prevent annotators from becoming mired in analysis. The general instructions for annotation are summarised below. For further details, including the phenomenon-specific instructions, see the complete annotation guidelines in Appendix B.

For each line of text, annotators were instructed to exhaustively annotate all identified instances of the phenomenon. In order to reduce cognitive load, it was suggested that they annotate each phenomenon in a separate pass through the text. When a phenomenon was observed, annotators were instructed to mark the character span they deemed to be relevant based on the phenomenon-specific instructions. In order to reduce annotation of unrelated text, annotators were asked to annotate only the minimum amount of text involved in the phenomenon, but to ensure that spans were not left incomplete. For instance, it might be tempting for annotators to exclude adverbial phrases at the start or ends of sentences, such as *after being ignored for two days* in (47), when annotating passive clauses, since the adverbial is not a necessary component of the passive construction. This was considered an incomplete annotation on the grounds that the annotation instructions called for annotation of the entire clause containing the passive, which clearly includes the adverbial phrase. Furthermore, it is harder to apply the argument for omitting the adverbial when it occurs between the subject and the verb, such as in (48). To maintain consistency then, the adverbial should be included in the annotation in both examples.

(47) After being ignored for two days, the cake was eaten by Pat.

(48) The cake, after being ignored for two days, was eaten by Pat.

A specific question that arose during the refinement of the annotation guidelines was what should be done for coordinated constructions, where both conjuncts could be seen as instances of the same target phenomenon, as in (49) and (50) where the complement of the verb *thought* involves sentential coordination in both cases. In such cases, it was necessary to determine whether the complement be annotated as one instance or two separate instances. Annotators were instructed to annotate as two separate instances if the sentence could be analysed such that the top-level constituent pertaining to the phenomenon was a coordinating constituent (such as in (49)) and to annotate as one single instance where the coordinating constituent is lower down. This decision was made on the basis that dividing the lower coordination up into two annotations could result in either incomplete annotations, or in the case

\[\text{20}\] The complete phenomenon reference document is included in Appendix A.

\[\text{21}\] This meant that constituents illustrating multiple phenomena were to be annotated with multiple distinct annotations. For instance, passivised relative clauses should be annotated as both passive clauses and relative clauses.
of (50) an implicit modification to the kind of phenomenon identified—the omission of the complementiser *that*.

(49) I thought *that* we would go to the shop and *that* we would get a sausage roll.

(50) I thought that we would go to the shop and we would get a sausage roll.

In the case of punctuation characters located at the beginning or end of a span, annotators were instructed to annotate the punctuation as binding tightly to the nearest word token, by including any punctuation characters adjacent to the beginning or end of the span up until a whitespace character is encountered. This decision was made primarily to ensure that no punctuation characters which may be relevant to the phenomenon were omitted. A further consideration is that retaining the punctuation also serves to maximise the downstream utility of this resource, where punctuation may or may not be deemed relevant, since the alternative form of the annotation can always be resolved simply by stripping initial and terminal punctuation from the span.

Text can also contain inherent syntactic ambiguity. An example of this is prepositional attachment ambiguity, where a preposition phrase could be either modifying the verb or functioning as the modifier of a noun in a sentence, with each possibility yielding a distinct semantic reading of the sentence. This can be seen in (51), which was taken from the WSJ component of the Phenomenal Corpus, where there is ambiguity between whether the preposition phrase *with the capital-gains provision* is functioning as an adjunct of the verb *amend* or a complement of the noun *Senate*. In this case, the natural reading is for the preposition phrase to modify for the verb, with the ambiguity being resolved for the reader through real-world knowledge that things, especially bills, tend to be amended with provisions, as well as the fact that there is generally speaking only one Senate, and so delimiting the scope of this referent would be redundant.\(^{22}\)

(51) And they threatened to try to amend any other revenue bill in the Senate with the capital-gains provision.

In cases where syntactic ambiguity could impact upon the outcome of the annotation and could not be resolved through context, annotators were instructed to resolve the ambiguity by assuming the ambiguous constituent attaches to the higher node of the corresponding derivation tree. In the case of prepositional ambiguity, this would mean treating the preposition as modifying the verb. This is consistent with the guidelines developed for the LinGO Redwoods treebank (Oepen *et al.* 2004).

\(^{22}\)This sentence possesses an additional instance of prepositional attachment ambiguity, with the preposition phrase *in the Senate* being ambiguous between modifying the verb *amend* or being a complement of the noun *bill*. 
Table 4.2: Breakdown of attested instances across annotators as well the number of instances present in the final gold standard annotations, across the two halves of WSJ section 8.

<table>
<thead>
<tr>
<th></th>
<th>WSJ-08 part 1</th>
<th>WSJ-08 part 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines</td>
<td>250</td>
<td>227</td>
</tr>
<tr>
<td>Annotator</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Passive</td>
<td>47</td>
<td>60</td>
</tr>
<tr>
<td>Gold</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>Relative</td>
<td>76</td>
<td>74</td>
</tr>
<tr>
<td>Gold</td>
<td>62</td>
<td>60</td>
</tr>
<tr>
<td>Complement</td>
<td>158</td>
<td>111</td>
</tr>
<tr>
<td></td>
<td>153</td>
<td>85</td>
</tr>
</tbody>
</table>

4.6.3 Annotations Collected from the WSJ

This section presents the output of the annotators for the WSJ section of the corpus. Table 4.2 shows the breakdown of the annotations made by the different annotators across the two parts of the WSJ section for the different phenomenon categories, as well as the corresponding numbers of annotations found in the gold standard, the creation of which is described in the following section. The breakdown across annotators is not presented for the text, as the annotation guidelines were in flux during its annotation, with annotator feedback being used to make revisions during the annotation process. For the final annotations of the text, see Table 4.3, which presents the final gold standard annotations.

As can be seen from Table 4.2, annotations were not recorded for the WSJ text for both imperative and interrogative phenomenon categories. The decision to omit these categories from the annotation of the WSJ was made after it became apparent that instances of these categories would be very much under-represented, or possibly completely absent, from the WSJ text. This was a function of the fact that imperatives and interrogatives are syntactic phenomena that are under-represented in newswire text as compared to other domains. This unfortunate result arose due to the selection of the target phenomena before the selection of the corpus data, which, in hindsight, was a poor methodological decision. This is discussed further in Section 4.8.2.

4.6.4 Creating the Gold Standard Annotations

The final statistics for the different categories across both components of the Phenomenal Corpus are presented in Table 4.3. For the component, the gold standard annotations were deemed to be the existing annotations at the end of the development of the annotation guidelines, as this process already required these annotations to be subject to significant scrutiny. For the
Table 4.3: The gold standard phenomenon annotations arrived at after the disagreement resolution stage.

<table>
<thead>
<tr>
<th>TSB</th>
<th>WSJ-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines</td>
<td>250</td>
</tr>
<tr>
<td>Passive</td>
<td>25</td>
</tr>
<tr>
<td>Relative</td>
<td>62</td>
</tr>
<tr>
<td>Complement</td>
<td>54</td>
</tr>
<tr>
<td>Imperative</td>
<td>5</td>
</tr>
<tr>
<td>Interrogative</td>
<td>16</td>
</tr>
<tr>
<td>Total</td>
<td>162</td>
</tr>
</tbody>
</table>

WSJ component of the corpus, the gold standard annotations were produced through a disagreement resolution process. This involved working through each sentence, comparing annotations by both annotators and resolving any disagreements, using the annotation guidelines as basis for arbitration. Where the guidelines did not provide sufficient guidance to make a decision, I consulted with Tim Baldwin, who was also involved in the development of the guidelines, and a decision was reached on a case-by-case basis.

Disagreements between the annotators largely fell into two broad categories: one of the two annotators marked a span while the other did not, or the two annotators both marked some portion of the span but did not agree on the boundaries. A more fine-grained classification scheme was devised for deviations from the gold standard by each annotator. This is discussed in the next section.

### 4.6.5 Error Analysis

In order to gain insight into the kind of mistakes that were made by the annotators, I performed an error analysis, comparing the annotations of each annotator with the gold standard annotations. Each deviation from the gold standard was categorised according to two different properties: the type of error and the phenomenon category the error pertained to. After manual inspection of all annotator errors, the following types of errors were identified:

**Missing instance:** a phenomenon instance that occurs in the gold standard was omitted by the annotator.

**Spurious instance:** a phenomenon instance was identified by the annotator but does not exist in the gold standard.

**Incorrect label:** a phenomenon instance was identified by the annotator and corresponds to an instance within the gold standard, however the
category assigned by the annotator is incorrect. In this case, the error is assigned to the correct phenomenon category found in the gold standard. If, in the gold standard, a clause is annotated with two different phenomena, and an annotator only identified one of them, this would yield a single missing instance error, not an incorrect label error.

**Span mismatch:** a phenomenon instance was identified by the annotator, however the character boundaries of the span do not match the corresponding instance in the gold standard. Span mismatches that were the result of omitting one or more punctuation characters were assigned to the category *missing punctuation*, which is described below.

**Conjunction split:** a phenomenon instance was identified by the annotator that contains a conjunction, however, within the gold standard, the corresponding span is annotated as two separate instances of the same phenomenon. In this case, this is only counted as one error and is not included as a *missing instance* error.

**Missing punctuation:** a phenomenon instance was identified, but in order to match an instance in the gold standard, one or more additional punctuation characters should have been included on either end of the span.

Table 4.4 and Table 4.5 show the error analysis for each pair of annotators across part 1 and part 2 of the WSJ text respectively. While these tables provide a comprehensive breakdown of the ways in which the annotators deviated from the gold standard, they don’t provide a good overview of the aspects of the task that were found to be difficult across all annotators. This can be seen more clearly in Table 4.6, which provides a breakdown of errors across all annotators, showing both the proportion of errors from each error type and phenomenon category, as well as the breakdown of error types for each phenomenon category.

As can be seen in Table 4.6, error analysis showed that the two most common classes of errors were missing an instance of a phenomenon that should have been annotated, followed by incorrectly delimiting the span of a phenomenon instance. This suggests that, for annotators, even before the potentially difficult question of how much text to annotate for a particular phenomenon instance becomes relevant, a difficulty was sighting the presence of the phenomenon in the first place.

### 4.6.6 Improving the Annotations

Given these high numbers of missing instances and incorrect span ranges within the recorded annotations, it is possible that some of these errors persisted even in the gold standard. Ideally, the gold standard could be vetted against some other resource providing a ground truth for the occurrence of phenomenon instances within
### Table 4.4: Annotator 1 and annotator 2’s errors for WSJ-08 part 1.

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing instance</td>
<td>14</td>
<td>7</td>
<td>15</td>
<td>36</td>
</tr>
<tr>
<td>Spurious instance</td>
<td>0</td>
<td>9</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>Incorrect label</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Span mismatch</td>
<td>6</td>
<td>2</td>
<td>15</td>
<td>23</td>
</tr>
<tr>
<td>Conjunction split</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Missing punctuation</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>20</td>
<td>19</td>
<td>38</td>
<td>77</td>
</tr>
</tbody>
</table>

(a) Annotator 1

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing instance</td>
<td>18</td>
<td>11</td>
<td>18</td>
<td>47</td>
</tr>
<tr>
<td>Spurious instance</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Incorrect label</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Span mismatch</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Conjunction split</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Missing punctuation</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>26</td>
<td>18</td>
<td>28</td>
<td>72</td>
</tr>
</tbody>
</table>

(b) Annotator 2
### Table 4.5: Annotator 1 and annotator 3’s errors for WSJ-08 part 2.

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing instance</td>
<td>12</td>
<td>6</td>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>Spurious instance</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Incorrect label</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Span mismatch</td>
<td>0</td>
<td>1</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>Conjunction split</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Missing punctuation</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>12</td>
<td>11</td>
<td>22</td>
<td>45</td>
</tr>
</tbody>
</table>

(a) Annotator 1

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing instance</td>
<td>6</td>
<td>11</td>
<td>31</td>
<td>48</td>
</tr>
<tr>
<td>Spurious instance</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Incorrect label</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Span mismatch</td>
<td>20</td>
<td>2</td>
<td>4</td>
<td>26</td>
</tr>
<tr>
<td>Conjunction split</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Missing punctuation</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>30</td>
<td>19</td>
<td>35</td>
<td>84</td>
</tr>
</tbody>
</table>

(b) Annotator 3

### Table 4.6: Relative percentages of the types of annotator errors across all annotators.

<table>
<thead>
<tr>
<th>Error type</th>
<th>Total</th>
<th>All (%)</th>
<th>Pass. (%)</th>
<th>Rel. (%)</th>
<th>Comp. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing instance</td>
<td>156</td>
<td>56</td>
<td>32</td>
<td>22</td>
<td>46</td>
</tr>
<tr>
<td>Spurious instance</td>
<td>26</td>
<td>10</td>
<td>0</td>
<td>62</td>
<td>38</td>
</tr>
<tr>
<td>Incorrect label</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Span mismatch</td>
<td>72</td>
<td>26</td>
<td>7</td>
<td>14</td>
<td>79</td>
</tr>
<tr>
<td>Conjunction split</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Missing punctuation</td>
<td>18</td>
<td>6</td>
<td>39</td>
<td>39</td>
<td>22</td>
</tr>
<tr>
<td><strong>All errors</strong></td>
<td>278</td>
<td>100</td>
<td>32</td>
<td>24</td>
<td>44</td>
</tr>
</tbody>
</table>

Table 4.6: Relative percentages of the types of annotator errors across all annotators. For each error type, the table shows: the total number of errors for that type, the ratio of this error type to all errors, and then a breakdown of the ratios of each phenomenon type to the numbers of errors for that error type.
the chosen text. Of course, the need to bootstrap a gold standard from the combined annotations of multiple annotators arose precisely because such a resource does not exist. This is a problem that faces any annotation task targeting a specific set of categories with bespoke criteria for identification.

While direct access to some kind of ground truth is not a possibility, one approach to improve the quality of the annotations in the resource is to use other resources to triangulate on a final set of annotations that, while still cannot be claimed to be “correct”, might at least boost the confidence that, over time, fewer errors would be discovered by users of the resource. In doing this, it is important, however, to remain faithful to the desideratum spelled out in Section 4.2.4, which stipulated that the evaluation resource should remain grammatical framework-independent.

Of the two types of errors likely to be persisting in the gold standard, I focused on the situation where the span of the annotation in the gold standard was incorrectly delimited. Given that all three of the phenomena under investigation pertain to the clausal level, they should yield spans that correspond to relatively uncontroversial (that is, grammatical-framework neutral) phrasal elements. Since reasonably accurate unlabelled phrase-structure annotations are easy to come by, these can be used to sanity check the span boundaries of the gold standard annotations. I used ERG annotations from DeepBank to create this alignment to Phenomenal Corpus annotations. After manually vetting the annotations that did not align with ERG annotations, the following numbers of annotations were updated with new character spans: 17 complement clauses, 3 passive clauses, and 1 relative-clause.

These modifications do not, I believe, leave this resource open to the claim that it presupposes the use of an external resource based on a specific grammatical framework because the criteria for identifying clause boundaries are relatively uncontroversial—certainly compared with internal structural analysis—and, in this case, the resource is only being used to bring possible errors to the attention of a human, who determines whether the span does need updating, based on the endogenous annotation guidelines from this project.

A limitation of this approach is that it only flags errors which were caused by span boundaries not aligning to what the ERG considered to be a phrasal boundary. It does not assist with locating issues where the boundaries of an annotation do form a coherent constituent, but are not consistent with the annotation guidelines. Furthermore, as already noted, it does not assist with the recovery of missing instances. In Section 4.7, I investigate ways of gaining insight into the quality of the resource, with the aim of ascertaining the likelihood of the resource containing such issues.

### 4.6.7 Data Packaging

In order to make the collected annotations available for use in other parts of the research in this thesis—as well as in downstream research—they needed to be packaged up in an appropriate format. I chose the profile format used by [incr tsdb()] (Open
and Flickinger 1998, Oepen and Carroll 2000), the tool used within the DELPH-IN consortium for test suite management and competence and performance profiling, which was discussed in Section 2.9.1. This format stores information within profiles, which collect together a group of test items, with each item containing the text from a sentence or fragment, along with accompanying metadata. In addition to storing information about the results of parsing runs over test items, these profiles can also associate items with phenomenon records, which indicates the presence of a specific phenomenon. The labels associated with each phenomenon instance are themselves records, meaning that the format supports the specification of custom phenomenon labels.

As discussed in Section 2.9.1, the \texttt{[incr tsdb()]} tool also supports querying the contents of profiles using an SQL-like query language. This means it is straightforward to specify queries that retrieve test items with specific phenomena. Furthermore, such queries can be combined with constraints over other information pertaining to the profile that is available within the \texttt{[incr tsdb()]} format. This enables the construction of queries such as “retrieve all items of length less than 15 words and which contain two relative clauses” and “retrieve all items that received at least one analysis by the grammar and contain a passive clause”. Since the former query only invokes constraints over the test items and phenomenon records, with no grammar involved, it can be performed with \texttt{[incr tsdb()]} alone. The second query does require the result of running a grammar over the profile with a parser, however it only requires the number of analyses produced by the grammar. This is indicative of the fact that the \texttt{[incr tsdb()]} tool and profile format is agnostic to the implementation of the precision grammar, as evidenced by its use in the CoreGram project, which was discussed in Section 2.6.6. At the same time, this format is a familiar target for those working within the DELPH-IN ecosystem, with various tools and resources interfacing with it—such as the PyDelphin Python library\footnote{https://github.com/delph-in/pydelphin}, making it convenient for use in this research.

The final annotations from both \textit{The Adventure of the Speckled Band} and WSJ text were packaged as two \texttt{[incr tsdb()]} profiles using the standard approach for converting a test corpus into this format, which is to map sentences individually onto consecutive test items. For each test item, phenomenon records were created for each annotation from the gold annotations\footnote{In the case of the annotations from DeepBank, which is already released as \texttt{[incr tsdb()]} profiles, the existing profile just had to be augmented with phenomenon records.}. As the \texttt{[incr tsdb()]} profile currently only supports item-level phenomenon records, the character level spans for each annotation was included as supplementary data external to the profile. These profiles are publicly available, as a part of the Typediff tool\footnote{https://github.com/ned2/typediff/}, which is an implementation of the syntactic phenomenon discovery methodology described in Chapter 5.

---

\footnotetext{https://github.com/delph-in/pydelphin}

\footnotetext{https://github.com/delph-in/pydelphin}
4.7 Assessing Annotation Reliability

This section further investigates the quality of the annotations produced, this time through the lens of annotation reliability, which, for this task, involves gauging the extent to which annotators agree on their identification of phenomenon instances and also the identification of span boundaries for these instances. If annotator agreement is high, this is indicative of reproducibility, as it suggests that the annotators’ mental models of the phenomena described by the annotation guidelines are consistent. If agreement is low, then either the task presented to the annotators is inherently difficult and will always yield a high amount of disagreement, or the coding scheme and accompanying annotation guidelines do not adequately constrain the phenomenon—or indeed, both could hold. In either case, some aspect of the methodology used in gathering the data will likely need to be revised. In this section I describe the steps taken to assess inter-annotator agreement over the recorded annotations of the Phenomenal Corpus. As will be seen shortly, commonly used approaches to measuring inter-annotator agreement tend to make assumptions regarding the nature of the annotation task which do not hold true for the task at hand.

Measuring inter-annotator agreement over hand-annotated data has a strong history in the field of computational linguistics. The most common kind of annotation task involves the labelling of text which has been pre-segmented into linguistic constituents, either by hand or automatically. Examples of such tasks include named-entity tagging, part of speech tagging, and dialogue act tagging. When measuring inter-annotator agreement in such tasks, agreement only need be assessed on the labels assigned by the annotators to the segments, with the segmentation process assumed to be reliable. The annotation task described in this chapter differs in that annotators were required to identify segments of text and label them. I will return to the challenge this difference poses in Section 4.7.2 after briefly outlining existing approaches to assessing inter-annotator agreement.

4.7.1 Chance-corrected Inter-annotator Agreement

A simple approach to measuring annotator agreement is to just use the percentage of agreement, also referred to as observed agreement (Artstein and Poesio 2008). For two annotators, this is simply the number of items on which the annotators agree divided by the total number of items. This measure has the significant drawback of not being comparable across studies as it does not take into account that some amount of agreement is the result of chance, which will vary with both the number of categories in the annotation scheme and the underlying distribution of categories.

---

26For a comprehensive survey of approaches to inter-annotator agreement in natural language processing, see Artstein and Poesio (2008).

27The tagging of sentences into categories describing the way in which it contributes to the discourse, such as statement, agreement, disagreement, and question.
within a dataset (Artstein and Poesio 2008). Firstly, agreement across coding schemes will, on average, be biased towards those with fewer categories—for example, two annotators randomly selecting between two categories will be more likely to have higher agreement than two annotators randomly selecting between three categories. Secondly, agreement will, on average, be higher when one category is more frequent, since the chance of disagreement is much lower.

There are a number of different coefficients for measuring inter-annotator agreement that take into account chance agreement. The three most notable of these are $S$ (Bennet et al. 1954), $\pi$ (Scott 1955), and $\kappa$ (Cohen 1960). All three take the same approach of finding the ratio of the amount of observed agreement beyond chance to the amount of agreement beyond chance that can be attained. As per Artstein and Poesio (2008), I will refer to the amount of agreement expected by chance as $A_e$ and the amount of agreement observed as $A_o$. Given that the amount of agreement above chance that was observed can be expressed as $A_o - A_e$ and the attainable amount of agreement above chance can be expressed as $1 - A_e$, the general form that these coefficients take is described by Equation 4.1. This formula yields agreement values between $-A_e$ and $1$, with $-A_e$ indicating no agreement, $1$ indicating perfect agreement, and with a result of $0$ indicating that agreement was as good as chance.

$$S, \pi, \kappa = \frac{A_o - A_e}{1 - A_e} \quad (4.1)$$

For two annotators, $A_o$ is easy to calculate as this is simply the observed agreement described previously: the number of items the annotators agree on divided by the total number of items. $A_e$, the amount of agreement expected by chance is more difficult, however, as it requires a model of chance annotation, a way of calculating $P(k|a_i)$, the probability that annotator $a_i$ will label an item with category $k$, which, without access to the actual distribution of each category, is unknown. Were such data available, it would suggest that a set of gold standard target annotations already existed.

The three different coefficients each differ in how they model expected agreement. $S$ side-steps this difficulty by assuming a uniform distribution, such that all categories are considered equally likely. A major downside to this approach is that by simply adding more categories, which may never even be used, the coefficient can be artificially inflated (Artstein and Poesio 2008). $\pi$ uses the annotations provided by the annotators to estimate the prior distributions of the categories, under the assumption that annotators share the same distribution. $\kappa$ also uses the annotations provided by the annotators to estimate the prior distributions of the categories, but by modelling individual distributions for each annotator, thus taking into account bias of individual annotators.

The coefficients discussed so far all have the limitation that they are restricted to measuring agreement between two annotators. This limitation arises from $A_o$ being defined as the proportion of items with agreement over the total items, since, with more than two annotators, there can be agreement and disagreement simultaneously
on the one item. Fleiss (1971) generalises \( \pi \) to multiple annotators by generating all possible pairwise judgements and taking the agreement on a particular item to be the proportion of agreeing judgements pairs over the total number of judgement pairs. This coefficient is often referred to as Fleiss’ kappa in spite of the fact that it is actually a generalisation of \( \pi \). Originating in the field of content analysis,\(^{28}\) it has become a popular choice for measuring inter-annotator agreement in computational linguistics after being introduced to the community by Carletta (1996), who referred to it as the kappa coefficient or \( K \). Due to their similar theoretical underpinnings (and perhaps to smooth over some of these terminological inconsistencies), the family of chance-corrected coefficients discussed here are often referred to as kappa-like coefficients in the literature. For consistency with the literature, I will refer to the coefficient that Fleiss derived from \( \pi \) as Fleiss’ kappa.

I selected Fleiss’ kappa as a candidate measure in order to support the possibility of evaluating annotation performed with more than two annotators.\(^{29}\) As already mentioned, \( A_o \) in Fleiss’ kappa is defined as the proportion of agreeing judgement pairs over the total number of pair-wise judgements across all annotators. The amount of agreement on a particular item \( i \) is given in Equation 4.2,\(^{30}\) where \( c \) is the total number of annotators, and \( n_{ik} \) is the number of times an item \( i \) is labelled as an instance of category \( k \) across all annotators. For each category \( k \), there are \( \left( \begin{array}{c} n_i \\ 2 \end{array} \right) \) pairs of agreeing judgements for item \( i \), which, when summed over all categories \( k \in K \) and divided by \( \left( \begin{array}{c} c \\ 2 \end{array} \right) \) —the total number of judgement pairs for each item—yields the agreement for item \( i \).

\[
\text{agr}_i = \frac{1}{\binom{c}{2}} \sum_{k \in K} \binom{n_{ik}}{2} = \frac{1}{c(c-1)} \sum_{k \in K} n_{ik}(n_{ik} - 1) \tag{4.2}
\]

\( A_o \) is then the mean of \( \text{agr}_i \) for all items \( i \in I \), as shown in Equation 4.3.

\[
A_o = \frac{1}{i} \sum_{i \in I} \text{agr}_i = \frac{1}{ic(c-1)} \sum_{i \in I} \sum_{k \in K} n_{ik}(n_{ik} - 1) \tag{4.3}
\]

Following on from this, \( A_e \) is defined as the probability that two arbitrarily selected judgement pairs for an item would be in agreement. Similar to \( \pi \), chance agreement is modelled using a single distribution, taken from the aggregate judgements of all annotators. \( A_e \) thus uses \( \hat{P}(k) \), the proportion of items assigned to category \( k \), which, as shown in Equation 4.4, is the total number of such assignments by all annotators \( n_k \) divided by the overall number of assignments, which is the number of items \( i \) multiplied by the number of annotators \( c \).

\[
\hat{P}(k) = \frac{n_k}{ic} \tag{4.4}
\]

---

\(^{28}\)Content analysis (Berelson 1952) is a broad set of methodological approaches characterised by the analysis of human generated text through the selection and identification of specific character-istics formalised as coding schemes.

\(^{29}\)It should also be noted that the \( \kappa \) coefficient was also generalised to more than two annotators by Davies and Fleiss (1982), however, it has not proven as popular as Fleiss’ kappa.

\(^{30}\)This treatment of Fleiss’ kappa is based on Artstein and Poesio (2008:563)
\[ \hat{P}(k) = \frac{1}{\hat{ic}} n_k \] (4.4)

The probability that two arbitrarily selected annotators assign an item to a particular category is assumed to be the joint probability of each annotator performing this assignment independently, which is to say \((\hat{P}(k))^2\). \(A_e\) is then the sum of the joint probability over all categories \(k \in K\), as shown in Equation 4.5.

\[ A_e = \sum_{k \in K} (\hat{P}(k))^2 = \sum_{k \in K} \left( \frac{1}{\hat{ic}} n_k \right)^2 = \frac{1}{(\hat{ic})^2} \sum_{k \in K} n_k^2 \] (4.5)

### 4.7.2 Challenges Presented by this Annotation Task

Before Fleiss’ kappa—or any of the kappa-like coefficients for that matter—can be used for assessing the reliability of the annotations in the corpus developed in this chapter, there are a few hurdles that must first be overcome.

**Unitising Continuous Data**

As mentioned previously, this task differs from prototypical linguistic annotation tasks in that annotators are both segmenting the data as well as labelling the resultant segments. This is problematic for the kappa-like coefficients because agreement is calculated on an item-by-item basis (or pairwise item-by-item basis in the case of the generalised coefficients), requiring that items have been identified in advance of the annotation so that annotators share the same pool of candidate items. In this task, however, there are no pre-established items shared across annotators. Krippendorff (1995) refers to this as the problem of unitising continuous data, in the sense that there is an underlying continuous stream of data that must be segmented into discrete units, as opposed to implying that the underlying data being modelled is a continuous distribution.

One approach to solving this problem is to massage the annotations into a format that can be used with the kappa-like coefficients. This can be done by tokenizing\(^{31}\) the text being annotated and treating the tokens as segments which annotators have been implicitly tasked with labelling, effectively taking the place of the pre-segmentation normally present in traditional annotation tasks. This tokenisation could be performed with either character or word-based tokenisation, however word-based tokenisation would have the disadvantage of hiding systematic annotator bias associated with character-level annotation decisions, such as the inclusion or exclusion of punctuation inside spans. An example of how this tokenisation would be performed is presented in Figure 4.2, which shows how a single line of text annotated

\(^{31}\)Krippendorff (1995) refers to this process as *digitising*, in the context of annotating binary categories.
Chapter 4: The Phenomenal Corpus

Text: We have the money to buy.
Tokens: 0000000000000000001111111

Figure 4.2: Implicit character-based tokenisation being applied to a line of text from the WSJ corpus containing a relative clause.

Annotator A: 0000111111110000000
Annotator B: 0000111111110000100

(a) First annotation possibility

Annotator A: 0000111101110000100
Annotator B: 0000111111110000100

(b) Second annotation possibility

Figure 4.3: Two different ways a pair of annotators might have annotated some hypothetical text that both yield the same kappa-like scores using the implicit tokenisation approach.

with a relative clause would be tokenised. This example, and those following, use binary categories, with only the one target category for labelling; thus a 1 indicates inclusion of the corresponding character within an attested relative clause span, and a 0 indicates exclusion. Under this methodology, the identification of a character as not belonging to a span is effectively the same as labelling it with a category indicating the absence of a relative clause. The use of Fleiss’ kappa with implicit tokenisation forms the first candidate approach for measuring inter-annotator agreement.

A troubling consequence of this tokenisation process, identified by Krippendorff (1995), is that it treats agreement over tokens excluded from spans as importantly as agreement over included tokens. For instance, Figure 4.3 shows two possible ways a pair of annotators—A and B—might have annotated some text, after character-based tokenisation has been applied. Both scenarios result in the same agreement scores, as they have the same number of pairs with different combinations of 0 and 1; one \( \langle 0, 1 \rangle \) pair has just swapped positions with a \( \langle 1, 1 \rangle \) pair. However, it is clear that agreement should be lower for the pair of annotations in Figure 4.3b as even though there is agreement on the single character length span on the right-hand side, on the left-hand side, annotator A has identified two distinct spans of shorter lengths than the corresponding span identified by annotator B. This demonstrates that this approach is blind to the length of agreement within a span, provided that the global amount of agreement is maintained.

Another problem Krippendorff (1995) identifies with this approach is that the tokenised form of the annotations don’t explicitly encode span boundaries, just whether
or not an annotator has indicated that it belongs to a span. Consequently, the distinctions an annotator could make in identifying multiple adjacent spans would be lost, appearing in the tokenised form as a single annotation. Figure 4.4 illustrates this, with two annotators dividing up the text with markedly different segmentation decisions but yielding the same digitisation—and thus the same agreement score. Once again, this is caused by the fact that when using a kappa-like coefficient over tokenised text, the final agreement score is determined by the aggregate agreement over tokens, but is agnostic to the relative location and proximity of these tokens to each other. While it is perhaps unlikely that the annotation task described in this chapter will yield many situations where instances of the same phenomenon will be attested by annotators to lie adjacent to each other, this may not hold true if additional kinds of phenomena were to be added. More generally, at least, this lack of sensitivity towards measuring agreement over the boundaries of spans adds further weight to the case that tokenisation combined with a kappa-like coefficient is more an expedient workaround rather than an appropriate way of determining the reliability of annotations that involve simultaneous segmentation and labelling.

In response to these deficiencies, Krippendorff (1995) developed a new coefficient, $\alpha_U$, specifically geared towards the task of assessing reliability over data that has been both segmented—or unitised—and labelled in the one annotation procedure. This coefficient was adapted from the $\alpha$ coefficient (Krippendorff 1980, Krippendorff 2004) which is similar to kappa-like coefficients described previously (especially $\pi$, with its expected agreement being modelled through a combined distribution of all annotators’ behaviour); however, instead of modelling observed and expected agreement, it models observed and expected disagreement, as illustrated by Equation 4.6,\footnote{This treatment of $\alpha$ is taken from Krippendorff (2007). A notable difference in the formulation provided here is that in its most general form, $\alpha$ is capable of handling missing values from annotators for items, whereas I assume that all items will be assigned a value—given that not annotating a unit can be considered an implicit annotation of the item being a non-instance.} where $D_o$ is characterised as the average over all item-wise disagreements found between annotators and $D_e$ is characterised as the average disagreement between two annotators for an arbitrary item over all pairs of possible values that could have been assigned.

Figure 4.4: Example of a pair of annotations with different segmentation, but both yielding the same representation under implicit tokenisation.
\[ \alpha = 1 - \frac{D_o}{D_e} \]  

(4.6)

\(D_o\) is shown in Equation 4.7, where \(n\) represents the total number of values assigned by all annotators (the number of annotators by the number of units), \(j\) and \(k\) both take values from the set of categories \(K\), and \(m\) represents the number of annotators with \(o_{jk}\) yielding, the number of \((j, k)\) judgement pairs—the number of judgement pairs assigned the value \((j, k)\) across all annotators. The difference function \(\delta_{jk}\) calculates the difference to be assigned to a particular judgement and can be assigned different metrics depending on the type of data being annotated. For nominal data, such as in this context, the difference function returns a binary value depending on the annotators’ agreement, as shown in (4.8). \(D_e\) is then shown in Equation 4.9, with \(n_j\) and \(n_k\) representing the total number of item judgements with the values \(j\) and \(k\), respectively.

\[
D_o = \frac{1}{n} \sum_{j \in K} \sum_{k \in K} \frac{o_{jk}}{m-1} \delta_{jk} 
\]

(4.7)

\[
\delta(j, k) = \begin{cases} 
0 & \text{iff } j = k \\
1 & \text{iff } j \neq k 
\end{cases} 
\]

(4.8)

\[
D_e = \frac{1}{n(n-1)} \sum_{c \in K} \sum_{k \in K} n_c \cdot n_k \delta_{ck} 
\]

(4.9)

\(\alpha_U\) extends \(\alpha\) by first dividing up the text into sequences of positively identified units and gaps between, as identified by each annotator. In the task at hand, the lengths of these sequences would be measured in characters, for the same reason that character-based tokenisation was previously chosen over word-based tokenisation—as well as being consistent with Krippendorff’s recommendation to use the smallest discernible difference in the lengths of a record’s section for the measurement of sections (Krippendorff 1995:71). A consequence of this binary unit/gap distinction is that \(\alpha_U\) must be computed on a per category basis. \(D_o\) is now determined by systematically comparing all sections from each annotator to every other annotators’ sections. \(D_e\) is determined by pooling together all the identified gap/unit sections to give the set of possible unitisations and then comparing all pairings over these unitisations.\(^{33}\)

In order to compare each section, \(\alpha_U\) makes use of a difference function which defines the amount of disagreement between two candidate sections of text as the square of the non-overlapping unit portions, with completely overlapping sections yielding 0 and completely non-overlapping sections yielding the square of the length of the section.

\(^{33}\)I sketch only the outline of \(\alpha_U\); for the full details, see Krippendorff (1995).
Figure 4.5: Defining the components of difference used in the $\alpha_U$ difference function.

Figure 4.5, which is taken from Krippendorff (1995), shows a slice of the textual continuum being annotated, that has been divided up in two different but intersecting sections by annotators $A$ and $B$, with $r_{A_j}$ indicating the $j^{th}$ section of the text identified by annotator $A$ and $r_{B_k}$ indicating the $k^{th}$ section of the text identified by annotator $B$. Given only these two candidate sections for consideration, $s_-$ indicates the left-most region of the text which only one annotator identified as belonging to a unit, $s$ indicates the intersecting regions of the text identified by both annotators as belonging to a unit, and $s_+$ indicates the right-most region of the text which only one annotator identified as belonging to a unit. The difference function itself is presented in Equation 4.10, which is also taken from Krippendorff (1995) and defines the amount of disagreement for the $j^{th}$ section of annotator $h$ and the $k^{th}$ section of annotator $i$. The difference function makes use of a unitising function $v_{hj}$, which is presented in Equation 4.11. This function returns 0 if the $j^{th}$ section of the text continuum $r$ is identified as a gap by annotator $h$ and 1 if identified as a unit.

Looking at the three different conditions in the difference function defined by Equation 4.10, in the first condition, when the two sections are both identified as units and overlap partially, then the function evaluates to the sum of the squares of the lengths of the two non-overlapping ranges. In the second condition, when one of the sections is identified as a unit, the other is identified as a gap, and the unit is completely contained within the gap, then the function evaluates to the square of the length of the unit. If neither of these preceding conditions apply, then either the sections are gaps, completely overlapping units, or do not relate to each other, in which case the function evaluates to 0. Figure 4.6 illustrates the application of the difference function to a range of possible scenarios between two annotator’s segments.
Figure 4.6: Examples of the difference function in action.
Chapter 4: The Phenomenal Corpus

Figure 4.7: A sentence containing nested phenomenon annotations.

![Figure 4.7: A sentence containing nested phenomenon annotations.](image)

Figure 4.8: A sentence containing nested phenomenon annotations from the same category.

![Figure 4.8: A sentence containing nested phenomenon annotations from the same category.](image)

Not only does this approach explicitly model agreement over the placement of span boundaries, it also has an additional advantage, which is that the squared distance function incorporates a form of fuzzy disagreement. As argued by Krippendorff (1995), some amount of disagreement at the edges of spans is to be expected and should indeed be tolerated. \( \alpha_U \) takes this into account by weighting down peripheral disagreement compared to that of core disagreement. I took \( \alpha_U \) to be the second candidate approach for measuring inter-annotator agreement.

**Overlapping Spans**

One more challenge presented by this annotation task is that kappa-like coefficients cannot handle the labelling of items with multiple categories. However, it is clear that, in this task, segments of the text must be able to be annotated with multiple categories as the collected phenomena can, and often do, overlap. Figure 4.7 shows a scenario from the gold standard WSJ annotations, where a segment needs to be assigned the labels of *passive clause*, *complement clause*, and *relative clause*. This problem is alleviated by determining agreement on a category by category basis, generating agreement scores for each phenomenon—as is already required by \( \alpha_U \). This is also methodologically desirable, since this yields agreement over the individual categories, allowing us to determine if the annotation data is more or less reliable for different categories and establish whether some phenomena might be harder to identify or delineate. This still leaves, however, the problem of nested phenomena from the same category, as shown in Figure 4.8—also from the WSJ gold standard annotations—which contains nested complement causes. In cases such as these, some segments must be annotated with the same phenomenon twice.

I solved this problem by implementing an initial process of line copying, whereby
for each overlapping pair of same-phenomenon annotations from an annotator, the containing line was repeatedly duplicated and appended to the end of the text, along with one of the overlapping annotations. If other annotators also had span(s) from the same category on that line, for each duplication and concatenation an annotation was optionally selected and moved to the copied line such that it maximised the resultant agreement. Figure 4.9 illustrates the result of the line-copying process being applied to the nested annotations found in Figure 4.8.

This yields two proposals for assessing inter-annotator agreement in this task: Fleiss’ kappa over character tokens and $\alpha_U$. In both cases, these are performed on a per category basis and require an initial process of overlap resolution through line copying, to ensure there are no overlapping annotations.

4.7.3 Inter-annotator Agreement Results

This section discusses the inter-annotator agreement results, which are presented in Table 4.7 below. I have not yet discussed how kappa-like coefficients are to be interpreted, besides noting that a value of 1 indicates perfect agreement, a value of zero indicates annotators agreement to be at the same rate as expected by chance, and a value less than 0 indicates agreement worse than chance. The question is: how high should this value be for the data to be considered sufficiently reliable—for us to be confident that annotators are successfully capturing the phenomena that the annotation guidelines attempts to pick out?

The prevailing practice within the field of computational linguistics is that kappa scores above 0.8 indicate good reliability and values between 0.8 and 0.67 indicate that tentative conclusions may be drawn regarding the reliability of the data (Artstein

34The decision to append copied lines to the end of the document was purely for ease of implementation. These could have been inserted anywhere within the document—provided a matching copy was inserted at the same place in the other annotator’s document—as the resultant inter-annotator agreement scores would be the same.
Chapter 4: The Phenomenal Corpus

204

and Poesio 2008:576). This de facto standard stems from Carletta's (1996) introduction of chance-corrected inter-annotator agreement to the computational linguistic community, which, in turn, makes use of Krippendorff's (1980) recommendations for interpreting kappa scores in the context of the field of content analysis. Artstein and Poesio (2008:576) argue that this likely constitutes an over-generous interpretation of Krippendorff's original statement, with the content analysis community having since moved towards higher kappa-like values being required for satisfactory reliability. However, Artstein and Poesio (2008) also note that the reliability needs of linguistic annotation performed in computational linguistics do not necessarily coincide with the needs of content analysis and, more generally, depend on the nature of the annotation task in question.

Complicating this situation further, the use of the squared difference function in $\alpha_u$ means that its results cannot be directly compared with those of the non-weighted kappa-like coefficients. Furthermore, the fact that the difference function can be defined according to the needs of the annotation task at hand means that this difficulty of comparison extends to different annotation tasks using $\alpha_u$, but with different weighting functions (Artstein and Poesio 2008:576). Krippendorff (1995) does provide a basic comparison of $\alpha_u$ with kappa-like scores—noting that $\alpha_u$ follows the same behaviour of 0 indicating chance agreement and 1 indicating perfect agreement—however, no explicit comment is made regarding the interpretation $\alpha_u$ with respect to degrees of reliability, presumably—however not necessarily—implying that the same recommendations for kappa-like scores should be used, even though their actual distributions will necessarily follow different shapes. I will not delve further into the debate on how to interpret kappa-like scores or weighted kappa-like scores; for that the reader is referred to Artstein and Poesio (2008).

Summarising the approach for interpreting the inter-annotator agreement results, in the case of Fleiss' kappa, there is the aforementioned heuristic of $\kappa \geq 0.8$ indicating good reliability and $0.67 < \kappa < 0.8$ indicating some reliability. In the case of $\alpha_u$, for want of any other available guidelines, I will apply the same similar ballpark heuristic used for Fleiss' kappa. This, combined with the relative comparison of agreement scores across the different phenomenon categories, affords at least a rough assessment of reliability.

Turning, then, to the results themselves, Table 4.7 shows the inter-annotator agreement results for Fleiss' kappa (presented as $\kappa$) and $\alpha_U$. It can be seen from the table that the agreement scores for Fleiss' kappa across the two parts of the WSJ text all lie within the nominal range for indicating some reliability—however, only just, in the case of passives for the second part—and with passives and complements in the first part and complements in the second all suggesting good reliability. As discussed, however, Fleiss' kappa with implicit character-tokenisation offers a poor model of agreement over variable-length annotation spans. Looking at the values of $\alpha_u$, which was specifically designed to model agreement over annotations that involve both unitising and labelling, it can be seen that scores are substantially lower. That
they are lower at all than the corresponding values of Fleiss’ kappa is not necessarily indicative of a problem, since the two distributions will be different. However, in the case of passives and complements, the scores are substantially lower, so much so, that the reliability of the annotations are indeed brought into question. While investigating potential reasons for these low scores, I uncovered a number of properties of the $\alpha_u$ coefficient that could be implicated.

Firstly, the difference function, which is characteristic of $\alpha_u$, turns out to have undesirable effects upon the coefficient’s score. For two annotators $A$ and $B$ for a particular phenomenon category, this can be seen when looking at a situation where annotator $A$ has not identified an instance that $B$ has. In this situation, when comparing annotator $B$’s unit with annotator $A$’s overlapping gap, the difference function will yield the square of the length of annotator $B$’s unit. Were it the case that annotator $B$’s identified unit was longer, then the difference function would yield a greater difference. In other words, missing annotations are penalised proportionally to the square of their would-be length. For the task at hand, this is counter-intuitive. If an annotator missed a phenomenon instance that should have been identified, they are not “more incorrect” for longer instances than shorter instances.

In order to see how this might impact the agreement score for passive clauses, Figure 4.10 shows two passive clauses from the second part of the WSJ text that were both identified by annotator 3, but not by annotator 1. The effect of removing the annotation in Figure 4.10a—which has a length of 122 characters—from annotator 3’s pooled annotations was to increase the agreement between the two annotators from 0.268 to 0.359. Whereas the effect of removing the annotation in Figure 4.10a—which is only 39 characters long—was to increase the agreement to only 0.271. This is a striking difference in the impact upon global agreement, produced through the selective removal of only one annotation.

Another property of $\alpha_u$ is that missing spans of less frequent categories are penalised more than missing spans of more frequent categories. This actually applies to all chance-corrected inter-annotator agreement coefficients and, while somewhat counter-intuitive, is simply a consequence of the correction for chance. The moti-

<table>
<thead>
<tr>
<th></th>
<th>WSJ-08 1</th>
<th>WSJ-08 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotators</td>
<td>1 &amp; 2</td>
<td>1 &amp; 3</td>
</tr>
<tr>
<td>Coefficient</td>
<td>$\kappa$</td>
<td>$\alpha_u$</td>
</tr>
<tr>
<td>Passive</td>
<td>0.828</td>
<td>0.675</td>
</tr>
<tr>
<td>Relative</td>
<td>0.783</td>
<td>0.568</td>
</tr>
<tr>
<td>Complement</td>
<td>0.806</td>
<td>0.742</td>
</tr>
</tbody>
</table>

Table 4.7: Inter-annotator agreement results, including both Fleiss’ kappa and $\alpha_u$ scores for the two WSJ parts.
It also would create a new IRA that would shield from taxation the appreciation on investments made for a wide variety of purposes, including retirement, medical expenses, first-home purchases and tuition.

(a) A passive clause annotation with a length of 122 characters

It should be considered against the law," said Danny Leish, a spokesman for the association.

(b) A passive clause annotation with a length of 39 characters

Figure 4.10: Two passive clauses identified by annotator 3, with differing lengths, which, when selectively removed, results in an undesirably large variation in impact to $\alpha_u$.

vation for chance-correction was to account for the fact that naive accuracy scores would inflate agreement over skewed data, with agreement over more frequent categories being more likely to occur by chance. Chance-correction avoids this problem, effectively, by requiring annotators to demonstrate increased agreement on rarer categories—meaning that chance-corrected coefficients are generally more sensitive to agreement on less frequent categories (Artstein and Poesio 2008:576). It is certainly true that passive clauses and relative clauses—the two categories with troublesome agreement scores—are much less frequent than complement clauses, however this on its own cannot be used to explain the low scores, since this would effectively be arguing against the use of chance-correction. However, it is possible that this could compound the previously identified issue of missing units being penalised proportionally to the square of their length, which is indicative of an undesirable symmetry in agreement penalties between overlapping units and non-overlapping units. In these cases, not only does the agreement score suffer more for longer non-overlapping units, but for the less frequent categories of passives and relatives, these penalties will be amplified.

Finally, a further quirk of the difference function used in $\alpha_u$ is that, with respect to maximising the agreement score, there may be situations where it is better for one annotator to have not annotated a unit that overlaps with the unit of another annotator, when the amount of disagreement the unit would have introduced—due to a long non-overlapping portion—would be larger than the disagreement yielded by not having annotated the unit. An example of this situation is presented in Figure 4.11.

$^{35}$I verified this empirically by introducing disagreement through the selective artificial addition of an additional unit of length 100 characters for annotator 3. When a single passive clause was added, agreement decreased by 0.056, and when a single complement clause—a more frequently occurring category—of the same length was added, agreement decreased by only 0.013.
This was in addition to a more parochial $4.5$ million authorization for a health center in South Carolina upheld by a 273-123 vote in the House last night.

(a) Annotator 1

This was in addition to a more parochial $4.5$ million authorization for a health center in South Carolina upheld by a 273-123 vote in the House last night.

(b) Annotator 3

Figure 4.11: Annotations of a passive clause by two annotators, for which the removal of annotator 3’s annotation counter-intuitively improves the global agreement score for passives.

where both annotators have identified the relativised passive clause upheld by a 273-121 vote in the House last night, however, annotator 3 has incorrectly included the noun phrase being relativised within the passive span. In this situation, the amount of disagreement yielded by the difference function is sufficiently large that global agreement is actually improved by 0.009 through the removal of annotator 3’s unit. For annotators 1 and 3, this was the only identified instance of overlapping units across annotators where the removal of one would improve the agreement score. This phenomenon is not then a likely contributor to the low agreement scores seen in this annotation task, however, it does contribute further evidence towards the mounting argument for $\alpha_u$ not being the most appropriate means of assessing inter-annotator agreement for this annotation task.

While $\alpha_u$ initially appeared a strong candidate for testing the reliability of annotations in this task, empirical observation has suggested that its extreme sensitivity towards the lower rates of recall that appear to be inherent to this task—which may also be exacerbated by the length of syntactic phenomenon annotations—might make the coefficient’s value too volatile for helpful interpretation. However, the inter-annotator agreement results at least made clear that, relative to the agreement scores for complements, agreement for relatives and passives are certainly lower and error analysis from Section 4.6.5 does not readily illuminate the cause of this.

One possibility that could be pursued here might be to perform inter-annotator agreement

---

36 For the removal of an overlapping unit to improve agreement, it might seem like it would be sufficient for the non-overlapping portion of the two units being compared to be longer than the overlapping portion. While this situation is sufficient to yield less overall observed disagreement, the removal of the longer unit results in a different division of that annotator’s textual continuum into gaps and units, and thus the expected disagreement is also impacted by the removal. In other words, this is a necessary but not sufficient condition.
He succeeded Pat Brown, who was fired by the Alley board 18 months ago.

Figure 4.12: An example of the IOBE tagging scheme applied to alternative annotations for a passive clause by two annotators.

agreement separately for only annotators’ units that do intersect—that is, filtering out units which were annotated by only one annotator and not annotated by the other—and reporting these scores alongside the number of units that were missing from either annotator’s annotations. While this might result in an approach to assessing reliability that is arguably more appropriate to this task, it would not shed any further light on the nature of any systematic disagreements between annotators, or even more relevantly, systematic difficulties the annotators had in internalising the nature of the phenomena the annotation guidelines were trying to capture. The next section presents one last approach to analysing the collected annotations in an attempt to provide this information.

4.7.4 IOBE Tagging Analysis

In order to gain insight into the causes of the low inter-annotator agreement scores reported in this section, I performed one final round of data analysis. This involved a departure from the span-based approach of $\alpha_u$, with its problematic difference function, and a return to the character-based digitisation process of implicitly treating each character as a token to be annotated. Again, this was performed on a per-category basis, with the line-copying procedure employed to avoid nested annotations from the same category. This time, tokens were additionally tagged using the IOBE tagging scheme, adapted from the IOB tagging scheme, which is often used in the context of training and evaluating chunk parsing systems (Munoz et al. 1999). Under this approach, character tokens are tagged $I$ when occurring inside an annotator’s identified unit, $B$ if the character is the beginning character in the unit, $E$ if it is the end character, and $O$ when occurring outside an identified unit. Figure 4.12 shows a sentence with the resulting IOBE tags generated from two attested passive clause annotations by annotators 1 and 3. The intention of using this tagging scheme is that it will provide more granular feedback on the location of disagreements than has been achievable with the evaluation methods used so far.

This methodology was first used to determine inter-annotator agreement for the two annotator pairs using Fleiss’ kappa coefficient. These results are presented in Table 4.8. Table 4.8a shows lower agreement between annotator 1 and 2 on $B$ and $E$

---

37Many thanks to Bob Carpenter for the suggestion of using IOBE tags.
Chapter 4: The Phenomenal Corpus

<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0.862</td>
<td>0.817</td>
<td>0.907</td>
</tr>
<tr>
<td>B</td>
<td>0.652</td>
<td>0.797</td>
<td>0.788</td>
</tr>
<tr>
<td>E</td>
<td>0.682</td>
<td>0.718</td>
<td>0.828</td>
</tr>
<tr>
<td>O</td>
<td>0.982</td>
<td>0.979</td>
<td>0.965</td>
</tr>
</tbody>
</table>

(a) Annotator 1 and 2

<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0.697</td>
<td>0.796</td>
<td>0.887</td>
</tr>
<tr>
<td>B</td>
<td>0.308</td>
<td>0.810</td>
<td>0.781</td>
</tr>
<tr>
<td>E</td>
<td>0.688</td>
<td>0.759</td>
<td>0.780</td>
</tr>
<tr>
<td>O</td>
<td>0.977</td>
<td>0.976</td>
<td>0.968</td>
</tr>
</tbody>
</table>

(b) Annotator 1 and 3

Table 4.8: Fleiss’ kappa scores for inter-annotator agreement over the two parts of WSJ text.

tags for passives and also $E$ tags for relatives. Table 4.8b also shows low agreement between annotator 1 and 3 for all unit-internal tags for passives—particularly so for $B$-tags.

That the $B$ and $E$-tags for passives have such similar agreement between annotator 1 and 2 suggests that the underlying disagreements contributing towards the lower scores might be more likely to pertain to phenomenon instances that are included by one annotator but entirely absent from the other annotator’s annotations, as opposed to disagreement due to overlapping but divergent annotations, since entirely absent spans introduce equal amounts of disagreement over these two tags. Through the same reasoning, it could also be argued that the large discrepancy between $B$ and $E$-tags for passives between annotator 1 and 3 might involve a number of disagreements over the length of the spans. This kind of inference does not, however, provide any insight into the difficulties faced by individual annotators.

In order to identify difficulties individual annotators faced, I compared the performance of each annotator’s IOBE tags against the IOBE tags generated from the gold standard by computing the precision and recall for each tag on a binary basis, as if this were evaluation of a classification task.\(^{38}\) I previously asserted that determining inter-annotator agreement was an important step in assessing the reliability of the collected annotations given that the gold standard does not represent a ground truth for this task. However, having investigated inter-annotator agreement and having

\(^{38}\)For example, when determining scores for the $B$ class, for a character that had been assigned a $B$-tag by the gold standard, if an annotator assigned it a $B$-tag, this would be a true positive, whereas $O$, $I$, and $E$-tags would be false negatives.
identified that there were potential issues with respect to reliability, returning to the use of the gold standard data presents a means of triangulating on the nature of the ways in which annotators deviated from expected behaviour. 39

The results of the evaluation against the gold standard are presented for the two parts of the WSJ text, in Table 4.9 and Table 4.10, respectively. Looking firstly at Table 4.9, it can be seen that compared to the other phenomena, annotator 1 and 2 both had lower recall for both B and E passive tags, in addition to annotator 2 having lower recall for relative E tags. This suggests issues with boundary identification for these phenomena. Comparing this with the error analysis in Table 4.4, both annotators did indeed have some span mismatches for these categories, however they also had more numbers of missing instances for these categories. This reveals a limitation of assessing precision and recall over IOBE tagged tokens, which is that it is difficult to disentangle the effects of span mismatches (which effect both precision and recall) from missing instances (which effect recall) and spurious instances (which effect precision). One datum that does appear discernible, however, is that annotator 1’s passive mismatches might be more likely to be the result errors on the left boundary.

Looking at Table 4.10, annotator 1 has consistently lower recall for I, B, and E tags. The corresponding perfect precision for these tags indicate that the low recall

39Of course, this does not provide information regarding ways in which annotators deviated from expected behaviour that might not be found in the gold standard.


<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>I</td>
<td>0.999</td>
<td>0.729</td>
<td>0.999</td>
</tr>
<tr>
<td>B</td>
<td>1.000</td>
<td>0.707</td>
<td>0.983</td>
</tr>
<tr>
<td>E</td>
<td>1.000</td>
<td>0.744</td>
<td>0.947</td>
</tr>
<tr>
<td>O</td>
<td>0.977</td>
<td>1.000</td>
<td>0.990</td>
</tr>
</tbody>
</table>

(a) Annotator 1

<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Relative</th>
<th>Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>I</td>
<td>0.820</td>
<td>0.826</td>
<td>0.866</td>
</tr>
<tr>
<td>B</td>
<td>0.444</td>
<td>0.390</td>
<td>0.902</td>
</tr>
<tr>
<td>E</td>
<td>0.829</td>
<td>0.744</td>
<td>0.864</td>
</tr>
<tr>
<td>O</td>
<td>0.985</td>
<td>0.984</td>
<td>0.978</td>
</tr>
</tbody>
</table>

(b) Annotator 3

Table 4.10: Precision and recall obtained when comparing annotator 1 and annotator 2’s IOBE tags against the gold standard for the second part of the WSJ text.

is due to missing instances rather than span mismatches. This is verified by the error analysis presented in Table 4.5. It can also be seen that annotator 3 struggled with identification of passive beginning boundaries, with very low recall and precision. This is again consistent with the error analysis from Table 4.5, which revealed annotator 3 to have a large number of passive span mismatches. The IOBE analysis has revealed information beyond what was available in the error analysis, namely that the majority of these mismatches were located at the left boundaries. Manual inspection of these errors indicated that this was largely the result of annotator 3 incorrectly including the head noun of passivised relative clauses. An example of this can be found in Figure 4.11.

When compared to the original error analysis described in Section 4.6.5, the only additional information that the IOBE-based error analysis affords is the identification of which boundary is more likely to be the location of errors. However, this information is not always intuitive to derive due to the signals of precision and recall both being affected by the two other error classes of missing instances and spurious instances, in addition to span mismatches. Another strategy for isolating the specific boundaries involved in span mismatches is to extend the error analysis process to distinguish between these two subclasses of span mismatches, since this distinction is readily identifiable and would result in much more accessible feedback than through the IOBE approach. This suggests, then, that the utility of the IOBE tagging strategy
lies in enhancing the tokenised form of inter-annotator agreement.

4.8 Discussion and Summary

This section contains, firstly, a critical discussion of the methodologies used for providing quality assurance in this chapter, followed by a discussion of the overall methodology used for developing the Phenomenal Corpus.

4.8.1 Discussion of Quality Assurance

In assessing the quality of the annotated corpus created in this chapter, I employed a range of different approaches. The first of these was manual error analysis, which involved comparing each annotator’s output against the gold standard annotations and classifying the errors into different categories. The next was three different techniques for determining inter-annotator agreement: Fleiss’ kappa over digitised character-tokens, the $\alpha_u$ coefficient, which models agreement over continuous sequences that are both segmented and labelled, and Fleiss’ kappa over digitised character-tokens assigned IOBE tags corresponding to their location with respect to annotation units. Finally, the same IOBE tagging methodology was also used to compare annotator output with the gold standard using precision and recall.

Quality of the Annotations

Following the use of these assorted strategies for quality assurance, what can be said about the quality of the resource created? The manual error analysis made clear that annotator output contained a number of problems, most notably missing phenomenon instances and misidentified instance boundaries. Inter-annotator agreement measures also showed that the reliability of the annotations between annotators might be questionable, with there being decidedly less agreement over passive and relative clauses than for complement clauses. The exact interpretation of the inter-annotator agreement scores was unclear, however they at least showed that the annotators were not reliably making the same errors. Ultimately the manual error analysis served to provide the best picture of the quality of the annotations.

It is clear from these various results that the consistent identification and delimiting of these syntactic phenomena is a challenging task. While some annotators were observed to be making systematic errors, many of the errors were simply due to the difficulty of spotting these syntactic phenomena in the wild. The investigation suggests that, for this task, annotation by a single annotator may well be insufficient and that the reliable annotation of this kind of syntactic phenomena might require at least double annotation, with a round of disagreement resolution used to create the gold standard annotations.
The residual question, then, is how many phenomenon instances might remain unannotated within the corpus? The number of missing-instance errors and the lower than desirable rates of agreement might intuitively suggest this to be high, however the combined annotations of both annotators could have identified the bulk of them. Additional phenomenon instances may be uncovered through a further round of annotations, however, it is likely that these would represent diminishing returns.

**Recommendations for Quality Assurance**

The particular focus I have placed on assessing quality assurance is perhaps of value less so for the current annotations in the Phenomenal corpus, but more in the context of re-using this methodology to extend the Phenomenal Corpus to cover more phenomena or, alternatively, if others were to adopt the methodology to develop new annotated corpora. Here, I use the results of this investigation to inform a set of recommendations for performing quality assurance during the development of annotated corpora similar to the Phenomenal Corpus.

As already suggested, for the reliable annotation of the kind of syntactic phenomena found in this corpus, it may be necessary to perform double annotation to compensate for the difficulties involved in this task. It is also possible that, with sufficient practice and feedback, annotator reliability might improve to the point where more efficient single annotation is sufficient. In a typical annotation task, inter-annotator agreement is often used over a double-annotated sample of the corpus as a proxy for the reliability of single annotators being deployed in parallel over different sections of the corpus, in order to maximise efficiency while still being able to monitor quality.\(^{40}\)

Even where double annotation is performed and a gold standard can be produced, the use of inter-annotator agreement can still play a role, as it can be used to provide more immediate feedback, since a resolution phase does not need to be performed.

The use of error analysis against the gold standard annotations, which was presented in Section 4.6.5, represented one of the most informative components of the investigated quality assurance processes, as it provided a picture of which phenomena annotators experienced issues with, in addition to identifying the nature of these errors. This process was resource intensive, however, as in addition to requiring the construction of a disagreement-resolved gold standard, it also involved a time-consuming manual process of error classification. An improvement here would be to fully automate the detection and classification of errors, making error analysis only dependant on the existence of gold standard annotations. For this annotation task, a set of error categories that can be readily detected programmatically and which also proved to be informative, is the following: the number of missing instances, the

---

\(^{40}\)The fact that, in this annotation task, the corpus was doubly annotated—allowing a gold standard covering the entire corpus to be created—distinguishes it from typical annotation evaluation contexts. It also explains why the use of a gold-standard based evaluation was seen alongside the use of inter-annotator agreement approaches.
The annotation task presented in this chapter offered something of a challenge for assessing annotator reliability through inter-annotator agreement, as, in being tasked with identifying phenomenon instances, annotators were required to both delimit their range and also label the resultant unit with a phenomenon category. This differs from typical annotation exercises, where text is pre-segmented and annotators must only assign these units labels. I investigated the $\alpha_u$ coefficient, which is explicitly designed to support modelling agreement over annotations in which annotators both segment and label continuous data. In their survey of inter-annotator agreement approaches for computational linguistics, Artstein and Poesio (2008) state that the $\alpha_u$ coefficient may prove to be useful for annotation tasks which meet these characteristics, however, also note that this has not been tested in the computational linguistics literature. My investigation suggested that, at least for this task, $\alpha_u$ was not especially useful for assessing inter-annotator agreement. While it did provide a course-grained indication that there were issues with reliability on some phenomenon categories, the agreement scores it produced were difficult to interpret. Notably, these appeared to be overly-sensitive to essentially irrelevant properties of the disagreements. Furthermore, the difference function that scales disagreements proportionally to their length—which is likely the cause of the over-sensitivity—meant that its scores cannot be directly compared to those produced by other kappa-like measures, resulting in more difficulty for interpretation given the paucity of information regarding how to interpret $\alpha_u$ scores in the literature. Investigation of the apparent volatility of the $\alpha_u$ scores suggested that this might be due to the behaviour of the squared difference function in the face of disagreements over long spans of text, which were not uncommon in this annotation task. However, ultimately, a more principled evaluation of the appropriateness of $\alpha_u$ for use in this task would require an a priori sense of how consistent the annotations are for use as ground truth—information that is not available.

The use of an IOBE tagging approach—which tagged the implicitly tokenised annotation spans with labels indicating whether they were located inside, outside, at the beginning of a span, or at the end of a span—enabled further insight into the nature of annotator discrepancies. I first used this approach with Fleiss’ kappa to measure inter-annotator agreement over the IOBE tags, which served to overcome the issue of digitised annotations not explicitly encoding span boundaries. These inter-annotator agreement results provided some more insight into disagreements, revealing, for instance, that both pairs of annotators agreed less on the left boundary of passives. Since these results don’t indicate which—if any—annotator might be producing errors that are resulting in disagreements, I also applied the IOBE approach to the comparison of each annotators output to the gold standard, determining the precision and recall of their IOBE tags against the similarly generated tags from the gold standard, on a per phenomenon basis. These results proved somewhat informative, in that they indicated which span boundaries annotators were frequently annotating
incorrectly, however it was difficult to readily ascertain this information, as these precision and recall values were affected by three distinct errors categories—missing instances, spurious instances, and span mismatches—the effects of which were hard to disentangle. Furthermore, all information other than the distinction between beginning and end span mismatches can be obtained from the more accessible error analysis. This suggests, then, that the IOBE style error analyses could be omitted in favour of the more accessible error analysis from Section 4.6.5, modifying it by splitting the span mismatch category into the two categories of left-span mismatch, and right-span mismatch, which are readily detectable.

Based on the preceding observations, I present the outline of a suggested methodology for performing quality assurance on this type of annotation task. This methodology applies to the results of two or more annotators annotating the same text, which, as the results of my investigation potentially indicate, may be necessary for this type of annotation task. If resources do not permit this however, the methodology can be applied to a subset of the corpus, with the results fed back to annotators working individually over parallel subsets of the corpus.

The methodology involves two distinct quality assurance processes, which are performed independently, beginning with the first and then optionally proceeding to the second. Both processes apply to a section of text that has been annotated by two or more annotators, with annotators both identifying segments of text that correspond to phenomenon instances and also labelling them with phenomenon categories. They are as follows:

1. Annotator reliability is assessed by measuring inter-annotator agreement over the annotations using a kappa-like measure such as Fleiss’ kappa or $\alpha$.\(^{41}\) This is performed by implicitly tokenising annotated spans and tagging these with IOBE labels corresponding to whether each token is located inside, outside, at the beginning, or at the end of a span. Inter-annotator agreement is the measured for each IOBE label on a per category basis.

2. Error analysis is performed over the two sets of annotations against gold standard annotations. This requires that a gold standard be first created through a process of disagreement resolution, in consultation with the annotation guidelines. The error categories selected for this process should help diagnose issues in the quality of annotations and should also be automatically detectable, in order to maximise efficiency. For the annotations recorded during development of the Phenomenal Corpus, a set of error categories meeting these two criteria are: the number of missing instances, the number of spurious instances, the

---

\(^{41}\)This is the $\alpha$ coefficient that the unitising variant, $\alpha_u$, is based off.

\(^{42}\)In this investigation I used character-based tokenisation, however it is possible that word-based tokenisation could have been more appropriate. The choice of tokenisation strategy will, in general, vary according to the nature of the annotations.
number left-span boundary mismatches, and the number right-span boundary mismatches.

This constitutes a two-tiered approach to quality assurance. The first tier of inter-annotator agreement, which can be applied automatically to any portion of the corpus that has received more than one set of annotations, provides rapid diagnostic information at no cost. This means it can be deployed automatically as soon as multiple annotators complete a section, flagging any reliability issues that might indicate problems with the annotation guidelines. The second tier of error analysis requires a disagreement-resolved set of reference annotations before it can be deployed. It is therefore more resource-intensive, but provides more informative feedback, being able to reveal specific errors that are associated with an individual’s annotations. This is potentially of value in the context of large-scale annotation endeavours, where it is likely worth investing in firmly establishing the quality of annotations over a small subset of the corpus, earlier in the project, before scaling out to the rest of the corpus. These two approaches to quality assurance that were both investigated in this chapter—inter-annotator agreement and error analysis—can therefore be used in a complimentary fashion, providing different levels of feedback and requiring different amounts of resources.

4.8.2 Discussion of Methodology

In terms of the overall methodology chosen to create the syntactic phenomenon corpus to support the research described in this thesis, one undesirable outcome was the presence of only three syntactic phenomena in the final corpus. This situation arose from the infrequency of interrogatives and imperatives within the WSJ text, which was ultimately caused by the target phenomena being selected before the corpus data—which was a poor methodological decision. That the distribution of syntactic phenomena varies across different textual domains is well established and should have been accounted for. In this case, the decision to use DeepBank needed to have been made first, followed by the selection of a set of syntactic phenomena likely to be prevalent within newswire text. Furthermore, to ensure that this situation did not arise, it would also have been prudent to perform at least a cursory investigation, checking that the candidate phenomena do occur with some kind of frequency in the selected corpus.

For the phenomena that were included within the corpus, instance-frequency manifests as a further methodological issue. This can be phrased as a question—namely, how many instances of a phenomenon are required to be identified within a corpus for reasonable coverage over the phenomenon space to be achieved? Or put another way, how far down the Zipfian long tail of syntactic phenomenon variants is it necessary to go? A somewhat less than informative answer to this question is that it almost certainly depends upon the application. Were the goal of this research to produce a
syntactic phenomenon discovery system that could augment a descriptive grammar such as Huddleston and Pullum (2002) and achieve reasonable coverage over its considerable inventory of syntactic phenomena, then the corpus produced in this chapter would almost certainly be insufficient. The aim of this research, however, is to identify practical approaches to syntactic phenomenon discovery and determine their efficacy. The Phenomenal Corpus only then needs to achieve a reasonable coverage over the core phenomenon variants that occur with higher frequency.

It is certainly true that, compared to many corpora, the Phenomenal Corpus only lives up to one sense of its moniker, covering a relatively small amount of text. Development of the corpus started to hit up against resource limitations, with much of the available time being dedicated to the development of an appropriate data collection methodology and with the actual annotation task itself proving to be time-consuming for annotators. This raises the question of whether it was ultimately a sensible decision to build an annotated corpus from scratch. In Section 5.7, I describe the construction of a small collection of text annotated with relative constructions, which was formed by extracting exemplars from Huddleston and Pullum (2002). This was also used for the purposes of supporting the development of a syntactic phenomenon methodology, but focused on more fine-grained phenomenon categories. This resulted in an annotated collection with different characteristics than those identified in Section 4.2, however it involved less effort to obtain, and the resulting annotations are likely more reliable.

Finally, with the perspective of hindsight, it also became clear that the decision to use character spans for annotation was also somewhat unfortunate. This decision was motivated by the need to align annotations with ERG tokens, which employs a non-standard tokenisation scheme. The use of character-based annotations, however, introduced a number of challenges for the quality assurance process, which could have been avoided by performing token-level annotations and then aligning these to ERG tokens using a post-processing step. In particular, the use of character-level annotations meant that the number of places of potential disagreement between annotators was increased, but not over distinctions that matter for the phenomena being targeted. Furthermore, the use of character-level annotations likely exacerbated the difficulties associated with the use of the $\alpha_u$ coefficient, which were discussed in Section 4.7.3, with the increased length of spans yielding potentially larger disagreements from the disagreement function. Pre-segmenting the text so that annotators would annotate sequences of tokens instead, would have avoided these issues, while still leaving open the possibility of alignment with ERG tokens through an automated alignment process. This could potentially be taken even further, by pre-segmenting the text using constituent boundaries from the PTB—which, at this level, is relatively theory-neutral. This would remove entirely the problem of assessing inter-annotator agreement over annotations produced by a process of segmenting and labelling, as annotators simply have to have label existing constituent segments as being instances
of the target phenomena or not.\textsuperscript{43}

4.8.3 Summary

In this chapter I described the creation of The Phenomenal Corpus, a corpus of text consisting of 250 lines of text from the Sherlock Holmes novel \textit{The Adventure of the Speckled Band} and 477 lines of text from the WSJ section of the PTB, annotated with instances of passive clauses, relative clauses, and complement clauses. This was created in order to support the development of syntactic phenomenon discovery techniques which are described in Chapter 5. In addition to describing the methodology used to create the Phenomenal Corpus, I outlined the desiderata that the evaluation resource should have and identified a range of existing resources that were potential candidates. However, after assessing each of these candidates against the established criteria, none were deemed to be suitable, thus motivating the creation of a new resource.

Since the annotation task used to collect the data for the Phenomenal Corpus differed somewhat from annotation tasks often found within computational linguistics, a significant portion of the chapter was dedicated towards the methodology used for collecting the data, which included the creation of the annotation guidelines and, notably, the methodology used for performing quality assurance. This is reflective of the considerable amount of time that was spent investigating appropriate techniques for assessing quality assurance of the collected annotations—prompted by the paucity of material in the computational linguistics literature on approaches for assessing the reliability of annotations in which annotators both identify linguistic units and also delimit their textual extents. Consequently, while not initially anticipated, one of the contributions of this thesis is an applied investigation into techniques for assessing quality assurance for these kinds of annotations, resulting in concrete recommendations, which are presented in Section 4.8.1

The establishment of a concrete methodology for the construction of a corpus annotated with syntactic phenomena leaves the door open for future work to scale up the corpus—with respect to both the amount of text covered and also the number of phenomena included. One application that a large-scale corpus of this nature could also be used for is the application of fine-grained parser evaluation, which is the motivation for the resources produced in Rimell \textit{et al.} (2009) and Bender \textit{et al.} (2011b), and which I describe in Section 2.8.6.

For the purpose of the current research, however, I use the Phenomenal Corpus to support the development of the Typediff methodology for syntactic phenomenon discovery, that is developed in Chapter 5. In particular, I employ it in Section 5.6, where it is used as a source of exemplars of syntactic phenomena. These are fed

\textsuperscript{43}Thanks to my two anonymous examiners who questioned my choice of character-level annotations and from one of whom came the suggestion to use pre-segmented constituent boundaries from the PTB for annotations.
into an implementation of the Typediff methodology in order to assess its capacity to identify internal precision grammar components that are associated with target syntactic phenomena.
Chapter 5
Syntactic Phenomenon Discovery

5.1 Introduction

In this chapter, I describe the development of an interactive methodology that allows a user to navigate the alignment between a given characterisation of a syntactic phenomenon and associated internal constraints of a precision grammar. This alignment process is an ancillary step towards the goal of augmenting descriptive grammars with precision grammars and also has the potential to improve the accessibility of precision grammars, by improving the discoverability of analyses for those unfamiliar with the grammar, and by allowing grammar engineers to navigate their grammars through the lens of alternate characterisations of syntactic phenomena.

As discussed in Chapter 2, linguistically-grounded precision grammars are rich sources of accurate syntactic analyses, which, along with their capacity to support the creation of high quality treebanks, makes them strong candidates for use in augmenting descriptive grammars with phenomenon-centric capabilities such as retrieving additional examples of syntactic phenomena. This has been argued for by Bender et al. (2012) and its application has been demonstrated through the work of Bouma et al. (2015), which I discuss in Section 2.7.3 and Section 2.9.2. A limitation of traditional treebank search tools—such as those described in Section 2.9.2—is that before they can be used to retrieve positive instances of a phenomenon from a treebank, the representation of the phenomenon within the treebank must first be known. Bouma et al. (2015) address this by using GrETEL (Augustinus et al. 2017), a query-by-example treebank search tool that was described in Section 2.9.3. Even with the use of this tool, however, they still report the alignment process that this task requires to be challenging for annotators.

Discovering how a particular target phenomenon is represented within a treebank produced by a precision grammar can be a challenging process due to a number of factors that I identified within Chapter 2. One contributing factor is that, while precision grammars encode many linguistic abstractions corresponding to syntactic phenomena, these are not always represented explicitly. Practical requirements on
how grammar components are organised, combined with the frequent interaction that is seen between grammatical phenomena means that constraints are often spread non-contiguously across the surface area of the grammar. Even when phenomena are represented explicitly within a grammar, these abstractions correspond to a particular analysis of the phenomenon that was selected by the precision grammar authors, which may differ from the provided analysis of the target phenomenon, either in a way that results in different distinctions within the phenomenon space, or in a manner that results in a fundamental incompatibility of analysis.

In order to better support the use of precision grammars for augmenting descriptive resources, there is a need for a practical methodology that enables users to navigate the alignment between a given syntactic phenomenon characterisation and associated internal constraints of a precision grammar. In addition to enabling precision grammars to be used for phenomenon retrieval in the context of electronic descriptive grammar enhancement and the exploration of linguistics research questions, such a methodology also has the potential to make the analyses found within precision grammars more accessible. Improving the discoverability of precision grammars themselves could present increased value for those unfamiliar with the grammar as well as for grammar engineers developing the grammar, allowing them to understand the grammar through the lens of a particular phenomenon characterisation.

In this chapter, I present a method for performing phenomenon discovery that supports the alignment of syntactic phenomenon analyses within a precision grammar and corresponding syntactic phenomenon characterisations, such as that found within a descriptive grammar. In line with the scope of this research, which was put forth in Section 1.1.1, this methodology focuses on the ancillary discovery and alignment process, rather than on the development of phenomenon detection rules that would be involved in the type of descriptive grammar augmentation I have argued for. In Section 6.3.3, I discuss how the discovery techniques put forth in this chapter could be evaluated in the context of this application in future work. The syntactic phenomenon discovery techniques described in this chapter are developed using precision grammars produced within the DELPH-IN consortium, but, as I discuss in Section 5.8.4, should be readily applicable to other precision grammars based on Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag 1994), and potentially to grammars based on other grammatical frameworks.

The rest of this chapter is organised as follows. In Section 5.2, I describe the overall strategy that I adopted for this methodology and then in Section 5.3, I outline the aspects of DELPH-IN grammar architecture that are pertinent to the details of the developed methodology. In Section 5.4, I describe the various components of the methodology itself, including a feature extraction process applied to analyses produced by DELPH-IN grammars, as well as the development of a user-interface that facilitates the identification of relevant grammar components. In Section 5.5, I describe the implementation of the Typediff methodology as a browser-based tool. In Section 5.6, I explore the application of the Typediff methodology to the Phenomenal
Chapter 5: Syntactic Phenomenon Discovery

Corpus from Chapter 4 and then in Section 5.7, I apply the Typediff methodology to a different collection of syntactic phenomena, this one containing relative constructions extracted from Huddleston and Pullum (2002). Finally, in Section 5.8, I discuss the strengths and limitations of this methodology, including its potential to extend to other precision grammars.

5.2 Proposed Approach

In this section, I draw from the background material presented in Chapter 2 in order to inform the general characteristics of the desired syntactic phenomenon discovery methodology. After identifying and motivating a range of properties it should have, I then put forth a concrete template of the approach that I adopt for the Typediff methodology.

One way of responding to the challenge of improving the discoverability of syntactic phenomena within precision grammars might be to redesign precision grammars so that they are organised around grammatical phenomena, as proposed by Oliva et al. (1999). This style of grammar engineering is already enabled through the CLIMB methodology for grammar engineering (Fokkens 2011), which was discussed in Section 2.6.6. Using CLIMB, grammars can be organised by grammatical phenomena at the metagrammar level, but then compiled to a concrete grammar with an arrangement that is required by the grammar processing tools. The ability to organise the grammar by phenomena could be used to try to make its syntactic phenomena more discoverable. However, besides the issue of needing to persuade grammar engineers to adopt a new way of constructing grammars, this does not solve the problem of needing to support arbitrary characterisations of syntactic phenomena, since a single arrangement of phenomena still needs to be selected at the metagrammar level. CLIMB can be used to maintain alternative parallel analyses, but attempting to use this feature in order to align with arbitrary descriptive linguistic resources would likely be impractical. The proposed approach to tackling the discoverability problem should therefore work with existing grammar engineering architectures. In this research I explicitly target HPSG-based precision grammars produced within the DELPH-IN research consortium. I discuss the applicability of the methodology described in this chapter with respect to other types of precision grammars in Section 5.8.

The need to navigate the alignment between a given syntactic phenomenon characterisation and its associated precision grammar constraints suggests an interactive process of discovery guided by a user, as opposed to an automatic process. This need for guided navigation arises from the complexity that is inherent to the grammatical phenomena of language systems, which I discussed in Section 2.3.5. This complexity is evidenced by the continual discussion and debate within the descriptive linguistics literature regarding appropriate ways to model various grammatical phenomena and indeed whole language systems. The task at hand essentially involves mapping
components of one such model of a language system to the relevant components of another model of the same language, a process that compounds the complexity already inherent within a single model. The need for an interactive methodology is also supported by the observation that, when using treebank querying for locating instances of syntactic phenomena, a vetting stage is typically required due to the high numbers of false positives within the candidate results (as seen in both Rimell et al. (2009) and Bender et al. (2011b), which were described in Section 2.8.6).

As I discussed in Section 2.9.2, one approach that has been frequently used to perform syntactic phenomenon discovery against a treebank is to employ signature-based treebank search tools,1 which involve the user supplying a query to be executed against a target treebank. While these tools do allow a user to interactively explore the quality of query results, as discussed in Section 2.9.2, they suffer from two notable limitations: the need to be familiar with a potentially arcane query language and the need to already know how the phenomenon is represented within the treebank. The former is primarily a limitation that pertains to accessibility, while the latter renders these tools unsuitable, since the representation of a given phenomenon within treebanks produced by a precision grammar is precisely what needs to be discovered. The query-by-example methodology for phenomenon discovery, which I discussed in Section 2.9.3, addresses both these issues by using sentences as input. Users then only need to provide known instances of the phenomenon in order to begin the discovery process. This improves accessibility by removing the need to be familiar with a query language, and, crucially, by also bootstrapping the discovery process without the need for an existing understanding of the representation of the phenomenon. The latter of these benefits is why the GrETEL tool was used by Bouma et al. (2015) in the process of deriving treebank queries that corresponded to target syntactic phenomena.

The query-by-example approach is also desirable from another perspective. Under this approach, the characterisation of the target syntactic phenomenon is effectively provided implicitly by the user through the choice of the exemplifying sentences selected as input. This means that the methodology will support a dynamic characterisation of the target phenomenon, which can respond to the needs of the current context—simply by changing the input sentences. In the context of aligning a particular description of a phenomenon in a descriptive grammar, the exemplars accompanying the description can then be used as both the query input and also the characterisation of the phenomenon. Likewise, in the context of a researcher attempting to locate additional instances of a syntactic phenomenon, the existing examples they already have form the characterisation of the phenomenon they are attempting to isolate.

Together, the properties identified above all point towards an interactive methodology for phenomenon discovery that incorporates query-by-example and invokes a grammar and parser to process user-provided sentences that exemplify the target

---

1See also Section 2.5.2 for a brief overview of treebank search tools.
phenomenon. This is the general methodology that I adopted and can be described more concretely as follows:

1. The user selects one or more input sentences or fragments exemplifying the target syntactic phenomenon.
2. These sentences are parsed using the target grammar.
3. Grammar components are extracted from the parser output and presented to the user.
4. With the aid of the tool, the user selects the components deemed to be pertinent to the target phenomenon.

Before going into the details of this methodology, in the next section I briefly outline some aspects of delph-in grammar architecture in order to provide some background for decisions made regarding which grammar components were extracted from parser output and why they are useful in the context of syntactic phenomenon discovery.

5.3 DELPH-IN Grammar Architecture

With their theoretical underpinnings grounded within HPSG, delph-in grammars encode grammatical constraints through typed feature structures and rely on a type hierarchy for capturing grammatical abstractions. It is because of this central role that types play in capturing grammatical abstractions that I selected them as the grammar components which are extracted from parser output as a part of the query-by-example methodology that is described in Section 5.4. In order to provide more context for this decision and to also motivate the ways in which types are used in latter portions of the methodology, this section briefly unpacks the formal machinery of typed feature structures and how DELPH-IN grammars use them to encode grammatical constraints and capture grammatical abstractions. For an in-depth, textbook treatment of the grammatical theory that is in many ways similar to that assumed in DELPH-IN grammars, see Sag et al. (2003), and for an introduction to the topic of implementing grammars using typed feature structures, see Copestake (2002b).

Typed feature structures are formal devices that are well-suited to representing grammatical information. Informally, a typed feature structure consists of a set of features, each paired with a value. Values can either be atomic or complex, consisting of structures such as a list of values or another feature structure. Additionally, every feature structure has a type, which applies constraints over the possible features the feature structure can take, as well as constraints over their values. This is illustrated in Figure 5.1, where the entire feature structure is of type \texttt{type}_0, and has two features, \texttt{FEATURE}_1 and \texttt{FEATURE}_2. \texttt{FEATURE}_1 has a value of type \texttt{type}_1, and \texttt{FEATURE}_2 has a
Figure 5.1: A typed feature structure represented as an attribute value matrix.

<table>
<thead>
<tr>
<th>type_0</th>
<th>FEATURE_1_{type_1}</th>
<th>FEATURE_2_{type_2}</th>
<th>FEATURE_3_{type_3}</th>
<th>FEATURE_4_{type_4}</th>
</tr>
</thead>
</table>

value which is a feature structure of type $\text{type}_2$. Within this nested feature structure, $\text{FEATURE}_4$ takes as its value a list, which has two elements, both which are of type $\text{type}_4$.

A typed feature structure can also be represented as a directed acyclic graph, with feature names corresponding to arcs, and feature values corresponding to nodes, as can be seen in Figure 5.2, which shows the graph that corresponds to the feature structure in Figure 5.1.\footnote{This also reveals that a list can be represented recursively as a feature structure, with \text{head} and \text{rest} features containing values that correspond to an item of the list and the tail of the list respectively.} As this level of formality is not necessary here, I will continue to use the more accessible attribute-value matrix (AVM) representation of feature structures that is used in Figure 5.1.

By using features that correspond to grammatical properties and by designing an appropriate feature geometry, typed feature structures can be used to encode grammatical constraints. This is illustrated by the AVM in Figure 5.3, which captures the constraints applicable to transitive verbs. The feature geometry within in this AVM is a simplified version of that found within the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011)—amongst other omissions, it notably lacks semantic information. In spite of this, this typed feature structure still evokes some of the core principles of HPSG that the ERG and other delphin grammars are based upon. By collecting together constraints over feature structures, types provide a mechanism by which such grammatical abstractions can be captured. This can be seen in Figure 5.3,\footnote{This AVM represents a truncated analysis produced by the 1214 release of ERG. Except where noted, all following ERG examples in this chapter are also derived from this release of the ERG.} with the type $\text{cat}$ capturing the syntactic category of a constituent by constraining the values of $\text{head}$ and $\text{val}$, which correspond, respectively, to the part of speech of the constituent and the valence properties of the constituent. The valence properties are defined by a feature structure of type $\text{valence}$, which has the two features $\text{subj}$ and $\text{comps}$, which—for a transitive verb—both take values of a list with a single element, indicating that the word requires a subject and a single complement. Both of these constituents are also constrained such that they are of type $\text{synsem}$, which collects together syntactic and semantic properties, and
Figure 5.2: The typed feature structure from Figure 5.1 represented as a directed acyclic graph.
by also further constraining the syntactic category. In the case of the subject, this is specified to be of type \texttt{nomp\_cat}, which corresponds to a nominal phrase, and in the case of the complement, it is \texttt{nomp\_cat\_acc}, which is an accusative nominal phrase.

The entire feature structure in Figure 5.3 has the type \texttt{v\_np\_le}. This is an example of a lexical type, which, as discussed previously in Section 3.2, can be thought of as a fine-grained part of speech, that captures selectional and other properties of the lexical entries it is assigned to within the lexicon. Figure 5.4 shows the definition of the lexical type \texttt{v\_np\_le} from the ERG, which is represented using Type Description Language (TDL; Copestake 2002b), the formalism DELPH-IN grammars use to encode grammar constraints as typed feature structures.

Characteristic of HPSG is the organisation of the grammar’s types into a hierarchy. This type hierarchy forms the backbone of the grammar, facilitating grammatical abstractions central to the analysis of the language, as well as supporting code reuse in the context of implemented precision grammars. Abstractions are facilitated through type inheritance, where, by inheriting from one or more supertypes, a subtype inherits the constraints provided by all of its supertypes. An abbreviated fragment of the ERG’s type hierarchy is presented in Figure 5.5. This shows some of the
types that inherit from \texttt{verb.synsem}, a supertype that captures the syntactic and semantic properties of different subcategorisations of verbs, meaning that it itself is a descendant of the type \texttt{synsem}. In Figure 5.5, one such subcategorisation is \texttt{np.trans.verb}, which corresponds to transitive verbs that take a noun phrase as a complement. This \texttt{synsem} type can be seen in Figure 5.4, where it is constraining the value of the \texttt{SYNSEM} feature, thereby applying the syntactic and semantic properties of a transitive verb to the lexical entry \texttt{v.np.le}, which can then be used in the lexicon to add lexical entries corresponding to this verbal subcategorisation. The complete fragment of the ERG’s type hierarchy rooted at \texttt{verb.synsem} contains 537 types. That all these types are predominately used within the definition of lexical types, such as \texttt{v.np.le}, is indicative of the highly lexicalised nature of HPSG, with the lexicon itself being richly structured.

The remaining aspect of DELPH-IN grammars that is relevant to the methodology I describe in Section 5.4 is how an analysis of a sentence is formed. The fundamental building blocks that an HPSG analysis is composed of are \texttt{sign} types, which can be divided into the two core subtypes of \texttt{word} and \texttt{phrase}. \texttt{sign} types represent pairings of surface text with grammatical constraints and semantic contribution. A complete analysis involves the successive combination of \texttt{sign} feature structures, from words through to phrases, until a \texttt{sign} feature structure can be constructed.
that spans the entire sentence.\footnote{The combination of two feature structures to yield a more specific feature structure which is subsumed by the two feature structures is performed through the unification operation, however the formal details of this operation are unnecessary for presenting this research. For an overview of grammar formalisms and frameworks that are based on unification, see Shieber (1986).} This process involves the application of lexical and phrasal rules, which operate on words and phrases respectively. Figure 5.6 shows a derivation tree produced by the ERG for the sentence \textit{The goat likes Kim}, with each node representing the application of a rule. Since the application of a rule results in a new feature structure of type \textit{sign}, each node can also be represented as its corresponding feature structure, which makes explicit the pairings of surface text with the \textit{sign} types found in the complete analysis. This can be seen in Figure 5.7, which presents the derivation tree from Figure 5.6 with the root node and the two nodes at the subsequent level expanded into AVMs containing simplified feature structures. The feature structures at these nodes have also been abbreviated, with for instance, the feature structure of type \textit{phr$_{synsem}$} in the root node being collapsed for brevity.

A complete analysis requires the construction of a valid feature structure of type \textit{sign} that spans the complete sentence.\footnote{While this presentation has not included reference to the semantic analysis that \textsc{delph-in} grammars produce, it is worth noting that both syntactic and semantic structures are constructed simultaneously in a monostratal process. An AVM that represented a complete analysis produced by the ERG would, amongst other information, include a representation of the semantic analysis of the sentence.}

In the following section, I outline the methodology that I developed that helps users align a given characterisation of a syntactic phenomenon with the relevant internal constraints of a precision grammar. As already established in Section 5.2, this

---

**Figure 5.6: Derivation tree produced by the ERG for the sentence \textit{The goat likes Kim}.**
Figure 5.7: The derivation tree produced by the ERG for the sentence *The goat likes Kim*, with the first two node levels presented as simplified feature structures.
methodology involves parsing input sentences with the target grammar and selectively extracting grammar components for presentation to the user. From the perspective of this component of the methodology, there are two salient aspects of DELPH-IN grammars to be taken from the above brief presentation of their architecture. The first is that the primary mechanism by which grammar abstractions are arrived at is by means of types and the type hierarchy. The second is that a successful analysis of a sentence involves the construction of a phrase-structure backbone, in which each node in the derivation tree is a typed feature structure of type \textit{sign} and every type originates from the grammar’s type hierarchy.

5.4 Typediff Methodology

This section describes the Typediff methodology for performing syntactic phenomenon discovery with precision grammars. As outlined in Section 5.2, this methodology adopts a query-by-example approach, in which users input sentences exemplifying target phenomena, which the system parses and extracts grammar components from the resulting output. The system then presents these grammar components to the user along with a number of affordances, which enable the user to identify components that correspond to the target syntactic phenomenon.

This approach can be broken down into the development of two distinct methodological components. The first of these involved the development of techniques for extracting, labelling, and weighting grammar components from parser output, which can be thought of as a feature extraction process. The second part of the methodology involved the development of an interactive interface that takes these extracted features as input and exposes them to users along with affordances developed specifically to assist users rapidly locate grammar components that are associated with the target phenomenon. An implementation of both of these components is discussed in Section 5.5.

In walking through the methodology, I use the English right node raising construction as the target phenomenon to motivate and illustrate the different steps. This construction involves two sequences—typically, but not exclusively, coordinated—that are parallel in structure and which do not form a complete constituent, with a constituent immediately to the right of the parallel sequences that functions as a shared argument to both sequences (Huddleston and Pullum 2002:1341). This is illustrated by (52), where the underlined sequences both share the argument \textit{employees}. An analogous sentence which does not feature the construction is shown in (53),

\footnote{The name \textit{right node raising} is attributed to Postal (1990:125) and is evocative of the transformation-based analysis put forth therein. Huddleston and Pullum refer to this construction as \textit{right nonce-constituent coordination} (Huddleston and Pullum 2002:1341). I use the former term for consistency with the ERG’s terminology. This terminological mismatch further underscores the difficulty involved in performing such alignments.}
with the argument manifesting locally in both sequences, resulting in two complete constituents.

(52) The manager remembered having to hire and being forced to fire employees.

(53) The manager remembered having to hire employees and being forced to fire employees.

The remainder of this section presents details of the feature extraction process, operations that can be performed over these features for locating those associated with a phenomenon, followed by a description of the interface that users interact with, explaining how these components contribute towards a tool that is able to assist users in more readily navigating the alignment of a syntactic phenomenon with its implementation within a precision grammar.

5.4.1 Type Extraction

Once an input sentence is selected, the first step in the feature extraction process is to parse the sentence with the target grammar. In developing this methodology, I used the ACE (Packard 2011) DELPH-IN grammar processor. As is common for precision grammars, DELPH-IN grammars typically yield multiple analyses for a given sentence. The Typediff methodology defaults to selecting the best analysis according to the parse selection model distributed with the grammar, with the user having the opportunity to select an alternative analysis in the event that the automatically selected one does not contain the desired analysis.

The grammar components I chose to extract from the parser output are the grammar’s types. This decision was based on the role that types play within DELPH-IN grammars in capturing grammatical abstractions, which was discussed in Section 5.3. The output that ACE and other DELPH-IN processors produces for a successful analysis of an input is a derivation tree and an MRS structure containing the semantic analysis. The derivation tree corresponds to the phrase structure backbone of sign types, which was outlined in the previous section, making it superficially similar to a traditional syntactic phrase structure tree, but is in fact much richer, containing all the information required to regenerate the sign type feature structures that are associated with each node. In order to extract the types found within a complete analysis of a sentence, I developed a program, which, using ACE libraries, takes as input a node of a DELPH-IN derivation tree and then for that node and every node beneath it in the derivation tree re-creates the corresponding sign-typed feature structure, before recursively extracting the identifier of every value’s type within the feature

\footnote{A statistical model trained over a treebank of disambiguated analyses produced by the grammar, which was discussed in Section 2.6.2.}
structure. When fed as input the root node of an analysis, this program would then extract every type involved in producing the analysis.

The algorithm employed by the type extraction program also needed to account for a requirement of delph-in processors, which is that, for any compatible pair of types in the grammar’s type hierarchy, they must share a single descendant that is higher than any other shared descendants in the hierarchy—referred to as a unique greatest lower bound (GLB) type (Copestake 2002b:39). This required property of the type hierarchy is achieved by a preprocessing step that the delph-in processors apply, which inserts GLB types as necessary. The feature structures of the final analysis therefore include these GLB types, meaning they would be extracted by the aforementioned process. This would be an undesirable outcome, as GLB types are not assigned identifiers indicating the types they inherit from, which would result in the masking of a subset of type identifiers from the output. I overcame this by employing a resolution process during type extraction, in which GLB types are replaced by the union of their closest non-GLB ancestors.

A residual consideration of the type extraction process arises from the fact that the same type can occur multiple times across the feature structures of an analysis. From the perspective of locating types associated with a syntactic phenomenon, incorporating type frequencies for each analysis would likely suffer from some confounding factors. Firstly, many types within a grammar correspond to frequently used abstractions that can be shared across diverse phenomena. For example, the sign type hcomp_rule is responsible for combining heads with their complements, meaning that it will be implicated by a great number of syntactic phenomena. Additionally, the length of a sentence affects the frequencies of types found within its analyses. Longer sentences increase the occurrences of types which may not be related to the target phenomenon, thereby increasing noise. To reduce these confounding affects, for each analysis the extraction process extracts the set of type identifiers from across all feature structures, ignoring the number of occurrences.

The large size and high degree of complexity associated with the feature structures of a complete analysis means that this type extraction process results in a large number of types for an input sentence. For example, when the right node raising exemplar (52) is used as input to the extraction process and the appropriate analysis is selected, 356 types are extracted. This is clearly too large a number of undifferentiated types to be of any practical value for performing discovery. In order to address this problem, I augmented the type extraction process with two further steps: a type-labelling component and a type-weighting component.

---

8Many thanks to Woodley Packard, the author of the ACE, who helped me develop this code.

9While the number of occurrences of each type within a single analysis is discarded, the number of analyses containing each type across a set of input exemplars does become relevant for the type weighting function, which is presented in Section 5.4.3.
5.4.2 Type Labelling

The aim of the type labelling is twofold: to reduce the number of candidate types by removing a subset that are unlikely to be relevant, and to organise the remaining types by the kind of abstraction they capture, allowing users to target these categories during the discovery process. Both of these goals were achieved by labelling types with a single label from a set of course-grained categories corresponding to high-level abstractions within the grammar.

Within the type hierarchy employed by Delph-in grammars, there are a large number of types that define abstractions which provide constructs for grammar engineers to work with. Rather than being linguistic abstractions, these are more engineering abstractions, including foundational components that are not provided by the underlying formalism as well as other kinds of “plumbing”. For example, the TDL formalism does not provide a list primitive, so Delph-in grammars enable this though the definition of recursive types with FIRST and REST features, as illustrated in Figure 5.2. As these types capture engineering rather than linguistic abstractions, it is unlikely that they would correspond to any syntactic phenomenon in particular and are therefore good candidates for being filtered out.

I identified six types in the hierarchy of the ERG—present across most Delph-in grammars—that capture high-level grammatical abstractions, and which play an important role for grammar engineers when implementing the analysis of phenomena. These types are sign, synsem, head, cat, relation, and predsort. They correspond to distinct linguistic constructs, defining mutually incompatible feature structures, meaning that for any given type in the grammar, it will inherit from only one of these types. Table 5.1 provides an overview of these types, outlining their function, the number of types within the ERG that inherit from each, and the corresponding coverage of the ERG’s 9085 types this represents.

The type labelling was performed by assigning types that inherited from any of the identified six types with the corresponding type name, and those which did not were labelled as other, which is also included in Table 5.1. This labelling schema meets the intended goals, by providing a means by which types that are unrelated to grammatical abstractions—those from the category other—can be readily filtered out, in addition to allowing users to selectively focus on candidate types that correspond to core grammatical abstractions. Returning to (52), filtering out other types from the 356 extracted types brings the number of candidate types down to 146, a somewhat more approachable number. Table 5.1 also shows the number of types from each of the categories that are extracted from (52), which provides a sense of the reduced number that a user would be presented with when filtering by these labels.
### Table 5.1

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>ERG</th>
<th></th>
<th>Exemplar (52)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Types</td>
<td>%</td>
<td>Types</td>
<td>%</td>
</tr>
<tr>
<td>sign</td>
<td>Capture the syntactic constraints and semantic contributions associated with different classes of words and phrases</td>
<td>3134</td>
<td>34.5</td>
<td>29</td>
<td>8.2</td>
</tr>
<tr>
<td>synsem</td>
<td>Abstractions across both syntactic and semantic constraints</td>
<td>1618</td>
<td>17.8</td>
<td>20</td>
<td>5.6</td>
</tr>
<tr>
<td>head</td>
<td>Part of speech categories used by the ERG, a proportion of which are disjunctive, such as <code>n_or_adv</code> and <code>nominal_or_verbal</code></td>
<td>494</td>
<td>5.4</td>
<td>40</td>
<td>11.2</td>
</tr>
<tr>
<td>cat</td>
<td>Broad subcategorisations, captured primarily by constraining HEAD and VAL features</td>
<td>36</td>
<td>0.4</td>
<td>9</td>
<td>2.5</td>
</tr>
<tr>
<td>relation</td>
<td>Core classes of semantic relations that are used to build the semantic analyses</td>
<td>77</td>
<td>0.8</td>
<td>13</td>
<td>3.7</td>
</tr>
<tr>
<td>predsort</td>
<td>Abstractions across the kinds of predicates that semantic relations introduce</td>
<td>2131</td>
<td>23.5</td>
<td>35</td>
<td>9.8</td>
</tr>
<tr>
<td>other</td>
<td>All other types that do not inherit from the above types</td>
<td>1595</td>
<td>17.6</td>
<td>210</td>
<td>59.0</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>9085</td>
<td>100.0</td>
<td>356</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 5.1: For each abstract type from the ERG that was used as a label, this table shows: the name of the corresponding type, a broad description of the type’s role in the ERG, its descendant types by number and percentage of the total types in the ERG’s type hierarchy, and the descendant types found in (52) by number and percentage of the total types found within its analysis. In the case of the label *other*, which is not an ERG type, the types are those that do not inherit from any of the other types.
5.4.3 Type Weighting

The second step performed to improve the discoverability of types associated with the target syntactic phenomenon was to apply a weighting function to each candidate type. This weight can then be leveraged by the interface in order to draw the user’s attention to types more likely to be relevant.

Developing a supervised model that could predict the probability of a type being associated with a syntactic phenomenon would require precisely the type of phenomenon-labelled training data the nonexistence of which motivates this work. Furthermore, even were training data to be created through an annotation process such as that presented in Chapter 4, it would correspond to a single characterisation of each phenomenon annotated—namely that found in the annotation guidelines. As already identified, however, a means of discovering dynamically characterised syntactic phenomena is required. I therefore opted against the development of a supervised approach.

Developing an unsupervised approach to weighting types by the likelihood of being associated with the target phenomenon is complicated by two properties of the input data. The first is that the type extraction process will typically be limited to a small number of examples as input, from as little as one example to perhaps no more than a handful, due to the interactive nature of the methodology. Secondly, the high fidelity with which precision grammars are able to model the complexity of language means that even once types unlikely to be associated with grammatical abstractions are removed, there will be a non-trivial number of remaining candidate types associated with the analyses of the various grammatical phenomena within the sentence, most of which will be unrelated to the target phenomena. One approach to mitigating this issue might be to boost the signal from potentially relevant types by simply adding additional positive exemplars to the input and weighting types by the proportion of sentences they occur in. This does not solve the issue however, as types unrelated to the target phenomena that are associated with frequently occurring phenomena are likely to also be present in additional exemplars, resulting in the weight of these irrelevant types increasing simultaneously.

As established in Section 2.3.6, the distribution of syntactic phenomena across a corpus exhibits a long tail, with a relatively small number accounting for a large proportion of instances and a large number of phenomena accounting for only a very small proportion. A similar property holds of the types from an HPSG grammar extracted from a treebank derived from that grammar. Rather than being an impediment however, the variation in the background frequency of types across a corpus contains information that can instead be exploited. The desired outcome of the weighting function is one that boosts the weight of types that occur commonly in positive exemplars, while decreasing the weight of types that occur frequently within a corpus. This desired property can be found in the tf-idf statistic (Salton and Buckley 1988), which has a strong history of use within the context of information retrieval tasks (Ju-
In the context of retrieving relevant documents from a collection, tf-idf is frequently used to weight terms by their importance to a document. The underlying intuition behind this weighting scheme is that terms found frequently within a given document are likely to be important to the document, while terms occurring frequently across a corpus are unlikely to be of particular relevance to the document. The tf-idf statistic achieves this by using the term frequency component to boost the weight of frequently occurring terms within the target document, while the inverse document frequency component offsets this by dampening the weight for terms that occur in many documents across a collection. The underlying intuition behind the tf-idf statistic can be applied to the Typediff methodology by conceptualising a set of input exemplars as a “document” that pertains to the topic of the syntactic phenomenon which the constituent sentences exemplify, and by the more straightforward interpretation of a collection being a treebank that is composed of many analyses, from which background frequencies of types can be extracted.

By modelling the input exemplars as bags of extracted types, the tf-idf function can be defined, as per usual, as the product of term frequency and inverse document frequency, but with these components being adapted for this context. I achieved this by deriving the term frequencies from the combined exemplar sentences and by deriving the inverse document frequencies from the analyses found within a treebank. The bag of types representation of an input exemplar is obtained through the process outlined in Section 5.4.1, with the exemplar being parsed dynamically and the selected analysis defaulting to the best analysis identified by the parse selection model, with the user able to select an alternative if necessary. Since this interactive process can’t be applied to a corpus, the inverse document frequencies were derived from the gold trees from the Redwoods treebank (Oepen et al. 2004), which, thanks to the dynamic treebanking methodology discussed in Section 2.6.5, are already synchronised with the 1214 release of the ERG and disambiguated, meaning that they represent the best analysis for their respective sentences that the grammar can provide. In the following description of the tf-idf weighting function, I use the term item to refer to an analysis of a sentence (either an input exemplar or a treebanked sentence) modelled as a bag of types generated using the type extraction procedure.

The function $tf-idf(t, I_E, I_T)$ from Equation 5.1 defines the tf-idf weighting for a given type $t$, a set of input exemplar items $I_E$, and a set of treebank items $I_T$. The function $tf(t, I_E)$ from Equation 5.2 yields the term frequency for the type $t$ from a set of exemplar items $I_E$, and is defined simply as the number of exemplar items that the type appears in. Finally, Equation 5.3 makes explicit the inverse document frequency, given a type $t$ and set of treebank items $I_T$. It is defined as the logarithmically scaled ratio of the number of items in the treebank $T_E$ to the number of items in the treebank containing $t$ (smoothed to avoid division by zero). A property of the weighting function defined by Equation 5.1 is that, when only a single input exemplar is provided, $tf-idf(t, I_E, I_T)$ takes on the same value as $idf(t, I_T)$. In this situation, no boosting of potentially relevant types through the term frequency
component occurs, as it will always have a value of 1, however the weighting function
still has the effect of reducing the weight of frequently occurring types, effectively
weighting types by their rarity in the treebank.

\[
\text{tf-idf}(t, I_E, I_T) = \text{tf}(t, I_E) \cdot \text{idf}(t, I_T) \quad (5.1)
\]

\[
\text{tf}(t, I_E) = |\{i \in I_E : t \in i\}| \quad (5.2)
\]

\[
\text{idf}(t, I_T) = \log \frac{|I_T|}{1 + |\{i \in I_T : t \in i\}|} \quad (5.3)
\]

(54) contains six exemplars of the right node raising construction and Table 5.2
presents two tables, each displaying the top 10 types when the type extraction pro-
cedure is applied to a different selection of exemplars from (54) and ranked using the
tf-idf formula from Equation 5.3. These are cut off at 10 types in order simulate a
ranking that could be readily scanned by a user interacting with the tool. For each
exemplar, the analyses returned by the parser were inspected and the best analysis
according to the parse selection model was verified as being an appropriate analysis
of the sentence.

(54) a. Linguistics concerns itself with describing and explaining the nature of
human language.
b. German is closely related to and classified alongside English and Dutch.
c. I selected and paid for second day delivery of my order.
d. An object is an entity which keeps or stores information in a database.
e. Keys are commonly used to join or combine data from two or more tables.
f. A database can set and hold multiple locks at the same time on the
different level of the physical data structure.

Table 5.2a shows the top 10 extracted and ranked types when only (54a) is pro-
vided as input. For this sentence, the type extraction process results in a total of 324
types, which is reduced to 126 once types labelled as other are removed. Since there
is only one input item, this yields the same ranking as ascending treebank coverage—
the rightmost column of the two tables in Table 5.2. Inspecting this ranking reveals
the sign type right_node_raise_vp_rule—which corresponds to the phrase rule li-
censing this construction—to be found in fourth position. This shows that by simply
weighting types with background frequencies from a treebank, a type that is clearly
relevant to this phenomenon has been made readily discoverable. There is room for
improvement, however, as the remaining types in this ranking are not associated
with the phenomenon characterised by the exemplars, with the possible exception of
vp_preditp_coord_top_phr which is discussed in the following section.
Chapter 5: Syntactic Phenomenon Discovery

Table 5.2: The different top 10 type rankings yielded by the tf-idf function from Equation 5.3 when different groups of exemplars are used as input.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Type</th>
<th>Supertype</th>
<th>tf-idf</th>
<th>Treebank (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v_nnp*-pp*_to_le</td>
<td>sign</td>
<td>6.60</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>n_-m-ed_le</td>
<td>sign</td>
<td>6.03</td>
<td>0.24</td>
</tr>
<tr>
<td>3</td>
<td>n_-pr-itself_le</td>
<td>sign</td>
<td>6.00</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td>right_node_raise_vp_rule</td>
<td>sign</td>
<td>5.56</td>
<td>0.38</td>
</tr>
<tr>
<td>5</td>
<td>refl_pron_rel</td>
<td>predsort</td>
<td>5.00</td>
<td>0.67</td>
</tr>
<tr>
<td>6</td>
<td>vp_predp_coord_top_phr</td>
<td>sign</td>
<td>4.78</td>
<td>0.84</td>
</tr>
<tr>
<td>7</td>
<td>_with_p_sel_rel</td>
<td>predsort</td>
<td>4.56</td>
<td>1.05</td>
</tr>
<tr>
<td>8</td>
<td>v_nnp-pp*_to_le</td>
<td>sign</td>
<td>4.54</td>
<td>1.06</td>
</tr>
<tr>
<td>9</td>
<td>empty_to_trans_verb</td>
<td>synsem</td>
<td>4.07</td>
<td>1.71</td>
</tr>
<tr>
<td>10</td>
<td>v_nnp-pp_e_le</td>
<td>sign</td>
<td>3.95</td>
<td>1.92</td>
</tr>
</tbody>
</table>

(a) Types extracted when (54a) is used as input.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Type</th>
<th>Supertype</th>
<th>tf-idf</th>
<th>Treebank (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>right_node_raise_vp_rule</td>
<td>sign</td>
<td>33.36</td>
<td>0.38</td>
</tr>
<tr>
<td>2</td>
<td>extracomp_rule</td>
<td>sign</td>
<td>15.75</td>
<td>7.24</td>
</tr>
<tr>
<td>3</td>
<td>hmark_e_phr_rule</td>
<td>sign</td>
<td>13.97</td>
<td>9.74</td>
</tr>
<tr>
<td>4</td>
<td>n_or_a_or_det</td>
<td>head</td>
<td>11.99</td>
<td>13.55</td>
</tr>
<tr>
<td>5</td>
<td>n_or_v</td>
<td>head</td>
<td>11.99</td>
<td>13.55</td>
</tr>
<tr>
<td>6</td>
<td>comp_spec_able</td>
<td>head</td>
<td>11.54</td>
<td>14.60</td>
</tr>
<tr>
<td>7</td>
<td>n_or_a_or_adv_or_det</td>
<td>head</td>
<td>11.29</td>
<td>15.24</td>
</tr>
<tr>
<td>8</td>
<td>vp_predp_coord_top_phr</td>
<td>sign</td>
<td>9.55</td>
<td>0.84</td>
</tr>
<tr>
<td>9</td>
<td>n_or_v_or_a</td>
<td>head</td>
<td>9.36</td>
<td>21.03</td>
</tr>
<tr>
<td>10</td>
<td>conj_relation</td>
<td>relation</td>
<td>9.21</td>
<td>21.54</td>
</tr>
</tbody>
</table>

(b) Types extracted when all exemplars in (54) are used as input.
Table 5.2b shows the type ranking produced when all exemplars of right node raising in (54) are used. The use of more than one exemplar item means that the term frequency component of Equation 5.3 can now boost the relative weight of types that occur in more than one exemplar—pushing them up the ranking. This is evident in the resultant ranking, with not only \texttt{right_node_raise_vp_rule} now being the highest ranking type, but with \texttt{extra_comp} and \texttt{hmark_e.phr_rule} occurring in positions two and three. These latter two types correspond to construction rules for extracted complements and coordinated verbal phrases respectively, which, while not exclusively involved with the phenomenon of right node raising, are associated with it. This makes them necessary—but not sufficient—for the presence of the right node raising construction, meaning that the emphasis they receive from the weighting function is desirable.

The examples presented here show that the tf-idf weighting clearly improves discoverability of types relevant to the phenomenon of right node raising. In Section 5.6, the entire Typediff methodology is tested against the phenomena within the Phenomenal Corpus, and in Section 5.7 its efficacy is assessed against a further more fine-grained set of phenomena. In Section 5.8, I include a critical discussion of the weighting used, addressing its limitations.

### 5.4.4 Type Filtering

The established methodology for extracting, labelling, and weighting types from exemplar items enables a range of operations that further improve the discoverability of grammar types associated with a syntactic phenomenon. One of these operations is filtering the ranked list of types by their type labels. When dealing with syntactic phenomena, \texttt{sign} types are frequently a locus of relevant abstractions within the grammar. Table 5.3 shows the result of filtering the types from the right node raising exemplars in (54) so as to only include \texttt{sign} types, again taking the top 10 weighted types. In this ranking, the first three types, which are the same as in Table 5.2b, occur in all exemplars. This filtering has also revealed a subsequent highly ranked group of three types with similar names—\texttt{vp.predp.coord.top.phr}, \texttt{vp.coord.nonfin.top.phr}, and \texttt{vp.coord.fin.top.phr}—which each occur in only one third of the exemplars. Given that the ERG may divide up the analysis of the right node raising construction into different variants, a question that the user might wish to answer is whether these types are mutually exclusive.

This can be explored by applying an additional filter, such that exemplars which do not include the type \texttt{vp.predp.coord.top.phr} are filtered out from being included for extraction. This reveals two salient pieces of information: (54a) and (54b) are the two exemplars containing this type, and within the resultant type ranking, \texttt{right_node_raise_vp_rule} and \texttt{vp.predp.coord.top.phr} are the top two types. Repeating this filtering process with \texttt{vp.coord.nonfin.top.phr} and \texttt{vp.coord.fin.top.phr} reveals that these types are indeed mutually exclusive, dividing up the exemplars.
### Table 5.3: The top 10 types from all exemplars in (54) when filtered to include only sign types.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Type</th>
<th>tf-idf</th>
<th>Exemplar (%)</th>
<th>Treebank (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>right_node_raise vp_rule</td>
<td>33.36</td>
<td>100.00</td>
<td>0.38</td>
</tr>
<tr>
<td>2</td>
<td>extracomp_rule</td>
<td>15.75</td>
<td>100.00</td>
<td>7.24</td>
</tr>
<tr>
<td>3</td>
<td>hmark_e_phr_rule</td>
<td>13.97</td>
<td>100.00</td>
<td>9.74</td>
</tr>
<tr>
<td>4</td>
<td>vp_predp_coord_top_phr</td>
<td>11.99</td>
<td>33.33</td>
<td>13.55</td>
</tr>
<tr>
<td>5</td>
<td>vp_coord_nonfin_top_phr</td>
<td>11.99</td>
<td>33.33</td>
<td>13.55</td>
</tr>
<tr>
<td>6</td>
<td>p_np_i-nm-poss_le</td>
<td>11.54</td>
<td>50.00</td>
<td>14.60</td>
</tr>
<tr>
<td>7</td>
<td>vp_coord_fin_top_phr</td>
<td>11.29</td>
<td>33.33</td>
<td>15.24</td>
</tr>
<tr>
<td>8</td>
<td>v_np*-pp*-ppfor_le</td>
<td>9.55</td>
<td>16.67</td>
<td>0.84</td>
</tr>
<tr>
<td>9</td>
<td>v_np*-pp_to_le</td>
<td>9.36</td>
<td>16.67</td>
<td>21.03</td>
</tr>
<tr>
<td>10</td>
<td>hadj_i_sl_rule</td>
<td>9.21</td>
<td>16.67</td>
<td>21.54</td>
</tr>
</tbody>
</table>

into pairs of two, and in each case the type being filtered on is ranked the highest after right_node_raise vp_rule. Within the ERG, these three types correspond to construction rules licensing a coordinated verb phrase, with each rule licensing coordination of verb phrases with head verbs in a different inflected form. This process, then, has revealed a possible way in which the handling of the phenomenon of right node raising by the ERG can be subdivided into more fine-grained categories, without requiring consultation of the ERG’s type hierarchy.

In the scenario just presented, the types that were identified as being able to divide up the right node raising analysis in the ERG were readily discoverable as they were visible within the top 10 when filtered by sign types. It may be the case, however, that types which could divide up the phenomenon space in an informative manner are located much further down the ranking. If, for example, a grammar type is identified as being associated with a target phenomenon but is found in only half the exemplars, this could suggest that the grammar analyses half the exemplars as a single phenomenon that aligns with the target phenomenon, but analyses the other half of the exemplars differently, potentially even further dividing up this group. In this scenario, a user would likely want to ascertain the grammar types that are associated with this latter subset of exemplars so as to establish how the grammar analyses them. A strategy to isolate these types is to this time filter out exemplars whose analysis do contain the type originally identified as being associated with the target phenomenon, causing types to only be extracted from exemplars whose analysis within the grammar is to be explored.

This example shows the potential utility of two distinct modes of type filtering within the phenomenon discovery process. The first operates at the level of the labels generated through the labelling process described in Section 5.4.4, allowing users to restrict their attention to types capturing specific kinds of grammatical abstractions.
The second mode allows users to either remove exemplar items that do not contain specific types, or to remove exemplar that do contain a specific type. This enables the Typediff methodology of type extraction and weighting to be applied to targeted subsets the grammars types, supporting the discovery of the way in which different grammar types divide up the input exemplars. Both of these modes of filtering are made available as affordances to the user within the Typediff interface, which is described in Section 5.4.6

5.4.5 Type Difference

Another operation that can be performed on the types extracted from exemplar items is to remove all types that occur in a secondary set of exemplar items. For two groups of exemplar items, each generating a set of extracted types $A$ and $B$, respectively, this is operationalised through set difference as $A - B$, with the resultant set of types then being ranked using the weighting function from Equation 5.3. The motivation for supporting this feature becomes evident when the $A$ types originate from a sentence exemplifying a target phenomenon and the $B$ types originate from a sentence that is as similar as possible but does not constitute an instance of the phenomenon. An example of this scenario is found in (55), with (55a) providing a positive instance of right node raising, and (55b) being the equivalent sentence but with the shared argument of the nature of human language now being realised explicitly within each conjunct. The use of such contrastive pairs to illustrate a phenomenon is common within descriptive grammars and is loosely analogous with the concept of a minimal pair from phonology, which consists of two words differing in only one phonological property but having different meanings. In this context, I will borrow the term to refer to a pair of sentences that differ with respect to a single syntactic phenomenon, remaining as similar as possible otherwise, and therefore ideally having the same meanings.

(55) a. Linguistics concerns itself with describing and explaining the nature of human language.

b. Linguistics concerns itself with describing the nature of human language and explaining the nature of human language.

Applying this methodology to the two sentences in (55) (treating (55a) as $A$ and (55b) as $B$) results in 27 output types, which is reduced to 19 when other types are filtered out—a notable reduction from the total 324 extracted from (55a) when it is processed alone. Table 5.4 shows the top 10 ranked types of the 19 that were produced after applying the set difference operation. The top two types in this ranking are now right_node_raise_vp_rule and extracomplement_rule, and when filtered to only include

---

10 This subtractive operation over types, which is realised through set difference, is the origin of the name Typediff.
Table 5.4: The top 10 types produced when the types extracted from (55b) are subtracted from (55a).

<table>
<thead>
<tr>
<th>Rank</th>
<th>Type</th>
<th>Supertype</th>
<th>tf-idf</th>
<th>Treebank (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>right_node_raise_vp_rule</td>
<td>sign</td>
<td>5.56</td>
<td>0.38</td>
</tr>
<tr>
<td>2</td>
<td>extracomp_rule</td>
<td>sign</td>
<td>2.63</td>
<td>7.24</td>
</tr>
<tr>
<td>3</td>
<td>n_or_a_or_det</td>
<td>head</td>
<td>2.00</td>
<td>13.55</td>
</tr>
<tr>
<td>4</td>
<td>n_or_v</td>
<td>head</td>
<td>2.00</td>
<td>13.55</td>
</tr>
<tr>
<td>5</td>
<td>comp_spec_able</td>
<td>head</td>
<td>1.92</td>
<td>14.60</td>
</tr>
<tr>
<td>6</td>
<td>n_or_a_or_adv_or_det</td>
<td>head</td>
<td>1.88</td>
<td>15.24</td>
</tr>
<tr>
<td>7</td>
<td>n_or_v_or_a</td>
<td>head</td>
<td>1.56</td>
<td>21.03</td>
</tr>
<tr>
<td>8</td>
<td>n_or_p_or_adv</td>
<td>head</td>
<td>1.36</td>
<td>25.76</td>
</tr>
<tr>
<td>9</td>
<td>n_or_p_or_v_or_adv</td>
<td>head</td>
<td>1.36</td>
<td>25.76</td>
</tr>
<tr>
<td>10</td>
<td>n_or_p_or_v</td>
<td>head</td>
<td>1.35</td>
<td>25.88</td>
</tr>
</tbody>
</table>

sign types, these are the only remaining types, indicating that are the only sign types involved in the analysis of (55a) that are not used in the analysis of (55b).

In terms of discovering the corresponding types of the ERG that are involved in the right node raising construction, this has provided further information, by revealing that the extracomp_rule discriminates between positive instances of the construction and their non-right node raising counterparts. Previously, the use of only positive exemplars in (54) revealed that both extracomp_rule and hmark_e_phr_rule are necessarily involved in a right node raising analysis, however set difference over a positive and a negative exemplar has revealed that of these two, extracomp_rule is arguably more characteristic of the construction within the ERG.

A further advantage of this approach is that it can require fewer input exemplars. Where (54) involved identifying six positive exemplars, (55) only involved two exemplars, one positive and one negative. This is a result of the different methods these two modes of discovery use for triangulating upon phenomenologically interesting types from the potentially large set of extracted types. The use of only positive exemplars combined with tf-idf weighting requires a sufficient number of exemplars to boost the relative weighting of potentially salient types through the term frequency component of Equation 5.3. The type difference mode, on the other hand, makes use of a masking approach, using a negative instance of the phenomenon to completely remove from consideration a large number of types.

\[^{11}\text{It is possible to arrive at a type ranking that has the pertinent types of right_node_raise_vp_rule, extracomp_rule, and hmark_e_phr_rule occurring in the top three positions by using less than six exemplars, however smaller numbers of exemplars become more sensitive to the particular sentences chosen. For example, (54a), (54c), and (54e) could have been used, however many of the other three-exemplar combinations from that set of six do not yield this result.}\]
There are some notable limitations of the type difference approach. Firstly, if one is not already provided, the process of identifying a suitable minimal pair for a target phenomenon may take some time—possibly more than the amount of time it would take to identifying and add further positive exemplars. Even more limiting is the fact that many syntactic phenomena may not be amenable to being isolated through a minimal pair, meaning that there will be no corresponding single negative sentence that can be used to conveniently mask a large number of likely irrelevant types. This can potentially be overcome by including additional $B$ items to try to cover the constituents required to produce the types that need to be masked. The concern here however is that given the nature of the discovery process, where the goal is not known in advance, this process becomes somewhat circular. In particular, the user does not have any way to ascertain when too much negative material has been added, resulting in the masking of potentially relevant types. This also raises another limitation, which is that the type difference operation does not discriminate over where in the sentence types occur. This means that types originating from $B$ items and introduced due to phenomena from unrelated parts of the sentence could potentially mask types that happen to be involved in the analysis of the target phenomenon in the $A$ item. This effect can be at least ameliorated by selecting positive and negative exemplars that are as small as possible—whilst still isolating the target phenomenon—so as to reduce the chances of both introducing extraneous items (through $A$ items) and masking relevant types (through $B$ items).

The two modes of syntactic phenomenon discovery enabled by the type extraction methodology—the exploration of types associated with positive exemplars through tf-idf weighting and the exploration of types by masking a positive exemplar’s types with a negative exemplar’s types—represent complementary avenues of exploration. They not only provide different kinds of information that can be combined together to build up hypotheses pertaining to which types are relevant to a phenomenon, but they also respond differently to different collections of input exemplars, the availability of which is likely to vary with the resources available within a discovery context.

5.4.6 The Interface

This section provides a brief walkthrough of the Typediff interface, explaining how the various features of the Typediff methodology that have been outlined in this section were incorporated.

Figure 5.8 presents a screenshot of the Typediff interface with the right node raising exemplars from (54) used as input. In order to arrive at the state shown in this screenshot, the user must first select the target grammar and treebank to use. For this example, the 1214 release of the ERG has been used, along with the accompanying release of the Redwoods Treebank. The user then enters and submits the desired positive exemplars as input, which Typediff parses using ACE. Assuming a successful parse is found for all exemplars, they are then displayed down the left hand
Figure 5.8: Screenshot of the Typediff interface with the right node raising exemplars from (54) used as input.

side of the interface. For each exemplar, the best parse according to the parse selection model is used for extracting types. If the user is not happy with an analysis for an exemplar, clicking on the exemplar reveals visualisations of the top 10 derivations\(^{12}\) which the user can select the desired analysis from.

On the right hand side of the interface, the user is presented with a ranked list of types. These represent the union of the types extracted from all input exemplars, using the type extraction methodology described in Section 5.4.1. To help users rapidly process the type ranking, these types are coloured according to the supertype labels they received from the labelling process described in Section 5.4.1. By default, the types are ranked according to the tf-idf weighting function Equation 5.1 described in Section 5.4.3, however the user can also rank the types by coverage over treebank items, coverage over input items, a predefined label ordering, as well as combinations of these, such as ordering by label and then by tf-idf weighting.

In addition to selecting alternative analyses for input exemplars, users can also explore the effects of various operations on the resultant type ranking. This includes the ability to manually disable individual exemplars, thereby excluding them from being used as inputs into the type extraction process, as well as being able to remove them entirely. Users can also perform the two types of filtering that were described in Section 5.4.1. The first enables filtering the current ranking to only include types of a specified label, such as only \texttt{sign} or only \texttt{synsem} types, and is applied by selecting

\(^{12}\)This can be increased using the \texttt{count} configuration parameter.
the label to restrict the ranking to from the top of the column displaying each type’s label. The second type of filtering allows users to restrict the set of exemplars being used to only those that do contain a specific type or to only those that do not contain a specific type, with these actions being exposed to users as buttons on each type in the ranking. These item-level type filtering affordances facilitate the discovery of how the input exemplars can be divided up by different sets of grammar types from the target grammar. The interface also presents another way of quickly ascertaining such divisions. When a user hovers over a type in the ranking, the corresponding input exemplars containing this type are highlighted, and when the type is clicked, this locks the highlighting until clicked again.\(^{13}\)

The screenshot shown in Figure 5.8 illustrates the use of Typediff to explore a syntactic phenomenon through the selection of positive exemplars and by leveraging the tf-idf weighting of types. As described in Section 5.4.5, the other mode of use that Typediff supports is leveraging a syntactically similar but negative exemplar to mask types within the type ranking that are less likely to be relevant to the phenomenon. This is illustrated by the screenshot in Figure 5.9, which shows the minimal pair from (55) used to triangulate upon types associated with the right node raising construction.

In enabling the type masking mode of use, rather than limiting the user to a single positive exemplar and a single negative exemplar, the interface was intentionally constructed to support a flexible range of operations over two different groups of exemplars. As in Section 5.4.5, these groups are labelled A and B, and the set difference operation of \(A - B\) is performed by default, meaning that types within the output ranking all originate from exemplars within A. Users can provide just a single exemplar for both groups, as in Figure 5.9, or can supply multiple input exemplars for either group, in which case the union of types is derived for each group before the set difference operation is performed. Users can also change the operation applied to A and B from the default of set difference to union or intersection. The latter of these provides users with the ability to derive the shared types between the two groups—weighted by tf-idf\(^{14}\)—which could yield valuable information when used to compare positive with negative exemplars. Additionally, the interface supports swapping the A items with B items, reversing the direction of the types being masked when set difference is being used.

Finally, the Typediff interface was also augmented with two external tools to assist users in identifying types from the grammar that are relevant to the target phenomenon. These are the treebank querying tool Fangorn (Ghodke and Bird

---

\(^{13}\)An additional feature that helps users discover the role of grammar types is that when sign types are hovered over, the corresponding span in the input text and the corresponding subtree within its derivation trees are also highlighted.

\(^{14}\)In this case, the term frequency for each type is taken to be the number of exemplar items from the A group that contain the type, on the basis that even though the type occurs in at least one B exemplar, it is still the nominally positive exemplars from A that are of interest.
2012), which was discussed briefly in Section 2.9.3, and the Linguistic Type Database (LTDB; Hashimoto et al. 2007), discussed in Section 2.9.1. Both of these tools are made available to the user from the visualisation of the derivation tree for each analysis. When a user hovers over a node within a derivation tree, a tooltip opens up, which, in addition to displaying the sign type of that node and its coverage over the active treebank, also includes links to both tools. This is illustrated in Figure 5.10, where the user is hovering over the node corresponding to the lexical entry for describe, revealing that in this analysis the lexical type used for this word was v np-pp* to le, which is found in 1.06% of the trees in the Redwoods Treebank. Below this information are the two links, query, and LTDB, corresponding to Fangorn and the LTDB respectively.

Following the query link takes the user to an installation of Fangorn that has been indexed with the Redwoods Treebank, automatically performing a query for trees with a node labelled as v np-pp*. This allows users to improve their understanding of the role this sign type plays within the grammar by examining vetted analyses containing the type. Users can also build upon this understanding by performing more specific queries using the query-by-annotation feature of Fangorn, which was discussed in Section 2.9.3. This allows users to construct new queries by annotating trees from existing results—without having to be familiar with the underlying query language used by Fangorn.

Following the LTDB link takes the user to the v np-pp* to le entry within an
Figure 5.10: The derivation tree for an analysis from the Typediff interface, showing the links to Fangorn and the LTDB that are available to users when hovering over nodes of the tree.
installation of the LTDB that has been populated with types from the ERG. This page presents users with an overview of the type that serves as its documentation within the ERG. This includes notes regarding its linguistic function, a sample of sentences from any indexed treebanks that the type is found in the analysis of (along with visualisations of the corresponding syntactic and semantic analysis), and the definition of the type within the ERG. As discussed in Section 2.9.1, this overview is intended to provide a single location that brings together information about how a type is used within a grammar, assisting grammar engineers to work collaboratively on the grammar. In the context of a user exploring how a given phenomenon is handled within a grammar, such information is likely to also be beneficial for enhancing their understanding.

5.5 The Typediff Implementation

I implemented both components of the Typediff methodology—the type extraction, labelling, and weighting, and the interactive interface to the type rankings—as a browser-based web application and released it as an open source software package. In addition to serving as a reference implementation of the Typediff methodology described in this chapter, this implemented tool also performed a number of valuable functions within my research. Before describing these functions, I first briefly outline how the tool was implemented.

The web application consists of a client, built using HTML, CSS, and JavaScript, which communicates with a server-side Python application built using the Flask web framework. In addition to serialising data between the client and server, the Python application also dispatches input exemplars received from the client to ACE for parsing and sends the resultant output to the custom ACE-derived program that extracts grammar types from each analysis. The resultant bag of type representations of parsed sentences are sent to the client, in addition to information pertaining to the grammar’s type hierarchy and the frequencies of types across the selected treebank. The client uses this data to label each type and derive its tf-idf weight in the context of the other exemplars, and then derive the type ranking for the group of exemplars. The client also implements and makes available to the user all the interactive features and affordances documented in Section 5.4.6, which facilitate the identification of types likely to be relevant to the target phenomenon. A further feature of the client is that its current state, including information such as which grammar is active and which sentences have been submitted to A and B categories, is recorded in the current URL in the browser’s location bar. This enables users to save the state of an informative Typediff configuration to rapidly recreate it at a later date, or to share amongst collaborators.

15https://github.com/ned2/typediff
The implementation of Typediff was valuable during the development of the methodology itself, where, initially, it provided a convenient way to verify that the adopted approach was a viable method for performing syntactic phenomenon discovery. After this point, it was also valuable in determining the impacts of changes to various components of the methodology, such as the addition of new interactive features, modifications to the weighting function, and changes to the layout of the interface. Additionally, the implemented tool was used to simulate two activities that correspond to the motivating context of aligning syntactic phenomenon characterisations with precision grammars. These two tasks, which are described in the following two sections, provided the opportunity to qualitatively assess the methodology’s usability and efficacy at the task of identifying types from a grammar that are associated with a particular syntactic phenomenon.

5.6 Phenomenal Corpus Exploration

In order to assess the efficacy of the Typediff methodology, I made use of the Phenomenal Corpus, which was described in Chapter 4. This involved treating annotated instances from the corpus as exemplars of their respective phenomena. These could be used as input into the implemented Typediff tool in order to identify types from the ERG that are associated with the phenomena. The descriptive characterisations from the annotation guidelines that were used to derive the recorded instances of syntactic phenomena in the corpus served to model the context of aligning descriptive characterisations of syntactic phenomena with associated precision grammar components. This presented an opportunity to assess the performance of the Typediff methodology in a scenario that simulates the motivating application of phenomenon discovery. This simulation is intended only to verify that Typediff is able to be used for performing syntactic phenomenon discovery and does not attempt to validate its use in a scenario which models the augmentation of a descriptive grammar, which would likely be restricted to more limited range of input exemplars. I discuss the need for an evaluation of the Typediff methodology that more closely reflects a real-world application in Section 6.3.3.

5.6.1 Experimental Setup

In order to perform this investigation, I enhanced the Typediff implementation by enabling it to interface with profiles produced by `incr tsdb()` (Oepen and Flickinger 1998, Oepen and Carroll 2000), the test suite performance profiling and management tool used by delph-in grammars, as this is the format used by the Phenomenal Corpus. This enhancement involved two components. The first was adding to the interface and server-side application the ability to select and retrieve items from a profile. This also included adding the ability to filter retrieved items to only those
containing a specific phenomenon annotation. Adding these features was necessary to be able to treat retrieved items as exemplars of distinct phenomena and was readily enabled by taking advantage of \texttt{[incr tsdb(\[])}\texttt{]}’s query language, which I described in Section 2.9.1. The second component then involved allowing Typediff to accept, as input exemplars, recorded item derivations extracted from profiles, instead of using user-supplied exemplars that are dispatched for parsing. This meant that I could take advantage of the disambiguated gold trees provided by DeepBank, the dynamically annotated treebank of WSJ text, described in Section 3.3. As I discussed in Section 4.5, using a disambiguated treebank as a source of syntactic phenomenon exemplars is advantageous in that it eliminates the need for a user to manually select an appropriate analysis from the potentially numerous analyses licensed by a grammar, in addition to reducing the risk of noise from the target phenomena not receiving appropriate treatment in the selected analysis.

The investigation itself involved my proceeding through the three phenomena annotated within the Phenomenal Corpus—passive clauses, relative clauses, and complement clauses—and trying to use Typediff to identify types from the ERG that are associated with the analysis of these syntactic phenomena. For each phenomenon, I began by loading all recorded annotations for the category into Typediff as \texttt{A items} and then proceeded to try to use the Typediff interface, including all the affordances described in Section 5.4, to locate salient grammar types. I restricted this process to only including annotated exemplars from the Phenomenal Corpus, which meant that, since the corpus only features positive instances of phenomena, I did not use Typediff’s type difference mode of use that requires negative exemplars. It should also be noted that the number of exemplars associated with each phenomenon does not correspond to the number of annotated instances from the Phenomenal Corpus, which was reported in Table 4.3, in Chapter 4. This is due firstly to a proportion of sentences in the corpus featuring more than one instance of the same phenomenon, and secondly, to a small number of sentences not receiving an analysis by the ERG. Table 5.5 provides an overview of this for each phenomenon category, showing the number of annotation instances in the Phenomenal Corpus, the number of potential exemplars this yields, and then the number of these exemplars with an analysis from the ERG—representing the size of each group used for this investigation.

5.6.2 Results

The results of this investigation proved to be informative. Each phenomenon category revealed different aspects of Typediff’s capacity for discovering relevant grammar types.
Passive Clauses

Entering the passive exemplars from the Phenomenal Corpus into Typediff provided an immediate window into the ERG’s analyses of this construction. In particular, three `predsort` types appearing at the top of the type ranking were revealed to be associated with all but three exemplars: `passive_rel`, `by_p_cm_rel`, and `parg_d_rel`. As `predsort` types capture abstractions across the kinds of predicates that semantic relations introduce, this indicates that part of the ERG’s analysis of passive clauses is exposed in the semantic representations it produces. Leveraging Typediff’s type filtering feature showed that in two of the exemplars not featuring these types, the past participle verb forms (denationalized and entrenched) could not be derived from any base verb and were assigned generic passive verb lexical entries by the unknown word handler, suggesting that the ERG’s analysis of unknown passives is not extended to its semantic representations. The third exemplar, on the other hand, was a spurious passive annotation from the Phenomenal Corpus. I therefore removed these three exemplars, which gave the aforementioned `predsort` types 100% coverage over the exemplar items.

The top 10 types from the type ranking that Typediff produced for the pruned set of exemplars is shown in Table 5.6a, along with their coverage over the analyses of the passive exemplars and coverage over the Redwoods treebank analyses. The comparatively low treebank coverage of the three `predsort` types suggests all are indeed associated with the analysis of the passive. The two `relation` types at the end of this type ranking fragment—`arg01_relation` and `arg2_relation`—also have full coverage with comparatively lower treebank coverage, indicating that these may also be associated with the semantic component of the passive’s analysis. The `synsem` type `passive_synsem`, jumps out as being clearly relevant, as does the `sign` type `v_pas-norm_lexrule`, which, inspection of the derivation tree makes clear, is the inflectional lexical rule that generates the past participle form of the verb in the presence of a passive construction. This type does not cover all passive exemplars however. Type filtering revealed that the lexical rules `basic_passive_verb_lr` and `prep_passive_verb_lr` together cover the small number of remaining exemplars. The

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Instances</th>
<th>Exemplars</th>
<th>Analysed exemplars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passive</td>
<td>102</td>
<td>90</td>
<td>85</td>
</tr>
<tr>
<td>Relative</td>
<td>141</td>
<td>111</td>
<td>101</td>
</tr>
<tr>
<td>Complement</td>
<td>283</td>
<td>210</td>
<td>197</td>
</tr>
</tbody>
</table>

Table 5.5: The number of instances of each phenomenon from the Phenomenal corpus, the number of exemplar items producing these, and the number of exemplars with an available analysis in DeepBank.
Table 5.6: The top 10 types produced for the passive and relative clause exemplars sourced from the Phenomenal corpus, along with the supertype label of each type, its coverage over the exemplars found in the group, and its coverage over the Redwoods Treebank. Coverage rates here indicate the percentage of sentences whose best analysis involves the relevant type.
remaining two sign types are unrelated to the passive construction, with their prominence in this ranking being a result of the exemplars containing a large number of proper nouns (indicated by n-pn-gen.le and np.name.cmpnd.rule) and reduced relatives (indicated by red.rel.cl.rule).

Relative Clauses

The type ranking produced by feeding the relative clause exemplars into Typediff is shown in Table 5.6b. Through inspecting the names of these types, it is apparent that the following sign types from the ranking are likely to be involved with the analysis of relative clauses: red_rel.cl.rule, hadj_i.relcl.pr.rule, hadj_i.relcl.npr.rule, extrasubj_fin.rule, filler_head_fin_rule.rel. However, unlike the passive—where relevant types were either involved in every exemplar, or close to—none of these types reach as high a coverage. While the exemplar coverage cannot show whether any of these types are in complementary distribution to each other—due to the occurrence of multiple instances within a sentence—they suggest the hypothesis that some of these types are associated with the handling of different relative clause variants. Further investigation using type filtering to determine how these types divide up the exemplars revealed this to be the case.

Typediff has then revealed an interesting difference between the syntactic phenomena of passive clauses and relatives clauses—as characterised within the Phenomenal Corpus—which is that the passive exemplars receive a largely homogeneous analysis, whereas the relative clause exemplars receive more heterogeneous analyses, being composed of a number of different distinct phenomenon variants. This resulted in a more challenging discovery process, but one that Typediff was still able to provide assistance with. While I don’t document Typediff’s exploration of these variants further here, in Section 5.7 I investigate fine-grained divisions within the ERG’s relative clause analyses using exemplars from a different source.

Complement Clauses

Lastly, turning to the exemplars from the Phenomenal Corpus annotated with complement clauses, feeding these into Typediff produced results that were different again from that of the previous two phenomena. The initial ranking that Typediff first produced for the complement clause exemplars was not as immediately informative as the rankings produced for both passive and relative clauses. After some exploration, only three types from the top 10 were identified as being relevant to this phenomenon. In the top two positions were pp_cp_fin.unspec.verb and v.pp*-cp_fin.imp.le, synsem and sign types respectively, which had identical frequencies, suggesting that they co-occur with each other. The lexical type v.pp*-cp_fin.imp.le allows verbs to take clauses as complements, as well as accepting an optional preposition phrase as a complement, licensing constructions such as
Kim said to Alex that they won. The third was the sign type subjhh_nonmc_rule, which licenses subordinate clauses, making it very much associated with this syntactic phenomenon. Further exploration of the type ranking revealed that exemplars could be divided up according to the lexical type that licensed the complement clause in the ERG’s analyses. I arrived at this observation by filtering out exemplars featuring the lexical type v.pp*-cp_fin-imp.le, which removed roughly half the exemplars. The type ranking produced by the resulting reduced set of exemplars promoted another lexical type licensing a complement clause to the top 10 types, and I was then able to repeat this process until fourteen lexical types were identified that covered all exemplars annotated as complement clauses. These complement-clause licensing lexical types are shown in Table 5.7, along with the number of exemplars they were found in, coverage over the treebank, as well as examples sentences illustrating each lexical type. They can be divided into two broad categories—whether the complement takes the form of a verb phrase (indicated by the presence of vp in the type identifier) or whether it takes the form of a complementiser phrase (indicated by the presence of cp in the type identifier), which is characterised through a complementiser, such as to and that. In order to arrive at an understanding of the functionality of these lexical types, I found Typediff’s inclusion of links to their documentation within the LTDB to be particularly useful.

5.6.3 Observations

Through these explorations I was successfully able to use Typediff to identify ERG types associated with phenomenon characterisations defined by a collection of exemplars. The rankings produced through the type weighting function promoted relevant types to within the top 10 for each of the three phenomena, particularly so with passives and relatives. Of notable utility was the type filtering feature, which, for all three of the phenomena, was used to explore phenomenon variants by filtering out a type identified as being relevant, and then exploring the resultant type ranking produced from all exemplars not featuring this type.

Another outcome of this investigation was the discovery of the negative impact that unrelated phenomena could have on the type rankings. This was seen in the ranking produced for the passive, with sign types associated with proper nouns and reduced relatives. This is likely an artefact of these phenomena being more frequent in newswire text. The noise from these phenomena would likely be ameliorated by using treebank coverage statistics from DeepBank rather than Redwoods, as the corpus frequencies would then be sourced from the same domain as the exemplars. My use of Redwoods was not a principled decision but the result of using the same configuration as that used throughout the development of Typediff. While unfortunate, this served to highlight the importance of matching the domains of the exemplars and the corpus used to derive treebank statistics.

The discoveries regarding associations between ERG types and syntactic phenomen-
Chapter 5: Syntactic Phenomenon Discovery

<table>
<thead>
<tr>
<th>Type</th>
<th>Exemplars</th>
<th>Treebank (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v.pp*-cp_fin-imp_le</td>
<td>99</td>
<td>0.55</td>
</tr>
<tr>
<td>v_vp_seq_le</td>
<td>27</td>
<td>4.22</td>
</tr>
<tr>
<td>v.pp*-cp_le</td>
<td>22</td>
<td>1.30</td>
</tr>
<tr>
<td>v_np-vp_oeq_le</td>
<td>12</td>
<td>0.82</td>
</tr>
<tr>
<td>v_cp_prop_le</td>
<td>11</td>
<td>1.24</td>
</tr>
<tr>
<td>v_np-vp_aeq_le</td>
<td>8</td>
<td>0.51</td>
</tr>
<tr>
<td>v.pp*-cp_unsp_le</td>
<td>6</td>
<td>0.09</td>
</tr>
<tr>
<td>v_np-vp_sor_le</td>
<td>5</td>
<td>0.29</td>
</tr>
<tr>
<td>v_vp_ssr_le</td>
<td>4</td>
<td>0.31</td>
</tr>
<tr>
<td>v.pp-vp_ssr_le</td>
<td>2</td>
<td>0.23</td>
</tr>
<tr>
<td>v_cp_fin-inf-q_le</td>
<td>1</td>
<td>0.95</td>
</tr>
<tr>
<td>v_np*-np-vp_it-s_le</td>
<td>1</td>
<td>0.12</td>
</tr>
<tr>
<td>v_p-vp_seq_le</td>
<td>1</td>
<td>0.54</td>
</tr>
<tr>
<td>v_np-cp_q_le</td>
<td>1</td>
<td>0.02</td>
</tr>
</tbody>
</table>

1. He says [he has never even dined with gangsters.]
2. It’s not clear whether Aska plans to [buy more shares.]
3. This time, however, some analysts think [he could face a real battle.]
4. The first Barney’s shop is scheduled to open in Japan next year.
5. They believe [the media, including “Batibot,” have played a crucial role.]
6. DIG is the vehicle being used [to pursue to acquisition.]
7. And neither Democrats nor Republicans are predicting [that the capital-gains forces can produce enough votes.]
8. The purchase of the additional 48% stake is expected [to cost more than 11 billion francs ($1.77 billion).]
9. GOP leaders continued [to press for a vote on the amendment to the Eastern Europe aid measure.]
10. Mr. Morishita’s main business certainly appears [to be thriving, although he won’t disclose numbers.]
11. We asked police to investigate [why they are allowed to distribute the flag in this way.]
12. It doesn’t take much [to provoke an intense debate.]
13. According to chief scriptwriter Rene Villanueva, “Batibot” doesn’t set out [to advance the cause of Filipino.]
14. But when asked [what exactly he does in business,] he immediately takes offense.

Table 5.7: Lexical types licensing complement clauses that were discovered using Typediff. Exemplars for each lexical type are also presented with the verb licensing the complement clause being underlined, and the complement delimited by braces.
ena that were made in this exploration would be of particular utility to someone who is unfamiliar with the grammar. Assuming the existence of a set of phenomenon exemplars, Typediff’s data-driven approach allows users to identify types associated with the grammar’s analysis of a target phenomenon, without having to be familiar with any of the grammar’s source code. The discoveries are less likely to be of interest to a grammar engineer familiar with the grammar, however. If they are not already aware of the specifics of how these phenomena are analysed by the ERG, then their ability to navigate through the source code of the grammar would result in a more concrete understanding of the grammar’s analysis. Typediff could, however, provide a convenient interface to bootstrap code-based exploration, as it brings together potentially relevant types from disparate locations in the grammar into the one interface. In the following section, I perform another investigation with the ERG, this time applying Typediff to syntactic annotations corresponding to phenomenon characterisations that are designed to be of interest to a grammar engineer working with the ERG.

5.7 Relative Construction Exploration

In this section I describe another exploration into the use of Typediff to identify ERG types associated with syntactic phenomena. While the previous exploration used the broad phenomenon categories sourced from the Phenomenal Corpus, this investigation makes use of categories that are characterised by more fine-grained intra-phenomenon distinctions. Applying Typediff to these categories was intended to model a scenario that would be of more relevance to a grammar engineer. In particular, this was conceptualised as a situation where a grammar engineer wants to understand how the analyses and distinctions made within their grammar correspond to analyses found within a descriptive grammar.

5.7.1 Exemplar Extraction

For this exploration I chose to further explore the phenomenon of relative clauses. In Section 5.6.2, the type rankings produced by Typediff revealed this phenomenon—as characterised in the Phenomenal Corpus—to be analysed as a range of distinct construction variants, suggesting that this represents a good candidate for the exploration of more fine-grained phenomenon distinctions with Typediff. I chose *The Cambridge Grammar of the English Language* (Huddleston and Pullum 2002) (“Huddleston and Pullum”, from hereon) as a source of exemplars that illustrate fine-grained distinctions between different types of relative clauses. This selection was made, firstly, on the basis that the extracted exemplars would support the modelling of a grammar engineer aligning syntactic phenomena from a descriptive grammar with corresponding analyses in their own grammar. Secondly, it features a thorough analysis of relative
constructions that makes use of a well-defined set of analytic distinctions, as well as a large number of exemplars illustrating these, making it a good candidate for extracting instances of fine-grained syntactic phenomena.

The bulk of Chapter 12 of Huddleston and Pullum is dedicated to an analysis of English relative constructions. Huddleston and Pullum use the term *relative construction* to refer to the broader category that includes both relative clauses which modify independent head nouns and fused relatives, such as (56b). In fused relatives, the relativised constituent is not present as an antecedent to a relativising clause—as it is in (56a)—but is instead part of a complete relativising construction, which is not a verb phrase but either a noun phrase or a preposition phrase. In this exploration I include fused relatives.

(56) a. It would mean abandoning that which we hold most dear.
    b. It would mean abandoning what we hold most dear.

In their analysis, Huddleston and Pullum employ two orthogonal contrastive dimensions, which they use to classify different types of relatives (Huddleston and Pullum 2002:1033). The first dimension is *formal*, classifying constructions by whether they use a relativising word such as *who* or *which*, the subordinator *that*, or are a bare relative, having no relativising word. The second dimension is *relational*, classifying constructions by their relationship with the construction that contains them. This dimension distinguishes between integrated, supplementary, cleft, and fused relatives, with the first two—integrated and supplementary—roughly corresponding to the traditional distinction of restrictive and non-restrictive relatives, which I described in Section 4.4.1. A third dimension, which is given less salience, but which I use for this dataset, is the *grammatical function* of the element that is being relativised from the relative clause, such as subject, object, or predicative complement. I used these three dimensions to create a coding scheme involving the three categories of formal, relational, and grammatical function. For each of these categories I derived a set of possible classes such that for every relative construction, it can be assigned a distinct value for each category. This coding scheme is shown in Table 5.8 and Table 5.9.

Some of the contrasts in Huddleston and Pullum involve hierarchical divisions (for example, sub-classifying *wh*-relatives as *wh*-simple, and *wh*-complex, with the latter being further subdivided). To simplify annotation, within the coding scheme, I flattened these hierarchical relationships within the coding scheme. Since it could still be useful to explore broader categories such as *wh*-complex, I added some additional derived classes. Annotations for these derived classes can be automatically generated based on the implied hierarchy. These are listed under *derived classes* in Table 5.8.

Armed with this coding scheme, I extracted all exemplars from the portion of Huddleston and Pullum’s Chapter 12 that is dedicated to the treatment of relative

---

16All examples in this section are from Chapter 12 of Huddleston and Pullum (2002:1031)
### Chapter 5: Syntactic Phenomenon Discovery

#### Table 5.8: The formal and relational dimensions from the coding scheme used to annotate relative constructions extracted from Huddleston and Pullum (2002)

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>No.</th>
<th>Exemplar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formal</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>that</td>
<td>51</td>
<td>This is the book that she recommended.</td>
</tr>
<tr>
<td>bare</td>
<td>40</td>
<td>That’s not really the reason she left him.</td>
</tr>
<tr>
<td>wh-simple</td>
<td>151</td>
<td>This is the letter which drew our attention to the problem.</td>
</tr>
<tr>
<td>wh-complex-1</td>
<td>14</td>
<td>She is the ideal person in whom to confide.</td>
</tr>
<tr>
<td>wh-complex-2</td>
<td>10</td>
<td>He came up with a strange plan, the purpose of which escapes me.</td>
</tr>
<tr>
<td>wh-complex-3</td>
<td>2</td>
<td>Several MPs were interviewed, chief among whom was the Chancellor of the Exchequer, Douglas Durack.</td>
</tr>
<tr>
<td>wh-complex-4</td>
<td>0</td>
<td>I became disturbed by a ‘higher criticism’ of the Bible, to refute which I felt the need of a better knowledge of Hebrew and archaeology.</td>
</tr>
<tr>
<td>wh-complex-5</td>
<td>0</td>
<td>They take a rigorous examination, passing which confers on the student a virtual guarantee of a place at the university.</td>
</tr>
<tr>
<td>wh-complex-6</td>
<td>12</td>
<td>The report contains statements whose factual truth is doubtful.</td>
</tr>
<tr>
<td>wh-complex-7</td>
<td>18</td>
<td>I may be late, in which case I suggest you start without me.</td>
</tr>
<tr>
<td>wh-complex-8</td>
<td>5</td>
<td>I haven’t yet met the people in whose house I would be staying.</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>303</td>
<td></td>
</tr>
</tbody>
</table>

| **Relational**        |     |                                                                           |
| integrated-non-inf    | 150 | The penknife that he was trying to cut it with was blunt.                 |
| integrated-inf        | 18  | The best place from which to set out on the journey is Aberdeen.          |
| supplementary         | 50  | They accused him of being a traitor, which he undoubtedly was.            |
| cleft                 | 8   | It’s to avoid such a conflict of interest that I’m resigning.             |
| fused                 | 77  | We’ll give whoever needs it a second chance.                              |
| **Total**             | 303 |                                                                           |

Table 5.8: The formal and relational dimensions from the coding scheme used to annotate relative constructions extracted from Huddleston and Pullum (2002)
### Table 5.9: The grammatical function dimension and derived classes from the coding scheme used to annotate relative constructions extracted from Huddleston and Pullum (2002).

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>No.</th>
<th>Exemplar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grammatical Function</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td>83</td>
<td><em>You should take the first appointment that is available.</em></td>
</tr>
<tr>
<td>object</td>
<td>94</td>
<td><em>Anything that you say may be used in evidence against you.</em></td>
</tr>
<tr>
<td>predicative-comp</td>
<td>22</td>
<td><em>I don’t think it is the good investment they consider it.</em></td>
</tr>
<tr>
<td>comp-of-prep</td>
<td>14</td>
<td><em>I’ll use whichever edition I can get hold of.</em></td>
</tr>
<tr>
<td>adjunct</td>
<td>82</td>
<td><em>He wrote most of his poetry during the years while he was in Paris.</em></td>
</tr>
<tr>
<td>genitive-subj-det</td>
<td>1</td>
<td><em>One cannot tailor a suit for a client whose measurements remain unknown.</em></td>
</tr>
<tr>
<td>comp-of-aux-verb</td>
<td>4</td>
<td><em>They advised me to call the police, which I did immediately.</em></td>
</tr>
<tr>
<td>oblique-comp</td>
<td>3</td>
<td><em>They advised me to call the police, which I did immediately.</em></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>303</td>
<td></td>
</tr>
</tbody>
</table>

| **Derived Classes** |     |                                                                           |
| relative            | 303 |                                                                           |
| integrated          | 168 |                                                                           |
| wh                  | 212 |                                                                           |
| non-wh              | 91  |                                                                           |
| wh-complex          | 61  |                                                                           |
Chapter 5: Syntactic Phenomenon Discovery

constructions (pages 1033–1078). Since descriptive grammar exemplars used to illustrate specific phenomena are typically only labelled with the phenomenon being illustrated, I was mostly only able to derive a value for one of the dimensions for each exemplar. Limiting the annotations to only these would have resulted in a sparsely populated dataset, so the missing values for the other dimensions were added through manual annotation. This annotation was performed by Dan Flickinger, the maintainer and primary developer of the ERG, who assisted me with this investigation.

This process resulted in a total of 417 sentences and fragments being extracted, 386 of which are marked as grammatical based on their status in Huddleston and Pullum. Of the those grammatical, 333 are also positive instances of relative constructions. I packaged up these extracted exemplars into a profile, that included labelled phenomenon records derived from the annotations. Additionally, phenomenon records for the derived classes were automatically generated and added to the profile. The exemplars were parsed with the current development version of the ERG. This version of the grammar was used so that improvements in coverage since the 1214 release could be taken advantage of and so that any improvements arising from this investigation could be fed back into the development of the grammar. The ERG was able to parse 351 of the 386 grammatical items—a coverage rate of 91%. This resulted in a total of 303 successfully analysed relative construction instances.

As a final step in the preparation of this dataset, the parsed exemplars were then treebanked by Dan Flickinger. Treebanking the generated analyses meant that, as with the Phenomenal Corpus, each exemplar would have a single high-quality analysis that Typediff could use.

The numbers of instances of each phenomenon category as present in the resultant profile are included in Table 5.8 and Table 5.9. This profile is publicly available, along with the Phenomenal Corpus profile and Typediff implementation.

5.7.2 Experimental Setup

The experimental setup for this investigation largely mirrored the one used for the investigation of the phenomena from the Phenomenal Corpus, which was described in Section 5.6.1. The profile of relative construction exemplars extracted from Huddleston and Pullum was made available from the Typediff interface, allowing the user to selectively load all exemplars of different phenomenon classes. One additional modification was made to the Typediff interface to support this investigation. This added an annotation feature which allowed the user to proceed through the set of phenomenon classes to be annotated, recording types identified to be relevant, in addition to any comments about the process they used. Also included within the information captured for each annotation was the current URL of the browser at

\[\text{Compiled from the ERG's trunk branch on 19/11/2017.}\]

\[\text{https://github.com/ned2/typediff}\]
the time the annotation was made. This was included for use during analysis of the results, as it enabled inspection of the final state of the interface that the user used to identify the annotated types.

Again, one of the goals of this exercise was for a user to proceed through the different classes of relative constructions, attempting to use the Typediff interface to identify ERG types relevant to its analysis of the target phenomenon. An additional goal this time was to observe the ways in which the fine-grained phenomenon distinctions from Huddleston and Pullum mapped onto ERG analyses, in addition to observing the Typediff features that facilitated these alignments. For this exercise, I recruited Dan Flickinger, whose role as lead developer of the ERG puts him in a good position to, firstly, validate that Typediff helps locate types that are known to be relevant to the various relative constructions and, secondly, ascertain whether the use of Typediff to navigate the types of the ERG through phenomenon characterisations from an external linguistic resource results in novel insights pertaining to the analyses of the grammar that were not previously known to the grammar engineer. In the discussion of the results of this exercise following, I refer to him as the test user.

5.7.3 Results and Observations

In this section, I describe the outcomes of this investigation. Rather than presenting details such as the grammar types identified and strategies used for all the phenomenon classes, I instead report on observations made by the test user based on his experience completing this task, which are informative from the perspective of profiling the strengths and weaknesses of the Typediff methodology.

The test user was able to use Typediff to locate types he expected to see for the target phenomena, provided that the ERG makes equivalent distinctions. In the case of supplementary relatives (largely similar to non-restrictive relatives), for example, the ERG does not make this distinction and so there was nothing for Typediff to locate. The test user also reported that when targeting the broader derived categories, such as the combined group of all relative constructions, it was harder to pin down relevant types. This is consistent with observations from the application of Typediff to the relative clause instances from the Phenomenal Corpus, which was described in Section 5.6.2. For the phenomena in this collection, the test user determined that corresponding ERG types were largely sign types (including lexical types), with some synsem types, and almost never any of the types associated with semantic representations.

In terms of the features offered by Typediff, the test user found a large number of them to be useful in triangulating upon the desired types. He made effective use of both modes of interaction—the exploration of types through positive exemplars and resultant tf-idf-based rankings, and the masking of positive exemplar types with negative exemplar types. When using the masking approach with negative exemplars, it was observed that while this feature was quite powerful, it was particularly sensitive
to the choice of negative exemplars and so careful thought had to be put into their selection. The test user also made use of the filtering capability, using it in the context of identifying types associated with the bare relative class. Another useful feature was Typediff’s ability to remove unwanted exemplars, which proved valuable in eliminating various sources of noise.

Finally, some more novel use cases for Typediff emerged from this investigation. One was that this Typediff exploration uncovered three errors from the treebanking of the exemplars, where the wrong analysis was recorded. In each case, this was discovered through the lack of an expected type appearing in the ranking when exploring a specific phenomenon class, leading to the discovery that the expected type did not appear in the input analysis in the first place. This revealed an unanticipated use for Typediff, which is that its application to a treebanked and phenomenon-annotated corpus can reveal treebanking errors in which an item is assigned an analysis that does not contain an appropriate analysis of the phenomenon it is known to have an instance of.

The second of these novel use cases was identified by the test user exploring the ERG’s treatment of the integrated relative phenomenon class from Huddleston and Pullum. The test user located four sign types licensing different forms of this construction, but could not find a type that unifies this construction as it is characterised by Huddleston and Pullum. From the perspective of an outside user of the grammar who wants to work with ERG analyses, the introduction of such a unifying type would make working with these analyses more accessible, as it would be immediately clear that the four variants are instances of a broader category that they are more likely to be familiar with. This suggests a Typediff application that involves grammar engineers identifying improvements that could be made within their type hierarchies to improve accessibility. However, a caveat to this is that, in the motivating example, there may actually be such a unifying type across the four variants of integrated relatives, but taking the form of a supertype, which is not exposed in the final yield of a complete analysis and therefore not extracted by Typediff. This application would therefore require Typediff to be extended to support the exploration of supertypes by consulting the grammar’s type hierarchy. I discuss this potential future enhancement of Typediff in Section 6.3.1.

5.8 Discussion of the Methodology

In this section, I discuss the Typediff methodology in the light of having used it to perform two different investigations, both involving the task of aligning syntactic phenomenon characterisations with ERG types. This focuses on the topics of: the utility of the methodology, identified profiles of potential Typediff users, limitations of the methodology, the applicability of the methodology to other precision grammars, and the relative construction dataset that was produced in Section 5.7.
5.8.1 Utility

The two investigations from Section 5.6 and Section 5.7 both showed Typediff to be effective at locating relevant types from the ERG. The tf-idf-weighted type rankings generated by a collection of positive phenomenon exemplars often resulted in types relevant to the target phenomenon being located within the top 10 types. This validated the hypothesis that frequencies of grammar types from positive exemplars can be combined with frequencies from across a corpus to discover grammar-internal correlates of syntactic phenomena, and also showed that it holds against a range of phenomena.

The investigation in Section 5.7 showed that a user of Typediff who was not involved in its development was able to make effective use of the various affordances provided by the tool’s interface, notably including the two modes of use identified in Section 5.4—exploring type rankings produced via positive exemplars only, and using type difference to mask irrelevant types using negative exemplars. This confirmed the attested utility of these two modes as well as a number of the other affordances described in Section 5.4.6. The decision to adopt a query-by-example approach was also validated by this investigation. The accessibility of the exemplar-driven interface meant the test user was able to make effective use of the tool with only a small amount of accompanying documentation, which contrasts with the barrier to entry associated with tools that employ formal query languages. Consistent with the argument put forth in Section 5.2, the query-by-example methodology also allowed the tool to be easily applied to two datasets containing phenomena defined through different characterisations, requiring only a set of attested exemplars to delimit the range of target phenomenon.

5.8.2 Typediff Users

Focusing specifically on applications within the grammar engineering context, three distinct profiles of potential Typediff users were identified during the course of the methodology’s development and exploration over the two different sources of syntactic phenomena. These users are as follows:

1. Grammar novice
2. Grammar expert
3. Treebank maintainer

The grammar novice is someone who is unfamiliar with a specific precision grammar but wishes to explore the analysis of one or more syntactic phenomena and possibly their underlying implementation within the grammar’s source code. For such users, it can be difficult to gain a foothold on how the grammar’s analyses cohere.
The type hierarchy of an HPSG grammar can be a complex data structure that is hard to visualise. Furthermore, types involved in the analysis of syntactic phenomena are frequently located in disparate locations within the source code. Typediff’s ability to rapidly locate types from across a grammar that are relevant to specific syntactic phenomena provides a way for users unfamiliar with a grammar to bootstrap their understanding of its analyses. Furthermore, Typediff’s exemplar-driven interface provides an accessible entry point for beginning this process, only requiring users to provide a selection of exemplar sentences.

The grammar expert is a grammar engineer involved in the development of a specific precision grammar. While such users typically have a strong familiarity with the analyses produced by the grammar and their implementation, Typediff still represents a valuable tool for two distinct applications. These uses of Typediff were identified in the process of the second investigation, described in Section 5.7, which involved a grammar engineer applying Typediff to a familiar grammar. The first application involves using Typediff to verify that types expected to be associated with specific syntactic phenomena are indeed found within the analysis of exemplars attested to be instances of the respective phenomena. The high degree of complexity found within precision grammars—often the result of interaction between phenomena—means that analyses produced by a grammar do not always have the properties that the grammar engineer intended. Grammar engineers already have a range of techniques for tackling these challenges—such as regression testing using treebanked analyses, which was described in Section 2.6.4. The aforementioned use of Typediff represents a complementary form of quality assurance to these existing techniques, allowing grammar engineers to verify that their grammar is producing the expected analysis for specific syntactic phenomena. The second use of Typediff involves the identification of candidate modifications to the grammar’s type hierarchy that do not change the behaviour of the grammar, but instead make the grammar’s analyses more accessible. The example of this that was identified in Section 5.7.3 involved making explicit the relationship between four types that all corresponded to variants of a broader phenomenon, but with that broader phenomenon not leaving a signature within the final analysis. This use of Typediff is therefore a way of leveraging a tool that improves the discoverability of syntactic phenomena, in order to identify modifications to the grammar that enhance the inherent discoverability of syntactic phenomena within the grammar.

The third user that was identified is someone who is responsible for monitoring and improving the quality of a precision-grammar derived treebank. This might often be the same person as the grammar expert, but corresponds to a different role. During the investigation described in Section 5.7, which involved relative construction exemplars extracted from Huddleston and Pullum, it was discovered that were three items in the profile had been incorrectly treebanked, which had resulted in target phenomena not manifesting within these analyses recorded in the profile. This was discovered through the process of exploring a set of positive phenomena exemplars...
and finding the lack of an expected type in the type ranking. This is a similar process to the grammar engineer verifying their analysis, however in this case, it is used for the task of identifying errors within precision grammar-derived treebanks.

### 5.8.3 Limitations

A notable limitation of the Typediff methodology is that it does not currently expose supertypes from the type hierarchy to the user. The types that are currently extracted are those present within the final analysis for a sentence produced by the grammar. In the type hierarchy, these are typically expressed as leaf types. As was discussed in Section 5.3, by inheriting from ancestor types (generally referred to as supertypes), leaf types can gain the constraints that are applied by supertypes. It is therefore common for supertypes to encode broader grammatical abstractions that are shared by multiple phenomena—syntactic or otherwise. Currently, such broader categories of phenomena expressed only through supertypes are not discoverable by Typediff. This was observed to be a limiting factor in Section 5.3, when Typediff was unable to determine whether the grammar explicitly encodes a type relating to the larger category of integrated relatives, while it could discern multiple fine-grained variants of this construction. I briefly outline some strategies for exposing supertypes within Typediff in Section 6.3.1, in the context of future work.

A challenge identified in both the explorations of Section 5.6 and Section 5.7 was the issue of noise being introduced by longer and larger numbers of exemplars. One contributing factor to this is that, as more exemplars are added, the chances of a misanalysed exemplar being included increases. Another issue is that longer and greater numbers of exemplars also increases the chances of irrelevant types appearing higher in the rankings due to other incidental phenomena occurring frequently in the exemplars. This occurs because of the corresponding increase in tf-idf weight for these types due to the increased term frequencies. This was seen in Section 5.6.2, with types relating to proper nouns unexpectedly appearing within the top 10 types when investigating passive clauses. This issue can be ameliorated by ensuring that, if all exemplars are sourced from the same domain, then treebanks used to extract corpus frequencies are also from that same domain, or alternatively that exemplars are sampled from heterogeneous domains. It may be however, that an artefact of using the tf-idf weighting function is that the number of input exemplars can only be scaled so far before noise introduced via other phenomena begins to prevent successful discovery.

The flip side of this problem is associated with the use of the inverse document frequency component of the tf-idf function, which has the effect of assigning rarer types higher weights. For many phenomena of interest, this simple heuristic works. However, when the target phenomenon is indicated through one or more types that occur frequently within a corpus, the tf-idf weighting could be counterproductive. For example, when targeting the construction of relativised predicative complements, the
type \texttt{subj_nonmc\_rule} is implicated within the ERG’s analysis, however it occurs in 36\% of sentences within the Redwoods Treebank, and so is likely to be drowned out by rarer types. This can be ameliorated by increasing the number of positive exemplars, thereby increasing the term frequency for this type. However this leaves a discoverability issue, where the user will not know in advance when to do this.

The choice to use types as the grammar components that are extracted from analyses also has an impact on the kinds of phenomena that are more readily discoverable by Typediaff. Some syntactic phenomena may only be reflected in the grammar’s analysis through the composition of a set of more basic (and therefore also more frequently used) types, rather than being realised through a single, more discoverable, phenomenon-specific type. These two characteristics that render syntactic phenomena more discoverable by Typediaff—being associated with a phenomenon-specific abstraction and having a lower frequency of occurrence—are likely related. Together, they evoke the notion of \textit{peripheral phenomena}, as contrasted with \textit{core phenomena}. This distinction, which can be found within formal and descriptive linguistics, sees core phenomena occurring frequently, constituting the regular systematic rules of a language, and peripheral phenomena being characterised by occurring less frequently and being marked by their deviation from the behaviour of the core phenomena. Rather than invoking an (often controversial) interpretation of this distinction as a sharp binary one, it can also be construed as a continuum, with core and periphery belonging to the same system (Michaelis 2012). Phenomena further towards the periphery occur less frequently and are more likely to require additional constraints for their analysis, which, in HPSG grammars, is likely to result in the introduction of phenomenon-specific types. Under this conception, the Typediaff methodology is therefore likely to perform better over syntactic phenomena that lie closer to the periphery than the core.

5.8.4 Applicability to other Precision Grammars

The Typediaff methodology was developed primarily against the ERG, however it makes no assumptions regarding the target language. Due to the interoperable nature of most \textsc{delph-in} grammars with the various processors,\textsuperscript{19} I was able to verify that the Typediaff tool could interface with a number of \textsc{delph-in} grammars. Of these, through brief experimentation, I established that it was able to perform meaningful syntactic phenomenon discovery using the Jacy Japanese grammar (Siegel \textit{et al.} 2016). I hypothesise that grammar variables more likely to affect Typediaff’s performance than the target language are: grammar size, grammar complexity, and potentially also grammar engineering style. There is almost certainly a threshold of grammar size for which grammars falling below will be unlikely to derive value from the Typediaff methodology. The harder-to-quantify dimension of grammar engineering style may be

\textsuperscript{19}See Section 2.6.2 for an enumeration of the main \textsc{delph-in} processors.
relevant in how different grammars use the type hierarchy. Some grammars leverage the type hierarchy less than others for capturing reusable abstractions, potentially resulting in fewer extracted types for Typediff to work with.

The Typediff methodology was developed to target precision grammars developed within the DELPH-IN consortium. However, besides processing software that can support real-time parsing, the only two broad requirements for applying the Typediff methodology to a precision grammar are: a means of extracting grammar components from a successful analysis, and a treebank to derive corpus frequencies for these components. In terms of the methodology described in Section 5.4, the only step that is specifically tailored to the architecture of DELPH-IN grammars is the use of types extracted from the feature structures of complete analyses. Since the use of typed feature structures is a hallmark of HPSG grammars, it is likely that the Typediff methodology could be applied directly to other precision grammars that invoke HPSG theory. A potential candidate here is the use of grammars from the CoreGram project (Müller 2015), which was discussed in Section 2.6.6.

The Typediff methodology could also be adapted to support precision grammars that do not make use of typed feature structures through the development of an alternative feature extraction algorithm. This could be done simply by extracting the identifiers from derivation tree nodes, which would amount to restricting the types extracted from a DELPH-IN grammar analysis to only include sign types. It would be desirable, however, to develop additional grammar-framework specific feature extraction methodologies that are able to extract more information from the analyses produced by precision grammars.

5.8.5 The Relative Construction Dataset

The dataset of relative constrictions from Huddleston and Pullum, which was described in Section 5.7.1, and which was created in order to apply Typediff to more fine-grained phenomenon categories, is also relevant in the broader context of my research into syntactic phenomenon discoverability. As a collection of text annotated with phenomenon instances, it is worth comparing the Huddleston and Pullum dataset (as I shall refer to it) with the Phenomenal Corpus.

In addition to targeting more fine-grained syntactic phenomenon categories than found in the Phenomenal Corpus, another differentiating property is that the Huddleston and Pullum dataset took less time and resources to construct than the Phenomenal Corpus. This was primarily due to the fact that, by leveraging an existing collection of exemplars, the presence of relative constructions was already flagged and did not have to be identified amidst unannotated corpus text. Furthermore, I

---

20Such as precision grammars that invoke the grammatical frameworks of Lexical Functional Grammar (LFG; Bresnan and Kaplan 1982), and Combinatory Categorial Grammar (CCG; Steedman 2000), which was discussed briefly in Section 2.8.5, in the context of describing CCGBank (Hockenmaier and Steedman 2007).
did not require the character spans of the instance to be identified, which was a large source of the difficulties in assessing the reliability of the annotations in the Phenomenal Corpus. More concretely, the properties of the Huddleston and Pullum dataset can be summarised by comparing them with the criteria that was stipulated for the Phenomenal Corpus in Section 4.2:

1. The text is **not** sourced from corpus data.

2. The text is **exhaustively** annotated.

3. The annotations are **not** performed at the character level.

4. The annotations are **grammatical framework-independent**.

The application of Typediff to the phenomenon instances from the Phenomenal Corpus was valuable in that it provided the opportunity to assess Typediff’s performance against broadly characterised phenomenon categories, and also because it revealed that Typediff can be prone to noise related issues when using corpus data. However, given the relative ease of construction of the Huddleston and Pullum dataset, for some tasks requiring syntactic phenomenon annotations, and which don’t require the use of corpus data, the methodology used to derive the Huddleston and Pullum dataset may be preferable.

### 5.9 Summary

In this chapter, I described the development of a methodology for aligning specific characterisations of syntactic phenomena with associated grammar components in analyses produced by precision grammar. I firstly characterised the overall approach as adopting an interactive query-by-example methodology, in which users identify exemplars of the target phenomenon to be parsed by the relevant grammar, followed by the extraction of candidate grammar components from the resulting parser output. I argued that the use of a human-in-the-loop approach allows for management of the complexity involved in aligning two different analyses of a language, while the query-by-example approach removes the circularity of needing to already know the grammar’s representation of a phenomenon and also removes the accessibility issue of needing to be familiar with a query language. By defining each syntactic phenomenon through a set of exemplars, the query-by-example approach also meets the need for the methodology to support dynamic characterisations of phenomena, which is necessary for precision grammar augmentation with a range of different descriptive grammars, that could invoke various linguistic theories within their analyses.

I then presented the Typediff methodology, which involves the extraction of grammar types from exemplar analyses, the labelling of these types with course-grained abstractions from the grammar’s type hierarchy, and the weighting of types with a
tf-idf-based function that offsets type frequencies from exemplars with corpus frequencies. The result of this process is a ranked list of types that promotes types more likely to be relevant to the target syntactic phenomenon. I also described the Typediff interface, which enables affordances such as filtering the list by type labels and also filtering the initial exemplars by the presence or absence of a specific type. I then identified two distinct, but complementary, modes of syntactic phenomenon discovery that Typediff enables, each revealing different information and responding to different kinds of exemplar groups. The first mode involves using multiple positive exemplars to promote relevant types up the ranking through the tf-idf weighting function. The second mode involves the use of negative exemplars to mask irrelevant types from positive exemplars.

I also presented a walkthrough of a browser-based implementation of the Typediff methodology, describing how it makes available to users the various features of the discovery methodology, including the two different modes of use. The Typediff tool provided a means for verifying that the type rankings produced by the extraction, labelling, and weighting components were useful in locating types relevant to target phenomena. Given the interactive nature of the Typediff methodology, it was also important to have a means of assessing its usability within a simulated context of syntactic phenomenon discovery.

After describing the Typediff methodology, I explored its application to two different collections of text annotated with syntactic phenomena. These were the Phenomenal Corpus, described in Chapter 4, and a collection of exemplars extracted from Huddleston and Pullum that were annotated with fine-grained phenomenon categories pertaining to relative constructions. In both investigations, Typediff was found to be able to successful identify relevant types from the ERG for a range of syntactic phenomena. These investigations served to validate the general approach that was adopted, as well as reveal some of its strengths and weaknesses.

Lastly, I discussed various aspects of the Typediff methodology that the investigations revealed. In addition to discussing Typediff’s efficacy at the phenomenon alignment task, I also outlined the three different candidate Typediff users that were identified: the grammar novice, the grammar expert, and the treebank maintainer. Some limitations of the methodology were also identified and discussed, focusing in particular on barriers to discovery and types of phenomena that Typediff is likely to be biased towards. I then argued that the Typediff methodology is likely applicable to HPSG grammars of other languages and speculated on strategies for extending it to grammars based on other grammatical frameworks, before finally discussing the dataset of relative constructions extracted from Huddleston and Pullum and comparing it with the Phenomenal Corpus.

In summary, the outcomes of the research described in this chapter are a novel methodology for performing syntactic phenomenon discovery within HPSG precision grammars and an accompanying reference implementation of the methodology. This methodology, called the Typediff methodology, supports the interactive alignment of
dynamic characterisations of syntactic phenomena with corresponding internal components of precision grammars. Its efficacy at performing this task was validated by applying it to two text collections annotated with instances of a range of differently characterised syntactic phenomena. In addition to confirming the utility of the Type-diff methodology, these two investigations also highlighted current limitations which, when addressed, will greatly increase the methodology’s capacity for supporting the motivating application of using precision grammars to augment descriptive grammars with syntactic phenomenon detection capabilities.
Chapter 6
Conclusion

In this thesis, I explored approaches for improving the discoverability of syntactic phenomena within HPSG precision grammars. This firstly involved the construction of a corpus annotated with syntactic phenomena, which was accompanied by an investigation into strategies for assessing the quality of these annotations. This corpus provided data needed to drive the second component of this research: the development of an interactive methodology for aligning syntactic phenomenon characterisations with associated components of precision grammars. These contributions were motivated firstly by the downstream application of using precision grammars to enhance descriptive grammars with syntactic phenomenon detection capabilities, for which phenomenon alignment and discovery techniques are necessary. Secondly, this methodology for syntactic phenomenon discovery also serves to improve the accessibility of precision grammars themselves, both for those unfamiliar with the grammars, as well as for grammar engineers engaged in their development. Both components of this research resulted in the production and public release of tangible artefacts—a corpus of syntactic phenomenon annotations and an implemented tool for syntactic phenomenon discovery within precision grammars.¹

The presentation of this research included, in Chapter 2, a treatment of background material that both contextualised this research and also established the literature it builds upon. Chapter 3 provided an overview of the resources that were used to support this research: the LinGO English Resource Grammar (ERG; Flickinger 2000, Flickinger 2011), DeepBank (Flickinger et al. 2012), and The Cambridge Grammar of the English Language (Huddleston and Pullum 2002). Chapter 4 then described the development of a corpus annotated with syntactic phenomena for supporting the development of discovery techniques, in addition to detailing an investigation into strategies for performing quality assurance of these annotations. Finally, Chapter 5 presented the methodology for syntactic phenomenon discovery I developed in this research, in addition to describing an accompanying implementation of the method-

¹https://github.com/ned2/typediff
In the rest of this chapter, I firstly summarise the two main areas of contribution of this research in Section 6.1 and Section 6.2. Then in Section 6.3, I discuss how this research could be extended in the future, before presenting some concluding observations in Section 6.4.

6.1 The Phenomenal Corpus

Chapter 4 presented the development of the Phenomenal Corpus, a small corpus of text from a Sherlock Holmes novel and a section of the WSJ part of the PTB (Marcus et al. 1993), annotated with occurrences of relative clauses, passive clauses, and complement clauses. This corpus was created in order to support the development of a methodology for syntactic phenomenon discovery. It was designed to support two possible use cases. In the first use case, it would constitute a collection of sentences which could be used as phenomenon exemplars that are suitable for input into the discovery process. In the second use case, the corpus would support precision and recall style evaluation over retrieved sentences, enabling the measurement of the performance of techniques for phenomenon retrieval. In this research, I used the corpus for the former use case, with the latter remaining a possible use case for future research. I chose to construct a new corpus on the basis that, after identifying the necessary properties for supporting the two use cases, no existing corpora that met these requirements were found to exist. The requirements that were identified are as follows: the annotated text is sourced from corpus data, each sentence is exhaustively annotated with all occurrences of phenomena covered by the corpus, annotations are performed at the character level, and the annotations are independent of formalised grammatical frameworks.

The methodology used to collect the annotations involved the text being broken up into sections, with each section being annotated by two people. Annotators were required to identify instances of the target phenomena by both marking the corresponding character spans and labelling them. A set of gold standard annotations were subsequently created through a disagreement resolution process. This proved to be a task for which annotators struggled with recall, as error analysis revealed that the most common issue for annotators was failing to identify phenomenon instances that were included within the gold standard.

A sizeable portion of Chapter 4 was dedicated towards an investigation into strategies for assessing the quality of the annotations that were collected. This resulted from difficulties using standard approaches for measuring annotator reliability. A notable challenge was that inter-annotator agreement coefficients typically used within computational linguistics do not support tasks in which annotators simultaneously segment and label text, as was the case here. I therefore investigated two alternative approaches: adapting existing measures by implicitly tokenising annotated segments
and the use of $\alpha_U$ (Krippendorff 1995), an inter-annotator agreement measure specifically designed to handle this situation, but which has seen little adoption within the computational linguistics community. These investigations showed that $\alpha_U$ produced volatile agreement scores, which seemed to be partly the result of characteristic properties of the collected annotations interacting with properties of the $\alpha_U$ coefficient. These results, combined with a lack of guidance on how to interpret $\alpha_U$ values, meant that I am unable to recommend the use of this approach for determining inter-annotator agreement in the context of annotation tasks like this one.

I therefore investigated the use of alternative approaches for assessing annotation quality. In particular, I employed a methodology that involved implicitly tokenising annotators’ segments and then annotating these tokens with IOBE tags determined by the length of annotators’ spans. I applied this firstly to derive inter-annotator agreement over these tags and then I performed precision and recall evaluation against the similarly tokenised gold standard. In both cases, this was performed on a per phenomenon basis. This IOBE tagging approach provides a finer resolution perspective on the behaviour of annotators and is able to illuminate systematic issues associated with specific locations in each annotator’s output. However, it proved to be more useful for augmenting inter-annotator agreement measurement than it was for enhancing error analysis, as most of the information yielded by the IOBE-based error analysis could be readily obtained in a more accessible form through the existing error analysis approach.

On the basis of these results, I developed recommendations for a two-tiered approach for performing quality assurance within annotation tasks such as this. Given a section of text that has been annotated by two annotators, the first tier involves the use of a traditional kappa-like coefficient, such as Fleiss’ kappa (Fleiss 1971), to determine inter-annotator agreement over implicitly tokenised annotation spans, which are also tagged with IOBE tags. This enables immediate feedback regarding possible issues in reliability. The second tier requires that these annotations be subjected to a disagreement resolution process to produce a gold standard. From this, an automated error analysis can be performed, which provides a more concrete picture of the errors made by individual annotators, the nature of these errors, and whether they are associated with any phenomena in particular. These two different tiers complement each other, with the first flagging more general issues faster, and the second tier requiring more resources, but providing a deeper picture of potential annotation issues.

Finally, I also described the packaging of the annotations within the Phenomenal Corpus so as to make them available for use in downstream research. The corpus was released as two [incr tsdb()] profiles, the data format used by [incr tsdb()] (Oepen and Flickinger 1998, Oepen and Carroll 2000), a test suite management and competence and performance profiling tool for grammar engineering, which I discussed in Section 2.6.2. I chose this format due to its support for custom phenomenon annotations, a flexible query language, and grammar implementation-agnostic design.
6.2 Syntactic Phenomenon Discovery

Chapter 5 presented my research into syntactic phenomenon discovery with precision grammars. In it, I described the development of an interactive methodology that assists a user to navigate the alignment between a given characterisation of a syntactic phenomenon and the associated internal constraints of an HPSG precision grammar. This methodology—the Typediff methodology—is motivated primarily by the application of augmenting descriptive grammars with precision grammars and their treebanks. The generative capacity of precision grammars, combined with the linguistic fidelity of their analyses, means that they can be leveraged to detect the presence of syntactic phenomena. As I observed in Section 2.7.2, this capacity could be used to produce electronic descriptive grammars with phenomenon query facilities, phenomenon-based navigation, and enhanced integration with text collections. Before a precision grammar can be used for the detection of a syntactic phenomenon, however, its corresponding signature within the representation of its output analyses must first be identified. This can be a challenging process, as the precision grammar may make different distinctions within its analysis of the target phenomenon and its analysis may even be incompatible with that implied by the target phenomenon. An example of the former situation is described by Bouma et al. (2015), who use a precision grammar to augment an electronic descriptive grammar with queries for retrieving syntactic phenomena from a range of corpora. The Typediff methodology facilitates the process of alignment necessary for such applications, by helping users navigate the challenges of aligning the target phenomenon with the internal components of the precision grammar. The Typediff methodology can also be used to improve the discoverability of syntactic phenomena for users of the grammars themselves, allowing people unfamiliar with the grammar to navigate its internal components through familiar syntactic phenomena. Likewise, it also enables grammar engineers to view the complex artefacts of precision grammars through the lens of alternative syntactic phenomenon characterisations from those found in the grammar. A third user who can also benefit from the Typediff methodology is a treebank maintainer, where Typediff can be used to identify incorrectly annotated trees within precision grammar-derived treebanks.

I argued in Section 5.2 that the methodology for facilitating syntactic phenomenon discovery should be interactive, so that users could navigate the complexity associated with the task of aligning two potentially different characterisations of syntactic phenomena. I also argued in this section for the use of a query-by-example approach, in which users interact with the tool by providing exemplars of target phenomena as input. This is desirable from the perspective of accessibility, since, as discussed in Section 2.9.3, query-by-example interfaces don’t require users to be aware of the representation of the target phenomenon, nor do they require familiarity with a query language. Furthermore, this ostensive definition of target syntactic phenomena enables the rapid characterisation of a dynamic range of phenomena, simply requiring
the user to source exemplar sentences, without needing to invoke any form of framework for constraining phenomena.

The Typediff methodology itself, which was presented in Section 5.4, involves the extraction of grammar types from the analyses the grammar produces for input exemplars. Types from the grammar’s type hierarchy were selected as features for extraction due to their being the primarily locus of linguistic abstraction within HPSG grammars. The remaining steps of the methodology involves labelling types with high-level HPSG abstractions and also weighting types with a tf-idf function I adapted for use with analyses that are represented as a bag of types. The resulting ranked list of types that this methodology produces for a given set of input exemplars is designed to promote types relevant to the target phenomenon. Users are also able to manipulate the contents of the list by applying various filtering operations that leverage type labels. I identified two distinct modes of use of the Typediff methodology, which form complementary strategies for triangulating upon grammar types likely to be associated with the target phenomenon. The first involves the specification of a sufficient number of positive items, which, through the tf-idf weighting, results in more likely candidate types being promoted to the top, while the second mode of use involves specifying at least one positive exemplar of the phenomenon and at least one negative exemplar, with the extracted types from the negative exemplar masking irrelevant types from the positive exemplar through set difference.

The implementation of the Typediff tool, which I discussed in Section 5.5, not only serves as a reference implementation of the Typediff methodology but also supported its development, providing a convenient means of gauging the utility of the ranked type lists produced through different configurations. Additionally, the tool was also used to assess the efficacy of the methodology within two scenarios that simulated the motivating context of aligning syntactic phenomenon characterisations with associated precision grammar internals. The first, which was presented in Section 5.6, involved attempting to use Typediff to identify grammar components within the ERG that are associated with the three syntactic phenomena from the Phenomenal Corpus. The second scenario, which was presented in Section 5.7, again involved the attempted alignment of syntactic phenomena with ERG internals, but this time using exemplars of relative constructions extracted from Chapter 12 of Huddleston and Pullum (2002), which were annotated with fine-grained phenomenon labels derived from distinctions that make up Huddleston and Pullum’s analysis of relative constructions.

The results of these two exercises indicated that Typediff showed efficacy at identifying relevant types for syntactic phenomena from both collections, confirming the utility of the tf-idf weighting function in locating relevant types as well as validating the utility of the various affordances provided to the user. These investigations also revealed three distinct users of precision grammars who could benefit from the resulting increase in the discoverability of their syntactic phenomena: grammar novices, grammar experts, and treebank maintainers. Additionally, these two explorations
Chapter 6: Conclusion

revealed some limitations of Typediﬀ, such as the lack of support for aligning phenomena captured through supertypes and the observation that the tf-idf rankings appear to be sensitive to noise from misanalysed items and larger numbers of exemplars which can start biasing the results towards unrelated phenomena. Additionally, based on these investigations, I also hypothesised that Typediﬀ performs better over peripheral phenomena—phenomena that occur less frequently and are more likely to be associated with phenomenon-specific types.

6.3 Future Work

In this section, I identify and discuss different directions that further work in the area of this research could take. This includes work addressing limitations of the investigations presented in this thesis, as well as extensions to this research.

6.3.1 Improvements to the Typediﬀ Methodology

There are a number of ways in which the Typediﬀ methodology could potentially be improved so as to enhance the efficacy of the phenomenon discovery process. As discussed in Section 5.8, a limitation of Typediﬀ is that it does not expose supertypes that are ancestors of types extracted from grammar analyses. This is likely necessary to capture broader phenomenon categories that are encoded in the grammar’s hierarchy through types that provide shared constraints, but which do not constitute fully realised types exposed in ﬁnal analyses. This enhancement is also necessary to support the application of using Typediﬀ to identify improvements to a grammar’s type hierarchy that improve its accessibility, which was identiﬁed in Section 5.7. The challenge for supporting this feature is that when including ancestors of types found in Typediﬀ’s rankings, a limit upon how far to climb the hierarchy is required in order to prevent noise from unrelated types swamping any signal. Without a cutoff, each type in the ranking would introduce every ancestor type up until the root type, with supertypes becoming increasingly shared across types, until the root type is introduced by every type present in the ranking. Preliminary explorations into supporting the inclusion of supertypes suggested that this is less of a concern for Typediﬀ’s type difference mode, where shared supertypes across the positive and negative exemplars are removed by the set difference operation, having the effect of removing a large amount of this noise. In the case of using only positive exemplars, there is some work to be done in restricting the supertypes that are presented to the user. One approach to solving this problem could be to extend the existing methodology to extract treebank frequencies of supertypes and then derive tf-idf weightings for each supertype in the context of a set of exemplars—enabling the construction of ranked lists of supertype. An alternative bottom-up approach could be to identify supertypes that have multiple descendants in the ranking, and which also minimise the aggregate
distance (within the type hierarchy) to all these descendants, under the assumption that closer supertypes are more likely to capture abstractions corresponding to the desired broad syntactic phenomenon categories.

Another avenue of improvement is the exploration of alternative weighting functions to apply to extracted grammar types. In this research I used the tf-idf-based function defined in Section 5.4.3, but I did not investigate the use of any other functions for increasing the relative weight of relevant types. Within the information retrieval literature, there are a number of different weighting variations involving both term frequency and inverse document frequency. For instance, term frequency scaling is useful in some contexts (Manning et al. 2008:126) and there exists a range of alternative formulations of inverse document frequency (Zobel and Moffat 1998). It may very well be the case that some variation of the tf-idf statistic performs better than the prototypical formulation for this particular task.

Within the user interface, an additional feature that could be made available to users is the ability to exclude portions of exemplar sentences from which types are extracted. Within the current methodology, users are limited to either extracting all or no types from each exemplar, however the target phenomenon will often only pertain to a smaller span of the exemplar sentence. This affordance would allow users to reduce the amount of noise within the generated type rankings that are the result of types extracted from parts of the sentence known to be irrelevant. Within the underlyingTypediff implementation (but not the interface), it is straightforward to restrict extraction of types to specific subtrees within a derivation, so this feature could be exposed via the interface by allowing users to prune irrelevant nodes, as well as restrict extraction to the yield of only a single node. Another way this feature could be incorporated would be to leverage character-based annotations of phenomenon exemplars—such as those recorded in the Phenomenal Corpus, but which were not taken advantage of in this research. By aligning character spans of input exemplar sentences to the nearest phrasal boundaries within their analyses produced by the grammar, as was performed in Section 4.6.6 when trying to improve annotations of the Phenomenal Corpus, the extraction of types could be automatically restricted to only those phrases that correspond to annotations of the phenomenon.

The Typediff interface currently includes the two different affordances of type filtering and type difference, which were described in Section 5.4.4 and Section 5.4.5 respectively. These both allow users to narrow down on the specific role that different types play within the analyses of a precision grammar. The application used to illustrate the value of the type filtering feature was that of assisting the user in discovering which types are involved in specific variants of a more general syntactic phenomenon through their presence or absence within groups of exemplars. Another strategy that could be employed for discovering the subdivision of more general phenomenon categories with respect to their type signature could be to identify groups of types that meet the dual criteria of never co-occurring within the same exemplar and also
covering the whole exemplar space of a given phenomenon. Types that meet these conditions are candidates for subdividing the target phenomenon into more specific variants of the general phenomenon. While this heuristic may not always correctly identify groups of types that correspond to subcases of a parent phenomenon, it is one that could be readily automated and incorporated into the Typediff interface as an additional signal that users can take advantage of, and which could complement the existing discovery affordances.

### 6.3.2 Exploring the Impact of Different Variables

In this research, I developed the Typediff methodology and assessed its utility primarily against one grammar, the ERG, using syntactic phenomena from the Phenomenal Corpus and from *The Cambridge Grammar of the English Language*. In order to establish Typediff’s utility in contexts outside of these alignment scenarios, as well as to gain a better understanding of the impact different variables can have on performance, Typediff should be tested against a range of dimensions. This could firstly include investigating the performance with varying numbers of positive exemplars. In Section 5.8, I discussed Typediff’s poorer performance over larger groups of exemplars, observing that this was likely an artefact of the tf-idf weighting function. It would be desirable to determine whether there is an optimal range of input exemplars through a more rigorous investigation across a range of phenomena.

It would also be desirable to evaluate the Typediff methodology against a range of phenomena beyond those explored in this research, potentially by using a similar methodology as that used in Section 5.7 to extract labelled exemplars for other phenomena—whether from *The Cambridge Grammar of the English Language* or other descriptive grammars. The Typediff methodology should also be tested against a range of other precision grammars, in particular, a selection of grammars targeting typologically diverse languages, in order to explore how abstractions captured in the type hierarchies of these grammars impact phenomenon discovery. It would also be desirable to gain a sense of the impact of the size and complexity of grammars, as it is likely that Typediff requires a certain amount of analytic coverage and complexity within a grammar’s type hierarchy in order to be of value. While it would be difficult to isolate these variables from other confounding factors, by evaluating Typediff over a range of different grammars that vary in size, a picture of the impact that these variables might have could emerge. The DELPH-IN consortium’s catalogue of implemented grammars provides a reasonable selection of languages and grammar sizes, which this evaluation could be performed over. Further horizontal expansion could involve the application of Typediff to grammars from other precision grammar projects. A natural extension would be to grammars built within the CoreGram

---

2 Thanks to one of my anonymous examiners for the suggestion of this Typediff enhancement.

3 [http://moin.delph-in.net/GrammarCatalogue](http://moin.delph-in.net/GrammarCatalogue)
The task of evaluating the utility of a methodology such as Typediff is a challenging one due to the complexity involved in aligning syntactic phenomenon characterisations with implemented grammars, in addition to the inherently subjective definition that the term utility must take in the context of an interactive tool used for discovery. However, any concerted exploration that applies Typediff against a range of variables such as those outlined would benefit from at least an operationalisation of the notion of utility in this context, if not the development of more objective measures. Additionally, when exploring modifications to components of the methodology, such as changing the weighting function, it would certainly be valuable to establish a set of regression tests that, for each group of exemplars delimiting a specific phenomenon, records the resulting type ranking, along with the position of relevant types. This would enable rapid feedback regarding the impact of changes that cause the generated type rankings to change.

6.3.3 In Situ Evaluation

The tasks that were performed in Section 5.6 and Section 5.7 provided a simulation of the motivating task of aligning syntactic phenomenon characterisations with precision grammar internals. Phenomenon discovery, however, is ancillary to the task of phenomenon detection, which is necessary to support the descriptive grammar enhancements which were identified in Section 2.7.2. One of these enhancements is the augmentation of descriptive grammars with the capacity to retrieve additional exemplars of phenomena from within a corpus. To evaluate the utility of the Typediff methodology in the context of realising this feature of electronic descriptive grammars, an in situ evaluation should be performed. An experimental setup that would support this could take a similar form to that of Bouma et al. (2015), who annotate both prose text and exemplars from a descriptive grammar with treebank queries that retrieve additional examples pertinent to the phenomena being labelled from supplementary corpora. Typediff could be provided to annotators developing the queries, who, in the task described by Bouma et al. (2015), reported difficulties in performing the necessary alignment between descriptive grammar analyses and treebank representation. Success in this task would be indicated by annotators reporting a more efficient query construction process when Typediff is used. Also of importance here would be the assessment of the performance of the resulting queries in terms of their precision and recall over gold standard phenomenon annotated treebanks.

In addition to evaluating phenomenon detection (and not just discovery), another key difference between this experiment and the use of Typediff for exploring the Phenomenal Corpus, which was described in Section 5.6, is that an extensive source of carefully pre-annotated phenomenon exemplars would not be available to annotators. This would then present the opportunity to assess the utility of the Typediff dis-
covery methodology under more realistic conditions, since the need to construct the Phenomenal Corpus in the first place was premised on the general lack of exactly this type of data. Annotators would instead be supplied with smaller selection for each phenomenon, such as might be found explicitly labelled in a descriptive grammar, or they could potentially be required to use other methods for bootstrapping their own small collection of positive instances of phenomena.

A query development and annotation task such as this could also be enhanced via the development of a treebank query tool that supports more complex queries over the syntactic derivations produced by DELPH-IN grammars. Currently, querying these structure with a treebank querying tool such as Fangorn (Ghodke and Bird 2012)—which is incorporated in the Typediff interface—is limited to queries whose constraints are defined by relationships across node identifiers. As argued by Bender et al. (2012), it would be desirable to support queries that can leverage more information within DELPH-IN derivations. It would be useful, for example, to constrain feature values—like HEAD and CAT—within feature structures at specific nodes in the derivation tree being queried. Further work here is required to either produce a new treebank querying tool that supports these constraints or to devise a strategy for transforming DELPH-IN analyses into a format compatible with existing treebank querying tools. Also of potential utility is the incorporation of techniques that enable the querying of semantic representations produced by DELPH-IN grammars, such as that of Kouylekov and Oepen (2014).

The aforementioned evaluation is geared towards the motivating application of descriptive grammar augmentation with precision grammars. The other motivating application for the syntactic phenomenon discovery techniques developed in this research is that of improved accessibility of the analyses within precision grammars themselves. The exploration of Typediff that was presented in Section 5.7.2, which involved the maintainer of the ERG using Typediff to explore exemplars from Huddleston and Pullum (2002) that were annotated with fine-grained phenomenon labels, indicated that the Typediff methodology shows potential for this kind of application, with Typediff both yielding expected results and enabling novel insights. A more in-depth evaluation of this particular use case of Typediff is required in order to properly assess its utility for this application. This would need to involve a number of users, and ideally, these users would include more diversity of experience with the target grammar. This would mean that, in addition to expert users of the grammar, the ability of Typediff to improve grammar accessibility for more casual users of the grammar—such as grammar engineers unfamiliar with the target grammar or linguists less experienced with grammar engineering—could also be assessed.

### 6.3.4 Unsupervised Detection of Phenomenon Categories

During development of the Typediff methodology I observed a number of occasions where, after a sufficient number of positive exemplars of a target phenomenon
had been added, the type rankings would stabilise. At this point, not only would many of the top-ranked types be relevant to the target phenomenon, but adding additional exemplars would not alter the relative ordering of the top 10 types. Another avenue of research could involve exploring whether this observed behaviour might be exploited for the purpose of the unsupervised detection of phenomenon distinctions made by precision grammars within a group of exemplars. Given a group of exemplars that correspond to a phenomenon characterisation, a clustering algorithm could potentially be developed that subsamples exemplars into clusters, which are then iteratively expanded or reduced until local convergence points are reached. Similarity measures for comparing exemplars could also be developed for driving clustering approaches. These could employ weighted vector space representations of exemplar analyses, such as cosine similarity (Jurafsky and Martin 2009:769), or they could continue to work with the type ranking model, using techniques for comparing ranked lists, such as rank-bias overlap (Webber et al. 2010). A potential proof of concept of such unsupervised approaches could be to target the two different relative construction datasets created in this research, which, as my investigations revealed, map onto a range of finer-grained distinctions within the ERG.

6.3.5 Other Applications of Typediff

The Typediff methodology could potentially be applied to other applications outside of the areas of electronic grammaticography and grammar engineering. In the areas of genre and authorship detection, which are types of text classification tasks, the distribution of syntactic constructions within a text has been identified as being associated with genre and author-specific styles and has therefore been employed as features within approaches used to perform this type of classification (Efstathios 2009). The high degree of linguistic fidelity and fine-grained syntactic distinctions that are characteristic of precision grammars means that they represent a compelling source of features for classification tasks that revolve around stylistics. In particular, the use of sign types produced by DELPH-IN grammars during the analysis of a sentence could be explored. This feature extraction process itself would not require the use of Typediff, since all that is needed is extraction of the node labels from the derivation trees. However, Typediff could play a role in performing manual feature analysis and tuning. The same features and affordances that were designed for users to be able to associate specific types with a target phenomenon, could also be used to identify types, or categories of types that are associated with a particular target genre or author. The notable features being that Typediff allows users to interactively parse collections of sentences and explore the distributions of types (whether sign or otherwise) and compare their distribution across different categories of labels, while also providing nuanced controls that allow users to filter against different criteria. More generally, this use of Typediff for such feature tuning could be potentially helpful for any task where the use of syntactic features is beneficial.
6.4 Concluding Remarks

This research, while being primary motivated by the goal of augmenting descriptive grammars with precision grammars, has implications for a number of research focal points. I developed the Phenomenal Corpus in response to the lack of existing resources with the properties that I identified as being required for supporting the investigation of syntactic phenomenon discovery techniques. With the existence of this corpus, there is now a resource that others investigating this space can also leverage. A further application that a rigorously annotated corpus of syntactic phenomena such as this could be used for is that of construction-focused parser evaluation, examples of which are Rimell et al. (2009) and Bender et al. (2011b), and which were described in Section 2.8.6. The methodology I used to construct the corpus and, in particular, the recommendations for quality assurance of the annotations also constitute a reference that can be consulted for producing further corpora annotated with additional syntactic phenomena. A notable component of the work involved in producing this corpus was the investigation of a suitable approach for gauging annotator reliability through the lens of inter-annotator agreement. I explored the use of the $\alpha_U$ coefficient, which has the requisite property of supporting annotation tasks in which annotators both segment and label continuous data. Artstein and Poesio (2008) identified this coefficient as a candidate strategy for assessing annotator reliability in computational linguistics tasks that exhibit this property, however, it has seen little adoption within the field. My investigation revealed issues with the use of $\alpha_U$ for assessing annotator reliability in this task, with it producing results that were both volatile and difficult to interpret, and as such, I am unable to recommend it, at least for annotation tasks resembling the one used to construct the Phenomenal Corpus.

My research into syntactic phenomenon discovery techniques resulted in the development of a novel method for interactively aligning dynamic characterisations of syntactic phenomena with corresponding components of HPSG grammars. This was also accompanied by an implementation of this methodology as an open source software package. The primary motivation for the development of this methodology is the enhancement descriptive grammars with features enabled by the capacity of precision grammars to generate linguistically accurate analyses of text. Using precision grammars for phenomenon detection, however, first requires supporting techniques for managing the complexities involved in aligning two different theories of the same language. The Typediff methodology is therefore valuable in the context of tasks such as that presented in Bouma et al. (2015), where it can be used to overcome difficulties in the process of discovering how syntactic phenomena are analysed and represented within a precision grammar, thereby facilitating the construction of queries that retrieve examples of syntactic phenomena from treebanks produced by the grammar.

The role that Typediff can play in augmenting descriptive grammars with links into accompanying corpora makes this research of notable relevance to the area of electronic grammaticography. In particular, my use of precision grammars and their
derived treebanks for realising components of this outcome, along with the work of Bouma et al. (2015), adds support for Bender et al.’s (2012) argument for enhancing descriptive grammars with precision grammar derived-treebanks. This synergy between descriptive and precision grammars also represents a compelling strategy for realising Thieberger’s (2008) exhortation to improve the integration of descriptive grammars with their text collections, by making the primary data their analyses are based on more accessible from within the descriptive grammar. The inclusion of links to instances of syntactic phenomena also constitutes a complementary form of augmentation to that seen in Musgrave and Thieberger’s (2012) conversion of an extensively cross-referenced descriptive grammar into a hyperlinked electronic descriptive grammar. The methodology used by Musgrave and Thieberger (2012) to produce these hyperlinks relies on the existence of manually created cross-references between description and text collection instances. Once treebank queries are generated through a process such as Bouma et al.’s (2015), on the other hand, the capacity for retrieval extends dynamically to cover any similarly treebanked text. The caveat being, however, that this is premised on the existence of a precision grammar implementation and appropriate corpora, which is often not practical for under-resourced languages.

More generally, the research presented in this thesis, and in particular, the Typediff methodology for syntactic phenomenon discovery, can be seen as a response to Bender et al.’s (2012) call for improving the accessibility of precision grammars. The use of phenomenon discovery techniques to improve the accessibility of precision grammars has broader implications beyond their augmentation of descriptive grammars, however. Tools that implement Typediff-like methodologies have the potential to improve the utility of precision grammars from a range of perspectives. Maintainers of precision grammar derived-treebanks can employ such tools for error detection, in order to help with quality assurance. Grammar engineers can use them to verify expected analyses as well as navigate precision grammars by phenomenon characterisations that differ from the analyses within their grammars, potentially supporting improvements to the grammar’s abstractions that further improve their accessibility. Finally, from the perspective of people who are unfamiliar with the formalism used by the grammar, or just unfamiliar with the analyses of a particular grammar, the exploration of a precision grammar through exemplars of familiar syntactic phenomena provides an intuitive interface that can help linguists and other researchers decode the linguistic knowledge implicit within precision grammars.
Appendix A

Grammatical Phenomenon Reference

Introduction

This document serves as a typological reference for the linguistic phenomena chosen to be included in the phenomena corpus used in this project. For each phenomenon, a brief overview of its function is presented as well as an outline of the range of variability it exhibits across languages of the world. Instead of trying to provide an exhaustive account of the different manifestations of each phenomenon across all documented languages—a task that rapidly runs headlong into thorny typological issues—this document aims to delimit the scope of each phenomenon such that a) the chosen definition captures what is considered “interesting” about the phenomenon, b) a sizeable number of languages are included, and c) establishing whether a given item contains an instance of a phenomenon is a tractable task in the context of the manual annotation of a large number of items. To this end, each section also includes a set of criteria for the inclusion of a candidate item as an instance of the phenomenon.

Passive Voice

The passive construction (or passive voice) is a construction found across many languages of the world. Broadly speaking, it is a valence-changing construction that alters the way thematic roles are mapped onto grammatical roles so as to foreground the referent that is the patient. From an information structure perspective, it is similar to object topicalisation and object left-dislocation in that these constructions all serve to foreground the patient. Another reason speakers may choose to use the

\footnote{This overview of the passive construction is largely based on Siewierska (2011) and Keenan (1985a).}
passive is when wanting to avoid referring to the agent or if the identity of the agent is unknown.

In a prototypical passive construction, the grammatical subject of a clause expresses the patient role—contrasting with the active form, in which the grammatical subject expresses the role of agent—and the agent is either realised as an oblique phrase or omitted entirely. While this is thought of as a prototypical passive construction, in many languages that have a passive construction, it is not possible to express the the argument that corresponds to the subject in the active variant.

(57a) shows a transitive active sentence in which the patient, the Higgs Boson, is the grammatical object, whereas in the passive form, illustrated by (57b), it is the grammatical subject, and agent is found in the oblique prepositional phrase by scientists. (57c) is the alternate passive form in which the agent has been completely omitted.

(57)  
   a. Scientists discovered the Higgs Boson.  
   b. The Higgs Boson was discovered by scientists.  
   c. The Higgs Boson was discovered.

Dissecting the Passive

The passive construction can be broken up into two distinct components. The first, agent demotion, sees the agent demoted from subject to an oblique, or completely omitted, having the effect of reducing the valency of the predicate by one. The second, patient promotion, promotes the patient from direct object to subject. Clauses that feature both of these components (such as those in (57b) and (57c)) are referred to as personal passives. There are also passive clauses, referred to as impersonal passives, that feature only agent demotion. A language that contains impersonal passives is Kannada, as illustrated by (58) (Sridhar 1990:215), where (58a) is the active form, and (58b) is the corresponding impersonal passive form. Some languages have only the impersonal passive. It is also possible for languages to create passive constructions out of intransitive verbs, resulting in a zero-place predicate. These are necessarily impersonal passives as they will have no grammatical subject.

(58)  
   a. ya:ro:  i:  nirNayav-annu khaNDisidaru  
      someone this resolution-ACC denounce.PST.3PL.HUM  
      “Someone denounced this resolution.” [kan]  
   b.  i:  nirNayav-annu khaNDisala:yitu  
      this resolution-ACC denounce.INF.BECOME.3N  
      “This resolution was denounced.” [kan]

A further type of passive that is identified in the literature, is that of the indirect passive found in Japanese (Washio (1987) and Oshima (2003)). The indirect passive
is distinctive in that the grammatical subject is not an object of the main verb, but rather an undergoer or experiencer of the event described by the main verb as well as being in some relation with one of the arguments of the main verb. This is illustrated by (59) where (59a) is a canonical passive (sometimes referred to as a direct passive) and (59b) and (59c) are examples of the indirect passive. The indirect passive is often referred to as an adversative passive or just adversative, as this construction often (but not always) implies an adverse effect on the undergoer. The indirect passive can be formed from both transitive and intransitive verbs as illustrated by (59b) and (59c) respectively.

(59) a. sensei-ga seito-ni ker-are-ta  
teacher-NOM pupil-by kick-PASS-PST  
“The teacher was kicked by the pupil” [jpn]
b. sensei-ga seito-ni kuruma-o ker-are-ta  
teacher-NOM pupil-by car-ACC kick-PASS-PST  
“The teacher is such that his car was kicked by the pupil” [jpn]
c. sensei-ga seito-ni nak-are-ta  
teacher-NOM pupil-by cry-PASS-PST  
“The teacher is such that his pupil cried” [jpn]

Due to the undergoer role that the grammatical subject plays in the indirect passive, it is not possible to analyse the indirect passive in terms of agent demotion and patient promotion. This is evidenced by the fact that the indirect passive has no active counterpart. While the indirect passive is not readily analysable as a standard passive construction, the direct and indirect passive are formally equivalent in Japanese, and it has been argued that they are in fact manifestations of the same construction and that the differences in interpretation between the two arise from contextual cues and conversational implicature (Washio (1987) and Oshima (2003)).

The strategies employed by the world’s languages to encode the passive construction can be divided up into the two different categories of analytic and synthetic passives. In an analytic passive, a participle form of the main verb is used along with an auxiliary. This is how the passive is constructed in English, as illustrated in (57b) and (57c)). With synthetic passives on the other hand, the main verb is marked via inflectional morphology. This is how passives are formed in Swahili, as illustrated by (60) (Ashton 1947:224), which uses the suffix -w- to mark the passive construction.

(60) a. Hamisi a-li-pik-a chakula  
Hamisi 3SG-PST-cook-IND food  
“Hamisi cooked the/some food.” [swa]
b. chakula ki-li-pik-w-a (na Hamisi)  
food 3SG-PST-cook-PASS-IND (by Hamisi)  
“The food was cooked (by Hamisi).” [swa]
Criteria for Inclusion

In this study both personal and impersonal passives were included. This means that for a clause to be counted as a passive construction it must at a minimum involve the process of patient promotion. The anticausative construction, which can be similar in appearance to the passive construction but does not involve an agent, was excluded. Also excluded, were indirect passives from Japanese which cannot be converted into an equivalent active form.

Imperative Sentence Type

This section deals with imperatives, the types of sentences that convey some kind of command, instruction or plea. Conveying these kinds of speech acts are so important to social interaction that it has been suggested that all languages possess some kind of dedicated form for achieving them (Sadock and Zwicky 1985). It is however a difficult task to tease apart imperatives from other closely related sentence types, as it is possible for languages to use the same dedicated imperative form to realise multiple imperative-like speech acts, while it is also possible for non-dedicated imperative forms to achieve specific imperative or imperative-like speech acts.

Here I will define some broad categories of imperative and imperative-like sentence types in the same way as Sadock and Zwicky (1985). All the sentence types illustrated in (61) share in common that they are used by a speaker to express a desire about a future state of affairs. The optative—which is generally considered to be a separate construction—can be seen as contrasting with the imperative and hortative in that it is used to indicate a general wish or preference for a state of affairs to come about whereas the hortative and imperative are used when appealing to an audience to assist in bringing about the desired future state of affairs. If the audience is in control of the state of affairs then this is the imperative. Otherwise, the construction is referred to as the hortative, the function of which is to provide encouragement towards bringing out the desired state of affairs.

(61) a. May you live long and prosper. \textit{optative}
b. (You) live long and prosper. \textit{imperative}
c. Let’s live long and prosper. \textit{hortative}
d. Let them live long and prosper. \textit{hortative}
e. Don’t (you) live long and prosper. \textit{prohibitive}

Imperative and imperative-like clauses can also be characterised by which participant can be the agent of the prescribed action. In what I have identified as the

\footnote{This overview is based on Sadock and Zwicky (1985), van der Auwera et al. (2011) and Birjulin and Xrakovskij (2001).}
canonical imperative this is generally the second person singular or second person plural. In the case of hortatives, this is typically the first person plural or third person singular or plural.

A problem with dividing imperative-like sentences up into categories such as the imperative and hortative is that while they are separate constructions in some languages such as English, they often overlap. For instance, as illustrated by (62), in Tatar, both the third person singular and plural imperative verb forms are homogeneous with the second person singular and plural imperative verb forms (Birjulin and Xrakovskij 2001:20). Even in English there is some ambiguity between these two categories. For instance, *Let him go* could be said to have the form of a hortative, however when issued towards an addressee that is in direct control of the freedom of the referent of the pronoun *him*, it clearly has imperative force. This ambiguity can also be seen in (61d), where once a scenario is constructed in which the utterance is being issued towards someone who has the power to decree whether the referents of the pronoun *them* do in fact live long and prosper, it also takes on an imperative interpretation.

(62) a. bar-∅
   go-2sg
   “Go.” [tat]

b. bar-syn-∅
   go-3-sg
   “Let [him] go.” [tat]

Languages frequently have alternate ways of conveying imperatives, often so as to provide a more polite form of imperative or for use in other specific conversational contexts. These indirect speech-acts use may make use of an entirely separate form or construction from the imperative but gain their imperative force via conversational implicature. This can be seen in (63a) and (63b) which make use of declarative and interrogative sentences respectively in order to achieve an imperative effect.

(63) a. You will take a seat.

b. Can you take a seat?

Another example of this is that of a colloquial Japanese imperative that uses the past participle of the verb. As shown in (64), taken literally it forms a declarative sentence, but in the correct context and combined with high stressed pitch in the final syllable of the verb, it is interpreted as an imperative (Svähn 2010:1). For simplicity, these kinds of imperative speech acts that require pragmatic effects to gain their imperative force were also excluded.

(64) *Doi-ta, doi-ta!*
   move.away-PST move.away-PST
   “Get out of my way!” (lit.) ‘(You) got out of my way!’ [jap]
Imperative sentences can also be further subdivided by their polarity. For example, prohibitives convey a desire that a particular state of affairs does not come to pass. However including these types of clauses in the same category as imperatives can potentially be misleading since some languages do not use the imperative form to construct prohibitives. Additionally, many languages use a different method of indicating negation in a prohibitive construction than would be used in the corresponding negative declarative clause. For these complicating reasons, prohibitives were excluded from this study.

An important way in which the imperative varies across languages is whether the language possesses a dedicated imperative form. This can either be a morphological form, where there is a specific verb form used only for the purpose of indicating the imperative, or it can be a syntactically dedicated form, where the particular arrangement of the constituents in the clause is used only for the purpose of indicating the imperative. In English the imperative is not morphologically dedicated, as the verb takes the form of a bare infinitive, however the entire construction—a bare infinitive verb in initial position without a subject—is syntactically dedicated. An example of a language that does not possess a dedicated imperative is Nunggubuyu, which makes use of the future tense to express imperatives (Heath 1984).

It is also possible for there to exist multiple imperative paradigms simultaneously. Lingala has a dedicated imperative inflection for the second singular, but has use of an entirely separate subjunctive-based paradigm for all other persons (including also the second singular) which can be used for the purpose of making requests in addition to giving orders (Meeuwis 1998). The difficulty in such cases lies in deciding, for the purposes of this study, whether such alternative paradigms will be counted as the imperative.

**Imperative Strategies**

The ways in which languages of the world realise imperative speech acts vary considerably, however there are some significant similarities. The strategies for realising the imperative can be divided up into two broad categories: those which use a morphological imperative and possess one or more imperative verbal paradigms and those which use only syntactic and prosodic strategies to encode the imperative. Those that lie in the latter category are exemplified by isolating languages of South East Asia and West Africa (Birjulin and Xrakovskij 2001).

Of those languages that possess morphological imperatives, Sadock and Zwicky (1985) provide some common features. One is that quite common is a reduction in affixes on the main verb, resulting in a verb form with fewer affixes than would be found in the equivalent declarative sentence. Many languages employ a strategy of removing all verbal affixes to indicate imperative constructions and it is common for tense and aspect distinctions to be dropped, and to a slightly lesser extent verbal conjugations. Another common feature of imperative strategies is for subject pronouns
and/or subject agreement affixes to be dropped.

A large number of languages possess dedicated imperative verb forms for the second person. These languages vary as to whether they possess imperative forms for both second person singular and plural, just one of these, or do not distinguish between the two. Of these strategies, van der Auwera et al. (2011) gives the following numbers for a survey of 547 languages: 292 languages have dedicated forms for both second person singular and plural imperatives, 42 have dedicated forms for second person singular imperatives but not for plural, 2 have no dedicated form for second person singular imperative but do for plural, 89 have dedicated forms for second person imperatives but do not distinguish between them, and 122 do not possess dedicated imperative forms for second person imperatives.

Criteria for Inclusion

In this study, attention was restricted to clause forms dedicated to serving an imperative function, which is to say that they are uttered when appealing to an audience to bring about a desired state of affairs and the audience is deemed to be control of this state of affairs. However, where a language uses the same form for both imperatives and other imperative-like constructions, such as hortatives, these will also be included. Another way this can be thought of is that if the clause has a literal meaning that is not imperative, then it will be excluded. As already mentioned, negative imperatives (prohibitives) were excluded due to the complexity they introduce. Also ignored, were imperative types that gain their imperative force via conversational implicature, unless this is the primary means the language has for realising the imperative.

Interrogative Sentence Type

This section deals with sentences whose purpose is to solicit information from the addressee, generally referred to as interrogative sentences. By looking at the types of responses that interrogatives solicit, they can be broken up into three subtypes:

(65) a. Do you like coffee?  
    b. Do you like coffee, or tea?  
    c. What kind of coffee do you like?

Polar, or yes-no questions as exemplified by (65a) require a response which indicates either the truth or falsity of the queried proposition. In other words, the available responses correspond to the set containing the equivalents of yes and no.

---

3This overview is largely taken from Sadock and Zwicky (1985).
There are many different strategies used by languages in order to encode polar questions, but by far the most common is that of final rising intonation, most often (but not necessarily) used in conjunction with some other feature.

Alternative questions are similar to polar questions, but differ in that a response indicating either true or false would not be acceptable. Instead, the set of available responses are found within the question itself, as in (65b), in which the addressee could respond with either coffee or tea. An implication of the alternative interrogative is that the speaker thinks the distinct propositions are mutually exclusive.

Wh-questions or content/information questions, on the other hand, do not limit the available responses to simple true or false statements or to a set list of possible responses; instead they are open-ended in the kind of response they solicit, as exemplified by (65b). They are referred to wh-questions because they generally make use of specific interrogative markers, which in English almost all begin with wh. In some languages polar questions and wh-questions are formed using the same syntactic construction, in others they are not.

There is also a particular type of interrogative construction, the tag question (as illustrated by (66)), which can be thought of as a construction that forms polar interrogatives out of declarative sentences by adding an interrogative fragment, single word or fixed phrase.

\[(66)\]  
\[a. \text{ Coffee is fantastic, isn’t it?} \]
\[b. \text{ Tu aim-es la musique, n’est-ce pas} \]
\[\text{you like-2SG the.FEM music, NEG’be.3SG-it NEG} \]
\[\text{“You like music, don’t you?” [fra]} \]

Criteria for Inclusion

In this study, all the different kinds of interrogatives above were included: polar, alternative, content questions and tag questions. One particular type of interrogative excluded are those that occur as subordinate clauses, such as indirect questions like those found in (67). This was excluded on the basis that of particular interest here is the way languages encode interrogative force. While these subordinate clauses do involve interrogatives, the matrix clauses they belong to do not possess interrogative force.

\[(67)\]  
\[a. \text{ The researcher asked whose coffee he had spilled.} \]
\[b. \text{ The technician wondered how much the equipment was worth.} \]

Complement Clauses

Complement clauses are subordinate clauses that are characterised by the two salient properties that they function as an argument to a predicate of a higher clause.
and that they possess some or all of the constituent structure of a clause. For example, the clause *that you are a light sleeper* functions as the object of the English verb *understand* in (68a) and as the subject of the English verb *frustrate* in (68b).

(68) a. I understand that you are a light sleeper.
    b. That you are a light sleeper frustrates me.

Complement clauses may take the form of a sentence-like clause, as in (69a)—in which case they tend to represent a fact or proposition—or they may be reduced in some way, as in (69b) and (69c), in which the subject is omitted and the verb takes a non-finite form.

(69) a. I remembered (that) I don’t like durian.
    b. I remembered to feed the trolls.
    c. I remembered seeing her.
    d. I remembered his singing the song.

Languages vary as to which grammatical relation a complement clauses can function as. For instance, Modern Hebrew disallows complement clauses acting as the subject of transitive clauses and languages such as White Hmong, Tariana, Goemai and Kambera only permit complement clauses to function as direct objects Dixon et al. (2008).

Across all languages that possess complement clauses, it is found that not all verbs license complement clauses as arguments. Indeed the set of such verbs in a language is generally significantly reduced. As is shown in (70a) and (70b), the English verb *eat* cannot take a complement clause as any of its arguments. Verbs may also restrict which argument the complement clause can function as, as is the case with the English verb *annoy*, which only permits complement clauses as subjects, as shown by (70c) and (70d). Furthermore, verbs may require specific complement clause forms, as illustrated by the English verb *know* in (70e) and (70f), which requires a complete sentence-like clause.

(70) a. *I eat that we lost.
    b. *That we won I lost.
    c. That we lost annoys me.
    d. *I annoy that we lost.
    e. I believe that you are sublime.
    f. *I believe to fall in love.

Criteria for Inclusion

In accordance with Dixon et al. (2008), the following criteria was used as the basis for inclusion in this corpus:
1. The constituent possesses the internal structure of a clause, with it requiring a predicate at a minimum, and any arguments of this predicate are marked in the same way as they would be in a main clause.

2. The constituent functions as an argument of a predicate in a higher clause.

Consequentially, languages that do not possess complement clauses and that rely on other complementation strategies were excluded. This also means that gerundals, such as in (69d), were excluded.

A potential cause for confusion may be found in English clauses such as his sinking the Titanic, which may at first appear to be a nominal phrase due to the superficial similarity with the NP his sinking of the Titanic. These can be teased apart, however, by noticing that in the case of the former, sinking is a gerundal verb which takes the Titanic as an argument, whereas in the latter case sinking is a deverbal noun (or nominalised verb) modified by the Titanic. A useful test in such cases is whether the constituent will accept an adverb, as this will not be possible if the constituent is an NP.

(71) a. I enjoyed his sinking the Titanic. complement clause
    b. I enjoyed his sinking of the Titanic. noun phrase
    c. I enjoyed his sinking the Titanic slowly.
    d. *I enjoyed his sinking of the Titanic slowly.

Some further complexities involved in delineating the scope of this phenomenon are illustrated by the examples in (72). While (72a) clearly meets our criteria for complement clause status, (72b) and (72c) are interesting in that the constituents in question are both clearly complements and are both clearly clauses, however they fail to meet the criterion of functioning as an argument of a higher clause. In the case of (72b) the clause is a complement of the noun attempts and it is the entire prepositional phrase at my attempts to go to the shops that is the argument to the verb laughed.

(72) a. I love to go to the shops.
    b. She laughed at my attempts to go to the shops.
    c. I am happy to go to the shops.
    d. I drink when going to the shops.

In the case of (72c) the clause is a complement of the adjective. Here it could potentially be argued that the subordinate clause is functioning as an argument to the copula predicate is happy. In order to simplify identification of this phenomenon and improve cross-linguistic comparison, it was stipulated that the predicate the subordinate clause is a complement of must be a verbal predicate. The revised criteria for complement clause status is thus presented below.
1. The constituent possess the internal structure of a clause, with it requiring a predicate at a minimum, and any arguments of this predicate are marked in the same way as they would be in a main clause.

2. The constituent functions as an argument of a verbal predicate in a higher clause.

**Relative Clauses**

Relative clauses are a specific type of subordinate clause which serve to restrict or elaborate on one of the referents in the higher clause. Additionally, one of the arguments of the subordinate clause shares its reference with the constituent on which the subordinate clause is dependant on—typically a nominal.\(^4\)

(73)  

| a. I see the tree **that** has a thick trunk.  
| b. He chose the boy **who** eats worms.  
| c. He chose the boy **whom** I saw.  
| d. You know that band **whose** tunes I like? |

As can be seen in (73), the head noun can be linked to either the subject, as in (73a) and (73b), the direct object position, as in (73c), or some other kind of argument, such as a genitive, as shown in (73d). Languages differ in which argument roles can be relativised, however there are cross-linguistic generalisations that can be made Keenan and Comrie (1977).

One strategy for linking relative clauses to their nominals is via the use of a relative pronoun, as illustrated in (73). This is only possible for languages with post-nominal relative clauses (Keenan 1985b). Such languages often permit dropping the relative pronoun in what is referred to as a *reduced relative clause*.

Relative clauses can also be restrictive or non-restrictive. A restrictive relative clause serves to refine and further delimit the reference of the head noun it modifies. A non-restrictive relative clause does not restrict the reference, but instead elaborates upon the information being provided about the referent picked out by the head noun. In standard English orthography, the distinction between the two is made by placing commas around the non-restrictive relative clause, as illustrated in (74).

(74)  

| a. The dog **which** has a long tail sat down.  
| b. Their house, **which** had a lovely fence, sat on a hill. |

---

\(^4\)This is overview is largely from Andrews (1985).
Criteria for Inclusion

A clause will be considered an instance of this phenomenon if it is a subordinate clause modifying a noun or noun phrase, and which contains an argument that shares its reference with the nominal it modifies. Both restrictive and non-restrictive relative clauses were included, as were reduced relative clauses.
Appendix B
Grammatical Phenomenon Annotation Guidelines

Introduction

These guidelines were prepared by Ned Letcher for the development of a corpus of linguistic phenomena to be used for the purpose of investigating techniques for the automatic detection of linguistic phenomena in precision grammars. This project is based at The University of Melbourne and University Washington. For more information please contact Ned Letcher at ned@nedletcher.net.

Overview

Firstly, thank you for taking the time to annotate a portion of our linguistic phenomenon corpus! The main aim of this corpus is for it to drive the development of techniques for the automatic detection of linguistic phenomena within precision grammars. We hope that such techniques will increase the utility of both precision grammars and their associated treebanks as linguistic resources—you are working towards a better future!

In this annotation task you will be asked to annotate utterances with linguistic phenomena that occur within them. By linguistic phenomena we mean interesting constructions or features that linguists concern themselves with when documenting languages. Below you will find general instructions on how to perform the annotation as well as specific instructions for each phenomenon.
Annotation Instructions

For each item, you need to annotate all instances of the phenomena you have been instructed to annotate. This means that if an item is not annotated with a particular phenomenon, it is implied that that is does not exhibit that phenomenon. We recommend doing each phenomenon in a separate pass, as it can be hard to keep track of the criteria for all the phenomena simultaneously. This means that after reading the remainder of this section you can read the instructions for each phenomenon as you get to it. Note that for the task you have been assigned, you may not be required to annotate all the phenomena listed in Section B.

How to Annotate

For each instance of a phenomenon, you will annotate the span of the surface string that you deem to be relevant to the phenomenon. For example, in (75) you would annotate the span *a car was hit by a truck*, as a passive clause.

(75) As I walked down the street, [a car was hit by a truck,] and rolled over thrice.

You should try to annotate the minimum span involved in the phenomenon, however annotations must still be complete. For example, when annotating an entire clause, you should include adverbials at the beginning and end of an item, even though they may not appear to relate to the phenomenon in question—they still are a part of the clause. While it is likely that annotation span boundaries will tend to coincide with perceived constituent boundaries, this may not necessarily be the case across all phenomena/languages.

Using the annotation tool

The annotation tool you will be using is brat, which runs in the browser. In order to annotate a span, simply select the desired text span with the cursor; then in the resulting pop-up, box select the phenomenon to annotate the span with. Alternatively, you can hold the control key, double-click on the word at the start of the span, release control and then while holding the shift key, click on the point you want the span to end. This can be useful if you are having difficulty selecting the final punctuation character in a span.

Double-clicking on the labels of existing annotations will allow you to move the span or delete it. If you need to annotate discontinuous sections of text, first annotate one section, then double click on the label of the newly created span, then select “Add Frag.”. You can then select the second span for the annotation. The two sub-spans will then be connected by a thick dashed line.

For further help, see the brat manual\(^1\).

\(^1\) [http://brat.nlplab.org/manual.html](http://brat.nlplab.org/manual.html)
Appendix B: Grammatical Phenomenon Annotation Guidelines

Punctuation

Punctuation that occurs within a span will obviously be included within that span, however the question arises of what should be done when punctuation occurs at the beginning or end of a span. In general our approach is that punctuation should bind tightly and be included within a span if it sits adjacent to the span and is not clearly associated with the preceding/following word. For instance, in (76) when annotating interrogative clauses, the span that would be annotated is “‘What time is it?’.”

(76) [“‘What time is it?’] I asked.”

Note that this tight binding of punctuation should be respected even if the data has been tokenized so as to put whitespace between words and punctuation characters. If it is ambiguous which direction a punctuation character should bind, then it should attach to the token to its left.

Coordination

When a candidate for annotation involves a coordination, annotate as two separate instances of the phenomenon if the coordination is at the top level of the constituent pertaining to the phenomenon in question. If the coordination occurs below the top level constituent then annotate this as one instance. For instance, when annotating complement clauses, you would annotate (77a) as two instances and (77b) as one. Likewise, when annotating relative clauses, (77a) would be annotated as two instances and (77b) would be annotated as one instance.

(77) a. I thought [that we would go to the shop] and [that we would get a sausage roll.]
b. I thought [that we would go to the shop and get a sausage roll.]
c. I like people [who tell the truth] and [who eat shrimp.]
d. I like people [who tell the truth and eat shrimp.]

Ambiguity

In the case of genuine ambiguity, such as prepositional attachment ambiguity, annotate under the basis of attaching higher rather than lower.

Phenomenon-specific Instructions

The following sections outline how we are defining each phenomenon and how they should be annotated. Read through the overview of each phenomenon before beginning its annotation paying particular attention to the instructions in the How
Passive Clause

The passive construction is a valence-changing construction that alters the way thematic roles are mapped onto grammatical roles so as to foreground the patient role.

(78)  

  a. Scientists discovered the Higgs Boson.
  
  b. The Higgs Boson was discovered (by scientists).

(78) shows a transitive active sentence and the corresponding passive form in which the patient, *the Higgs Boson*, has moved from direct object to become the subject, and the agent has moved from the subject to the oblique prepositional phrase *by scientists*.

The passive construction can be broken up into two distinct processes. The first, agent demotion, sees the agent demoted from subject to an oblique, or completely omitted, having the effect of reducing the valency of the predicate by one. The second, patient promotion, promotes the patient from direct object to subject. For a clause to be counted as a passive construction it must at a minimum involve the process of patient promotion.

Watch out for clauses like (79) where the participle *broken* is not functioning as a passive participle but as an adjective derived from a passive participle. Where it is clear that such participles are functioning as adjectives, the clause will be ignored. Sometimes the syntactic environmental together with textual context is insufficient to resolve this ambiguity, in which case mark it as a passive.

(79)  The door was quite broken.

How to annotate it?

- Annotate the span covering the entire passive clause.
- Include the agent phrase if there is one.
- Passive relative clauses should be included.
- Exclude clauses with adjectives formed from past participles.
Imperative Clauses

In this study we will restrict our attention to clause forms dedicated to serving an imperative function, which is to say that they are uttered when appealing to an audience to bring about a desired state of affairs and the audience is deemed to be control of this state of affairs. Such an example is presented in (80).

(80) Live long and prosper.

In this study we will exclude negative imperatives (prohibitives) and we will also ignore imperative types that gain their imperative force via conversational implicature, unless this is the primary means the language has for realising the imperative. This means that examples such as in (81) will not be counted as imperatives.

(81) a. You will take a seat.
    b. Can you take a seat?

How to annotate it?

- Annotate the span covering the entire imperative clause.
- Do not include prohibitives.
- Do not include utterances which gain their imperative force via conversational implicature.

Interrogative Clauses

Interrogative clauses are those whose purpose is to solicit information from the addressee. By looking at the types of responses that interrogatives solicit, they can be broken up into three subtypes:

(82) a. Do you like coffee?                         polar question
    b. Do you like coffee, or tea?                alternative
    c. What kind of coffee do you like?          wh-question

There is also a particular type of interrogative construction, the tag question (as illustrated by (83)), which can be thought of as a construction that forms polar interrogatives out of declarative sentences by adding an appropriate word or fixed phrase.

(83) Coffee is fantastic, isn’t it?

In this study we will include all the different kinds of interrogatives identified above: polar, alternative, content questions and tag questions. One particular type of interrogative we will exclude are those that occur as subordinate clauses, such as indirect questions like those found in (84).
(84)  a. The researcher asked whose coffee he had spilled.  
      b. The technician wondered how much the equipment was worth.

How to annotate it?

- Annotate the span covering the entire interrogative clause.
- Include polar, alternative, content and tag questions.
- Exclude indirect questions.
- Exclude declarative clauses end in a question mark and do not display the internal structure characteristic of interrogatives.

Complement Clause

For this study, we are defining complement clauses as subordinate clauses that are characterised by the following two properties:

1. The constituent possesses the internal structure of a clause, with it possessing a predicate at a minimum, and any arguments of this predicate being marked in the same way as they would be in a main clause.

2. The constituent functions as an argument of a verbal predicate in a higher clause.

For example, the clause that you are a light sleeper functions as the object of the verb understand in (85a) and as the subject of the verb frustrate in (85b).

(85)  a. I understand that you are a light sleeper.  
      b. That you are a light sleeper frustrates me.

Complement clauses may take the form of a sentence-like clause, as in (86a)—in which case they tend to represent a fact or proposition—or they may be reduced in some way, as in (86b) and (86c), in which the subject position of the complement clause is realised as a gap.

(86)  a. I remembered (that) I don’t like durian.  
      b. I remembered to feed the trolls.  
      c. I remembered seeing her.  
      d. I remembered his singing the song.
A potential cause for confusion may be found in English clauses such as *his sinking the Titanic*, which may at first appear to be a nominal phrase due to the superficial similarity with the NP *his sinking of the Titanic*. These can be teased apart, however, by noticing that in the case of the former, *sinking* is a gerundal verb which takes *the Titanic* as an argument, whereas in the latter case *sinking* is a deverbal noun (or nominalised verb) modified by *the Titanic*. A useful test in such cases is whether the constituent will accept an adverb, as this will not be possible if the constituent is an NP.

(87)  
\[ \text{a. I enjoyed his sinking the Titanic.} \] complement clause  
\[ \text{b. I enjoyed his sinking of the Titanic.} \] noun phrase  
\[ \text{c. I enjoyed his sinking the Titanic slowly.} \]  
\[ \text{d. *I enjoyed his sinking of the Titanic slowly.} \]

Sometimes it can be difficult to differentiate between clauses which are functioning as arguments of a verb, or as adjuncts. For example, in (88), the clause *to let the cat out* might be perceived to be a complement. It should not be annotated however, on the basis that it is not being selected for by the verb *opened*, but rather functioning as a prepositional phrase, modifying either the verb phrase or perhaps the entire main clause.

(88)  
\[ \text{I opened the door to let the cat out.} \]

**How to annotate it?**

- Annotate the span covering the entire complement clause.
- Include any complementizer such as *that* in the span.
- Do not include the verb of the matrix clause in the span.
- Ignore clauses that are the complements of things other than verbs—if the clause attaches to a noun, it is possible it should be annotated as a relative clause.
- Include extracted complements such as in *“The cat sat,” said George.*
- Where the head verb licensing the complement clause appears as a parenthetical, with components of the complement clause on either side, such as in (89), you will need to use discontinuous annotations, as described in Section B.

(89)  
\[ \text{[George,] he said, [is a rather amusing individual.]} \]
Relative Clause

Relative clauses are a specific type of subordinate clause which serve to restrict or elaborate on one of the referents in the higher clause. In this study we will only look at relative clauses which modify nouns. The relativisation is achieved via one of the arguments of the subordinate clause sharing its reference with the head noun of the constituent on which the subordinate clause is dependant.

(90)  
   a. I see the tree that has a thick trunk.  
   b. He chose the boy who eats worms.  
   c. He chose the boy whom I saw.  
   d. You know that band whose tunes I like.  
   e. I am the person to talk to.  
   f. The place where I live is awesome.  
   g. The reason why I left is silly.

Relative clauses also come in the form of reduced relative clauses. This can occur when the relative pronoun is omitted, as in (91a), but also when the relative clauses is formed with the participial form of the verb, as in (91b).

(91)  
   a. He chose the boy I saw.  
   b. You know that band suing the little boy?

Relative clauses can also be restrictive or non-restrictive. A restrictive relative clause serves to refine and further delimit the reference of the head noun it modifies. A non-restrictive relative clause does not restrict the reference, but instead elaborates upon the information being provided about the referent picked out by the head noun. In English the distinction between the two is made by placing commas around the non-restrictive relative clause, as illustrated in (92).

(92)  
   a. The dog which has a long tail sat down.  
   b. Their house, which had a lovely fence, sat on a hill.

A phenomenon that is superficially similar to restrictive relative clauses, but are not instances of relative clauses, are nouns that take a complement headed by that, as illustrated by (93). These are excluded through the observation that the head noun does not function as an argument of the predicate in the subordinate clause.

(93)  
   The belief that the moon is made of cheese is fanciful.

Another phenomenon that is functionally similar to relative clauses, but syntactically distinct, is that of noun phrases in apposition, in which two noun phrases are placed side by side and in which one serves to modify the other, such as in (94). In English, the noun phrase on the right modifies the reference of the one of the left. These are not counted as relative clauses as the modifying phrase is a nominal rather than a clause.
Appendix B: Grammatical Phenomenon Annotation Guidelines

(94) a. Jessica, the daughter of a gardener, also lived on the street.
     b. My friend Jessica also lived on the street.

How to annotate it?

- Annotate the span covering the entire relative clause.
- Include relative pronouns linking the relative clause to the head noun.
- Do not include the head noun that the clause modifies in the span.
- Include non-restrictive relative clauses and reduced relative clauses.
- Do not include the comma to the left of non-restrictive relative clauses in the span.
Bibliography


BIBLIOGRAPHY


BOUDA, PETER, and JOHANNES HELMBRECHT. 2012. From corpus to grammar: how DOBES corpora can be exploited for descriptive linguistics. In Electronic Grammaticography, ed. by Sebastian Nordhoff. Honolulu, USA: University of Hawai‘i Press.


COHEN, K. BRETONNEL, WILLIAM A. BAUMGARTNER JR., and LAWRENCE HUNTER. 2008. Software testing and the naturally occurring data assumption in natural language processing. In Software Engineering, Testing, and Quality Assurance for Natural Language Processing, 23–30, Columbus, USA.


Ferdinand Saussure, Edited by Charles Bally, and Albert Sechehaye. 1986. Course in General Linguistics. LaSalle, USA: Open Court.


Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Letcher, Ned

Title:
Discovering syntactic phenomena with and within precision grammars

Date:
2018

Persistent Link:
http://hdl.handle.net/11343/214022

File Description:
Complete Thesis

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.