

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Willy

Title:

The effects of sampling and semantic categories on large-scale supervised relation extraction

Date:

2012

Citation:

Willy (2012). The effects of sampling and semantic categories on large-scale supervised relation extraction. PhD thesis, Department of Computing and Information Systems, Engineering, The University of Melbourne.

Persistent Link:

<https://hdl.handle.net/11343/38090>

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.

The Effects of Sampling and Semantic Categories on Large-Scale Supervised Relation Extraction

A thesis presented

by

Willy

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

November 2012

Produced on archival quality paper

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: _____

©2012 - Willy

All rights reserved.

Thesis advisor(s)
Timothy Baldwin
Tara McIntosh

Author
Willy

The Effects of Sampling and Semantic Categories on Large-Scale Supervised Relation Extraction

Abstract

The purpose of relation extraction is to identify novel pairs of entities which are related by a pre-specified relation such as **hypernym** or **synonym**. The traditional approach to relation extraction is to building a dedicated system for a particular relation, meaning that significant effort is required to repurpose the approach to new relations. We propose a generic approach based on supervised learning, which provides a standardised process for performing relation extraction on different relations and domains. We explore the feasibility of the approach over a range of relations and corpora, focusing particularly on the development of a realistic evaluation methodology for relation extraction. In addition to this, we investigate the impact of semantic categories on extraction effectiveness.

Contents

Title Page	i
Abstract	iii
Table of Contents	iv
List of Figures	vii
List of Tables	ix
Citations to Previously Published Work	xii
Acknowledgments	xiii
Dedication	xiv
1 Introduction	1
1.1 Supervised Relation Extraction	3
1.2 Research Contributions	6
1.3 Thesis Structure	7
2 Background	9
2.1 History of Information Extraction	10
2.1.1 The Message Understanding Conferences	11
2.1.2 Automatic Content Extraction	15
2.1.3 Large-scale Document Collection IE	16
2.2 Relation Extraction	17
2.2.1 Relation Types	18
2.2.2 Styles of RE	22
2.2.3 Our RE Task	28
2.3 RE Approaches	28
2.3.1 Handcrafting Lexico-syntactic Patterns	29
2.3.2 Automatic Bootstrapping Approaches	32
2.3.3 Supervised Learning	38
2.3.4 Semi-Supervised Approaches	43
2.3.5 Relation-Independent RE Approaches	46
2.4 Pattern Representations	50
2.5 RE Evaluation	52
2.5.1 RE Datasets	52

2.5.2	Open Task vs. Closed Task	57
2.5.3	RE Evaluation Metrics	58
2.5.4	Interpreting Type-Level Evaluation Metrics	59
2.6	RE Research Comparability	61
2.7	RE Applications	62
2.8	Summary	64
3	Resources	66
3.1	WordNet	66
3.2	Corpora	73
3.2.1	English Gigaword Corpus	73
3.2.2	Wikipedia	73
3.3	Pattern Representation	74
3.3.1	MINIPAR	74
3.4	Machine Learning	81
3.4.1	Support Vector Machines	82
3.4.2	Classifier Evaluation Methods	85
3.5	Evaluation Metrics	86
3.6	Summary	87
4	Supervised Relation Extraction	89
4.1	System	91
4.1.1	Task Overview	91
4.1.2	Preprocessing	92
4.1.3	Labelling Relations	93
4.1.4	Gold-Standard Dataset Formation Process	99
4.1.5	Feature Representations	101
4.2	Comparison with Snow <i>et al.</i> (2005)	104
4.3	Extending Relation Sets	105
4.4	Experiments on Supervised Relation Learning	107
4.4.1	Effects of Using Different System Parameters	107
4.4.2	Effects of Using Different Corpora	111
4.4.3	Different Levels of Relations Difficulty	112
4.5	Summary	113
5	Effects of Sampling	116
5.1	Experiments on Different Sampling Procedures	117
5.1.1	Intersecting Corpora Instances	118
5.1.2	One Million Noun Pairs Experiment	122
5.2	Revisiting Our RE Task	137
5.2.1	Revisiting the Relation Labelling Process	137
5.2.2	Revisiting the Training and Evaluation Processes	142

5.3	Experiments	146
5.3.1	Results on the Development Partition	147
5.3.2	Observations on System Parameter Settings	156
5.3.3	Results on the Test Partition	160
5.3.4	Error Analysis	161
5.4	Experiment on Navigli and Velardi (2010)'s Dataset	170
5.4.1	Data Description and Hypernym Extraction Task	170
5.4.2	Navigli and Velardi (2010)'s Approach	171
5.4.3	WCLs for Definition and Hypernym Extraction	176
5.4.4	Applying Our RE Methodology to the Navigli and Velardi (2010) Data	176
5.4.5	Result	179
5.5	Summary	182
6	Semantic Categories for Hypernymy RE	186
6.1	WMEB	189
6.1.1	Background	190
6.1.2	System Architecture	191
6.1.3	Stop Categories	194
6.2	Using WMEB to Induce Semantic Lexicons	194
6.2.1	WMEB as a Blackbox	194
6.2.2	Forming and Organising Semantic Categories for WMEB	194
6.2.3	Results	202
6.3	Postclassification with WMEB	204
6.3.1	Methodology	204
6.3.2	Experiment	206
6.4	Integrating WMEB Lexicons into RE	211
6.4.1	Methodology	212
6.4.2	Experiment	216
6.5	Metaclassification	221
6.6	Summary	223
7	Conclusion	225
7.1	Conclusions	225
7.1.1	Our RE Methodology	226
7.1.2	Accuracy of Our Supervised RE System	227
7.1.3	Using Semantic Categories to Aid Hypernymy RE	229
7.2	Future Work	230
A	Examples of patterns extracted from MINIPAR output	250
B	List of the Seed Terms for each Category used in Bootstrapping	258

List of Figures

1.1	An overview of supervised RE	4
2.1	The difference between type and token-level RE	25
2.2	Framework of a bootstrapping RE system (figure adapted from Li <i>et al.</i> (2011))	34
2.3	Snippet of the infobox of <i>Australia</i> Wikipedia article	42
3.1	Examples of WORDNET concept definitions and synsets	67
3.2	Examples of word/sense pairs in WORDNET synsets	68
3.3	A snippet of the concept network in WORDNET	69
3.4	A dependency tree representation of the parse output in Table 3.1. The dashed lines represent the additional links for conjunctions . . .	77
3.5	<i>noun</i> ₁ has two satellite links, with one non-satellite link (the child nodes are a stick of length 3). <i>noun</i> ₂ has one satellite link, with one non-satellite link (the child nodes attached are not a leaf node or a stick of length 2)	81
3.6	A linearly separable dataset with possible separating hyperplanes . .	82
3.7	The margin difference between hyperplane $\vec{H}_1$ and $\vec{H}_2$	83
4.1	Flowchart for labelling antonym and synonym	95
4.2	Flowchart for labelling hypernym	96
4.3	System accuracy on the three relations, with $R \in \{1, 10, 25\}$	108
4.4	System accuracy on the three relations, with $R \in \{50, 100\}$	109
5.1	System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{1, 10, 25\}$	122
5.2	System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{50, 100\}$	123
5.3	Aggregate sort index distribution of 1-MIL instances and hypernymy INTERSECTION instances on various R settings	129

5.4	Aggregate sort index distribution on hypernymy INTERSECTION instances on various R settings whose negative instances are randomly sampled from data	131
5.5	System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{1, 10, 25\}$. The negative instances are randomly sampled from data	132
5.6	System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{50, 100\}$. The negative instances are randomly sampled from data	133
5.7	The accuracy of various WIKIPEDIA models based on two sampling procedures on the 1-MIL dataset	134
5.8	The accuracy of various WIKIPEDIA models based on two sampling procedures on the 1-MIL dataset	135
5.9	Aggregate sort index distribution on the three experiment partitions .	145
5.10	The Word-Class Lattice for the three definitional sentences in Table 5.22. Hypernym vertices are indicated with rectangles. The original words (and which sentence they are from) for each word class are also presented	175
5.11	A snippet of the MINIPAR output for typical definitional sentences found in the NV data as represented by <i>A bat is a mammal in the order of Chiroptera</i> . There is no directed path between the TARGET word <i>bat</i> with its hypernym <i>mammal</i>	178
6.1	Overview of WMEB architecture, showing the expanded ANIMAL semantic lexicon using <i>cat</i> , <i>fish</i> , and <i>dog</i> as the seed terms	192
6.2	An example of postclassification, highlighting the overridden predictions for (<i>snake</i> , <i>animal</i>) and (<i>cat</i> , <i>mammal</i>)	205
6.3	The recall, precision, and f-score of model trained on WIKIPEDIA, $\text{pos}=\text{w}$, $\text{neg}=\text{b}$, $Q = 10$, on the development partition after postclassifying using first- X extracted terms of WMEB lexicons	206
6.4	Normalised proportion of first- $X\%$ extracted terms of the five largest WMEB-induced semantic lexicons found in our evaluation data	208
6.5	An overview for the preprocessing experiment	213

List of Tables

2.1	An example of a filled template used in MUC-3	13
2.2	A predicate to describe <code>starsIn(<i>MovieActor</i>, <i>Film</i>)</code> used in KnowItAll	44
2.3	Pattern templates for binary relations used in KnowItAll	45
3.1	A sample of parse output by <code>pdemo</code>	76
3.2	Dependency paths between <i>composer</i> and the listed destination nouns	79
4.1	Example noun pairs and their frequency of occurrence	93
4.2	Annotation results using the proposed relation labelling heuristic. “+” = positive label, “-” = negative label, “Ign” = Ignored noun pairs due to: (i) “-W”: one or both nouns do not appear in WORDNET, (ii) “-S”: one or both nouns do not appear in SEMCOR, or (iii) “-sense”: the relation link is between non-first senses of both nouns	98
4.3	Comparisons of class distribution for instances extracted from GIGAWORD and WIKIPEDIA for various thresholds P of Θ_{n_i, n_j}	100
4.4	The number of patterns extracted from GIGAWORD and WIKIPEDIA for various thresholds Q of Λ_{pat_k}	102
4.5	The final feature vector representations for noun pairs from Table 4.1	103
4.6	Result comparisons for hypernymy between our systems and Snow <i>et al.</i> (2005). Snow <i>et al.</i> (2005) did not report their recall and precision	105
4.7	Patterns normally associated with the given relation (NB: for hypernymy, n_j is a hypernym of n_i)	106
4.8	Comparison between our systems and the baselines’ accuracy over GIGAWORD and WIKIPEDIA	106
5.1	Comparisons of class distribution for instances present in both GIGAWORD and WIKIPEDIA for various thresholds P of Θ_{n_i, n_j}	119
5.2	Fractions of each corpus instances present in both GIGAWORD and WIKIPEDIA for various thresholds P of Θ_{n_i, n_j}	120
5.3	Examples of noun pairs occurring in the data, sorted by their aggregate rank	121

5.4	Comparison of the instances class distributions for all relations in 1-MIL	125
5.5	Examples of noun pairs excluded from the experiment because the relation link is not between the first senses of both nouns. Notice that the noun pair is excluded only in one relation, but is labelled as a negative pair for the other relations	140
5.6	The class distributions of all eligible instances and the instances found in the three experimental partitions	143
5.7	The ratio of negative to positive of all eligible instances and the instances found in the three experiment partitions	144
5.8	The number of patterns extracted from GIGAWORD and WIKIPEDIA for various values of Q	147
5.9	Results for antonymy models trained on the training partition and evaluated on the development partition, treating white instances as positive and grey and black instances as negative	148
5.10	Results for antonymy models trained on the training partition and evaluated on the development partition, treating white and grey instances as positive and black instances as negative	149
5.11	Results for synonymy models trained on the training partition and evaluated on the development partition, treating white instances as positive and grey and black instances as negative	151
5.12	Results for synonymy models trained on the training partition and evaluated on the development partition, treating white and grey instances as positive and black instances as negative	152
5.13	Results for hypernymy models trained on the training partition and evaluated on the development partition, treating white instances as positive and grey and black instances as negative	154
5.14	Results for hypernymy models trained on the training partition and evaluated on the development partition, treating white and grey instances as positive and black instances as negative	155
5.15	Results for various models trained on the training partition on WIKIPEDIA features and $Q = 10$, evaluated on the subset of the development partition where grey instances are excluded from evaluation	158
5.16	The system parameter settings of the models chosen to classify the test partition	160
5.17	Results for the best models trained on the training partition and evaluated on the test partition, treating white instances as positive and grey and black instances as negative in evaluation	161
5.18	Results for the best models trained on the training partition and evaluated on the test partition, treating white and grey instances as positive and black instances as negative in evaluation	162

5.19	The number (and its proportion to the entire size) of false negatives, based on their corpus frequency	163
5.20	The number (and its proportion to the entire size) of true positives, based on their corpus frequency	163
5.21	15 most frequently-paired hypernyms in the test partition and the success rate of our best hypernym model in predicting such instances . .	164
5.22	Examples of sentences and the annotation markings. Target terms are in bold face, while their <i>hypernyms</i> are in italics	172
5.23	Macro-averaged precision scores in hypernym extraction task over the NV data for three different RE models	180
6.1	List of SEMANTIC CATEGORIES in our experiments. Each category appeared with at least 10 different hyponyms in the data	196
6.2	Groups (and their size) of semantically-close categories in our experiment	199
6.3	Example arrangement of semantic categories for multiple WMEB runs .	201
6.4	The number of categories relative to the size of their WMEB-induced lexicons	203
6.5	Postclassification improvement to the basic hypernymy model using WMEB semantic category lexicons	208
6.6	Comparison of our system's accuracy before and after post-classification on instances involving the 15 most frequently appearing semantic categories in the test partition	210
6.7	Accuracy of preprocessing model using semantic category information compared to the basic hypernymy model	218
6.8	Comparison of our system's accuracy before and after the preprocessing process for the 15 most frequently appearing semantic categories in the test partition	219
6.9	The accuracy of our various RE models using semantic category information compared to the basic hypernymy model	222
A.1	List of patterns extracted from a MINIPAR output in Figure 3.4. Note the case-folding of the nouns	257
B.1	List of the seed terms for the SEMANTIC CATEGORIES as sorted by their frequency in the GOOGLE WEB 1T corpus	265

Citations to Previously Published Work

Portions of Chapters 2, 3, and 4 have appeared in the following paper:

Willy Yap and Timothy Baldwin (2009) Experiments on Pattern-based Relation Learning, In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, Hong Kong, China, pp. 1657–1660.

Acknowledgments

There are many people who have supported my PhD journey in different ways. First, I would like to thank people who are dearest to me: my parents and siblings for their constant support, patience, and encouragement since the very first day I arrived in Australia; and my partner Aimee, for her love and her knack for easing all my troubles during my study.

Various institutions have supported me during my PhD. I would like to thank NICTA, for funding my scholarship and research trips. I am also grateful for the access to the advanced computing facilities of VPAC, VLSCI, and NICTA to perform many of my experiments.

I would like to thank the people and various visitors of the LT group, past and present. You have all been wonderful for your willingness to annotate the data, discuss research problems, organise social events (*bla-di-das*, especially!), provide endless supplies of cakes, biscuits, chocolates, and various delicacies, among many others. I will cherish all the fond memories we have shared together.

I have been very fortunate to have interacted with various researchers in the NICTA BioTALA group who have provided insights and feedbacks to my research. In addition, I am grateful for the members of my advisory committee Justin Zobel and Steven Bird for their sharp analysis and visions to shape my research direction.

Finally, I dedicate my most heartfelt gratitude to my two supervisors. My secondary supervisor Tara McIntosh, provided guidance and support through some of the most difficult moments during my PhD. Her expertise in **WMEB** has been a tremendous help for my research in Chapter 6. I am also thankful for her commitments to my cause despite having moved to Seattle. I reserve my last thanks for Tim Baldwin, who has been a model primary supervisor. He has always successfully made time to our research despite his increasingly packed schedule. His broad and deep knowledge of many areas have been invaluable for shaping my research, which has in turn helped me to grow as a researcher. His patience and belief in me have kept me motivated throughout my research. Thank you Tim, for being the best supervisor I could possibly have asked for.

Von Herzen - Möge es wieder - Zu Herzen gehn!

- LvB

Chapter 1

Introduction

This thesis is concerned with the task of Information Extraction (IE), which aims to identify and extract information of interest from natural language text, and represent it in a machine-readable format. There is a myriad of information that can be extracted from text. This thesis is specifically interested in the task of Relation Extraction (RE), where the aim is to extract information on the relationships (relational information) between a pair of entities (e.g. *Mozart* was born in *Salzburg*) from (unstructured) text documents, and represent it in a machine-readable relational tuple (e.g. `person-birthplace(Mozart, Salzburg)`). The collections of relational tuples are often stored in a relational database.

Knowledge of relational information is very valuable to various Natural Language Processing (NLP) applications. For example, the tuple `person-birthplace(Mozart, Salzburg)` can help a question answering system to answer factoid questions such as *Where was Mozart born?*. Relational databases containing `person-birthplace` and `person-born.in.year` tuples can also be generalised to answer more complicated queries such as *Provide a list of people born in Melbourne in 1986*.

The utility of a relational database is dependent on the number of available relational tuples and their quality: usually the more high quality relational tuples, the more useful they are. Manually building such a resource is expensive, as one may need to scan through a large number of documents to identify tuples. RE research aims to address this problem by providing automated methods that extract as many

relational tuples as possible from a document collection. This is a hard task, as there are many possible relations of interest (e.g. a specific **person-birthplace** relation vs. a more general **synonymy** relation), each with their own characteristics. As a result, most RE research (including this thesis) tends to focus only on extracting tuples that are related by one or a small set of predefined relations of interest.

RE systems rely heavily on the patterns occurring between a pair of entities in text to determine whether they are related by a given relation or not. This idea stems from Hearst (1992)’s observation that there exists a set of patterns that are frequently used and unambiguously defines the **is-a** relation between noun pairs. Consider the following snippets from sample sentences:

1. ... *composers* **such as** *Bach* and *Beethoven* ...
2. ... *Perl* **or other** *programming languages* ...

where the patterns **such as** and **or other** are very useful in identifying examples of **is-a** relations, e.g. *Bach* and *Beethoven* are *composers*, or *Perl* is a *programming language*. Using these patterns, it is then possible to directly extract the relational tuples **is-a**(*Bach*, *composer*), **is-a**(*Beethoven*, *composer*), and **is-a**(*Perl*, *programming language*) from the respective sentences.

While these patterns are undoubtedly useful, they are usually specific to one relation, and therefore ineffective at identifying examples of other relations. For example, the patterns above are not helpful for identifying examples of **person-birthplace** relation. Following Hearst’s idea, one is therefore required to identify a different set of patterns specific to each relation of interest. Repeating Hearst’s manual steps to identify relation-identifying patterns for every relation of interest is an expensive process, because it requires a domain expert to perform the time-consuming task at scanning and generalising the contexts in which the sample entity pairs occur.

Hearst’s approach is dependent on the assumption that there exists a set of relation-identifying patterns for the relations of interest. However, this is not true for some relations, where the contexts they often appear in are ambiguous and not specific enough to those relations (Hearst 1992; Girju *et al.* 2006). For example, one

may consider using the following patterns to extract the **antonymy** relation (entities with opposite meanings):

1. ... work hard **from** *morning* **to** *evening* ...
2. ... you can answer with **either** *yes* **or** *no* ...

However, the same patterns can also be used to describe non-antonymy relations, which may result in the extraction of incorrect relational tuples. For example:

1. ... drive **from** *Melbourne* **to** *Darwin* ...
2. ... completed by **either** *Joe* **or** *Mary* ...

The examples above highlight the limitations of Hearst's approach. First, the generalisation process only learns from positive pairs (i.e. sets of pairs which have been previously known to have a given relation). Therefore, it does not take into account that the identified patterns, specific to the positive pairs, may also extract other pairs not related by the given relation.

Second, the extraction process is entirely dependent on the patterns, i.e. looking at whether an entity pair occurs in the identified patterns or not. It ignores the other contexts that the entity pair occurs with in the data, which otherwise can be used to leverage the information as a confidence for extracting the entity pairs. For example, we can be more confident in extracting an entity pair that occurs very often in a range of the identified patterns, as opposed to another entity pair that only occurs once in one of the identified patterns but appears frequently outside the context of the identified patterns.

1.1 Supervised Relation Extraction

There have been several approaches to alleviate the challenge of learning patterns for RE (which we will discuss in Chapter 2). One popular idea (and also the focus of this thesis) is to approach RE as a supervised task, which automates the generalisation of the context the entity pairs occur in with a machine learner based on evidence from

(+) (tiger, animal)	<i>... animals such as tiger ... (15)</i> <i>... tiger and other animals ... (7)</i> <i>My cousin's favourite animal is the tiger (1)</i>	Training
(+) (senior, undergraduate)	<i>... is targeted for undergraduates, especially seniors ... (14)</i> <i>... an eccentric mix of senior citizens and young undergraduates (3)</i>	
(-) (expert, data)	<i>Michael is the company's expert in data (4)</i> <i>Expert predicted that the economy data ... (25)</i>	
(?) (green, colour)	<i>Cars that are green in colour (20)</i> <i>Light colours such as green, ... (12)</i> <i>Green as a colour, represents ... (8)</i>	Test

Figure 1.1: An overview of supervised RE

a set of documents. For any relation, given a set of labelled (positive and negative) training examples and their patterns, the machine learner produces a classifier that can make a binary classification of any given pair of entities to decide whether they are related by the specified relation or not.

Figure 1.1 provides a simplified overview of the RE task for the *is-a* relation. Learning from the three training examples and their patterns, the classifier produced is required to make a binary prediction of whether the pair *(green, colour)* is related by the *is-a* relation, based on its set of contextual patterns.

The example highlights a few challenges for the learner to generalise the patterns. First, not all the patterns for the positive pairs are actually relation-identifying. For example, the third pattern for *(tiger, animal)* is not specific to the *is-a* relation. As such, the learner may risk extracting entity pairs that do not have the *is-a* relation. Second, there may be word ambiguity in the training examples provided. For example, the word *senior* in *(senior, undergraduate)* may either mean “a final year University student” or “someone elderly”. While it is clear that the positive annotation refers to the former meaning, there can be occurrences of *senior* in the data that refer to the latter (e.g. the second pattern for *(senior, undergraduate)*), which are not relation-identifying.

A supervised system often learns from the frequency indicating how often an entity pair occurs with a certain pattern when generalising patterns (we refer to this frequency as *pair-pattern cooccurrence frequency* and it is indicated by the numbers

in parentheses following each pattern in Figure 1.1). Using this, the learner can determine the importance of certain patterns. Patterns frequently and commonly occurring in all entity pairs of a certain label can then be associated as high-quality patterns that are indicative of their labels. This style of learning from both positive and negative pairs in a supervised approach is in contrast to Hearst’s approach which exclusively learns from positive pairs.

Although as humans we can confidently predict that (*green*, *colour*) has the *is-a* relation, it is not as straightforward for the classifier. This is due to the diversity of its patterns. For example, its first pattern is similar to the first pattern of (*expert*, *data*), which is a negative example. On the other hand, the second pattern of (*green*, *colour*) is similar to the first pattern of (*tiger*, *animal*), which is a positive example. Finally, there is a further challenge of the third pattern not appearing in the patterns of the training examples. For a given pair, the classifier takes into account all the patterns that pair occurs with in the data, and makes its prediction based on the weight of the noun pair’s patterns, as learned from the training examples. This is in contrast to Hearst’s approach whose extraction process is entirely dependent on whether a noun pair occurs with a certain pattern or not.

One of our research objectives is to propose a general supervised RE methodology for the task above. The desiderata of our methodology is that it only assumes the existence of training and test pairs for relations of interest, and the corpus data that our system can learn from. The methodology is not dependent on relation-specific tools or resources, so as to preserve its ease of adapting it to any other relations with minimal effort.

Different RE research often makes different experimental assumptions, making direct result comparisons inappropriate. This may be caused by different factors, such as determining which entity pairs constitute positive and negative examples, the number of positive and negative examples provided to the machine learner during training, the corpora used to extract the entity pairs, system parameters, and learning or evaluation methods. Because of this, results across different research are often largely directly incomparable, and it is difficult to assess what are the best approaches to RE (Pyysalo *et al.* 2008). In our research, we hope to avoid the above problems by

using assumptions that are consistent throughout our experiments, to enable better result comparability.

1.2 Research Contributions

This thesis addresses a number of issues pertinent to supervised RE, focusing on RE over large corpora. We propose an experimental methodology to RE that is applicable to any relation of interest, allows direct and sensible RE comparison of different resources, tools, and relations within its standardised extraction process, and, most importantly, whose design is reflective of the challenges in extracting relation tuples from large-scale corpora.

An important element of this thesis is the prior work by Snow *et al.* (2005) who successfully applied their RE approach to large-scale corpora for the *is-a* relation. We take their approach as a starting point in our research on our way to proposing our RE methodology. Much of our research carefully examines their methodology, especially their task setup and its impact on the experimental methodology.

We observe that Snow *et al.* (2005)’s methodology made a number of simplifying assumptions that greatly impact on the interpretability of their reported RE results. We first address these assumptions to develop a robust evaluation methodology that is much more representative of the task. Following that, we apply Snow *et al.* (2005)’s system to our evaluation, and discover its limitations when applied to a more realistic setting.

Using our evaluation methodology, we also look at the utility of applying a supervised RE system to large-scale RE. As part of this, we experiment with various relations, corpora, and system parameters, and draw conclusions on their impact on RE accuracy.

Finally, we propose various approaches to using semantic categories to improve the accuracy of our system on the *is-a* relation, using the output of **WMEB**, an automatic lexicon induction system to predict semantic categories of nouns (McIntosh and Curran 2008). Our first approach directly utilises the induced lexicons from **WMEB** to make a generalised prediction for all noun pairs belonging to a certain semantic

category, instead of individually predicting the label for each noun pair. Our second approach integrates the semantic category information during the learning process in generalising the patterns for semantically-related pairs. The third approach combines the predictions from our first and second approaches via a metaclassification process to further improve RE accuracy.

1.3 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 provides the necessary background to the research presented in this thesis, and locate our work relative to previous research. It covers a wide range of RE topics, such as the different commonly extracted relation types and their characteristics, the pros and cons of popular RE approaches, and issues relating to RE evaluation and result comparability. Chapter 3 describes the resources used in our research, such as the gold-standard labelling resource, corpora, and external tools used for parsing and machine learning.

Chapter 4 first discusses our effort to reimplement Snow *et al.* (2005)’s supervised methodology on large-scale corpora. Having confirmed the success of our implementation, we then investigate how to extend the methodology to two other relations. Throughout this chapter, we identify limitations of Snow *et al.* (2005)’s task setup and their implications for the interpretation and comparability of RE results.

Chapter 5 describes our proposal for RE methodology that enables more robust interpretation and comparability of results. The methodology also includes a task setup that better reflects the challenges in applying a supervised methodology to large-scale corpora. We then investigate the accuracy of our new methodology on large-scale corpora, and find that its results are much less competitive, especially compared to our earlier findings in Chapter 4. Error analysis and further experiment on a smaller dataset suggests that the methodology suffers from data sparseness and class imbalance.

Chapter 6 discusses various strategies for improving the accuracy of our system using semantic categories, focusing on the hypernymy relation. These strategies combine outputs from WMEB (McIntosh and Curran 2008) and our supervised RE system

to generalise both the learning and predictions of individual noun pairs into groups of pairs sharing similar semantic categories.

Finally, Chapter 7 summarises our research and contributions to the field, and proposes various avenues for further work.

Chapter 2

Background

RE is a large and well-researched field. Therefore, there exist many different styles and approaches to RE, each catering to different relations and research objectives.

In this chapter, we review RE research in general, and discuss our research with respect to the larger scheme of general RE research community. We start by discussing the historical motivation for Information Extraction (IE), and how it influenced the development of RE in Section 2.1. We also outline the evolution of document-level RE tasks such as the Message Understanding Conference (MUC) and Automatic Content Extraction (ACE), which laid the foundations for the current day RE research.

The subsequent rise of the web encouraged a shift in research direction from the traditional small-scale to large-scale RE task. Further, the diversity of the information on the web presents ample opportunities to expand RE work into a wide range of other relations: from generic lexical relations, such as **hypernymy**, to factoid relations such as **person-birth_place**. This gives rise to a variety of RE learning styles, as different relations are usually approached differently. These different RE learning styles often have different assumptions on the task setups, such as the importance of explicit or implicit evidence, the datasets, or whether a set of relations are to be specified prior to extraction. We discuss the various properties of relations and learning techniques in Section 2.2.

The accuracy of an RE system is highly dependent on the success of identifying a set of patterns that unambiguously highlights the relations between the two entities.

Section 2.3 explores various approaches in identifying such set of patterns for a given relation, ranging from manual, semi-supervised, and supervised approaches, each with its own advantages and disadvantages. In order to improve extraction accuracy, researchers have also experimented with integrating additional information on top of the information available from the patterns' surface strings, or to use them in some other more useful representations. This is discussed further in Section 2.4.

Various aspects in RE evaluation are presented in Section 2.5. We discuss the various datasets commonly used for evaluation. In addition to this, we describe a number of popular evaluation metrics used. More importantly, we explore the potential pitfalls in interpreting RE results from these evaluation metrics due to the task setups.

Several factors such as different corpora, learning styles and approaches, may cause difficulties when comparing performances of different systems, which in turn, may lead to difficulties in judging the overall RE progress. Section 2.6 discusses various factors that may lead to system incomparability and its impact to RE.

Finally, various NLP applications that have been shown to benefit from the knowledge extracted from RE systems are described in Section 2.7. Moreover, we discuss several future avenues in how RE techniques can be potentially employed to influence and improve other NLP applications.

2.1 History of Information Extraction

The genesis of RE research can be traced back to the development of Information Extraction (IE). Unlike Information Retrieval (IR) whose main focus is to find relevant documents, IE focuses on finding the relevant information inside documents (Jackson and Moulinier 2002). The general process in IE usually involves scanning text for information relevant to some interest, extracting the information, and presenting it in a structured form. To do so, IE needs to “understand” the text well enough to extract the information of interest. Having said that, the level of text understanding in IE is not as ambitious as in full text understanding, which seeks to capture all information in the text (Freitag 2000; Hobbs and Riloff 2010). In this light, we can

think of IE as a “shallow” version of full text understanding.

IE research is motivated by the need to efficiently extract structured data from large amounts of text. Historically, typical organisations interested in IE applications were military intelligence organisations and newswire agencies. In recent times, researchers in the biomedical domain have been investing in IE in order to keep up with the tide of information from the large (and growing) amount of biomedical literature. Some examples of IE-related biomedical research are ontology construction and curation (McCrae and Collier 2008), and identifying protein-protein interactions in the published papers (Özgür *et al.* 2008; Pyysalo *et al.* 2008; Miyao *et al.* 2009).

2.1.1 The Message Understanding Conferences

The Message Understanding Conferences (MUC) are one of the first influential shared tasks in IE, which arguably played a large role in formalising IE tasks and research. They were organised by the Defense Advanced Research Projects Agency (DARPA) over seven conferences between 1987–1997. DARPA’s motivation in organising MUC was to encourage research in automated text analysis of military messages (Grishman and Sundheim 1996). This is reflected by the use of documents regarding naval sightings (MUC-1 and MUC-2) and Latin America terrorist operations (MUC-3 and MUC-4). The documents used in MUC are in free, unstructured text.

Over these seven conferences, MUC successfully conceptualised IE as a *template filling* task. A MUC template defines all the information of interest that must be extracted from the text. It contains various slots, each representing specific information to be extracted from a given document. The MUC task is to then provide answers for each slot, by scanning and extracting information from unstructured text and represent it in a structured form (template).

From MUC-1 to MUC-5, the conference was centred on the *event extraction* task, which involves detecting a certain event and extracting information regarding the event. Below is an example of text provided in MUC-3 for the task, whose text

domain is terrorism in Latin America.¹ Table 2.1 shows the filled template (Chinchor *et al.* 1993):

DEV-MUC3-0011 (NOSC)

Lima, 9 Jan 90 (EFE) – [TEXT] Authorities have reported that former Peruvian Defense Minister General Enrique Lopez Albuja died today in Lima as a consequence of a terrorist attack.

Lopez Albuja, former Army Commander General and Defense Minister until May 1989, was riddled with bullets by three young individuals as he was getting out of his car in an open parking lot in a commercial center in the residential neighborhood of San Isidro.

Lopez Albuja, 63, was driving his own car without an escort. He was shot eight times in the chest. The former minister was rushed to the Air Force Hospital where he died.

To reflect the growing complexity of IE tasks, new slots were added and the structure of the templates and the evaluation metrics were refined in each subsequent conference. On top of this, new tasks were introduced: MUC-6 introduced *named entity recognition* and *coreference resolution*, while MUC-7 introduced several tasks such as *scenario template*, and *template relation* (Grishman and Sundheim 1996; Chinchor 1998). All of these tasks were still based on the concept of template filling. However, as each task has different objectives, the template structure for these tasks are also different.

MUC Contributions & Shortcomings

Research in IE owes much to the seven iterations of MUC. These conferences defined the IE task and provided formal evaluation procedures for it. As previously discussed, MUC's contribution in conceptualising IE as template filling task has been highly influential. The other significant contribution is the introduction of precision, recall, and f-score as standard IE evaluation metrics in MUC-2. Since their introduction, these metrics have established themselves as the preferred evaluation metrics.

¹All of the original text is in uppercase; we normalise the casing for better readability

Slot	Answer
Message: ID	DEV-MUC3-0011 (NOSC)
Message: Template	1
Incident: Date	09 Jan 90
Incident: Location	Peru: Lima (city): San Isidro (neighborhood)
Incident: Type	Attack
Incident: Stage of execution	Accomplished
Incident: Instrument ID	—
Incident: Instrument Type	Gun
Perp: Incident category	—
Perp: Individual ID	Three young individuals
Perp: Organization ID	—
Phys Tgt: ID	—
Phys Tgt: Type	—
Phys Tgt: Number	—
Phys Tgt: Foreign Nation	—
Phys Tgt: Effect of incident	—
Phys Tgt: Total number	—
Hum Tgt: Name	Enrique Lopez Albuja
Hum Tgt: Description	Former Army Commander General and Defense Minister
Hum Tgt: Type	Former government official / Former active military
Hum Tgt: Number	1: Enrique Lopez Albuja
Hum Tgt: Foreign Nation	—
Hum Tgt: Effect of incident	Death: Enrique Lopez Albuja
Hum Tgt: Total number	—

Table 2.1: An example of a filled template used in MUC-3

During the initial stage of organising MUC-6, MUC organisers identified several key issues associated with the previous MUC conferences. One key issue is that MUC participants at that time were focusing too much on specific aspects of individual MUC tasks (which only focus on a specific domain, e.g. newswire articles on terrorist activity in South America). This issue, along with the fact that MUC systems are only evaluated in a task or domain-specific setting, leads to questions around the contribution of these task-specific approaches to solving the general IE task. Relating to this, there was also concern about the lack of system portability, where it had been shown that substantial effort is required to adapt IE systems participating in MUC tasks to a different domain or task, as most of these systems are heavily-tailored for a specific domain or task.

Next, the MUC-6 organisers were concerned with the considerable effort invested by participants in developing their systems. For example, the IE task in MUC-5 had become so complex that most participants spent at least 6 months developing their systems (Grishman and Sundheim 1996). This prompted the organisers to question whether the effort invested to developing the systems could be properly justified. All these key issues highlighted the shortcomings of approaching IE as a single monolithic task.

To address these issues, MUC-6 organisers divided the single monolithic IE task into several subtasks, in which each subtask is hypothesised to individually contribute to the general IE performance. These subtasks are:

- **Named entity recognition:** a task to identify words in the text representing a person, organisation, or location name (**entity names**), a date or time stamp (**temporal expressions**), or a currency or percentage figure (**numeral expressions**).
- **Template element filling:** a task to identify basic information related to person or organisation entities from text and fill the template associated with each entity mention.
- **Coreference resolution:** a task to identify strings that represent coreferring noun phrases. For example, identifying that the word *his* refers to *Chopin* in

the following sentence: *Chopin arrived in Paris in September 1831, not sure whether he would settle there for good.*

- **Scenario template:** a task to capture domain- or task-specific information similar to the previous installments of MUC (e.g. capturing information regarding terrorist attack in MUC-3).

A new template was created for each subtask. As the individual subtask is simpler than the monolithic IE task, the templates used in each subtask of MUC-6 are also comparatively compact as compared to the complex hierarchical structure used in MUC-5.

By breaking IE into multiple subtasks, MUC organisers brought about a significant change of research direction in IE. By allowing participants to only participate in subtasks they are interested in, MUC encouraged researchers to dedicate concentrated effort in solving smaller IE subtasks rather than a shallower approach to the general IE task. This is part of the *divide and conquer* vision, that is to solve smaller relevant subtasks before putting them together to solve the overall IE problem (Cardie 1997; Freitag 2000).

The last of the MUC conferences, MUC-7, preserved all the subtasks introduced in MUC-6 and introduced the template relation (relation extraction) task (Chinchor 1998). All of these subtasks introduced continuously throughout MUC series are still actively researched, reflecting the clear influence of MUC to IE research.

2.1.2 Automatic Content Extraction

After the completion of the final MUC in 1998, the Automatic Content Extraction (ACE) program was created to continue the legacy of MUC.² ACE started as a pilot study in 1999, with the first formal evaluation in 2004. Initial ACE research objectives closely resembled those of MUC-6 and MUC-7 (Doddington *et al.* 2004). The influence of the later MUC conferences (MUC-6 and MUC-7) was clearly reflected

²<http://www.itl.nist.gov/iad/mig/tests/ace/>

in the experimental design and subtasks proposed in ACE. In its initial evaluation, the main ACE subtasks were entity recognition, relation extraction, and event extraction.

ACE, however, expanded the research scale by increasing the task complexity, investing considerable effort in meticulously defining each subtask and its components, in addition to collecting and annotating the data according to detailed task definitions. In subsequent evaluation workshops, ACE reintroduced the value detection (numeral expression) and temporal expression tasks introduced in MUC-6, but with finer-grained annotation (LDC 2005; LDC 2007). Its contribution to RE research can also be reflected in the resources it produced: it has annotated textual, audio, and image data in three languages (Arabic, Chinese, and English).

The number of subtasks was scaled back in 2008, and limited to entity recognition and relation extraction tasks (LDC 2008). The task, however, proposed an evaluation with an additional layer of complexity that involves a cross-document experiment setting. When evaluating in this setting, the task assumes its documents context to be global (across documents) rather than local (within the bounds of a single document) traditionally assumed in previous MUC or ACE subtasks. The cross-document setting provides an additional challenge in that systems are required to link entity or relation information across separate documents.

In 2009, ACE was incorporated into the Text Analysis Conference (TAC) under the Knowledge Base Population (KBP) track (McNamee and Dang 2009). The objective of the task is to evaluate the ability of automated systems to discover named entities from a large-size corpus (around 1 million newswire articles), and represent this information in a standardised knowledge base.

2.1.3 Large-scale Document Collection IE

In the mid 1990s, researchers had started to notice the increasing availability of large-scale document collections, especially documents from millions of web pages. These collections contain vast amount of potential information and present a different challenge than the small-scale document-oriented IE tasks tackled in MUC or ACE.

MUC and early ACE tasks assume a small-scale document collection, of the order

of hundreds of newswire articles. Because of this, the conference organisers could afford to perform extensive data annotations, requiring annotators to carefully examine each sentence in the document. This hand-annotation process is very expensive, or rather impractical, to be applied to large-scale document collections.

The document collections used in MUC and ACE are domain-specific. Systems participating in these tasks tend to take advantage of the domain information and exploit resources that can increase the domain knowledge (Riloff 1993). As a result, the approaches used are inherently domain-specific and cannot be easily adapted to different domains. In contrast, large-scale document collections may contain documents from multiple domains, where a domain-specific approach may not be effective and thus encourages the development of generic domain-independent approaches.

The challenges of large-scale document collections prompted some researchers to divert their focus from MUC and ACE tasks, and exploit the vast amount of information in large-scale document collections to propose various approaches to IE. Some examples include generating extraction patterns that reduce the need for corpus annotation (Riloff 1996), automatically building a knowledge base from web pages (Craven *et al.* 1998), or using machine-learning techniques to IE (Freitag 2000).

Since then, there has been a push towards research in *Machine Reading* (MR). The task of MR is defined as the autonomous understanding of text, of exploratory and open-ended nature, and with the objective of forming a coherent set of facts based on a textual corpus and a background theory (Etzioni *et al.* 2006). An MR system aims to extract high quality knowledge from large-scale corpora with minimal human supervision, but without the prior domain knowledge commonly assumed in the majority of IE research. Another aspect of an MR system is the desideratum to be able to continuously learn from experience, either from new documents, or from user feedback (Poon *et al.* 2010).

2.2 Relation Extraction

This thesis focuses on the task of relation extraction (RE). RE extraction can be defined as a subtask of IE which is concerned with identifying pairs of entities with a

prespecified relation from a given text. Instances of RE system outputs (or *relation mentions*) are expressed in a relation tuple: **relation**(*entity-1*, *entity-2*). For example, **person-born_in_year**(*Mozart*, *1756*) implies that *Mozart* was born in *1756*.

RE systems have been developed for a wide variety of purposes. As a result, it is unsurprising to see the diversity of RE tasks explored in the research area, which is clearly reflected in the large number of different relations of interest in the field and the variety of styles. We discuss these two aspects in the next sections.

2.2.1 Relation Types

RE research has explored a wide variety of relations. Some examples include **location_of**, **employee_of**, and **manufacturer_of** as introduced in MUC-7 (Chinchor 1998), semantic relations such as **antonymy**, **synonymy**, **hypernymy** (alternatively **is-a**), **meronymy** (Turney 2008; Girju *et al.* 2006; Girju *et al.* 2007), general facts such as **person-born_in_year** or **city-capital_of_country** (Paşca *et al.* 2006), and other more specific relations such as **gene_associated_with_disease** (Özgür *et al.* 2008), **author-book_title** (Brin 1998), **chemical_reaction** or **succession** (Pantel and Pennacchiotti 2006). We compare and contrast the broad attributes of the relations of interest in the next section.

Binary vs. Complex Relations

A binary relation is any relation which takes two arguments (or entities), in which the relation expresses the association between the two arguments. **Hypernymy** between a pair of words or **author-book_title** are examples of binary relations. The majority of RE research focuses on these.

In comparison, a complex relation is any n -ary relation among n pre-specified typed entities, where usually $n > 2$. McDonald *et al.* (2005) provide a few examples of complex relations such as **person-job-company** consisting of three entities: a person's name, job title, and workplace; or a **genomic_aberration_event** relation that are interested in extracting four entities of a genomic variation event: the type of the event, its location, its initial state, and its altered state. Xu *et al.* (2007) use the **Nobel.Prize.award** relation where the objective is to extract instances of the name of

the recipient, the area of the Nobel Prize, and the year of the award. For example, the following sentences:

Mike Smith works as a programmer in InvestTech Corp.

At codons 12, the occurrence of point mutations from G/A to T/G were observed.

Einstein was awarded the 1921 Nobel Prize in Physics for his discovery of the law of the photoelectric effect.

would give rise to these tuples respectively:

`person-job-company`(Mike Smith, programmer, InvestTech Corp)

`genomic_aberration_event`(point mutation, codon 12, G, T)

`Nobel_Prize_award`(Albert Einstein, Physics, 1921)

Symmetric vs. Asymmetric Relations

Another attribute of a relation is the effect of the *entity order* in relation to the tuples. A *symmetric relation* is a relation where the order of the entities does not affect the interpretation of the relation tuple. On the other hand, in an asymmetric relation the order of the entities affects the interpretation of a relation tuple.

To illustrate the difference between the two, consider the symmetric relation `synonym`, and relation tuple `synonym(salary, wage)`, where *salary* occupies the first entity slot and *wage* occupies the second slot. The relation tuple `synonym(wage, salary)`, where we reverse the order of *salary* and *wage* of previous tuple, is *equivalent* to `synonym(salary, wage)`, as `synonym` is a symmetric relation. On the other hand, consider the asymmetric relation `person-born_in_year`. In this relation, it is implied that the first entity slot refers to a *person*, while the second slot refers to *year*. Following this, `person-born_in_year(Mozart, 1756)` is a correct relation tuple, while `person-born_in_year(1756, Mozart)` is not.

Hierarchical vs. Flat Relations

The contrast between hierarchical and flat relations is based on the ability of a relation to be expressed in a hierarchy (for example, an organisation structure hierarchy). In a hierarchical relation, a particular entity can be described as being “above”,

“below”, or “at the same level” relative to other entities in the same hierarchy. On the other hand, entities in a flat relation do not have hierarchical awareness relative to the other entities.

Using graph theory, we define a *hierarchical* relation to be a relation that can be expressed via a *directed acyclic graph* (DAG), in which the relations between the nodes are *transitive*. A binary relation is said to be transitive if when an entity *a* is related to an entity *b*, and *b* is related to an entity *c*, then *a* is also related to *c*. A *flat* relation is a relation that either cannot be expressed via an directed acyclic graph, or is not transitive.

Hypernymy (is-a) is an example of a hierarchical relation. It is both transitive (e.g. *violin is-a stringed instrument*, *stringed instrument is-a musical instrument*, and therefore *violin is-a musical instrument*), and can be arranged in a directed acyclic graph representing a hierarchy. The WORDNET (Fellbaum 1998) hypernym network is a prime example of an arrangement of hypernymy hierarchy (see Section 3.1 for our WORDNET discussion).

Synonymy, on the other hand, is a flat relation. While **synonymy** is a transitive relation, the relations between **synonym** entities are cyclical, i.e. *enemy* is a synonym of *foe*, and therefore *foe* is a synonym of *enemy*. This is in contrast to **hypernymy** where it is incorrect to deduce that *musical instrument* is a *violin* from the relation *violin is a musical instrument*. Other flat relations such as **author-book_title** and **person-born_in_year** are not transitive.

Domain-Specific vs. Domain-Independent Relations

In RE research, there is a strong correlation between the domain of the documents used and the relations of interest. The correlation stems from the fact that if one is interested in extracting a certain relation, one is more likely to perform extraction process on documents containing a large number of mentions of the particular relation of interest. For example, to extract biomedical relations such as **protein-protein_interaction** or **gene_associated_with_disease**, it is preferable to use biomedical documents rather than newswire articles. Relations that have limited and narrow

scope and are most likely to gain benefit from the use of similar domain documents in the system, are referred to as *domain-specific* relations.

On the other hand, there are relations that can be extracted from any domain or grouping of domains. These relations tend to be broad-coverage factual information that are applicable to a large number of domains. They are often referred to as *domain-independent* relations. Some examples are `hypernym` and `location_of`.

This distinction, however, does not imply that the choice of documents in an RE task is or should always be dictated by whether a relation is domain-specific or domain-independent. In practice, it is possible to perform RE on any relation using documents from any domain, regardless of whether the relation is domain-specific or domain-independent. The choice of the documents (and their domain) in accordance with the relation of interest is simply to maximise the benefit and applicability of employing an RE system.

Functional Attribute

A binary relation `rel` between two entities: *entity-1* and *entity-2* is said to be functional if *entity-1* is mapped to exactly one unique value of *entity-2* (Ritter *et al.* 2008; Lin *et al.* 2010). An example of a functional relation is `person-biological_father`, which expresses the relation that there is only one biological father for a particular person. While theoretically simple, detecting the functionality of a textual relation is challenging due to several linguistic phenomena. We illustrate the challenges with an example below.

Consider the functional relation `was-born-in`, and the mentions `was-born-in(Mozart, Salzburg)`, `was-born-in(Mozart, Salzburg, Austria)`, `was-born-in(Mozart, Austria)`, and `was-born-in(Mozart, 1756)`. First, we can identify the problem of different argument types for the second argument, where the semantic of the second arguments are different between *Salzburg* (city), *Austria* (country) and *1756* (birth year). In this sense, the functional relation is distinguished at the semantic-level rather than at the word-level. Another problem is synonymy where *Salzburg* and *Salzburg, Austria* refer to the same entity, but at the token-level they are not the same.

Temporal Attribute

A relation is said to have a temporal aspect if we can associate a specific time to its relation mentions. Relations with a temporal aspect usually express a certain event. For example, we can associate a certain time with a composer writing a particular composition. On the other hand, semantic relations such as **synonym** or **antonym** do not have temporal attributes.

Tasks for detecting temporal attributes of a given event relation have been formulated. The first run of ACE 2004 introduced this task by detecting a temporal attribute of a relation based on explicit evidence in text (Mitchell *et al.* 2003). The temporal attributes are expressed as a TIMEX2 tag,³ and can be classified into three categories: absolute time (e.g. *2 January 2004*), relative time (e.g. *before 5 May 2010*), or unspecified time (e.g. *eight years*).

The Knowledge Base Population (KBP) track at TAC 2011 proposed a similar task where it requires participants to always decode temporal expressions into a date format (e.g. *20051231* for *31 December 2005*) (TAC 2011). This representation, however, fails to capture temporal information that does not explicitly state the date information (e.g. *I have been working here for 192 days*) or fuzzy temporal expressions (e.g. *I went to England recently*) in text.

There are also different tasks targeted at determining temporal relations between pairs of events. Unlike the examples proposed in ACE or TAC, the task interest does not lie in identifying a specific time associated with an event. Instead, it focuses on classifying when a certain event occurred relative to a second event (Mani *et al.* 2006), e.g. whether an event happened **before**, or **after**, or **simultaneous** to the other event. The classification of the temporal relations is based on the annotation of the TIMEBANK corpus (Pustejovsky *et al.* (2003)).

2.2.2 Styles of RE

In this section, we compare and contrast different learning styles of RE. First, we compare the different RE styles due to the distinction and usage of explicit and

³<http://timex2.mitre.org/>

implicit evidence. Following this, we look at the distinction on the extraction level, where we contrast extractions on the type and token-levels. Next, we look at the role of redundancy in RE, and how it impacts on the learning style of an RE system. Finally, we discuss the difference between RE tasks that require a prespecified relation as an input to their systems, as contrasted with tasks that do not.

Explicit vs. Implicit Evidence-Based RE

All relations in RE require textual evidence as the basis for extraction. Textual evidence can be divided into two types: *explicit* and *implicit* evidence (Mitchell *et al.* 2003). Evidence is considered explicit if there is an explicit mention expressing the relations between the two entities in the text. The entity mentions must also be explicit, whereas an anaphora mention is considered to not be explicit. Consider the following sentences:

Beethoven is a composer. He is also a pianist.

There is explicit evidence expressing *is-a*(*Beethoven*, *composer*) in the example sentences above. However, there is no explicit evidence of a relational between *Beethoven* and *pianist*. One needs to realise that *He* refers to *Beethoven* (using *coreference resolution*) to infer *is-a*(*Beethoven*, *pianist*).

An anaphor is an example of implicit evidence, where the textual evidence is not explicit, and contextual inference is required to resolve the unclear connection. Consider extracting *is-a* relations from the sentence:

Beethoven went to Vienna in 1792 to further establish himself as a composer, not knowing that he was destined to continue the legacy of Haydn and Mozart who have previously triumphed there.

In this sentence, the evidence for inferring *is-a*(*Beethoven*, *composer*) is explicit. On the other hand, there is no explicit textual evidence to infer *is-a*(*Haydn*, *composer*) and *is-a*(*Mozart*, *composer*), although such relations are clear once one considers the context of the sentence.

Performing implicit-evidence-based RE presents challenges during feature extraction and system evaluation. For the purposes of both feature extraction and system evaluation, one must find and annotate the contextual inferences latent in the corpus. This proves to be a big barrier as manually annotating a corpus (especially a large-size corpus) is a time-and resource-intensive process. An alternative to manual annotation is employing a state-of-the-art automatic contextual inference system, which is more efficient, but less accurate compared to human annotation. As an indication, the performance of state-of-the-art anaphora resolution system is around 60% accuracy evaluated against human annotations (Mitkov and Hallett 2007; Pradhan *et al.* 2011). Considering these difficulties, it is unsurprising that most research in RE (including this thesis) assumes explicit evidence as the basis for extraction.

Type vs. Token-Level RE

One can perform RE on the *type-level* or *token-level*. To illustrate the difference between them, consider predicting the given relation **rel** label for a certain entity pair (*entity-i*, *entity-j*) which occurs X times in the corpora. Type-level extraction only produces a single prediction for the mention **rel**(*entity-i*, *entity-j*) using the evidence collected from all X occurrences (which is the focus of this thesis). In contrast, token-level extraction produces different predictions for each mention of **rel**(*entity-i*, *entity-j*), based on the evidence collected from the individual mention. Figure 2.1 highlights this difference.

The main factor for performing RE at a type or token-level depends on the importance of specific individual evidences as an extraction basis for the associated task. A token-level RE is necessary in a task where specific individual evidences are necessary for extraction, e.g. detecting a **protein_mutation_event** in the biomedical domain (Cohen *et al.* 2009), where the relation is valid when described in a certain way, but not valid in other contexts. Hendrickx *et al.* (2010) refer to this kind of task as being *in context*.

In contrast, a type-level RE is more suited for a task where the individual evidences

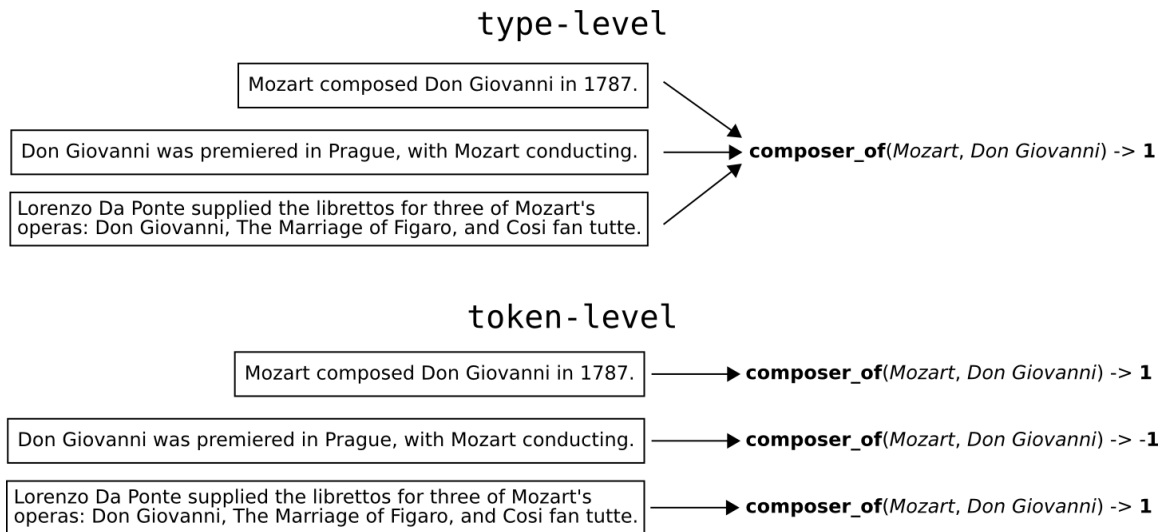


Figure 2.1: The difference between type and token-level RE

are not as important, and the presence of one positive evidence is enough as the basis for extraction. For example, consider the type level task in Figure 2.1, which is interested in detecting a relation between a composer and the name of one of their compositions. One positive evidence for *Mozart* and *Don Giovanni* in the documents is enough as an extraction basis for **composer_of(Mozart, Don Giovanni)**. Hendrickx *et al.* (2010) refer to this kind of task as being *out of context*.

This distinction plays a large role in the annotation effort for both tasks. In a token-level task, one is required to annotate every mention of a given entity pair in the documents, which can be very resource-expensive for a large-scale document collection. This is because each individual mention of that entity pair must be evaluated in context, based on the specific evidence individually associated with that mention. As such, there is the possibility of assigning different labels for different mentions. Due to the expensive annotation process, the token-level task is usually applied to a small-sized document collection.

A similar effort is unnecessary for type-level RE due to its task assumption. As a result, type-level tasks usually consult a knowledge base to determine the label of an entity pair, without verifying in the documents that there is an evidence to support the labelling as provided by the knowledge base. In other words, if a knowledge base

labels an entity pair to have a certain relation, type-level RE assumes that there is at least one evidence in the data describing the relation between that entity pairs, and vice versa. By consulting a knowledge base, the annotation effort for type-level RE is comparably less expensive than token-level RE.

The difference in the task assumptions presents different challenges for token-level and type-level RE. The challenge in a token-level RE is to make the most information out of the context that an entity pair appears in as a basis for extraction. Therefore, token-level systems usually focus on a deeper linguistic and domain analysis of the given context.

On the other hand, type-level RE is usually applied to large-scale document collections, in order to find an evidence for extraction. This is based on the hypothesis of *information redundancy*, where the evidence to extract a certain relation mention can be expressed in various ways and multiple times in the documents (Poon *et al.* 2010). Therefore, the larger the document collection is, the more redundant information there is, and subsequently, there is a higher chance to find quality evidences (Etzioni *et al.* 2005; Downey *et al.* 2010). As a result of its dependence on redundant information, the challenge in a type-level RE is to manage and sift through the potentially large amount of information when predicting relation labels. Further, unlike the token-level task, there is a possibility that type-level task has to predict a relation label for an entity pair without any evidence at all. This can happen when the document collections used for extraction do not have any mentions of that entity pair.

Single-Document vs. Multi-Document RE

MUC and ACE defined RE as the task of identifying relations between entities within a single document. We refer to this task design as *single-document RE*. The single-document RE task is thus limited only to the information available in the given document.

With the availability of large text corpora (e.g. web documents or large collections of newspaper articles), researchers have developed a new style of RE exploiting their

information abundance. This new style of RE treats large collection of documents as a single extraction source, which we refer to as *multi-document RE*.

The distinction between single-document and multi-document RE is closely related to that between token and type-level RE. The practicality of token-level annotation is relative to the size of the documents used. One cannot be reasonably expected to perform token-level annotation for corpora greater than 1 GB in size. Therefore, it is unsurprising that single-document and multi-document RE is usually at the token-level and type-level respectively.

Single-document RE tends to be applied to a group of documents from the same domain, where the domain is known in advance. Further, the relations are usually domain-specific. In contrast, documents provided to multi-document RE tend to be from a number of different domains and sources. In this light, multi-document RE usually focuses on domain-independent (broad-coverage) factual relations.

Relation-Specific vs. Relation-Independent RE

So far, we have assumed that the relations provided to an RE task are specified prior to learning. We refer to this style of learning as *relation-specific* RE. In this setting, individual systems are built for each relation of interest. These systems can (and usually do) use, exploit, or learn knowledge that is specific to the particular relation.

The limitation of relation-specific RE systems is their lack of portability. As they are built specifically for a particular relation, they are not easily adaptable to other relations. Additional effort to acquire new extraction patterns, training examples for the new relations, and retraining the systems are usually required in order to adapt them to new relations (Sekine 2006; Banko and Etzioni 2008).

Recently, researchers have argued that current relation-specific RE is ineffective in extracting information from large-scale corpora which contain a large number of different relations, as relation-specific RE systems require repurposing for each relation it intends to learn (Banko and Etzioni 2008; Hoffmann *et al.* 2010; Wu and Weld 2010). As an alternative, they proposed a different learning style that

is capable of extracting as many relations as possible without requiring that the relations be specified in advance. We refer to this style of learning is referred to as *relation-independent* RE.

As the relations are not provided as input, relation-independent RE systems need to learn how to recognise the existence of various relations, and in turn, label them. This adds another dimension to relation-independent RE not found in relation-specific RE. Also, as it can identify and label unseen relations, relation-independent RE systems do not need to be repurposed for different relation types.

Banko and Etzioni (2008) argued that relation-independent RE systems are suited to tasks where the objective is to find as many interesting relations as possible from large-scale corpora. On the other hand, relation-specific RE systems are more appropriate for targeted extraction, in order to find as many instances of a relation type from text. We discuss various approaches to relation-independent RE in Section 2.3.5.

2.2.3 Our RE Task

In this thesis, we are interested in looking at lexical semantic relations such as **hypernymy**, **synonymy**, and **antonymy** (see Section 3.1 for a detailed description of these relations). These relations are binary and domain-independent, and are not functional or temporal in nature. **Antonymy** and **synonymy** are symmetric and flat relations, while **hypernymy** is an asymmetric and hierarchical relation.

In terms of style, our research task is based on explicit evidence found in a large collection of documents (multi-document RE). The annotation and extraction in our experiments is performed at the type-level. The system built will be relation-specific, i.e. there will be a system for every relation that we are interested in. However, the overall procedure in building the different individual RE systems is the same.

2.3 RE Approaches

In this section, we discuss various approaches proposed to address RE, focusing on techniques related to our RE task.

2.3.1 Handcrafting Lexico-syntactic Patterns

The idea of using patterns for RE originated from the work of Hearst (1992) in automatically acquiring hyponym (or inversely, hypernym) pairs from text. The basis of her work is the hypothesis that there exists a set of lexical relations that are linguistically expressed in frequently used and unambiguous lexico-syntactic patterns, and consequently, it is possible to automatically extract such relations from text using these patterns. Hearst identified six lexico-syntactic patterns which she exploited to extract hyponym pairs from text:

1. NP_0 **such as** $NP_1, NP_2, \dots$, **and** | **or** NP_n
2. **such** NP_0 **as** $NP_1, NP_2, \dots$, **and** | **or** NP_n
3. $NP_1, NP_2, \dots, NP_n$, **and** | **or other** NP_0
4. NP_0 , **including** $NP_1, NP_2, \dots$, **and** | **or** NP_n
5. NP_0 , **especially** $NP_1, NP_2, \dots$, **and** | **or** NP_n

In these patterns, the noun phrases $NP_1, NP_2, \dots, NP_n$ are identified as the hyponyms of the noun phrase NP_0 . Hearst discovered the first three patterns by observing the contexts of hyponym mentions in text. Next, she used *is-a(England, country)* as the *seed* example and discovered pattern 4 and 5 based on her manual observations that these patterns often cooccur together in context with the seed example. Following this, Hearst used a new seed *is-a(tank, vehicle)* to discover pattern 6.

Hearst demonstrated the utility of these six patterns, by automatically extracting 152 possible relations from GROLIER'S AMERICAN ACADEMIC ENCYCLOPEDIA (Grolier 1990), and then evaluating the quality of these relations by comparing them to the WORDNET noun hierarchy. The relation set contained 226 unique words, of which 180 were in WORDNET. WORDNET contained 106 hyponym relations for these 180 word pairs, of which Hearst's system successfully identified 61 relations.

Hearst, however, did not achieve similar success when applying her method to meronymy (part-whole relation). The main problem with meronymy was that the

patterns found using her first method are ambiguous in that they do not tend to uniquely identify meronymy and can express relations other than meronymy (Hearst 1992).

Although highly accurate, Hearst’s hyponym patterns do not always produce correct relation pairs, as one (or more) of the patterns may be used to express other relations. For example, pattern 5 incorrectly identifies a hyponym relation between *family* and *child* for the sentence *Michael brought his entire family, including his newly born third child, to visit his dad in Germany last summer*, where a meronymy relation is more appropriate.

Further, Hearst’s approach tends to suffer from low coverage. This is due to the fact that the approach is not able to detect valid relation pairs that are not expressed in one of the identified patterns (Cederberg and Widdows 2003; Yap and Baldwin 2009).

Postprocessing of Extracted Instances

One of the avenues researchers have taken to improve Hearst’s patterns is to improve the accuracy and the coverage of the RE system by postprocessing the initial extracted instances (Berland and Charniak 1999; Girju *et al.* 2003; Cederberg and Widdows 2003). Examples of the postprocessing strategies are ranking of the extracted instances and picking the top- N ranked instances, requiring the extracted instances to be in a semantic category, or expanding the coverage of extracted instances by inference.

Berland and Charniak (1999) crafted a set of patterns for meronymy based on the Hearst (1992) methodology and used them to extract meronymy pairs from text. Their task is evaluated by examining the accuracy of the extracted “part” objects relative to each of their six seed “whole” objects (*book*, *building*, *car*, *hospital*, *plant*, and *school*). Using these manually created patterns, they noticed that the patterns proposed are not always error-free, i.e. some patterns extracted part objects that are not related to the whole object. To address this, Berland and Charniak (1999) proposed postprocessing extracted list by (i) removing candidates which contain words

ending in *-ing*, *-ness*, or *-ity*, as such words typically indicate a quality rather than physical object (meronymy usually contains relations between physical objects); and (ii) ranking the extracted part objects individually relative to their corresponding whole objects according to some statistical metrics (e.g. the log-likelihood ratio (Dunning 1993)).

Girju *et al.* (2003) argued that some of the lexico-syntactic patterns used by Berland and Charniak (1999) are ambiguous, and are not specific enough to differentiate the meronymy relation from other relations. However, they argued that the ambiguity of such patterns can be reduced by placing a semantic constraint on the noun phrases appearing in the lexico-syntactic patterns. For example, using one of the extraction patterns for meronymy NP_0 **of a** NP_1 where NP_0 is the part object, and NP_1 is the whole object, it is possible to avoid extracting the incorrect pair **meronymy**(*Tackling*, *game*) from the sentence *Tackling is part of a game* by requiring that the semantic class of NP_0 must be a **location** and NP_1 must be an **object**.

To remove the manual effort in identifying these semantic constraints, Girju *et al.* (2003) proposed to use machine learning. Using a set of training noun pairs examples tagged with a WORDNET semantic class, the system successfully learnt more than 20 semantic constraints that are either strongly indicative or, conversely, strongly non-indicative of a part-whole relation.

Cederberg and Widdows (2003) aimed to improve the precision of Hearst’s approach based on the hypothesis that a hyponym and its hypernym are semantically similar to each other. They proposed to use Latent Semantic Analysis (LSA) (Deerwester *et al.* 1990; Schütze 1998) to measure such similarity between candidate pairs for extraction, and exclude pairs that have low LSA similarity score.

Glozzo *et al.* (2007) hypothesised that terms extracted from a corpus tend to be semantically relevant. Assuming domain information about a document, they used LSA to measure the domain similarity score between each of extracted instances, where extracted instances that tend to cooccur in the same domain’s documents are assigned higher score than instances that do not. For example in the chemistry domain, **is-a**(*ethanol*, *alcohol*) is assigned a higher score than **is-a**(*ethanol*, *entity*), as *ethanol* and *alcohol* are from the chemistry domain, while *entity* is a generic term

that can be used in any domain. A threshold-based filtering can then be applied to remove incorrect noun pairs from the extractions.

Cederberg and Widdows (2003) also proposed a method to improve the coverage of a hyponymy RE system, motivated by the fact that not all of the relation pairs are expressed in one of Hearst’s patterns. Their expansion method focuses on an inference technique based on word semantic similarity. Consider the following sentences:

The influence of Mahler’s compositional style can be traced in the compositions of major 20th century composers such as Schoenberg, Webern, Berg, Shostakovich, and Britten.

and

Leonard Bernstein’s vast repertoire includes his critically-acclaimed recordings of symphonies by Beethoven, Mahler, Sibelius, and Shostakovich, among others.

In the first sentence, there is explicit evidence for *Shostakovich* being a *composer*. In contrast, there is no explicit evidence in the second sentence that *Beethoven*, *Mahler*, *Sibelius*, and *Shostakovich* are composers, although based on the context they appear in, one can infer that there is some semantic similarity between *Beethoven*, *Mahler*, *Sibelius*, and *Shostakovich*.

Cederberg and Widdows (2003) proposed to combine the prior knowledge extracted from the first sentence that *Shostakovich* is a *composer*, with the information that *Beethoven*, *Mahler*, and *Sibelius* are similar to *Shostakovich*, to infer that *Beethoven*, *Mahler*, and *Sibelius* are composers. In order to analyse word semantic similarity, they used techniques developed to automatically build semantic lexicons via bootstrapping from a set of seed examples (Riloff and Shepherd (1997), Roark and Charniak (1998), and Widdows and Dorow (2002), for example). We discuss bootstrapping techniques further in Section 2.3.2.

2.3.2 Automatic Bootstrapping Approaches

A different extension to the Hearst (1992)’s approach to improve its low coverage is to automatically bootstrap a set of extraction patterns for a particular relation of

interest from a set of seed examples. The idea behind this approach is to iteratively identify more patterns that are highly associated with a particular relation, and use these patterns to extract more relation pairs occurring in such patterns from text. This is in contrast to Cederberg and Widdows (2003)’s approach, where the expansion is based on word semantic similarity and inference.

The main motivation for automatic bootstrapping approaches is to remove the manual effort in identifying useful extraction patterns. The manual pattern identification process in RE is time consuming, and requires domain-specific knowledge of the relation of interest. In the automatic bootstrapping approach, this manual effort is replaced with various ranking strategies to ensure that only high quality patterns are picked. In doing so, it also offers greater portability as it can be easily adapted to different domains and relations.

The use of bootstrapping in RE stems from its successful application in automatic semantic lexicon construction. In terms of RE, we can view semantic lexicon construction as a *unary relation* task, where one of the main objectives is to identify instances that are similar to the seeds used to initiate the iteration procedure. Some prominent examples of automatic semantic lexicon construction systems using the bootstrapping approach are Riloff and Shepherd (1997), Riloff and Jones (1999), Paşca *et al.* (2006), and McIntosh and Curran (2008).

Figure 2.2 illustrates a standard bootstrapping framework used for RE. This particular framework is often referred to as the *dual extraction model* (Li *et al.* 2011). The input to the system is a set of seed pairs (or patterns). Assuming seed pairs as the input to the bootstrapping system, the next step is to find occurrences of such pairs in text. Following this, the system generalises the context of the pairs. These patterns are then ranked and filtered by some criteria, and only high quality patterns are retained as extraction patterns.

The system then scans these high quality patterns from text, and uses them to identify more candidate pairs expressed in those patterns. Similar to the patterns, these candidate pairs are subject to a rank and filter step, and only high quality pairs are retained as the system’s output. The iterative process is then repeated until a stopping criteria is met.

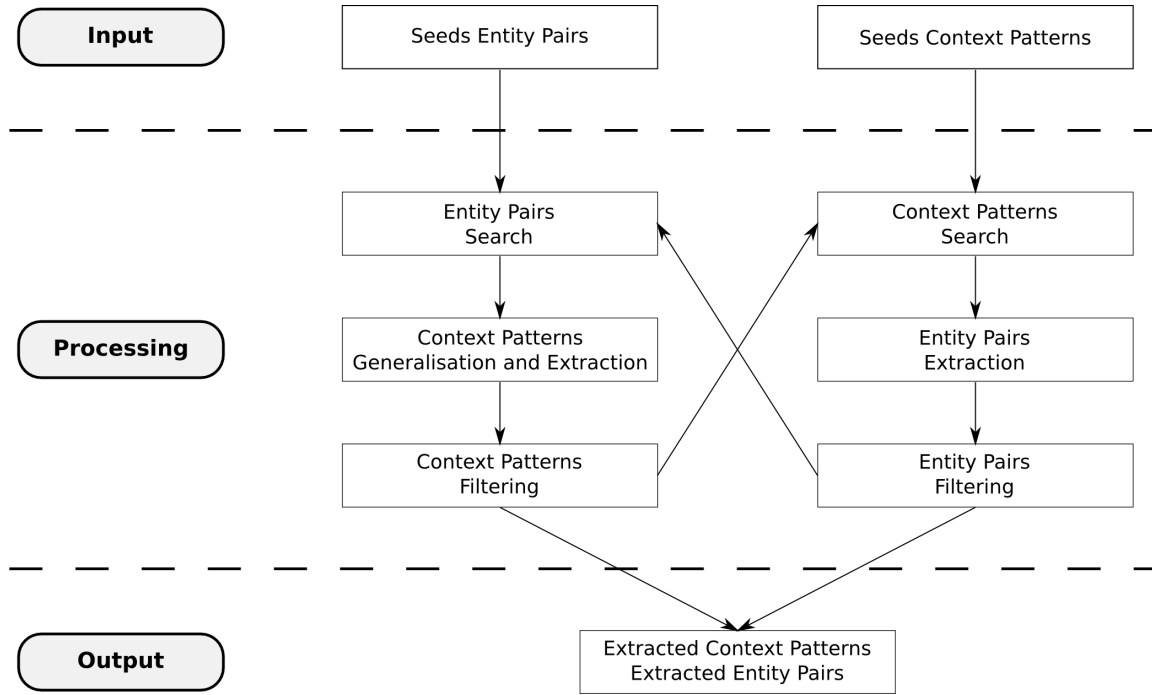


Figure 2.2: Framework of a bootstrapping RE system (figure adapted from Li *et al.* (2011))

There exists a considerable number of systems that adhere to this framework, most prominently: DIPRE (Brin 1998), Snowball (Agichtein and Gravano 2000), and Espresso (Pantel and Pennacchiotti 2006). Differences between bootstrapping systems are mainly attributable to two important aspects of the framework: the filtering processes of extraction patterns and candidate pairs.

We list five general types of pattern filtering methodology below. The first four methodologies are categorised by Blohm *et al.* (2007),⁴ while the fifth is novel to this thesis:

1. *Syntactic assessment*: the filtering process is based purely on syntactic criteria.

For example: Brin (1998) uses the pattern length as a component in his filtering process. Pantel *et al.* (2004) employ minimal edit distance (of insertions,

⁴In their paper, Blohm *et al.* (2007) listed five methodologies which consists of *inter-pattern comparison* methodology in addition to the four methodologies described here. However, we merge the *inter-pattern comparison* and *performance-based assessment* as the difference between them is unclear.

deletions, and replacements) in generating and filtering extraction patterns.

2. *Support-based assessment*: rank the quality of new patterns by counting the number of seed pairs that it can successfully extract. For example: Brin (1998), Agichtein and Gravano (2000), Ravichandran and Hovy (2002) and McIntosh and Curran (2008) require that their new patterns are supported by a minimum number of seed pairs.
3. *Performance-based assessment*: rank the quality of new patterns by comparing an iteration's outputs to that of the previous iterations. For example: Agichtein and Gravano (2000) assign a pattern confidence score by calculating the percentage of correctly extracted pairs using that particular pattern in a given iteration with the set of known extracted pairs from the previous iteration.
4. *Instance-Pattern correlation*: rank the quality of new patterns based on their correlation with the context of pairs known to correspond to the relation of interest. This usually involves calculation of a correlation statistic between the pairs and the patterns, such as: log-likelihood (in Pantel *et al.* (2004)), Point-wise Mutual Information (PMI) (in Etzioni *et al.* (2005), Pantel and Pennacchiotti (2006), and McIntosh (2010)), or chi-square distribution (in McIntosh *et al.* (2011)).
5. *Semantic assessment*: the filtering process is based on the semantic similarity of the patterns. For example: Stevenson and Greenwood (2005) and Stevenson and Greenwood (2006) rank new patterns based on their semantic similarity to a previously known set of seed patterns. Paşca *et al.* (2006) rank patterns highly if they contain a term commonly found in the set of candidate patterns.

Some RE systems use more than one filtering method to control the quality of the patterns. Blohm *et al.* (2007) investigated the effectiveness of several pattern filtering methods. In their experiment, they looked at the performance of performance-based assessment (as implemented by Agichtein and Gravano (2000)), instance-pattern correlation (PMI metrics as implemented by Etzioni *et al.* (2005) and Pantel and Pennacchiotti (2006)), a simple metric based on support-based assessment, and a random

filtering process. Their conclusion is surprising in that the instance-pattern correlation methods (e.g. the PMI metric) that require considerable calculation do not outperform random filtering at a level of statistical significance.

Similar to pattern filtering, there exists a considerable number of pair filtering methods. Overall, they are quite similar to the five general types of pattern filtering methods listed above. Some of them depend on syntactic or/and semantic assessment, in which the candidate pairs are required to correspond to particular named entity types (Agichtein and Gravano 2000), or a semantic class (Girju *et al.* 2003). There are other utilities that depend on instance-pattern correlation (Riloff and Jones 1999; Pantel and Pennacchiotti 2006), or support-based assessment (Agichtein and Gravano 2000; Xu *et al.* 2007) where pairs are ranked based on the scores of patterns that produce them. Paşca *et al.* (2006) filter their candidate pairs by measuring their similarity to the seed pairs, which involves the calculation of support-based assessment, instance-pattern correlation and syntactic assessment. Finally, Li *et al.* (2011) propose a graph-based method to filter candidate pairs that combines the components of instance similarity, pattern similarity, and instance-pattern correlation.

Semantic Drift

One of the main causes of extracting incorrect pairs and patterns in the bootstrapping process is lexical ambiguity. For example, in using (*flower, plant*) as a seed example to extract hyponym pairs, the ambiguous term *plant* may introduce error in the bootstrapping process as occurrences of *plant* in the sense of *manufacturing plant* may be extracted, rather than *organism plant*. Lita and Carbonell (2004) termed this phenomenon *semantic drift*. Once extracted, ambiguous terms can quickly infect the data leading the system to extract more incorrect patterns or pairs in subsequent iterations.

As the majority of bootstrapping systems are prone to semantic drift (Curran *et al.* 2007; Xu *et al.* 2009), it is important to block this phenomenon from occurring in the early iterations. The pattern and pair filtering processes described above are examples of procedures to delay the occurrence of semantic drift.

In order to reduce semantic drift, Kozareva *et al.* (2008) proposed hypernym extraction of a given semantic class via bootstrapping using *doubly-anchored* patterns which use information on a semantic class and a seed example to form an extraction pattern. The extraction pattern is formed based on one of the Hearst’s patterns, producing the pattern *semantic_class_name* **such as** *seed_class_member* *and* *, where * are other instances similar to *seed_class_member* that belongs to the *semantic_class_name*. For example, by using a doubly-anchored pattern *presidents* **such as** *Ford* *and* *, the system reduces the chance of semantic drift in extracting instances of car manufacturers, while maintaining the system’s precision in extracting presidents’ names at the same time.

There are other bootstrapping frameworks that are designed to reduce the impact of semantic drift. A popular approach is multi-category bootstrapping which involves extraction of multiple semantic categories or relations simultaneously. The idea is to detect competition between pairs and patterns in order to direct the relations instances away from each other (McIntosh 2010). That is, if there is a relation instance that has been identified by multiple relations as a candidate for extraction, the instance is *excluded* from extractions as it may be ambiguous, and hence, introduces semantic drift. Multi-category bootstrapping methods, such as the ones proposed in Thelen and Riloff (2002), Yangarber (2003), Downey *et al.* (2004), Curran *et al.* (2007), McIntosh and Curran (2009) and Carlson *et al.* (2010), have been shown to outperform single category bootstrapping methods that solely rely on several patterns and pair filtering processes at reducing the impact of semantic drift.

Finally, for bootstrapping tasks, there is no strict rule whether one should initiate the bootstrapping process with seed patterns or pairs.⁵ Orthogonal to this, there is little or no guidance in selecting seeds (pairs or patterns) that can benefit the system the most, although it has been shown that the success of bootstrapping approaches depends on the seed pairs used (McIntosh and Curran 2009). In fact, most researchers tend to use seeds selected based on the annotators’ intuition. This

⁵Xu *et al.* (2009) compared the output of their bootstrapping system that starts with a set of seed patterns to the output of the same system but initiated with a set of seed pairs. They found a high level of overlap between the top-ranked output of the two, and concluded that both approaches produced similar results.

may cause problem in evaluating a system’s true performance, or when comparing it against other systems. McIntosh (2010) proposed to investigate the bootstrapping algorithm’s sensitivity to the initial seed pairs, and their true performance by evaluating them multiple times, each using different sets of seed pairs.

2.3.3 Supervised Learning

A variant on bootstrapping approaches is to learn patterns for extraction by supervised learning. The majority of supervised learning approaches focus on *feature engineering* to train high-quality classifiers (Culotta and Sorensen 2004; Kambhatla 2004; Zhou *et al.* 2005; Snow *et al.* 2005; Turney 2008; Ritter *et al.* 2009). The main features used are based on the frequency of pattern-pair cooccurrence (Snow *et al.* 2005; Turney 2008). Other features include using lexical, syntactic, and semantic features not found in the pattern representation, such as part-of-speech, word n-grams, and capitalisation (Culotta and Sorensen 2004; Kambhatla 2004; Zhou *et al.* 2005; McNamee *et al.* 2008; Ritter *et al.* 2009). These features are then used by various machine learners, such as logistic regression, decision trees, support vector machines (SVMs), and conditional random fields (CRFs).

Other supervised learning research focuses on using *kernel methods* (Vapnik 1995; Cristianini and Shawe-Taylor 2000). Instead of engineering features, kernel-based methods map the pattern representations into a set of kernels for each training instance. Next, a kernel function is calculated to compute the similarity between two instances via dynamic programming. Zelenko *et al.* (2003) and Culotta and Sorensen (2004) argue that kernel-based approaches implicitly explore a feature space larger than feature-engineering approaches. Several kernel representations have been used in RE tasks, such as a shallow parse kernel (Zelenko *et al.* 2003), a dependency tree kernel (Culotta and Sorensen 2004), and a subsequence kernel (Bunescu and Mooney 2007).

Snow *et al.* (2005)’s methodology is an example of a supervised approach to RE. They proposed a supervised machine learning system that automatically learns useful patterns for hypernymy based on annotated corpus data. Using a set of both positive

and negative instances, they collect all patterns cooccurring with the noun pairs and calculate the frequency of their cooccurrence within the corpora.

It is reasonable to expect that not all of the collected patterns are indicative of the relation of interest. To learn the utility of each collected pattern, Snow *et al.* (2005) used a logistic regression-based learner on the frequency of the collected pattern-pair cooccurrence. Patterns which cooccur frequently with positive examples are assigned more weight, and conversely, patterns which cooccur with negative examples are weighted down.

Following the Snow *et al.* (2005) methodology, we can break down the major steps of supervised learning into:

1. Training:

- a. Collect entity pairs from corpora. Label whether an entity pair has a relation or not based on a gold-standard.
- b. Collect sentences in which the pairs cooccur.
- c. Extract the features of entity pairs based on these sentences.
- d. Train a classifier based on these features.

2. Test:

- a. Classify entity pairs in the test set using the classifier.

One novel aspect of supervised learning is the role of negative examples during the learning process. It provides an additional dimension not explored by the traditional pattern-based and bootstrapping approaches, which usually only learn from positive examples (with the exception of multi-category bootstrapping-based systems such as Yangarber (2003), Downey *et al.* (2004), Curran *et al.* (2007), and McIntosh and Curran (2009) that use positive examples of one relation to be negative examples for all other relations).

The classification in a supervised system is more flexible as compared to the traditional and bootstrapping approaches, as it relies on weighing up all patterns (positive and negative) that a particular pair cooccurs with as evidence for extraction (Snow

et al. 2005; Culotta *et al.* 2006). This is in contrast with a traditional approach that performs classification based solely on whether an entity pair appears with a given set of (positive) lexico-syntactic patterns or not. This property is arguably very useful when learning relations that are linguistically expressed in ambiguous patterns, e.g. qualia structure (Yamada *et al.* 2007). Finally, supervised learning allows for the possibility of fine-grained control over the learner, via parameter turning, for example.

One of the major disadvantages of supervised learning is the requirement of having training data. Training data for certain relations can be difficult to acquire, and further, the process of labelling the training data itself is expensive (Mintz *et al.* 2009; Jiang 2009). These two factors limit the adaptability of a supervised RE system to another domain. Related to this, as it requires training data for each relation, supervised learning has attracted criticism that it cannot be easily scaled to extract the large numbers of relations that can be found in large-scale corpora.

Jiang (2009) attempted to address the domain adaptability problem in RE by *transfer learning*, following the works of, for example, Blitzer *et al.* (2006) and Daume III (2007) that successfully apply transfer learning to other NLP tasks. Jiang (2009) studied how one can leverage training data for other relation types to aid in the extraction of a new relation type, for which training data is scarce. This is based on the observation that different relation types share certain common syntactic structures, where if such knowledge is learned in several relations, it can be transferred to a new relation.

Recently, *distant supervision* has generated a lot of interest in overcoming the problem of acquiring training data. In the traditional sense (as implemented by Wu and Weld (2007) and Hoffmann *et al.* (2010)), the idea of distant supervision is to exploit information from documents that contain descriptions of a particular entity and a summary of the document-level relational information (expressed as relation tuples) as a potential source for data gathering.

Both Wu and Weld (2007) and Hoffmann *et al.* (2010) used Wikipedia articles in their experiments, as by nature, Wikipedia articles contain descriptions of a particular entity. However, not all Wikipedia articles contain a document-level relational information summary in the form of Wikipedia *infoboxes*. As distant supervision requires

the existence of such summary, previous research only exploit Wikipedia articles with infoboxes.

Using the Wikipedia article of *Australia* as an example, we can find the document-level relation information summarised in the article’s infobox (snippet shown in Figure 2.3), in the form of *Attribute-Value* pair, e.g. *Capital-Canberra*. Therefore, we can think of a Wikipedia infobox *Attribute* as a relation. While this information is explicitly unary relation between *Capital* and *Canberra*, it is implicitly expressing a binary relation between the infobox *Attribute-Value* pair with the article’s title, i.e. $\text{Capital}(\text{Australia}, \text{Canberra})$.

To construct the training data, they align sentences in Wikipedia articles with their corresponding infoboxes information. An article sentence that contains one of the infoboxes attributes and values can be used as a positive example for that particular attribute (relation). Conversely, sentences that do not contain such information can be used as negative examples. As pointed out by Wu and Weld (2007), the alignment process is not without its own challenges. Issues such as strings denoting identical semantic tags (e.g. *Census yr* vs. *Census Year*, conflicting information between sentences and infoboxes, or attributes containing several values strings (e.g. *Australian* and *Aussie* are both listed as the demonyms of *Australia*) may effect the quality of the produced training data.

Wikipedia contains a large number of articles with infoboxes, and each infobox contains different relational information. From this, it is possible to learn a classifier for each Wikipedia attribute by iteratively gathering training sentences for each attribute from Wikipedia articles. As a result, distant supervision using Wikipedia may produce large amount of training data for a range of different relations, which may ease the bottleneck problem of hand-labelling data for supervised training.

The traditional implementation of distant supervision requires collection of documents each describing a particular entity, a summary of that document in the form of relational tuples, and the alignment of sentences in the documents to the summary. These requirements prevent a large number of traditional RE resources, such as newswire articles, to be used as a potential source for distant supervision as they neither describe a particular entity nor have an infobox-like summary.

Commonwealth of Australia	
	
Flag	Coat of arms
Anthem: "Advance Australia Fair" ^[N 1]	
	
Capital	Canberra
Largest city	Sydney
Official language(s)	None ^[N 2]
National language	English (<i>de facto</i>) ^[N 2]
Demonym	Australian, Aussie ^{[3][4]}
Government	Federal parliamentary democracy and constitutional monarchy
- Monarch	Elizabeth II
- Governor-General	Quentin Bryce
- Prime Minister	Julia Gillard
Legislature	Parliament
- Upper House	Senate
- Lower House	House of Representatives
Independence	from the United Kingdom
- Constitution	1 January 1901
- Statute of Westminster	11 December 1931
- Statute of Westminster Adoption Act	9 October 1942 (with effect from 3 September 1939)
- Australia Act	3 March 1986

Figure 2.3: Snippet of the infobox of *Australia* Wikipedia article

In light of these constraints, recent research adopts a weaker form of distant supervision that removes some of the above requirements. As most documents do not have a infobox-like summary, relation-rich resources such as FREEBASE (Bollacker *et al.* 2008) and YAGO (Suchanek *et al.* 2007) are used as substitutes. Unlike the infoboxes, these resources are not tied to particular documents, resources cannot be

aligned to the documents' sentences in the same fashion as the traditional distant supervision systems. In this form, the traditional alignment process is compensated by allowing any sentences from any documents to be heuristically aligned to the relation-rich resources used as a possible source for training data (Mintz *et al.* 2009; Yao *et al.* 2010; Nguyen and Moschitti 2011). This may lead to a less precise training data collection compared to the data produced by traditional distant supervision method.

In this thesis, we focus on applying a weaker form of distant supervision learning to RE. More specifically, one of our aims is to investigate the extensibility of the supervised learning methodology of Snow *et al.* (2005) by applying it to other relations, using WORDNET (Fellbaum 1998) as our knowledge base. Although the main focus of the thesis will be on the supervised approach, we also incorporate features from automatic bootstrapping approaches in Chapter 6.

2.3.4 Semi-Supervised Approaches

A major weakness of supervised approaches is their heavy reliance on training data, which leads to poor domain adaptation. Semi-supervised RE approaches have been proposed to address this issue. The common theme among these systems is to reduce the need to specifically repurpose an existing RE system for each relation type, and to perform extraction with minimal human supervision, training data, or input. By this definition, automatic bootstrapping can be classified as a semi-supervised approach. We discuss other semi-supervised approaches below.

Some researchers describe some of the approaches below as minimally-supervised, as they usually do not require manually-tagged training data. Instead, such approaches depend on either a set of well-known patterns or a minimal amount of information manually prepared — such as a topic of interest, or various sets of seeds known to represent a particular relation — to generate data for further processing (machine learning or clustering).

There has been some research that has categorised itself as unsupervised approaches to RE, as their systems do not require relations of interest to be specified

Predicate: $\text{starsIn}(\text{MovieActor}, \text{Film})$	
relation labels:	“stars in”, “star of”
<i>MovieActor</i> labels:	“actor”, “movie star”
<i>Film</i> labels:	“film”, “movie”

Table 2.2: A predicate to describe $\text{starsIn}(\text{MovieActor}, \text{Film})$ used in **KnowItAll**

prior to extraction (Etzioni *et al.* 2005; Rosenfeld and Feldman 2006; Yan *et al.* 2009; Min *et al.* 2012). While such approaches can generate training data (semi)-automatically, the extraction systems still need to train on the generated data to perform the extraction. For this reason, we do not consider such approaches to be fully-unsupervised, preferring to refer to them as Relation-Independent RE approaches, which we discuss in Section 2.3.5.

KnowItAll

One distinguished semi-supervised RE system is **KnowItAll** (Etzioni *et al.* 2005). It is unique in self-supervising the extraction bootstrapping process, unlike other systems that rely on seed pairs to initiate the process (as described in Section 2.3.2). It takes as input a set of predicates, containing class labels and relation labels describing the relation of interest (an example for the relation $\text{starsIn}(\text{MovieActor}, \text{Film})$ is shown in Table 2.2).

Using a set of extended Hearst’s patterns, **KnowItAll** collects instances of all entity-classes (e.g. $\{\text{Marlon Brando}, \text{Al Pacino}, \text{Robert Duvall}\} \in \text{MovieActor}$; $\{\text{The Godfather}, \text{To Kill a Mockingbird}\} \in \text{Film}$) participating in the relation as described by the predicate. Next, it connects the two entity-classes using a set of predefined patterns (shown in Table 2.3) to produce *Marlon Brando stars in The Godfather* and other phrases. It then queries these phrases using a web search engine, and applies an extension of the PMI-IR algorithm (Turney 2001) to evaluate their validity.

<i>class1</i>	“is the”	relation	<i>class2</i>
<i>class1</i>	“,”	relation	<i>class2</i>

Table 2.3: Pattern templates for binary relations used in KnowItAll

Clustering Approaches

The basic hypothesis of using clustering to RE is the distributional hypothesis (Harris 1985): words that occur in similar contexts are similar in meaning. In RE, this concept is extended to entity pairs, where entity pairs which appear in similar contexts express a similar relation (Hasegawa *et al.* 2004; Pantel and Ravichandran 2004; Yamada *et al.* 2009). Therefore, it is possible to cluster entity pairs expressing a similar relation based on the similarity of their contexts, using a large corpus as the context cooccurrence source.

Caraballo (1999) applied an RE clustering approach in grouping similar nouns in a cluster before finding a common hypernym for the clusters produced. She clustered nouns based on the similarity of conjunction and apposition features of each noun measured via cosine similarity, based on the hypothesis that nouns in conjunctions and appositions tend to be semantically related. Next, a set of hypernym candidates is nominated for each of these clusters using Hearst’s patterns. Candidates which cooccur the most with the clusters members are then assigned as the labels of the clusters.

Pantel and Ravichandran (2004) extended Caraballo (1999)’s framework by using further syntactic features in representing their contexts and a clustering method termed Clustering by Committee (CBC) (Pantel and Lin 2002). Each generated cluster is assigned a *committee*, which is a set of terms that represents the cluster. Based on the cooccurrence feature vector of members of the committee, CBC then generates a set of cluster *signatures*. Next, the system searches for a set of patterns indicative of hyponymy relations within each cluster signature, and labels each cluster with the term that has the highest mutual information score between the set of patterns indicative of hyponymy and pattern committee. The hyponym relation can then be

derived between each cluster member (e.g. *tiger*, *bird*, *dog*) and the cluster label (e.g. *animal*).

Sekine (2006) proposed *On-demand information extraction* which requires only a user query for any topic to automatically create extraction patterns and outputs a list of extracted relation mentions related to the initial query. The framework executes the user query using an IR system, and returns a collection of text related to the query. The heart of his system is paraphrase discovery, which is a clustering process to merge patterns from the returned text collection expressing the same relations in a pattern cluster, based on the cooccurrence of named entity instances and common keywords in their pattern representations. These pattern clusters are in turn used as the basis for extracting relation mentions from text.

2.3.5 Relation-Independent RE Approaches

So far, the approaches we have described focused on the task of extracting as many instances as possible for a relation type that is specified in advance. Due to increasing demand for RE systems to automatically extract from large-scale corpora, there have been new lines of research that have concentrated on extracting not only as many instances as possible, but also as many relation types as possible. These approaches usually do not require the user to prespecify which relations to extract.

Hasegawa *et al.* (2004) investigated methods to discover relations from large corpora between a pair of named entities. Instead of clusters of nouns produced by Carballo (1999) and Pantel and Ravichandran (2004), Hasegawa *et al.* (2004)'s clusters consist of named entity pairs which correspond to a relation type. This work, however, is unable to automatically identify the relation type of the clusters, and relies on manual identification (in the previous example, the relation is *merger_and_acquisition* between two organisations). Therefore, this task is often referred to as the *Relation Identification task* (Rosenfeld and Feldman 2007). Since then, approaches have been proposed to improve on Hasegawa *et al.* (2004) using different clustering methods (Rosenfeld and Feldman 2007; Chen *et al.* 2005a; Zhang *et al.* 2005), feature selection techniques (Chen *et al.* 2005b), pattern representations (Zhang *et al.* 2005;

Yan *et al.* 2009), and developing methods to automatically label the clusters (Chen *et al.* 2005a; Chen *et al.* 2005b).

Shinyama and Sekine (2006) proposed *Preemptive information extraction* to automatically discover all feasible relations from text without human intervention. Given a collection of documents, the system initially clusters documents that express a specific event (e.g. hurricane *Katrina* hitting *New Orleans*) based on bag-of-word features. Named entity taggers, and coreference resolution are then applied to each member of the document clusters to identify interesting entities. Following this, the system iteratively compares the similarity of every document cluster pair to merge clusters that express a similar event (e.g. merging cluster containing documents about hurricane *Katrina* hitting *New Orleans* with a cluster containing documents detailing the progress of hurricane *Longwang* heading to *Taiwan*), based on the pattern similarity of the entities in each cluster pair. The system then outputs sets of entity pairs that express the same event (e.g. (*Katrina*, *New Orleans*) and (*Longwang*, *Taiwan*)).

Shinyama and Sekine (2006)’s approach has been criticised for not being scalable to large corpora, especially the Web (Banko *et al.* 2007). This is due to its heavy reliance on NLP tools, such as named entity taggers and coreference resolution which may not work well when applied to (potentially noisy) Web data. Their clustering process is also computationally expensive as it requires pairwise comparison between documents in the corpora.

Open Information Extraction

Banko *et al.* (2007) proposed a different RE task, termed *open information extraction*, which aims to extract as many relation tuples as possible, while still being highly-scalable. Their system, **TextRunner**, performed the RE task within a single pass over a large corpus, without requiring any human supervision (Banko *et al.* 2007; Banko and Etzioni 2008).

There are other systems based on the Open IE paradigm, such as: **StatSnowBall** (Zhu *et al.* 2009), and **WOE** (Wu and Weld 2010). The basic approach of these systems can be summarised as a three-step method (Etzioni *et al.* 2011):

1. **Label:** Label sentences sampled from a small subset of corpus as training data for a relation phrase extractor learner. The labelling process is done automatically using a set of heuristics.
2. **Learn:** Train a relation phrase extractor using the labelled sentences and entities found in the sentence.
3. **Extract:** Given a sentence as input, the extractor identifies candidate pairs from the sentence, and uses the learned extractor to determine whether the pairs are part of a relation or not. If the pairs are part of a relation, the extractor then labels the extracted relation.

Unlike distant supervision systems that generate labelled training examples specific to a relation, Open IE labelled examples are not tied to any relation. This is because the objective of Open IE in labelling training examples is to identify whether a sentence contains an interesting relation regardless of the relation type, while the corresponding step in distant supervision is to generate training examples of a certain relation type for a supervised learner.

We discuss the details of each step as implemented in the **TextRunner** system in the following. **TextRunner** uses the following set of heuristics in identifying whether sentences are likely to contain an interesting relation or not:

- There exists a link between *entity-1* and *entity-2* that is no longer than a prespecified length.
- The link from *entity-1* to *entity-2* does not cross a sentence boundary.
- Neither *entity-1* nor *entity-2* consists solely of a pronoun.

These heuristics are applied to a small corpus. Sentences and entity mentions found in the sentences that fulfill all these heuristics are labelled as positive examples. Conversely, if a sentence does not fulfill all of these heuristics, both the sentence and its entity mentions are labelled as negative examples.

Non-domain specific features of these training examples are then gathered from a corpus, and a supervised classifier is trained using Naïve Bayes (Banko *et al.* 2007) or

a CRF (Banko and Etzioni 2008). The features used are mainly lexical and syntactic features, and they specifically avoid semantic features produced by domain-specific parsers.

Next, the trained classifier is used to extract relation tuples from a large-scale web corpus. As Open IE does not rely on a set of predefined relations, the extraction sets may contain several textual versions of a similar relation type label. For example, the system may extract `is_headquartered_in(Microsoft, Redmond)` and `is_based_in(Microsoft, Redmond)`, which are essentially the same extraction. **Resolver** (Yates and Etzioni 2007), a system for resolving synonymous relations and objects, is applied to the output of **TextRunner** to resolve these variations on a similar relation tuple (e.g. to `headquarters(Microsoft, Redmond)`).

Although successful in extracting a large number of useful relations, Open IE extracts large volumes of uninformative and incoherent extractions (Fader *et al.* 2011). Uninformative extractions are extractions that omit critical information, e.g. extracting `got(University, funding)`, rather than `got_funding_for(University, cancer research)` from *The University got funding for cancer research today*. Incoherent extractions occur when the relation type labelled by the classifier does not have any meaningful interpretation, e.g. extracting `was_central_torpedo(Mark 14, scandal of the fleet)` from *The Mark 14 was central to the torpedo scandal of the fleet*. Fader *et al.* (2011) identified that these errors occurred during the relational phrase identifier stage due to the classifiers limitations, and proposed syntactic and lexical constraints to reduce these errors.

Another weakness of Open IE is that the relation labels are found by examining the text of sentences deemed to possess a relation and generating the labels using phrases found in those sentences. For example, in the sentence *Melbourne is located in Victoria*, an Open IE system may identify that the phrase *located in* is the relation label. Open IE, therefore, is unsuited to identifying relations in which the relation label is not textually expressed, or indicated by punctuation or document-level features (Banko and Etzioni 2008), such as antonymy and synonymy.

2.4 Pattern Representations

In previous sections, we have described various approaches to RE. One unifying factor among these approaches is their reliance on the context of where the entity pairs appear in detecting the existence of a relation between the pairs. The most common way to abstract the context is to use patterns that connect the entities.

One key issue in RE systems is how to represent these patterns effectively to maximise their utility in improving the system performance. The simplest way to represent the patterns is to use the raw text representation of the patterns (surface strings) between the entities pair, as appeared in the corpus, like Hearst (1992)’s approach.

The main disadvantage of this simplistic representation is that it tends to be inflexible. To generalise this representation, terms in the raw representation could be lexicalised, or replaced by a glob operator (e.g. wild card operator) (Brin 1998; Rosenfeld and Feldman 2006; Turney 2008). Navigli and Velardi (2010) proposed a different generalisation of surface patterns by constructing word-class lattices to extract hypernyms from definitional sentences. Some representations also include additional words that appear on either side of the entity pairs (i.e. words that are not included in the pattern) (Downey *et al.* 2004).

It has also been hypothesised that this simplistic representation does not benefit from the analysis of sentence structures in which the entities appear. This analysis might contain syntactic and semantic clues valuable to the RE task. Parsing should therefore help RE as it is able to provide this information, and its corresponding output can be utilised as the pattern representation (Sudo *et al.* 2003; Etzioni *et al.* 2005; Snow *et al.* 2005; Sekine 2006; Girju *et al.* 2006).

There are different types of parsing output available, each representing different semantic and syntactic information. Some RE systems prefer using shallow parsing, which may provide information such as part of speech, named entity tags, word chunks, or semantic classes (Agichtein and Gravano 2000; Etzioni *et al.* 2005; Sekine 2006; Girju *et al.* 2006; Banko *et al.* 2007; Kozareva *et al.* 2008). Other systems adopt full parsing to capture further semantic and syntactic information.

Some examples of full parsing are dependency parsing, phrase-structure parsing, and predicate-argument models (Miyao *et al.* 2009). Further, there are different ways of interpreting the outputs of a given parser. For example, Stevenson and Greenwood (2009) compared and contrasted six different representations derived from the outputs of a dependency parser.

Sudo *et al.* (2003) were the first to investigate the effect of different pattern representations on RE performance. They compared three different representations: dependency tree, predicate-argument, and chain-model representations, derived from the outputs of a dependency parser, and concluded that an identical pattern represented differently can affect system performance. Jijkoun *et al.* (2004) compared the performance of a Question Answering system using surface string and dependency parsing representations. They concluded that dependency parsing offers better system performance. Miyao *et al.* (2009) reported similar observations that full parsing-based methods offered an improvement over a protein-protein interaction RE task, although they claim that the findings cannot necessarily be generalised to other datasets or tasks.

In another comparison, Rosenfeld and Feldman (2006) demonstrated the utility of shallow parsing to improve extraction performance over a simple surface string representation. Elsewhere, Wu and Weld (2010) showed that their RE system performed better using dependency parsing features than shallow parser features. Finally, Yan *et al.* (2009) combined the representations of both dependency patterns and surface patterns, and found that their system performed better on the combined representations than the individual representations. Hachey (2009a) also concluded that combining dependency and surface representations is the best approach, although his investigation is performed for Generic Relation Identification (which only aims to perceive whether an entity pair has a relation or not — without labelling the relation). In the same experiment, he also investigated the ideal length for various pattern representations. The results of these research suggest that different representations provide complementary clues to RE, and system performance can benefit from combinations of different representations.

Although it has been shown to outperform surface string and shallow parse-

based representations, a full parse-based representation depends on parser availability. While parsers are generally available for edited English text such as newswire or encyclopedia articles, they may not be available for other domains or non-English text. This restricts the domain adaptability of a system that depends on a full parse-based representation (Banko *et al.* 2007). The use of parsers also increases the computational cost, and leads to reduced term extraction speed. Parsing a large amount of text may also take a considerable amount of time and have to be off-line, which may prevent the system from performing extraction on the fly. Finally, the performance of parsing-based systems depends on the parser’s performance, and is therefore susceptible to parsing errors (Yakushiji *et al.* 2006).

In this thesis, we employ full parse-based features, specifically, a dependency-tree path representation. We believe that this representation may improve system performance based on several previous findings (Jijkoun *et al.* 2004; Miyao *et al.* 2009; Wu and Weld 2010). Further, both the corpora used in our experiment use are made up of edited English text for which parsers are readily available (see Section 3.2 for details). We use MINIPAR (Lin 1998) as our parser, and discuss the details of employing it and forming our pattern representations in Section 3.3.1.

2.5 RE Evaluation

We have described various RE approaches in the previous section. Unfortunately, the large methodological and application variety in RE research has led to a number of different methods on how to evaluate RE. In this section, we survey a range of popular datasets or knowledge bases and various evaluation procedures and metrics used for RE evaluation.

2.5.1 RE Datasets

One of the earliest datasets used in RE evaluation is the MUC-7 template relation (TR) dataset. The MUC-7 dataset contains training and test data, each based on 100 newswire articles. The objective of MUC-7 TR is to extract relational information

on `employee_of`, `manufacture_of`, and `location_of` relations (Chinchor 1998). These relations are defined to be between a pair of named entities, where:

1. `employee_of` is a relation between a *Person* and an *Organisation*.
2. `product_of` is a relation between an *Artifact* and *Organisation*.
3. `location_of` is a relation between a *Location* and *Organisation*.

Prior to identifying the relation between the named entities, participants are required to identify the named entities for extraction from text. As such, the extraction success is dependent of the accuracy of the named entity tagger used.

The ACE dataset (from various years) adopted the same experiment setting as MUC-7, in that the task objective is to detect and characterise relations between identified entities (Doddington *et al.* 2004). ACE also uses newswire articles, although with more data than MUC-7. Initially, there were 5 relation types in ACE, which are further subdivided into 24 fine-grained subtypes:

1. **ROLE**, which captures the role of a *Person* within an *Organisation*, a *Facility*, or a *Geo-Political Entity*. It is subdivided into `management`, `general_staff`, `member`, `owner`, `founder`, `client`, `affiliate_partner`, `citizen_of`, and `other`.
2. **PART**, which captures the part-whole relationships between an *Organisation* or a *Facility* with a *Geo-Political Entity*. It is subdivided into `subsidiary`, `part-of`, and `other`.
3. **AT**, which captures the *Location* of a *Person*, an *Organisation*, a *Geo-Political Entity*, or a *Facility*. It is subdivided into `located`, `based-in`, and `residence`.
4. **NEAR**, which indicates that any named entity type is near to either a *Location* or a *Geo-Political Entity*. It is not subdivided.
5. **SOCIAL**, which indicates the social relations between two *Persons*. It is subdivided into *Parent*, *Sibling*, *Spouse*, *Grandparent*, *Other-Relative*, *Other-Personal*, *Associate*, or *Other-Professional*.

ACE provides a very detailed definition of each relation and their subtypes. It also strictly defines all of ACE’s named entity types. Similar to MUC-7, ACE also requires that entities be identified prior to characterising the relation between the entities.

SemEval-2007 Task 4 constructed a dataset for semantic relations between nominals. It contains seven different relations: *cause-effect*, *instrument-agency*, *product-producer*, *origin-entity*, *theme-tool*, *part-whole*, and *content-container* (Girju *et al.* 2007). The aim of this shared task is to classify semantic relations that occur in the context of a written English text (token-level). This setup is very different to document-centric shared tasks such as MUC-7 or ACE.

For each relation, the annotators provide a detailed description and sample sentences containing both positive and near-miss negative examples. However, unlike MUC-7 and ACE, the relations are not constrained to being between two named entity types. In fact, named entity types are not provided as part of the dataset.

The dataset itself contains separate training and test sets for each relation. The training data contains 140 sentences, while the test data contains at least 70 sentences. The distribution of positive and negative instances is balanced in both the training and test data. Within each sentence, two nouns are identified, which are annotated for the existence of a given relation between them. Below are training and test examples for *cause-effect*:

Training:

Zinc is essential for *growth* and cell division.
cause-effect(*Zinc*, *growth*) = “true”

Everybody in the region experienced the *tremors* after *midnight*.
cause-effect(*tremors*, *midnight*) = “false”

Test:

Givers gain moral strength and *happiness* from *giving*.
cause-effect(*happiness*, *giving*) = “?”

This task setup assumes RE to be a simple binary (positive/negative) classification task relative to a given relation. In other words, the system is not required to predict which relation(s) a pair of nouns may hold out of the seven possible relations (i.e.

it not a multi-class classification task). Also, unlike MUC-7 and ACE dataset, the system is not required to identify the candidate extraction nouns, as the nouns are provided as part of the test data.

SemEval-2010 Task 8 aimed to build on the effort of SemEval-2007 Task 4, but removed the assumption of a binary classification task (Hendrickx *et al.* 2010). The organisers argued that such an assumption is not practical in real-life conditions, where any given relation may occur in any sentence. They addressed this problem by proposing a multi-class classification between a pair of nominals. In doing so, the RE systems are required to classify which of the 10 available relations fit the identified noun pairs best.⁶

Non-Shared Task Resources

Shared task datasets have been a valuable reference point for RE evaluation. However, they are not suitable for researchers who assume a different RE setup and focus on different relations other than those annotated in the existing shared task datasets. As a result, researchers tend to use other knowledge bases that contain the relations they are interested in, or failing that, may resort to either developing their own datasets and/or depend on human judgements for evaluation to suit their task setups (Davidov and Rappoport 2008).

When using knowledge bases developed independently from a corpus, researchers tend to assume type-level RE setup. The RE system learns a model based on a (training) partition of the knowledge base with the assumption that every occurrence of a knowledge base entry in the corpus matches the knowledge base’s annotation. Having learnt the model, the system is then evaluated against the remaining (test) partition of the knowledge base.

WORDNET is a popular choice for type-level RE evaluation of lexical semantic relations, such as hypernymy, synonymy, and meronymy. One reason for its popularity is that RE research has a long history with the knowledge base population and maintenance task (see Section 2.7), and WORDNET has been used as the resource

⁶There are 9 primary relations, and one other relation in cases where the relation does not fit any of the other relations.

to be improved upon since the early RE research of Hearst (1992). Hypernymy has also been regarded as the standard (and arguably, most popular) relation in RE based on the amount of prominent literature that focuses on it (e.g. works by Hearst (1992), Ravichandran and Hovy (2002), Cederberg and Widdows (2003), Pantel *et al.* (2004), and Snow *et al.* (2005)). We use WORDNET as the knowledge base in this thesis, and discuss it further in Section 3.1.

There are several problems with using WORDNET as the relation knowledge base. First, despite reasonable coverage, WORDNET still suffers from low-coverage of proper nouns (Ritter *et al.* 2008; Hovy *et al.* 2009). Second, it is not trivial to use due to its relation annotations being at the synset-level for some relations (e.g. hypernymy and synonymy), and at the word sense-level for other relations (e.g. antonymy). This is evident in the contradicting definitions of hypernymy among the research community. For example, Ritter *et al.* (2008) consider that a noun pair has a hypernym relation if one of its possible sense combinations has a hypernym link. On the other hand, Snow *et al.* (2005) consider a noun pair to be hypernyms iff their WORDNET hypernym relations are between their first WORDNET senses (predominant senses). We discuss this implication further in Section 3.1.

For research focusing on other relations, researchers often use data from other sources or knowledge bases. Popular examples include FREEBASE (Bollacker *et al.* 2008), YAGO (Suchanek *et al.* 2007), or Wikipedia articles. In other tasks, researchers used analogy or synonym questions from standardised tests such as SAT or TOEFL to measure their RE accuracy (Turney 2008; Mohammad *et al.* 2008; Davidov and Rappoport 2008). An example of a SAT analogy question is (as adopted from Turney (2008)):

mason:stone ::

- a) *teacher:chalk*
- b) *carpenter:wood*
- c) *soldier:gun*
- d) *photograph:camera*
- e) *book:word*

There is also research that focuses on extracting instances of specific relations, in which there does not exist a suitable knowledge base. In this scenario, researchers often choose to construct their own datasets or manually evaluate their system’s accuracy.

2.5.2 Open Task vs. Closed Task

In a *closed task*, a statement that is not known to be true is assumed to be false, and a statement is true only if the knowledge base provides evidence for the statement. This contrasts with an *open task*, where the lack of evidence for a certain statement in a knowledge base does not imply that the statement is true or false; it is simply unknown. The possibility of not being able to confirm a statement (rather than simply rejecting it) is a reflection of the open task presumption that the knowledge base itself can be incomplete.

We present the following example highlighting the difference between a closed task and an open task. Suppose we have a knowledge base containing mentions of `composer_of(Mozart, Don Giovanni)` and `composer_of(Beethoven, Fidelio)`, and need to confirm the candidate `composer_of(Mozart, The Magic Flute)`. In a closed task, the knowledge base evaluates `composer_of(Mozart, The Magic Flute)` to be false, as it does not contain the relevant information. However, in an open task, the knowledge base considers the mention to be unknown.

Most research in RE tends to assume that the knowledge base used in the experiment is complete, therefore assuming the task to be closed. However, as one of the objectives in RE is to find additional relation tuples not found in the current knowledge base, it is inevitable that some systems will predict correct relation mentions that are not yet within the knowledge base. To deal with this scenario, some researchers relax the definition of the closed task by using domain-expert knowledge (human judgements) to evaluate whether those predicted mentions are correct or not (Snow *et al.* 2005; McIntosh and Curran 2008). This thesis assumes this relaxed definition of the closed task.

2.5.3 RE Evaluation Metrics

Since the introduction of MUC, IR metrics such as recall, precision, and f-score have been used and commonly accepted as the standard metrics for IE and RE evaluation. In this thesis, we use these metrics to evaluate our RE system (the definitions of these metrics are discussed in Section 3.5).

The ACE shared task chose to use their own scoring schema to evaluate RE accuracy. One of the main reasons is due to the task setup that requires a system to identify both the relation types and subtypes (Doddington *et al.* 2004). Further, the task requires a system to first identify the entities of interest in the documents using a named entity recogniser. Because of this the named entity recogniser’s accuracy has a direct effect on the final RE accuracy.

Most RE systems report their accuracy relative to a given relation. Shared tasks where multiple relations are involved (e.g. ACE and SemEval-2010 Task 8) may report a final system score by combining or averaging the system accuracy across individual relations. For example, ACE final score is derived by summing up the individual system accuracy on different relations, while the SemEval tasks’ overall score is measured by micro-averaging or macro-averaging the system’s precision, recall, or f-score scores on individual relations.

Researchers often manually judge the extraction results (usually with more than one evaluator) when evaluating relations of interest that do not yet have any existing knowledge base. In this scenario, an inter-annotator agreement is often reported. A downside of using human evaluators is that it is expensive and not practical for systems extracting large number of relation mentions. To address this, researchers often evaluate over a random subsampling of the extraction results or over the top N -ranked mentions (if the mentions are ranked).

Traditionally, given a relation, the extracted mentions are judged to be either correct or incorrect relative to that particular relation. However, researchers have considered varying the notions of correctness due to various factors such as capitalisation, the existence of extra linguistic items (e.g. an article or plural marking), or tuples that are too general or too specific (cf. *is-a(Pilot, Human)* vs. *is-a(Pilot,*

Experiment)) (Cederberg and Widdows 2003; Pantel and Pennacchiotti 2006). The partially correct instances are usually weighted according to a predetermined weight when calculating the final system score.

2.5.4 Interpreting Type-Level Evaluation Metrics

RE systems usually employ a corpus and a gold-standard resource as resources for learning and evaluation. In the token-level (e.g. SemEval shared tasks) or document-level (e.g. MUC or ACE) tasks, these resources are often developed and annotated together. Assuming the annotation to be correct, the information in the gold-standard is therefore correct and complete relative to the corpus for a given relation.

Type-level tasks often lack the resources where the corpus and the gold-standard are mutually developed and annotated. This stems from the tendency to perform type-level tasks with a large amount of text that cannot practically be manually annotated. As a result, researchers often use preexisting knowledge bases developed independently of the corpus. However, unlike the annotation at the token-level or document-level tasks, there is no guarantee of finding mentions of the gold-standard's entity pairs in the corpus. We examine this assumption to the calculation of recall and precision, and how it may effect their interpretations.

Because there is no guarantee a given entity pair will be attested in the corpus, there may be no evidence for a type-level RE system to base their predictions on. As a result of this possibility, it may not be able to correctly predict some positive gold-standard entity pairs, which lowers the recall. On the other hand, the system may also predict valid pairs that are currently not existing in the gold-standard (perhaps due to its limited coverage), which negatively affects the system's precision.

Symmetrical relations such as synonymy and antonymy pose a subtle issue for RE evaluation. The issue is whether the evaluation should ignore the relation order or not, given that the validity of the symmetrical relations is not affected by the entity order (e.g. `synonym(salary, wage)` is equivalent to `synonym(wage, salary)`). If we are to assume that the RE system has knowledge of the symmetrical nature of the relation, the evaluation can ignore the relation order: `synonym(salary, wage)` and

`synonym(wage, salary)` can be treated as one identical instance. On the other hand, if the RE system does not have knowledge of the relations' symmetrical nature, then the evaluation should take the relation order into account and ensure the two tuples (of the differently-ordered nouns) are treated as two different type instances.

It is very challenging to evaluate a system's true recall in a task that focuses on a relation for which there does not exist a knowledge base. This is because it is very difficult to enumerate all the correct relation mentions. While the manual annotation of the corpus is possible, it is very expensive and prone to annotator bias and error. An alternative to this is to measure the system's *relative recall* to other systems on the same corpus and relation (Pantel *et al.* 2004).

Open IE presents another difficulty for evaluation as its main objective is to extract as many interesting relations as possible from text. As there is no closed set of relations that can be defined universally (Downing 1977; Ó Séaghdha 2008), it is not possible to accurately evaluate Open IE's recall in extracting all possible relation types from a corpus. Current evaluation of Open IE focuses on the number of distinct relation types successfully extracted, and accuracy over a set of frequently appearing relations and a small-sized corpus (Banko *et al.* 2007; Banko and Etzioni 2008).

Sampling Bias for Evaluation

It is arguable that there is more utility in extracting correct low-frequency pairs that do not exist in the gold-standard than the high-frequency pairs that are already in the gold-standard (Hovy *et al.* 2009). However, this utility is often not reflected in the current RE evaluation procedures.

Some RE evaluation procedures choose to evaluate only on a subset of the extracted pairs, due to the large volume of the potential candidates. A popular RE evaluation procedure is to pick the top- X ranked pairs, and generalise the system's accuracy based on its accuracy over this set (Agichtein and Gravano 2000; Pantel *et al.* 2004; Paşca *et al.* 2006). Other procedures ignore low-frequency instances to avoid sparsity problem and only concentrate on evaluating over high-frequency noun pairs (Snow *et al.* 2005).

These procedures, however, are biased as they only evaluate their systems on entity pairs that appear frequently in the data (and are most likely to have high-quality patterns/features and also already in the gold-standard), while ignoring their system’s accuracy on the low-frequency pairs. As a result, the systems’ reported accuracy may not necessarily reflect their success in extracting low-frequency pairs, which arguably are more difficult to extract than the high-frequency pairs.

Hovy *et al.* (2009) argued that there is little utility to improve RE system accuracy based on evaluation procedures that concentrate only in evaluating pairs that already exist in the gold-standard, and in fact, the improvement based on such procedures may give the community a false sense of progress. Consequently, he suggested that RE systems’ accuracy should always be evaluated based on their success in predicting pairs that do not currently exist in the gold-standard. We examine the sampling issue for our RE evaluation more closely in Chapter 5.

2.6 RE Research Comparability

Following MUC and the increasing availability of large-scale document collections and annotation resources, there has been little attempt — with the exceptions of SemEval-2007 Task 4 and SemEval-2010 Task 8 — to formalise and define what relations are suitable for RE, and the style of RE that should be practiced. Since then, most research in RE has been driven by individual research objectives rather than a collective effort on a specific task or relation. As a result, a large variety of relations, learning styles and approaches, and evaluation procedures have been explored in the RE research area.

While the large variation in RE research has led to the development and advancement of RE techniques, it has the major drawback of causing system incomparability. This is because each research tends to perform RE over different experimental settings (Girju *et al.* 2007; Pyysalo *et al.* 2008). Pyysalo *et al.* (2008) even argued that the inability of directly comparing RE results inflicts a great cost on the research community, as one often has to reimplement systems and compare them under common experimental settings to reliably determine the relative merits of each system.

2.7 RE Applications

In a range of NLP systems, resources such as lexical databases, semantic taxonomies, or ontologies have been exploited as a knowledge base to boost the systems performance (Girju *et al.* 2006). WORDNET (Fellbaum 1998) and CYC (Lenat 1995) are two prominent examples of such resources that have been widely adapted in various NLP systems. While such resources have been shown to be a crucial knowledge source, they are usually constructed manually by domain-experts at great cost (Grefenstette 1994; Snow *et al.* 2006). Further, such resources usually suffer from low coverage, and only contain general-purpose knowledge that is not helpful for domain-specific applications.

Moldovan and Girju (2001) argued that the inability of building flexible and up-to-date knowledge bases with minimal human effort has obstructed many NLP developments, and urged the development of techniques for automatic knowledge base population (for a new domain or under-resourced languages, for example) and integration (e.g. integrating new information to an existing knowledge base) (Buitelaar and Cimiano 2008). This is a motivation for RE research, as the majority of the desired knowledge base information can be expressed as relations between entities.

Several researchers have successfully applied RE systems in building a knowledge base for a given relation by leveraging information from a corpus (Craven *et al.* 1998; Brin 1998; Paşca *et al.* 2006; Jing *et al.* 2007; McCrae and Collier 2008). There is also a considerable amount of RE research on extending or maintaining existing knowledge bases, with most focused on WORDNET (Hearst 1992; Caraballo 1999; Pantel and Pennacchiotti 2006; Girju *et al.* 2006; Snow *et al.* 2006; Hovy *et al.* 2009). These lines of research are also popular in the biomedical domain as the field struggles to keep biomedical ontologies up-to-date (McCrae and Collier 2008; Özgür *et al.* 2008; Pyysalo *et al.* 2008).

RE techniques, directly or indirectly, have been successfully integrated to question answering (QA) systems (Paşca and Harabagiu 2001; Moldovan *et al.* 2002; Ravichandran and Hovy 2002; Jijkoun *et al.* 2004; Girju *et al.* 2006; McNamee *et al.* 2008). The QA task involves finding an answer to a question posed in natural lan-

guage, e.g. *When was Beethoven born?* As many factual sentences containing this information are in the form of *Beethoven was born in 1770* or *Beethoven (1770–1827)*, it is possible to answer the **born-in** question using pattern learning techniques commonly employed in RE systems (Ravichandran and Hovy 2002; Jijkoun *et al.* 2004).

In related QA research, Paşca and Harabagiu (2001) describe various ways of exploiting WORDNET information in QA systems, such as determining the semantic class of the candidate answer, or reformulating the question. Similarly, Moldovan *et al.* (2002) used WORDNET to implement modules such as lexical-chaining to build their QA system. However, while demonstrating the positive contribution to QA systems, the impact of WORDNET is obstructed by its limited coverage (Paşca and Harabagiu 2001; McNamee *et al.* 2008). To address this, several researchers have experimented with integrating QA systems with a dynamically constructed knowledge bases that relies on RE techniques (Chklovski and Pantel 2004; Girju *et al.* 2006; McNamee *et al.* 2008).

McNamee *et al.* (2008) applied hypernymy learning techniques to automatically learn and build a knowledge base containing hypernymic relations between named entities from text. The focus on named entities is due to the importance of named entity recognition and the identification of their relations with other nouns to the QA task (Paşca and Harabagiu 2001). McNamee *et al.* (2008) then integrated the mined knowledge base with a QA system and demonstrated that the information from the knowledge base helped the QA system. They also remarked on the better utility of the automatically built named entity knowledge base when compared to WORDNET for the QA task, due to its better coverage of named entities.

The success of integrating RE techniques to QA systems prompted Etzioni (2011) to question the current search paradigm in IR. He argued that the current paradigm did not directly satisfy users' information needs, as most IR systems focus on returning all relevant documents (often millions of documents) to even a simple factoid query such as *What is the opus number of Beethoven's Appassionata Sonata?*

He noted an alternative paradigm where IR systems present the user with a direct answer to their queries, extracted from the web via an IE or a RE system and weighted relative to system confidence. While some major systems have implemented this

paradigm in limited ways (for example, in response to the query *Melbourne weather forecast*, popular search and answer engines such as Google⁷, Bing⁸, and Wolfram Alpha⁹ provide direct answer by showing temperature forecasts as a top search result in addition to the results returned from the search engine), their ability to directly answer users' queries are still very limited (none of these engines can provide a direct answer to *What is the opus number of Beethoven's Appassionata Sonata?*)

The applications of RE to other systems can perhaps be illustrated via Machine Reading (MR), which aims to extract knowledge automatically from unstructured text and use such knowledge in end-to-end NLP tasks such as question answering, textual inference, or text summarisation (Poon *et al.* 2010). As one of the main tasks is to extract knowledge from unstructured text, IE components, especially RE, play an important role in the advancement of MR research. Within the MR framework proposed by Poon *et al.* (2010), RE techniques have been used in various tasks such as identification of contrary statements from the web (Ritter *et al.* 2008) and improving the quality of Wikipedia infoboxes (Wu and Weld 2007). Further, outputs from an RE system have been used to learn first-order Horn-clauses that in turn are used to infer implicit relations from the web (Schoenmackers *et al.* 2010). Outside of this framework, RE techniques have also been used in text summarisation tasks as a medium to represent the relation between events in the source documents to generate better summary (Hovy and Lin 1998; Chklovski and Pantel 2004; Hachey 2009b).

2.8 Summary

In this chapter, we presented a survey of relation extraction (RE) research. We began with discussing the historical tie between information extraction (IE) and RE. The influence of the Message Understanding Conferences on IE and RE was also discussed. The decision of the MUC-6 organisers to break up IE into several multiple subtasks paved the way for the introduction and formalisation of the RE task in

⁷<http://www.google.com>

⁸<http://www.bing.com>

⁹<http://www.wolframalpha.com>

MUC-7.

At the conclusion of MUC-7, the RE research direction shifted from small-scale document collections to large-scale document collections due to the increasing availability of large-scale document collections. Owing to the diversity of information from large-scale document collections, researchers explored other relations not defined in MUC-7. We compared and contrasted the properties and characteristics of popular relations, looking specifically at binary vs. complex relations, symmetric vs. asymmetric relations, hierarchical vs. flat relations, domain-specific vs. domain-independent relations, and whether a relation is functional or temporal. The variety of relations explored in RE influenced the development of different RE styles. Some of these styles are explicit vs. implicit evidence-based RE, type vs. token-level RE, single-document vs. multi-document RE, and relation-specific vs. relation-independent RE.

Next, we discussed the importance of patterns for RE tasks and presented a survey of various approaches to identify patterns for a given relation from text. This includes methods such as Hearst (1992) which rely on manual pattern identification, automated relation-specific approaches that perform pattern identification via automatic bootstrapping, supervised, and semi-supervised approaches. Following this, we discussed the development of relation-independent approaches where the relations are not required to be prespecified prior to extraction. Also, we discussed the importance of how to represent patterns for RE, and how different representations of the same pattern may affect RE performance.

We next looked at various ways of evaluating RE performance. We described various standard datasets and evaluation metrics, and several key issues that may affect the interpretation of RE system results. We also discussed the lack of standards in RE research and experimental settings, and how these different factors may lead to system performance incomparability. And finally, we reviewed how RE systems have been successfully applied to improve other NLP tasks.

In the next chapter, we review all the resources used in this thesis, and discuss in detail how we employ them in our RE task.

Chapter 3

Resources

In this chapter, we discuss various resources we use and the how they are adapted in our RE task. To begin with, we discuss our relation labelling resources: WORDNET and SEMCOR. As part of this, we describe our relations of interest, and how we label our noun pairs using these resources. Following this, we describe the corpora used (GIGAWORD and WIKIPEDIA), and how we preprocess them. The issue of pattern representation is presented next, along with the discussion on how we interpret MINIPAR output to yield our pattern representations.

Machine learning forms an important part in constructing the classifiers for our relation extraction task. We describe Support Vector Machines (SVM), the machine learning algorithm prominently used in this thesis. Next, we present two methods for estimating classifier accuracy, and describe the evaluation metrics used in evaluating our classifiers.

3.1 WordNet

WORDNET¹ is an electronic semantic lexical database of English terms, created and maintained at Princeton University (Fellbaum 1998). It contains English nouns,

¹<http://wordnet.princeton.edu/wordnet/download/>; We use `WordNet::QueryData` Perl module by Jason Rennie to query the database. It is available from <http://search.cpan.org/dist/WordNet-QueryData/QueryData.pm>

Concept 1

Category: Noun**Definition:** "A *living organism characterized by voluntary movement*"**Synsets:** {animal, animate being, beast, brute, creature, fauna}

Concept 2

Category: Noun**Definition:** "A *cruelly rapacious person*"**Synsets:** {beast, wolf, savage, brute, wildcat}

Concept 3

Category: Adjective**Definition:** "Resembling a beast; showing lack of human sensibility"**Synsets:** {beastly, bestial, brute, brutish, brutal}

Figure 3.1: Examples of WORDNET concept definitions and synsets

verbs, adjectives, and adverbs. Throughout this thesis, we use version 3.0 of WORDNET.

WORDNET is best thought of as a collection of concepts. Each concept is characterised by its *definition*. In WORDNET, concepts are strictly categorised into one of four syntactic categories (noun, verb, adjective, and adverb), and are represented by WORDNET *synsets*. A WORDNET synset can be formally defined as a set of synonymous words which expresses a particular concept. Figure 3.1 highlights the relations between concepts and synsets in WORDNET.

The examples in Figure 3.1 illustrate the ambiguity of the words *beast* and *brute*, as they can be used to express more than one concept. Such a word is often referred to as *polysemous*, implying that the word has more than one meaning. On the other hand, a word with a single meaning (e.g. *t-shirt*) is known as *monosemous*. The individual meanings of a word are referred to as its *senses*.

WORDNET addresses the word ambiguity problem by including a sense number with all of its words, forming a *word/sense* pair. This is done by the following process.

Concept 1

Category: Noun

Definition: "A *living organism characterized by voluntary movement*"

Synsets: {animal₁, animate being₁, beast₁, brute₂, creature₁, fauna₂}

Figure 3.2: Examples of word/sense pairs in WORDNET synsets

First, using the concepts in WORDNET as the knowledge basis, WORDNET determines the number of senses for a particular word by simply counting the number of individual synsets the word appears in (recall that each synset expresses different concepts or meanings). Next, a sense-ordering process determines which sense number is assigned to each of these senses (see Section 3.1). This information is incorporated into WORDNET synsets forming a word/sense pair. Figure 3.2 presents this information (the subscript following each word denotes its sense number).

We only consider WORDNET noun synsets in our experiments as we are only interested in extracting relations between noun pairs. The main reason for this is that out of all WORDNET synsets, around 70% of them (82,115 out of 117,659) are of the noun class. Furthermore, out of the 147,278 unique strings in WORDNET, — counting a string that can be in more than one category as one unique string, e.g. *brute* (which can either be a noun or an adjective) — 117,798 of them are noun strings (around 80%). In addition, the number of nouns (especially proper nouns) is growing at a much faster rate than verbs, adjectives, and adverbs (Jackson 2002). This motivates the need for a reliable relation extraction system to help lexicographers determine the relations between (both existing and newly discovered) nouns.

WORDNET Relations

WORDNET is the primary source for labelling the existence of a relation between a pair of nouns. For each noun synset, WORDNET provides links to other noun synsets via various conceptual-semantic and lexical relations, forming a network of concepts. We use the *antonymy* (between word senses in synsets), *synonymy* (between synsets), and *hypernymy/hyponymy* (between synsets) relations in our experiments. A snippet

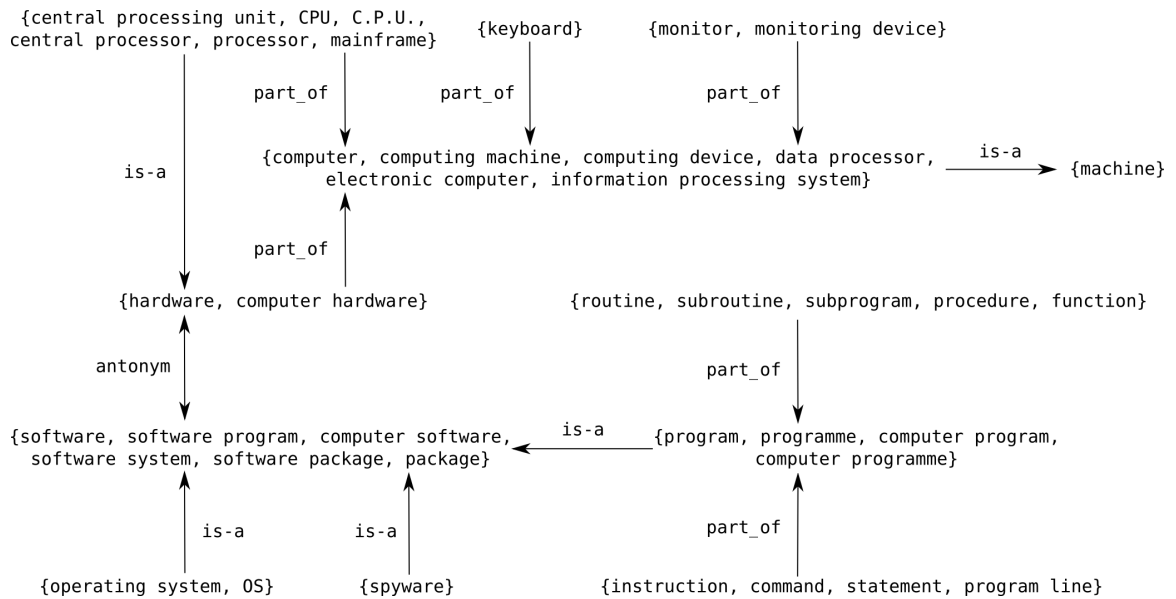


Figure 3.3: A snippet of the concept network in WORDNET

of the WORDNET network is shown in Figure 3.3.

A pair of synsets is considered to have a given relation iff WORDNET contains a semantic relation link of that type between the two synsets. We define the relations used in our experiments as:

- *Antonymy*: X is an antonym of Y iff X expresses the opposite concept of Y (e.g. $advance_4$ “the act of moving forward (as toward a goal)” and $retreat_7$ “the act of withdrawing or going backward”). Antonymy links are applicable only to specific word senses in synsets, instead of all members of the synsets.

Therefore, given that the antonymy link is only specified for $advance_4$ and $retreat_7$, there is no antonym relation between $procession_3$ and $retreat_7$, although both $advance_4$ and $procession_3$ belong to the same synset. Also, note that other senses of *advance* (e.g. $advance_1$ “a movement forward”, $advance_2$ “a change for the better; progress in development”, or $advance_5$ “an amount paid before it is earned”), do not have antonymy links with $retreat_7$, as they express different concepts — and hence belong to different synsets — to $advance_4$.

- *Synonymy*: X is a synonym of Y iff both X and Y express the same concept

(e.g. *chip*₇ and *microchip*₁ “electronic equipment consisting of a small crystal of a silicon semiconductor fabricated to carry out a number of electronic functions in an integrated circuit”). In WORDNET, synsets are groups of synonyms. Therefore, given a noun synset, synonymy links can be found in every possible pairing of the member of the synset. For example, synonymy links can be found in all possible pairings of *astronaut*₁, *spaceman*₁, and *cosmonaut*₁ “a person trained to travel in a spacecraft”.

- *Hypernymy* (alternatively, *is-a*): X is a hypernym of Y iff Y is a type of X (e.g. *musical_instrument*₁ “any of various devices or contrivances that can be used to produce musical tones or sounds” is a hypernym of *clarinet*₁ “a single-reed instrument with a straight tube”). If visualised as nodes in a tree, Y is the ancestor node, while X is the descendant node, with the edge between them representing the existence of the hypernymy link. In our experiments, we do not limit the hypernymy link to be between an immediate ancestor–descendant synset pair. We consider a pair of synsets to have a hypernym link regardless of the number of hypernym links between them. Therefore, if X is a kind of Y , and if Y is a kind of Z , then X must be a kind of Z .

Unlike antonymy, hypernymy links are applicable to all members of synsets.

It is worth noting differences in the nature of the three relations above. Both antonymy, and synonymy are symmetrical (i.e. *astronaut*₁ is a synonym of *cosmonaut*₁ $\Leftrightarrow$ *cosmonaut*₁ is a synonym of *astronaut*₁). Hypernymy, on the other hand, is asymmetrical (i.e. it is not true that *musical_instrument*₁ is a kind of *clarinet*₁).

The relation links in WORDNET are specified for a pair of synsets, that is, the relations are specific for a particular word sense (sense-level). In contrast, our experiments are concerned with extracting relations between two nouns at the word-level. This creates a tension in using WORDNET as the relation labelling resource, as there is no WORDNET sense information available for the nouns collected from our corpora. Therefore, to take advantage of WORDNET relation links, a mapping process between the word-level noun pairs collected in corpora to the sense-level WORDNET noun synsets is required. This mapping process is more commonly known as word

sense disambiguation (WSD) on WORDNET senses. However, WSD is in itself a challenge, and has been described to be “an AI-complete problem, whose solution is at least as hard as the most difficult problems in artificial intelligence” (Navigli 2009), which is not our main research focus. We discuss our heuristic of the mapping process further in Section 4.1.3.

Lexicographers often have different views of the number of possible meanings, i.e. *sense ambiguity* for a given word, and how to define, differentiate, and group those meanings (Wilks *et al.* 1996; Ravin and Leacock 2000). These views reflect the complex and difficult philosophical question of what a word meaning can represent. As a result, there does not exist a single authoritative source of word meanings, and any resources providing this information are inherently reflective of the resources creators’ view on word meanings. By using WORDNET in our experiments, we inevitably make a closed-world assumption of possible word meanings within the limits of WORDNET, and accept WORDNET as the authoritative source in defining, differentiating, and managing word meanings.

SEMCOR File

The main motivation in including sense number information in WORDNET synsets is to avoid confusion when referring to polysemous words. While highly dependent on the domain, it is arguable that some of a polysemous word’s senses appear more frequent than others (Magnini *et al.* 2002; Koeling *et al.* 2005). For example, *tissue* is more likely to refer to “part of an organism consisting of an aggregate of cells having a similar structure and function” than “a soft thin (usually translucent) paper” in biomedical articles.

WORDNET attempts to capture the sense distribution of a polysemous word in general text. Its sense-ordering policy is a reflection of this, as WORDNET orders and presents senses of a polysemous word based on how frequent a particular sense is used, with the most frequently used sense presented first (Tengi 1998). The basis of the sense frequency is the number of times that sense is found in the semantic concordance files (SEMCOR) used by WORDNET (Landes *et al.* 1998). SEMCOR can

be thought of as a representation of general text, and is designed to be balanced across literary styles and domains.

SEMCOR is built from two corpora: the BROWN CORPUS (Francis and Kučera 1982), and the complete text of Stephen Crane’s novella *The Red Badge of Courage*. The BROWN CORPUS contains 1 million American English words excerpted from 1960s American English documents (only around 20% of BROWN CORPUS is used in building SEMCOR), while the novella contains 45,600 words describing the American Civil War. Next, all words in the corpora were manually tagged to their sense entries in WORDNET version 1.5. As newer versions of WORDNET were released, there has been an updated version of these mappings.

Not all the words in SEMCOR are tagged (around half of all words of each corpus are tagged). This is because WORDNET only contains entries for nouns, verbs, adjectives, and adverbs. Similarly, not all WORDNET entries are found in SEMCOR, with only around 10% of the WORDNET version 3.0 unique nouns (11,846 out of 117,798) appearing in SEMCOR. These nouns have at least one sense SEMCOR-tagged. Furthermore, using SEMCOR as a comparison basis, these nouns can be thought of as more *commonly-used* than the others.

When queried for a particular word, WORDNET orders and presents its senses in descending order of the frequency of each sense in SEMCOR. The order for senses with the same SEMCOR frequency is determined by a random tie-breaking process. And finally, senses that do not appear in SEMCOR are presented last in random order. Therefore, the first sense of a word is most likely to be its most frequent sense. However, due to the random tie-breaking process, the ordering of some senses in WORDNET is questionable. For example, there are 2 senses of *tiger* with the same SEMCOR frequency count. Most readers would have not expected that the first sense of *tiger* in WORDNET refers to “a fierce or audacious person” rather than “large feline of forests in most of Asia having a tawny coat with black stripes; endangered”.

The order and role of WORDNET senses (especially the first senses) are important when labelling our gold-standard data in Section 4.1.3.

3.2 Corpora

We use two corpora in our experiments: the English Gigaword Corpus (newswire articles) and Wikipedia (encyclopedia articles). We use Gigaword corpus as relation extraction tasks or more generally information extraction tasks, have been historically performed on newswire articles (Grishman and Sundheim 1996). Wikipedia articles provide a contrast to the newswire articles as they contain more descriptive information on a certain topic, and hence it is hypothesised to be a good source for relation extraction tasks. Furthermore, Wikipedia articles are generally written by non-journalists, providing different writing styles compared to Gigaword articles.

3.2.1 English Gigaword Corpus

The 2003 English Gigaword Corpus (LDC2003T05),² GIGAWORD hereafter, is a collection of newswire articles from four distinct sources:

- Agence France Press English Service
- Associated Press Worldstream English Service
- The New York Times Newswire Service
- The Xinhua News Agency English Service

GIGAWORD contains over 4 million articles. After preprocessing and sentence tokenising (using Marcia Munoz’s Perl script³), there are around 84 million sentences in the corpus.

3.2.2 Wikipedia

Wikipedia is a popular online encyclopedia whose articles are contributed by its users. Wikimedia Foundation, the organisation operating Wikipedia, periodically generates database dumps of Wikipedia articles, which are downloadable from [http:](http://)

²<http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05>

³http://cogcomp.cs.illinois.edu/page/tools_view/2

[//download.wikimedia.org/](http://download.wikimedia.org/). We use the July 2008 XML dump version in our experiments (WIKIPEDIA hereafter). We preprocessed the XML dump and extracted only the articles' texts using `wikiprep version 1.0`⁴ (Gabrilovich and Markovitch 2006; Gabrilovich and Markovitch 2007). Finally, we applied Marcia Munoz's sentence tokeniser to tokenise WIKIPEDIA, resulting in around 38 million sentences (around half of the number of sentences in GIGAWORD).

3.3 Pattern Representation

As discussed in Chapter 2, the main features used in most RE systems are the patterns that appear between two entities of interest. There are different ways to express these patterns, using either the raw strings found in the corpora (Hearst 1992), or some preprocessed form of those strings. Some examples of possible string modifications are morphological normalisations (Turney 2008; Banko and Etzioni 2008), pattern generalisations (replacing a sequence of words with a wild card operator) (McCrae and Collier 2008), or to obtain the syntactic information of the strings by parsing them (Snow *et al.* 2005; Banko and Etzioni 2008; Hachey 2009a).

We experiment with the pattern representations using parse information derived from the MINIPAR dependency parser.

3.3.1 MINIPAR

MINIPAR is a fast and efficient broad-coverage dependency parser for English (Lin 1998).⁵ It produces a dependency tree for a sentence whose nodes contain all the words in the sentence. The nodes are connected by directed edges representing dependency syntactic relationships between them. We use the `pdemo` binary, provided as part of the MINIPAR distribution, to parse sentences in our corpora. Due to the limitation of `pdemo`, we ignore sentences with more than 1,024 characters. An example of a parse output of `pdemo` for the sentence *Prominent composers such as Ludwig van*

⁴<http://sourceforge.net/projects/wikiprep/>

⁵We use MINIPAR version 0.5, Linux 32 package in this thesis. It can be downloaded from <http://webdocs.cs.ualberta.ca/~lindek/minipar/>

Beethoven, Wolfgang Amadeus Mozart, Frederic Chopin, and Franz Schubert are also renowned for their skills on keyboard is shown in Table 3.1. The dependency tree version of this parse output is shown in Figure 3.4.

There are two types of nodes produced by MINIPAR: nodes representing surface words, i.e. words found in the parsed sentence (referred to as *surface nodes*), and nodes representing events (referred to as *artificial nodes* or *event nodes*). Surface nodes are indexed by integer numbers, while the artificial nodes are indexed by labels starting with an “E”. MINIPAR produces a lemma for each surface node, with the tilde symbols indicating the corresponding surface words are the lemmas. A lexical normalisation process is also involved in MINIPAR, as evidenced in Node 7 (*Beethoven* $\rightarrow$ *Ludwig van Beethoven*) in Table 3.1. Furthermore, as artificial nodes do not represent any surface words, we assign the string *d_n* (for *dummy_node*) as their lemmas.

We extract pattern representations from this output by the following steps. First, nouns are identified based on their POS-tags, lower case-folded, normalised to their lemmas, and combined into all permutations of noun pairs in the sentence (we refer to the first noun in resulting noun pairs as the *source noun* and the second noun as the *destination noun*). Next, we iteratively detect if a relationship link exists between all pairs of nouns. The link detection process for a noun pair involves finding the shortest path of four or less edges (treating the edges as undirected for the moment) from the source noun (node) to the destination noun (node). Noun pairs whose shortest path are longer than four edges are excluded. For each adjacent node, we collect the word’s lemma, POS-tag, and the relation between the words, and present them in the form of a dependency edge with the following structure:

$$\text{Lemma}_{w_i}, \text{POS}_{w_i} : \text{relation} : \text{POS}_{w_j}, \text{Lemma}_{w_j}$$

Note that the shortest path between two nouns can traverse non-noun nodes.

Nodes Index	Surface Word	Lemma	POS-tag	Parent Node	Relation
E1	()		U	*	
1	Prominent	~	A	2	mod
2	composers	composer	N	E1	
3	such	such as	Prep	2	mod
4	as	~	U	3	lex-mod
5	Ludwig	~	U	7	lex-mod
6	van	~	U	7	lex-mod
7	Beethoven	Ludwig van Beethoven	N	3	pcomp-n
8	,	~	U	7	punc
9	Wolfgang	~	U	11	lex-mod
10	Amadeus	~	U	11	lex-mod
11	Mozart	Wolfgang Amadeus Mozart	N	7	conj
12	,	~	U	11	punc
13	Frederic	~	U	14	lex-mod
14	Chopin	Frederic Chopin	N	11	conj
15	,	~	U	14	punc
16	and	~	U	14	punc
17	Franz	~	U	18	lex-mod
18	Schubert	Franz Schubert	N	14	conj
19	are	~	U	E1	punc
20	also	~	A	21	mod
21	renowned	~	A	E1	
22	for	~	Prep	21	mod
23	their	~	N	24	gen
24	skills	skill	N	22	pcomp-n
25	on	~	Prep	24	mod
26	keyboard	~	N	25	pcomp-n
27	.	~	U	*	punc

Table 3.1: A sample of parse output by `pdemo`

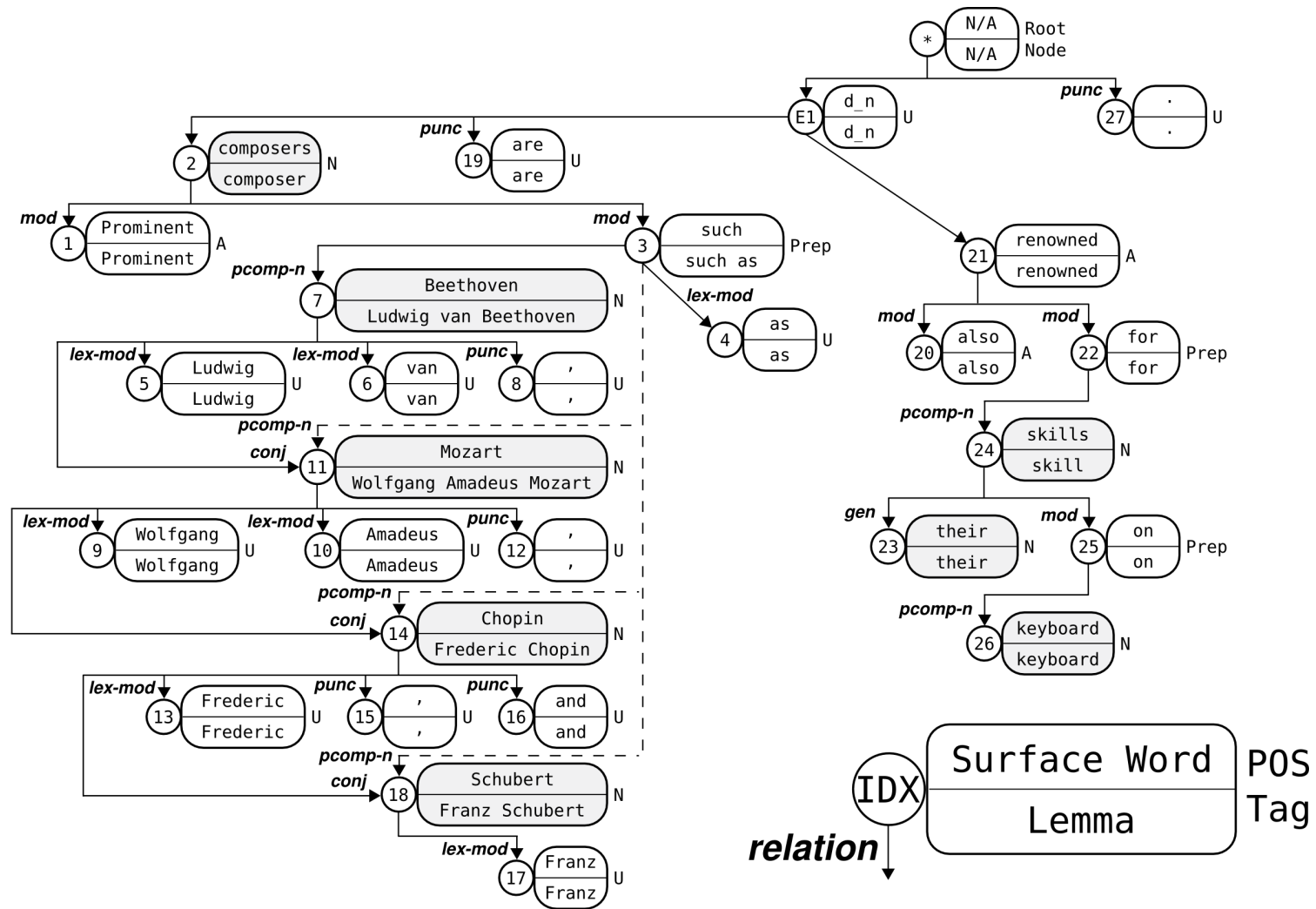


Figure 3.4: A dependency tree representation of the parse output in Table 3.1. The dashed lines represent the additional links for conjunctions

We concatenate all dependency edges from the source node to the destination node to form a dependency path for a noun pair. Finally, to generalise the dependency paths, we remove all the noun lemmas found in the dependency paths. For example, the dependency path between *skill* and *keyboard* in the example sentence above is `N:mod:Prep,on,on,Prep:pcomp-n:N`. These dependency paths serve as pattern representations in our experiments.

We categorise a dependency path into three types:

- **Directed top-down (dtd)** path. We categorise a dependency path to be directed top-down if the path between a pair of nouns can only be obtained by treating the edges as directed edges (based on their original directions), and the paths connect all the nodes from the source noun to the destination noun. We append the prefix `dtd--` to such dependency paths.
- **Directed bottom-up (dbu)** path. The reverse of a directed top-down path, which can only be obtained by treating the edges as directed edges but **reversing their original directions**. The resulting paths connect all the nodes from the destination noun to the source noun. We append the prefix `dbu--` to such dependency paths.
- **Undirected (ud)** path. We categorise a dependency path to be undirected if neither a directed top-down path nor a directed bottom-up path can be obtained. In other words, the dependency path can only be obtained by treating the edges as undirected edges. We append the prefix `ud--` to such dependency paths.

In the majority of our experiments, we limit ourselves to the directed paths (`dbu` and `dtd` paths). We avoid using undirected paths due to our use of large corpora which potentially generates a large number of paths resulting in a massive feature set that is not able to fit in the main memory. We extract undirected paths in an experiment involving a smaller corpus in Section 5.4.

Nouns connected with conjunction relations (such as the words *and*, or *or*, or by commas in listings — as identified by the MINIPAR `conj` relation) present a special

Destination noun	Dependency path
Ludwig van Beethoven	dtd--N:mod:Prep,such as,such as,Prep:pcomp-n:N
Wolfgang Amadeus Mozart	dtd--N:mod:Prep,such as,such as,Prep:pcomp-n:N,N:conj:N
Frederic Chopin	dtd--N:mod:Prep,such as,such as,Prep:pcomp-n:N,N:conj:N,N:conj:N
Franz Schubert	no link

Table 3.2: Dependency paths between *composer* and the listed destination nouns

case. *Ludwig van Beethoven*, *Wolfgang Amadeus Mozart*, *Frederic Chopin* and *Franz Schubert*, are examples of such nouns in Table 3.1. In MINIPAR, only the first noun (*Ludwig van Beethoven* in our example) of a conjunction possesses a relation link with a node outside the given conjunction (*such as* in this case).

This situation is not ideal during the link detection process because of the length constraint imposed on the dependency paths. In the example above, no link is detected between *composer* and *Franz Schubert*, primarily caused by the three `conj` edges between *Franz Schubert* and the first noun of this conjunction, *Ludwig van Beethoven*. The list of the dependency paths between *composer* and all the members of conjunctions are listed in Table 3.2.

As shown in Table 3.2, *Franz Schubert* is the only noun in the conjunction to not have a link with *composer*. Also note the repeated `N:conj:N` string in the dependency paths of *Frederic Chopin*, resulting in different dependency paths. Ideally, as the noun ordering in this conjunction is arbitrary, all the members of the conjunction should possess identical dependency paths (with any other nouns outside this conjunction).

In order to do this, we need to treat these nouns as a set, rather than list of individual nouns. We achieve this effect by connecting additional links between the first node not associated with the current conjunction with the non-first nodes of the conjunction (Snow *et al.* 2005), effectively serving as “short-cuts” during the link detection and dependency path formation processes. These additional links are represented by the dashed-lines in Figure 3.4.

We form the candidate noun pairs for extraction in our task by identifying noun

pairs that are linked by four or less edges within a sentence, ignoring other candidates linked with more than five edges. As such, all noun pairs in the extraction candidate set must have cooccured in the same sentence. However, as Stevenson (2007) pointed out, facts or relations can be expressed across sentence boundaries. While we agree with this assessment, extracting patterns between noun pairs across sentences in our experiments is challenging, considering MINIPAR is engineered to produce a dependency tree one sentence at a time.

Satellites

So far, we have only considered using information from the edges in the shortest path of a noun pair in our dependency paths. However, as evidenced in Hearst (1992), there are certain useful function words that provide (hypernymy) relation clues but are not found in the direct path between that particular noun pair (e.g. *such* in the pattern **such** *X* **as** *Y*). Moreover, the shortest path between two nouns in MINIPAR might not include nodes representing words between the two entities. For example, the shortest path connecting *X* and *Y* in the phrase *X and other Y* are **N:conj:N** without the crucial information of the function words *and* or *other* which signify the hypernymy relation.

In their work, Snow *et al.* (2005) addressed the two challenges above using a heuristic by introducing additional links (referred to as *satellite links*) connecting such words to the dependency path as produced by MINIPAR. They define a satellite link as an edge that connects a *leaf node* (a node without descendants) to the noun nodes found on either side of the dependency path. The leaf nodes themselves are child nodes of the noun nodes representing the nouns in the noun pair. In total, the satellite links can only connect two additional nodes to the dependency path, one per noun in the noun pair.

Unlike the heuristic introduced by Snow *et al.* (2005), we do not limit our satellites to a small set of function words. We define satellite links as links that connect every child node of a noun node, if and only if the child nodes are either a leaf node, or a node with only one child node (i.e. a *stick* of length 2). Figure 3.5 shows examples of

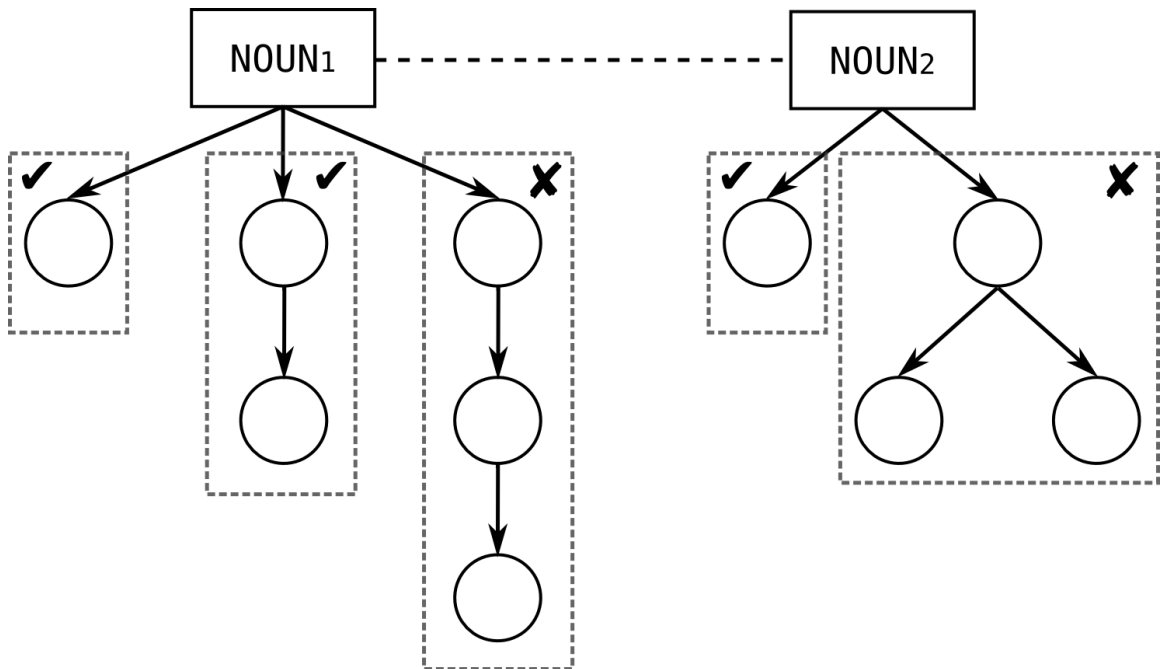


Figure 3.5: $noun_1$ has two satellite links, with one non-satellite link (the child nodes are a stick of length 3). $noun_2$ has one satellite link, with one non-satellite link (the child nodes attached are not a leaf node or a stick of length 2)

both satellite and non-satellite links.

The final pattern extracted from MINIPAR for a given noun pair is represented by three separate components: the dependency path between the two nouns in the pairing, and the satellites of each noun. Using Figure 3.4, we list all the noun pairs, the dependency paths found for every noun pair, and their satellites (if applicable) in Appendix A.

3.4 Machine Learning

Our RE task is a binary classification task, i.e. predicting whether a noun pair has a given relation or not. The approach used in this thesis is a *supervised classification task*, where a classifier to perform relation extraction is produced via *supervised learning*. The main goal of supervised learning is to learn a model from a *training set*,

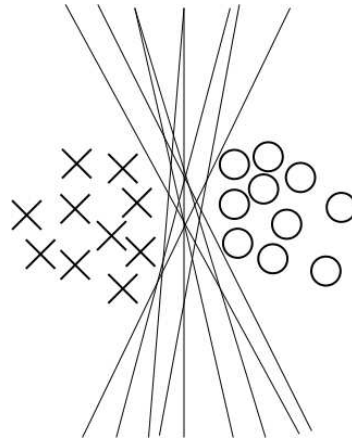


Figure 3.6: A linearly separable dataset with possible separating hyperplanes

which generalises the relationship between the attribute set and the class label of the training data. A classifier is then built using the mentioned model, and its accuracy is evaluated by applying it to the *test set*, of which the class labels are masked.

The main machine learning algorithm used in this thesis is Support Vector Machines (SVM), while holdout and (stratified) cross-validation methods are used to evaluate classifier accuracy. They are discussed below.

3.4.1 Support Vector Machines

To explain the idea of Support Vector Machines (SVM), it is first necessary to introduce the concept of an *optimal hyperplane*.

Suppose a dataset contains instances of two classes which are *linearly separable*, i.e. there exists a hyperplane such that it perfectly discriminates the instances of the two classes (see Figure 3.6 for an example of such dataset with two classes, represented by crosses and circles). The hyperplane acts as a decision boundary that separates the two classes.

However, as shown in Figure 3.6, there are infinite such hyperplanes for any linear separable dataset. The objective of the learning task is to find a hyperplane that best generalises the data. Vapnik (1995) proposed to find an *optimal hyperplane* that maximises the *margin* between the two classes of the dataset. The *margin* between

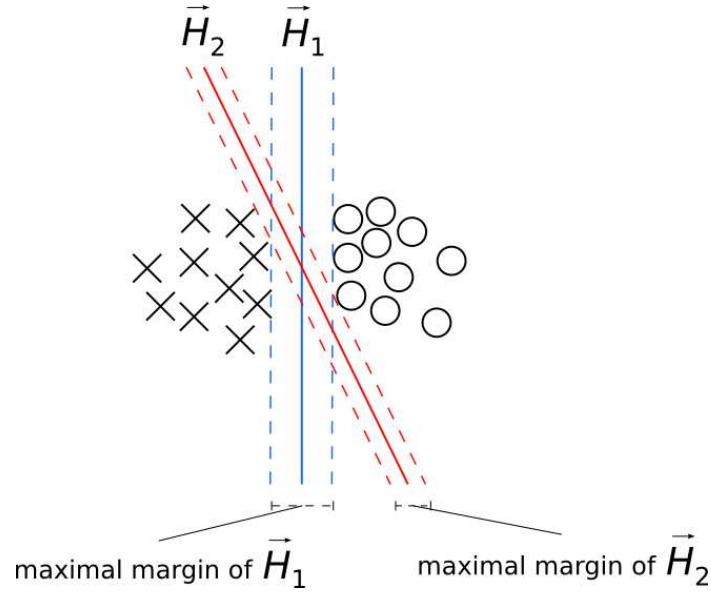


Figure 3.7: The margin difference between hyperplane $\vec{H}_1$ and $\vec{H}_2$

the two classes can be derived from the distance between the two instances — each represents one class — closest to the hyperplane. For an optimal hyperplane, this set of instances is called the *support vectors*. Figure 3.7 illustrates the margin comparison between two hyperplanes $\vec{H}_1$ and $\vec{H}_2$, with $\vec{H}_1$ being the optimal hyperplane.

The process of finding the optimal hyperplane can be formally defined as follows. Consider a dataset with n instances with two possible classes, $y \in \{-1, 1\}$. It can be represented as a set of tuples $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)$, where $\vec{x}_i$ and y_i correspond to the features and the class of i^{th} example respectively. A hyperplane separating the two classes can be defined by the following equation:

$$\vec{w} \cdot \vec{x} + b = 0 \quad (3.1)$$

Following Equation (3.1), it is possible to represent the support vectors, i.e. two closest hyperplanes (one hyperplane per side) parallel to the hyperplane as described in Equation (3.1) as:

$$\vec{w} \cdot \vec{x} + b \geq 1 \quad \text{if } y_i = 1, \quad (3.2)$$

$$\vec{w} \cdot \vec{x} + b \leq -1 \quad \text{if } y_i = -1 \quad (3.3)$$

The objective of learning support vector machines is to maximise the margin d between these two hyperplanes, which can be calculated via:

$$d = \frac{2}{\|\vec{w}\|} \quad (3.4)$$

Putting everything together, the entire learning task can be formalised as the following constrained optimisation problem:

$$\min_{\vec{w}} \frac{\|\vec{w}\|^2}{2} \quad (3.5)$$

$$\text{subject to: } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, i = 1, 2, \dots, n \quad (3.6)$$

Using Lagrange multipliers and Karush-Kuhn-Tucker (KKT) conditions (for details see Vapnik (1995) and Tan *et al.* (2006)), it can be shown that:

$$\vec{w} \cdot \vec{x} + b = \left(\sum_{i=1}^n \lambda_i y_i \vec{x}_i \cdot \vec{x} \right) + b \quad (3.7)$$

where λ_i are Lagrange multipliers, with $\lambda_i = 0$ for all $\vec{x}_i$ where $y_i(\vec{w} \cdot \vec{x}_i + b) \neq 1$. Using this formulation, support vectors are sets of $\vec{x}_i$ where $\lambda_i > 0$. After solving for support vectors $\vec{x}_i$, λ_i , and b , we can classify a test instance $\vec{z}$ by the following function:

$$f(\vec{z}) = \text{sign}(\vec{w} \cdot \vec{z} + b) = \text{sign} \left(\sum_{i=1}^n \lambda_i y_i \vec{x}_i \cdot \vec{z} \right) + b \quad (3.8)$$

Support vector machines based on the maximal margin concept can only be used on a linearly separable dataset. However, noise tends to occur in the training data which in turn may cause the data to be non-linearly separable (Cristianini and Shawe-Taylor 2000). In order to tolerate small amounts of noise, the training can be approached via the *soft-margin* approach by introducing positive-valued slack variables ($\xi > 0$) into the problem formulation.

In this thesis, we use BSVM⁶ as the machine learner (Hsu and Lin 2002). The default settings of BSVM are used throughout, except for the choice of kernel where we employ a linear kernel.

⁶<http://www.csie.ntu.edu.tw/~cjlin/bsvm/>

3.4.2 Classifier Evaluation Methods

Holdout Method

The holdout method provides a simple procedure in evaluating the classifier accuracy. The entire dataset is partitioned into two mutually exclusive fixed sets, named training and test sets. A classifier is then built using the training set only. Finally, the classifier accuracy is evaluated on the test set.

The amount of training and test data poses an interesting trade-off in supervised learning: more training data is preferable to produce a good classifier; and conversely, more test data is required to accurately estimate the classifier accuracy (Witten and Frank 2005).

In the holdout method, there is no rule on the split proportion between training and test sets, although it is common to either equally split the training and test data, or to assign two-thirds of the data to the training set, and use the remaining one-third for the test set.

A more important issue is to know whether the training and test sets are representative of the entire dataset or not. One significant aspect to check for representativeness is the class distribution. A training and test set with similar class distribution is arguably more representative of a dataset compared to another training and test sets where the class distribution is very different. In the latter case, the learning process is comparably more difficult (especially if one class is underrepresented in the training class), and it can subsequently affect the classifier accuracy. To avoid this problem, we can *stratify* the data, that is we sample the training and test sets by preserving the dataset class distribution in both the training and test sets.

Cross Validation

In the holdout method, both the training and test sets are fixed, and therefore, the classifier accuracy can only be evaluated once. If the training and test sets are not representative of our task, the evaluation result may not be an accurate estimate of the classifier accuracy. A general way of solving this is to repeat the process multiple times with different training and test sets at each iteration. The overall accuracy can

be derived from averaging the classifier accuracy on different iterations.

Cross validation is similar to the idea above. In k -fold cross validation, the dataset is split into k equal-sized folds or partitions (k is usually set to 10). It is also common practice to perform data stratification during the data partitioning. Following this, we train and test a different classifier for each of k iterations, where in each iteration a specific fold is used as the test set, while the remaining folds are used as the training set. The classifier accuracy of each iteration is then averaged to produce a single classifier's accuracy estimate.

Stratified cross validation arguably offers a better estimate of classifier accuracy compared to the holdout method because it makes effective use of the entire dataset for both training and testing purposes. By repeating the process multiple times, stratified cross validation minimises the variance in the classifier accuracy and subsequently offers a better estimate of confidence. Its downside is the computationally expensive iterative process.

3.5 Evaluation Metrics

We use evaluation metrics commonly used in Relation Extraction tasks, namely precision, recall, and f-score, in evaluating our RE system's accuracy. Originally developed in 1960s as comparative measures for different indexing systems, precision and recall have become standard metrics in evaluation of RE systems (Jackson and Moulinier 2002; Snow *et al.* 2005; Girju *et al.* 2006; Banko and Etzioni 2008).

Let us assume the gold-standard dataset has T instances for a specific relation. Suppose a relation extraction system extracts N instances, of which c is the number of instances found in the gold-standard. We define *precision*, and *recall* evaluation metrics in our experiment as:

$$\begin{aligned} \textit{Precision} = P &= \frac{c}{N} \\ \textit{Recall} = R &= \frac{c}{T} \end{aligned}$$

For a given relation and its gold-standard, the precision is an indicator of a system's accuracy in predicting instances with that relation. Recall is an indicator of

a system's coverage in extracting instances with that relation, relative to its gold-standard.

F-score is the harmonic mean of precision and recall. It is useful as a single indicator of the balance between the precision and recall metrics, and is usually defined as:

$$F\text{-score} = F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

where β is the relative importance given to precision and recall (Jackson and Moulinier 2002). In our evaluation, we set β to 1, giving equal weight to both precision and recall. We can then simplify our f-score definition to:

$$F\text{-score} = F_1 = \frac{2PR}{P + R}$$

3.6 Summary

This chapter presented a description of the various resources and their adaptation to the RE task addressed in this thesis. We started with a discussion of WORDNET as our relation labelling resource, and various WORDNET relations we experiment with. Next, we described the implications of the nature of WORDNET, which provides information at the sense-level, to our relation extraction task which operates at the word-level. As part of this, we also discussed SEMCOR, which provides the sense ordering information in WORDNET, and also the potential impact of WORDNET sense ordering on our task interpretation.

Next, we described both GIGAWORD and WIKIPEDIA, and various forms of pre-processing required in adapting them to this research. Following this, we focused our discussion on pattern representations. We described how we use MINIPAR, and how we form pattern representations based on its output. Furthermore, we discussed several shortcomings of the pattern representation, and introduce several heuristics to address them.

Finally, we reviewed the topic of machine learning and classifier accuracy. In particular, we briefly described SVM, the primary machine learning method in this

thesis. This is followed by discussion regarding two classifier evaluation methods. We concluded the chapter by presenting the details of evaluation metrics (precision, recall, and f-score) used throughout this thesis.

Chapter 4 describes how we use all the resources discussed in this chapter in building a relation extraction system. Related to this, we discuss the assumptions made in building the system, and investigate the impact of these assumptions on system accuracy.

Chapter 4

Supervised Relation Extraction

RE systems often rely on patterns that appear between a pair of entities as their primary feature representation (Pantel and Pennacchiotti 2006; Girju *et al.* 2006). The traditional RE approach exploits a small set of manually-crafted patterns that strongly indicate a specific relation type (e.g. Hearst (1992)). This process usually involves manual inspection of the data, which is time consuming. Furthermore, this expensive process needs to be iterated every time we aim to learn a different relation type.

To address this, supervised machine learning methods that automatically learn high-quality extraction patterns from the data have been proposed (Zelenko *et al.* 2003; Snow *et al.* 2005). Unlike the traditional hand-crafting approach, supervised methods shift the learning responsibility from humans to machine learners. In doing so, the amount of time to learn and induce patterns can be reduced, which in turn allows the possibility of learning from a larger dataset.

There are fundamental differences in the learning style of the traditional and the supervised RE approaches. In the traditional approach, the task objective is to identify a small set of patterns highly associated with a specific relation, by generalising from a set of positive examples. The extraction process from the data relies entirely on whether an entity pair appears in one of the patterns or not. In contrast, a supervised system learns from both positive and negative examples. The use of negative examples during learning provides another dimension not used by traditional approach, as

the negative examples can direct the learner away from extracting non-relation pairs. Further, the system does not rely on a small set of patterns for extraction. Instead, it relies on weighting information from a large set of patterns learnt from the training data.

The machine learner learns the utility of the patterns that appear with both positive and negative examples using corpus data. Arguably, not all of the patterns collected are of the same quality: some patterns are more relation-identifying (i.e. high-quality) than others. A reasonable expectation is to find a mixture of a small number of high-frequency patterns strongly indicative of the relation of interest (high-quality patterns) and a large number of low-frequency noisy patterns that do not indicate the given relation (low-quality patterns). In learning, the learner may assign more weight to the high-quality patterns, and conversely, weigh down low-quality patterns.

The supervised methodology is more flexible compared to the traditional approach. It can be used to identify a set of high-precision patterns similar to the traditional approach-based system by pattern-selection based on the pattern weights. Further, it allows the options of fine-grained control of the system behaviour by manipulating it using common machine learning techniques, e.g. changing the values for system classification parameters, or training on different domains or data.

An example of a supervised method is Snow *et al.* (2005)'s methodology which uses a machine learner to automatically learn patterns for extracting hypernym pairs based on corpus data. The learner generalises from a set of training examples containing both positive and negative noun pairs along with their pair-pattern cooccurrence frequency. Their system has been shown to perform better than WORDNET in identifying noun pairs with the hypernym relation from newswire articles. Moreover, it has been shown to be useful in a semantic taxonomy induction system (Snow *et al.* 2006).

The high performance, practicability, and adaptability of Snow *et al.* (2005)'s methodology presents ample opportunities for RE research. In this chapter, we build our own system based on their methodology using the resources described in Chapter 3. These resources are: the corpora for collecting the noun pairs, patterns, and

the noun pair-pattern frequency of occurrence (GIGAWORD and WIKIPEDIA), a relation labelling resource to label noun pairs (WORDNET and SEMCOR), a pattern representation tool (MINIPAR), and a machine learner (SVM).

The focus of this chapter is to investigate the extensibility of Snow *et al.* (2005)’s methodology by applying it to other relations as, to our knowledge, there has been no previous work in adapting this methodology to other relations. In order to do this, we build our own system following their methodology, as described in Section 4.1. We compare the results of our system in extracting hypernym tuples with those of Snow *et al.* (2005) in Section 4.2, and show that our system achieves comparable performance to Snow *et al.* (2005). Next, we investigate its extensibility by applying our system to antonymy and synonymy in Section 4.3.

When implementing our system, we identified several experimental parameters related to noun pairs sampling and feature generation. We hypothesise that these parameters are critical to the task setup, and investigate their effect to RE performance and the interpretation of our RE task in Section 4.4.

4.1 System

4.1.1 Task Overview

We define our RE task to be determining whether a noun pair corresponds to a given relation or not, based on the features associated with it, using supervised methodology.

Formally, given:

1. a noun pair (n_i, n_j)
2. (a) the vector of features $V = \{f_1, f_2, f_3, \dots, f_n\}$, where $f_i \rightarrow \mathbb{R}$
 (b) the instantiation of V for (n_i, n_j) : V_{ij}
3. a relation **rel**

our RE task is:

$$(n_i, n_j, \text{rel} | V_{ij}) \implies C, C \in \{-1, 1\}$$

where:

$C = 1$ indicates that (n_1, n_2) has a *rel* relation

$C = -1$ indicates that (n_1, n_2) does not have a *rel* relation

We perform RE on three semantic relations: antonymy, synonymy, and hypernymy, over noun pairs found in GIGAWORD and WIKIPEDIA. We use WORDNET and SEMCOR to label the relations between the identified noun pairs. The main feature used for learning is the patterns between the noun pairs, as represented by MINIPAR dependency paths.

4.1.2 Preprocessing

We follow a similar preprocessing step to Snow *et al.* (2005). First, we tokenise and split the corpora into sentences, parse the sentences, form the noun pairs, and extract the patterns between the noun pairs (see Chapter 3). Next, we form our pattern representations using our heuristic as described in Section 3.3.1, which is an extension of Snow *et al.* (2005)'s.

We discussed in Section 3.3.1 how we form noun pairs for extraction from the output of MINIPAR, where we require noun pairs to cooccur in a sentence, and be connected together by four edges or less (as represented by the MINIPAR output). However, not all these noun pairs are of interest, e.g. *dummy_node* strings as a by-product of MINIPAR (see Section 3.3.1). These noun pairs are filtered from our data by first removing those where one or both nouns are labelled as *dummy_node*. Next, we exclude noun pairs (n_i, n_i) as it is not useful to extract a relation between a pair of identical nouns.

We further restrict our nouns to only contain alphabetic characters, hyphens, or spaces, conforming to the regular expression `[a-zA-Z -]*`. This is because the majority of nouns in our relation labelling resource (WORDNET) do not contain any special characters, e.g. %, &, or #. The nouns are then lower-case folded. We also

Noun Pairs	Pattern 1	Pattern 2	Pattern 3
(n_1, n_2)	5	7	10
(n_3, n_4)	0	5	0
(n_1, n_4)	23	4	0
(n_2, n_3)	17	9	1
(n_5, n_6)	31	7	0

Table 4.1: Example noun pairs and their frequency of occurrence

exclude numbers because they mostly represent date information, which is uninteresting for our relations of interest. Our nouns, however, are not limited to single word tokens, as a significant number of proper nouns are made up of two or more words with spaces as the token separator.

After filtering, 53,082,302 and 71,005,164 unique noun pairs are extracted from GIGAWORD and WIKIPEDIA, respectively. For each of these noun pairs, our system keeps a record of the patterns cooccurring with them (and each pattern’s frequency of occurrence). For illustrative purposes, Table 4.1 shows the frequency record for 5 noun pairs across 3 different patterns.

4.1.3 Labelling Relations

Our next step is to label these noun pairs using WORDNET and SEMCOR. Noun pairs where one or both nouns do not have a WORDNET entry are excluded from our data.

Relation Labelling Heuristic

To use WORDNET information, we first need to map the nouns in the pairing to their WORDNET entries. However, as the nouns in our data are not labelled for WORDNET sense, the mapping process is essentially a word sense disambiguation task on WORDNET senses — a difficult AI-complete problem which is not our main

research focus (see Section 3.1).

To compensate for performing word sense disambiguation on our data, we adapt Snow *et al.* (2005)’s heuristic to map each noun to its corresponding *first sense* in WORDNET, i.e. its *most commonly used* sense (refer to Section 3.1 on sense-ordering in WORDNET). This heuristic is equivalent to the most frequent sense baseline, a hard to beat baseline, commonly used in WSD evaluation (Navigli 2009).

Having mapped the nouns to their WORDNET first senses entries, we label a noun pair based on the WORDNET relation information between the pair. For a given relation, we label a noun pair as a positive instance iff there is a relation link between the WORDNET first senses of the two nouns. Conversely, we label a noun pair as negative iff the given relation does not hold between any of the senses (first or otherwise) of the two nouns.

We consider two types of relations in our experiment: flat-symmetrical (antonymy and synonymy) and hierarchical-asymmetrical (hypernymy). Due to the structural difference (flat vs. hierarchical), there is a practical difference in the labelling process.

We describe the labelling process for antonymy and synonymy below. For a given noun pair (n_i, n_j) , we first query the relation links for the first sense of n_i (using `WordNet::QueryData`; see Section 3.1). We label the pair as positive iff the first sense of n_j is returned by WORDNET as part of this query. If the first sense of n_j is not found in the answer set for the relation query, we remove the first sense restriction, and check whether there is a link between a sense of n_i and a sense of n_j (but not between their first senses) by an iterative process. We label the pair as negative iff there is no such link between all possible sense combinations between n_i and n_j . Finally, we exclude the pair iff there is a link between n_i and n_j and the link is not between the first-senses of both nouns. This process is illustrated in Figure 4.1.

The labelling process for hypernymy is similar to antonymy and synonymy. However, due to its hierarchical nature, we look for possible hypernymy relations by detecting links recursively up the WORDNET hierarchy (considering all possible ancestors), i.e. we do not limit the hypernymy link to be only between an immediate ancestor-descendant link. Therefore, unlike antonymy and synonymy, the process of detecting hypernym links between a noun pair (n_i, n_j) may involve a set of non- n_i or

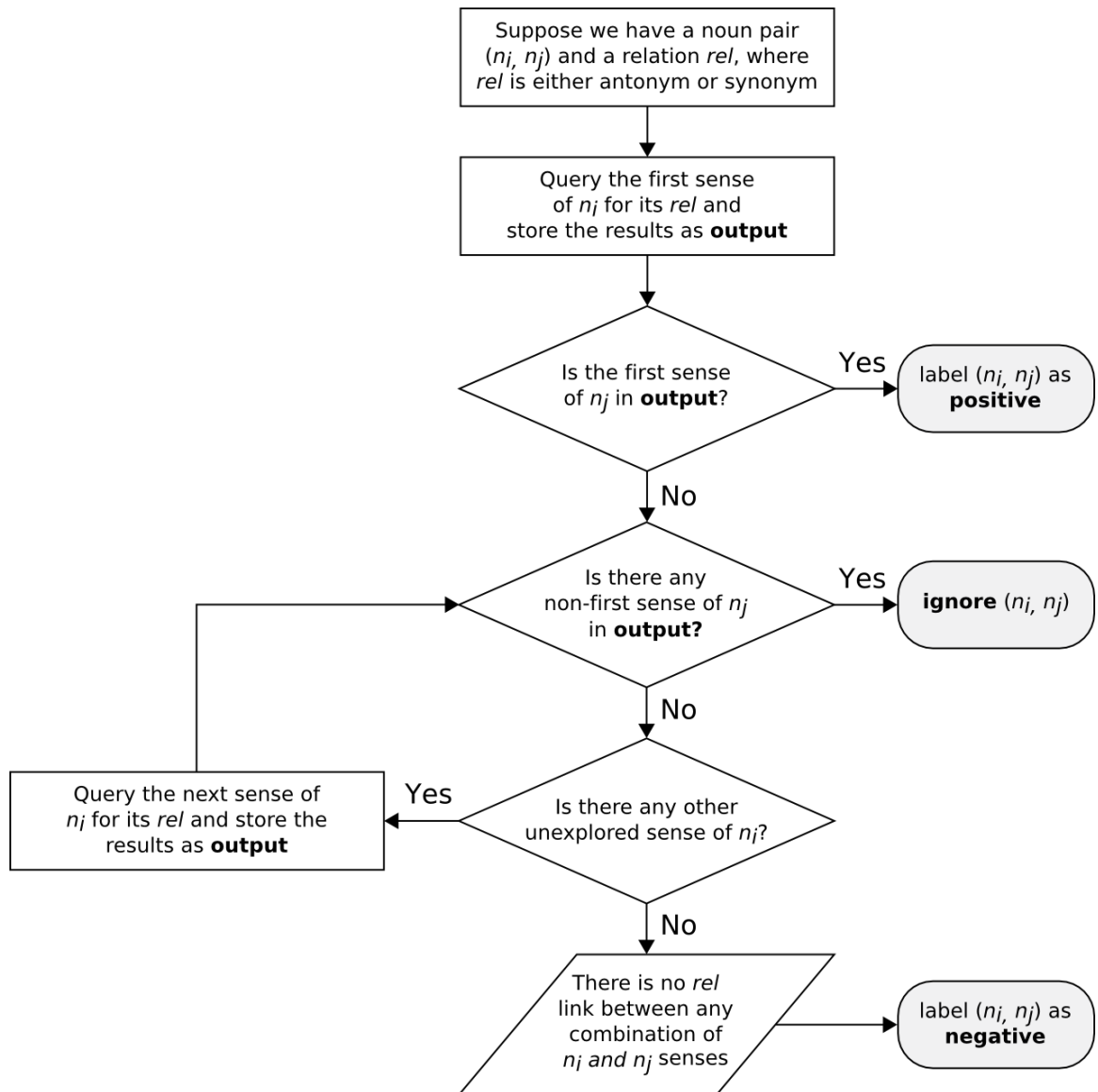


Figure 4.1: Flowchart for labelling antonym and synonym

n_j nouns. For example, when detecting the hypernym relation for $(car, vehicle)$, we may need to traverse *motor_vehicle*, *self-propelled_vehicle*, and *wheeled_vehicle*, which are all hypernyms of *car* and hyponyms of *vehicle*. Figure 4.2 illustrates the labelling process for hypernym.

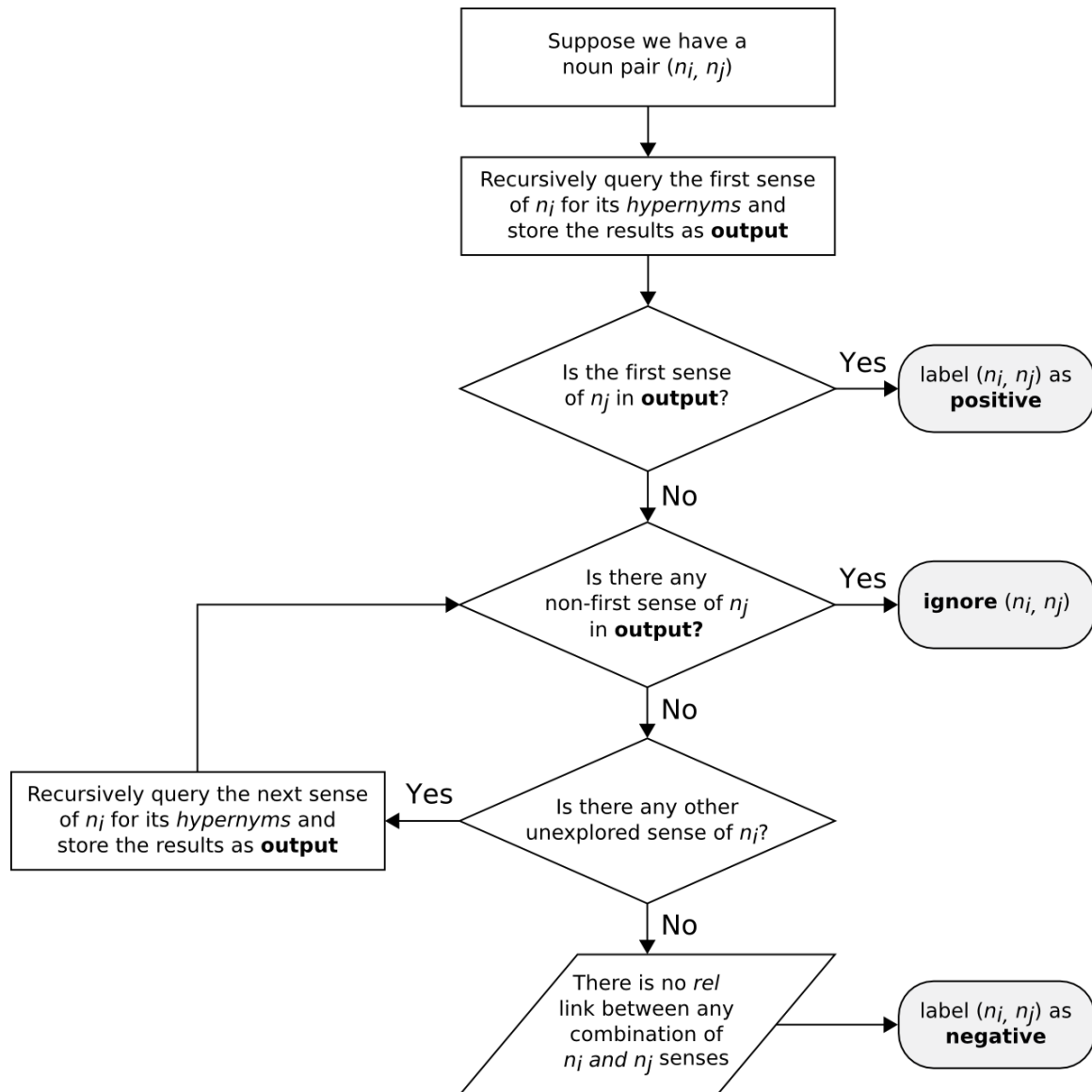


Figure 4.2: Flowchart for labelling hypernym

SEMCOR Role

There is a slight difference between our relation labelling process to Snow *et al.* (2005)'s in employing SEMCOR. In Snow *et al.* (2005), nouns must have exactly one SEMCOR-tagged sense. They choose to ignore nouns with more than one SEMCOR-tagged sense to reduce the number of false hypernym pairings due to polysemy. There are only 9,239 out of the 117,798 WORDNET nouns that meets this constraint.

In comparison, our heuristic is less strict: we only require nouns to have at least one SEMCOR-tagged sense. This is in response to Snow *et al.* (2005)’s assumption of ignoring polysemous nouns, which we consider to be unrealistic. As a result of this heuristic, there are 11,846 WORDNET nouns eligible for our experiment.

With the SEMCOR constraint, the possible noun pairs in both Snow *et al.* (2005) and our experiments are limited to commonly-used WORDNET nouns. While there is significant overlap in each case, our task is arguably more difficult and realistic as polysemy is more apparent.

Annotation Examples and Discussion

Using our labelling heuristic, we present annotation results of a number of noun pairs in Table 4.2.

We expect the majority of positive and negative noun pairs to be clearly attested, as our heuristic is very conservative. That is, due to the first-sense relation link requirements for positive instances, there is little likelihood that the positive label is assigned to a noun pair of which one or both nouns are used in a manner other than their WORDNET first senses. The conservativeness of our heuristic is even more pronounced for the negative instances, where the negative label is assigned to a noun pair only when no relation exists between all possible sense combinations of the nouns. The first six noun pairs of Table 4.2 are examples of such noun pairs, where the relations between them are clearly-attested.

There are two types of noun pairs excluded in Table 4.2. The first type is where the noun pairs are excluded across all three relations due to one or both nouns not appearing in WORDNET (**lgn-W** instances), or in SEMCOR (**lgn-S** instances). The second type is for pairings that are excluded as the relations between them are not via the first senses of nouns (**lgn-sense** instances). The difference between the two types is that the exclusions for the **lgn-sense** instances are only excluded for a specific relation, rather than excluded across all relations of interest, like the **lgn-W** and **lgn-S** instances. For example, the noun pair (*heir*, *successor*) is labelled as a negative instance for both antonymy and hypernymy, but is ignored for synonymy. We explore

Noun Pairs	ANTO	SYNO	HYPER
(<i>daughter, son</i>)	+	−	−
(<i>future, past</i>)	+	−	−
(<i>salary, wage</i>)	−	+	−
(<i>film, movie</i>)	−	+	−
(<i>new york, city</i>)	−	−	+
(<i>week, period</i>)	−	−	+
(<i>michael jordan, basketball player</i>)	Ign-W	Ign-W	Ign-W
(<i>altavista, search engine</i>)	Ign-W	Ign-W	Ign-W
(<i>narrowness, wideness</i>)	Ign-S	Ign-S	Ign-S
(<i>gangster, mobster</i>)	Ign-S	Ign-S	Ign-S
(<i>south australia, state</i>)	Ign-S	Ign-S	Ign-S
(<i>agreement, disagreement</i>)	Ign-sense	−	−
(<i>heir, successor</i>)	−	Ign-sense	−
(<i>chip, food</i>)	−	−	Ign-sense

Table 4.2: Annotation results using the proposed relation labelling heuristic. “+” = positive label, “−” = negative label, “Ign” = Ignored noun pairs due to: (i) “-W”: one or both nouns do not appear in WORDNET, (ii) “-S”: one or both nouns do not appear in SEMCOR, or (iii) “-sense”: the relation link is between non-first senses of both nouns

this distinction between the two types further in Section 5.2.1.

Looking at Ign-S and Ign-sense instances, our heuristic excludes noun pairs which may be perceived to be clearly-attested (e.g. *hypernym(chip, food)*, *synonym(gangster, mobster)* and *antonym(agreement, disagreement)*). Further, as the learning and evaluation processes only involve the list of eligible noun pairs, our experiment will not include pairs that have been excluded or ignored. This begs the question whether

our task setup is too limiting (an issue we address further in Chapter 5). However, for the time being, we continue our experiment with this heuristic.

4.1.4 Gold-Standard Dataset Formation Process

Having labelled the eligible noun pairs, we now form our datasets based on the two system parameters used by Snow *et al.* (2005). The first system parameter relates to the pattern diversity of the noun pairs, while the second relates to the noun pair label (class) distribution.

We first look at the system parameter relating to the pattern diversity of the noun pairs. We term this parameter as Θ_{n_i, n_j} , where we require noun pairs to occur across at least P unique patterns (i.e. $\Theta_{n_i, n_j} \geq P$). The idea behind this parameter is to avoid data sparsity during learning by removing noun pairs that do not cooccur with a sufficient amount of different patterns.

Table 4.3 shows the number of our noun pairs based on the number of patterns they appear with. Using noun pairs from GIGAWORD as an example, we can observe that we have more than 10 million pairs in total (i.e. $P = 1$, as each noun pair in our data must have occurred with at least one pattern). Further, we can observe that most of these pairs only occur with small number of patterns, as there are only around 2.8 million noun pairs that have occurred with at least five different patterns ($P = 5$).

Due to their large numbers, training classifiers on all of these pairs is resource-expensive (time and memory-wise). Furthermore, these noun pairs are arguably less informative in the learning process compared to the more frequently appearing noun pairs that occur across a variety of patterns. For these reasons, Snow *et al.* (2005) applied the threshold $P = 5$ in their experiment. We agree with their reasonings for the time being, but further hypothesise that this constraint is an important parameter in the task setup and experiment with other values of P ($P \in \{5, 10, 20\}$).

The second parameter we consider relates to the class distribution of the noun pairs. It is desirable that the class distribution used in a RE task reflects that of the class distribution in the real-world situation (Snow *et al.* 2005). To address this

Θ_{n_i, n_j}		GIGAWORD			WIKIPEDIA		
$\geq P$		ANTO	SYNO	HYPER	ANTO	SYNO	HYPER
$P = 1$	+	126	1,422	16,929	125	1,682	19,478
	-	10,562,560	10,554,733	10,497,835	9,816,811	9,808,366	9,745,560
$P = 5$	+	79	668	6,946	87	815	8,217
	-	2,740,248	2,736,147	2,709,974	2,281,994	2,277,394	2,248,195
$P = 10$	+	64	453	4,472	67	530	5,239
	-	1,510,174	1,507,386	1,489,846	1,176,422	1,173,234	1,153,973
$P = 20$	+	53	290	2,690	60	339	3,041
	-	786,326	784,556	773,549	563,152	561,115	549,503

Table 4.3: Comparisons of class distribution for instances extracted from GIGAWORD and WIKIPEDIA for various thresholds P of Θ_{n_i, n_j}

issue, Snow *et al.* (2005) hand-labelled 5,387 noun pairs picked from randomly-selected paragraphs in their data. They observed that the ratio of positive ($|\mathbf{P}|$) to negative hypernym pairs ($|\mathbf{N}|$) is 1:50.

As shown in Table 4.3, the number of negative pairs greatly outnumbers the number of positive pairs. Further, the ratio of positive to negative pairs is different among the three relations in our data, P values for the Θ_{n_i, n_j} parameter, and across the corpora. The ratio ranges from a factor of hundreds for hypernymy to well over tens of thousands for antonymy.

Note that the ratio of positive to negative pairs observed by Snow *et al.* (2005) is for the hypernymy relation, while we experiment with two additional relations. It is unreasonable to assume that the ratio for hypernymy is applicable to antonymy and synonymy, as the class distribution for the different relations is potentially quite different. However, considering that the process of hand-labelling antonymy and synonymy instances in order to observe their natural class distribution is resource expensive, we make the simplifying assumption for our initial experiments that the

ratio between positive and negative pairs is 1:50 for all of our relations.

As with the Θ_{n_i, n_j} parameter, the ratio parameter is also very important in shaping the composition of the dataset. To investigate the effect of this parameter, we experiment with various values of ratio R where $|\mathbf{N}| = R \times |\mathbf{P}|$ and $R \in \{1, 10, 25, 50, 100\}$.

Note that the actual ratio of negative to positive pairs in our data is greater than the biggest R value for our experiment ($R = 100$). Therefore, some adjustments are required to meet the ratio constraint. There are two approaches for this: increase the number of positive pairs or reduce the number of negative pairs. The first approach is not practical, as the positive pairs are gathered from the entire corpus and it is not possible to acquire more positive pairs without additional data. We prefer the second approach where it is a simple process to reduce the number of negative pairs. The only question is which negative pairs should be filtered.

In Snow *et al.* (2005), only the top $|\mathbf{N}|$ most frequently appearing negative pairs in the data are included in the dataset. The rationale behind choosing these instances is that they are arguably more informative than the less frequently appearing negative noun pairs, akin to the motivation of the Θ_{n_i, n_j} parameter. We apply this filtering method in our experiment.

Our final dataset contains all positive pairs satisfying the Θ_{n_i, n_j} constraint and the chosen negative pairs, which satisfy both the Θ_{n_i, n_j} and $|\mathbf{N}|/|\mathbf{P}|$ constraints. These pairs form our instances for the learning process.

4.1.5 Feature Representations

The next step after the instance selection process is to generate the features for each instance. The main feature used in the system is the frequency of occurrence of the noun pair across each pattern in the data. We show an example of this in Table 4.1, where we record the frequency of occurrence of five noun pairs across three different patterns. Recall that the patterns in our experiment are represented by MINIPAR dependency paths. In the current experiment, we extract features from the directed top-down and directed bottom-up paths (see Section 3.3.1 for details).

$\Lambda pat_k \geq Q$	GIGAWORD	WIKIPEDIA
$Q = 1$ (all patterns)	136,307,240	128,114,140
$Q = 5$	5,120,966	3,935,698
$Q = 10$	2,316,150	1,756,760
$Q = 20$	1,097,524	827,964
$Q = 50$	416,692	314,620

Table 4.4: The number of patterns extracted from GIGAWORD and WIKIPEDIA for various thresholds Q of Λpat_k

Similar to our instance sampling via the $\Theta n_i, n_j$ parameter, we hypothesise that patterns occurring only between a small number of unique noun pairs are less informative than patterns occurring across a large amount of unique noun pairs. In order to reduce the feature space during learning, we experiment with the Λpat_k parameter, where only patterns occurring with at least Q unique noun pairs are considered for learning (i.e. $\Lambda pat_k \geq Q$).

Table 4.4 shows the total number of unique directed dependency paths (patterns) found in both corpora. Similar to our observation on the noun pairs, we can observe that there are a large number of patterns that only cooccur with small number of noun pairs (compare the number of patterns in $Q = 1$ to $Q = 5$). To avoid learning from uninformative features, we apply the threshold $Q = 5$ as used by Snow *et al.* (2005). Further, as we hypothesise that the $\Lambda pat_k \geq Q$ parameter may effect the performance of RE, we experiment with additional Q values ($Q \in \{5, 10, 20, 50\}$).

Feature Mapping

In our data, we have:

Set of noun pairs $\{np_1, np_2, np_3, \dots, np_x\}$, where np_i stands for the i th noun pair.

Set of patterns $\{pat_1, pat_2, pat_3, \dots, pat_y\}$, where pat_j stands for the j th pattern.

Noun	Pattern 1						Pattern 2					Pattern 3				
Pairs	f_1^0	f_1^1	f_1^2	f_1^3	f_1^4	f_1^5	f_2^0	f_2^1	f_2^2	f_2^3	f_2^4	f_3^0	f_3^1	f_3^2	f_3^3	f_3^4
(n_1, n_2)	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0
(n_3, n_4)	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
(n_1, n_4)	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0
(n_2, n_3)	1	1	1	1	0	0	1	1	1	1	0	1	0	0	0	0
(n_5, n_6)	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0

Table 4.5: The final feature vector representations for noun pairs from Table 4.1

Frequency of occurrence where $freq(np_i, pat_j)$ indicates the frequency of occurrence of np_i with pat_j .

Feature set $\{f_1, f_2, f_3, \dots, f_z\}$, where f_k stands for the k th feature.

Instance feature vector $V_{np_i} = \{val(f_1), val(f_2), val(f_3), \dots, val(f_z)\}$, where V_{np_i} is the feature of np_i ; and $val(f_k)$ is the value of feature f_k .

In learning, we need to translate the set of patterns to features and assign a value to each feature. The challenge is to find an effective feature representation for the patterns and their frequency of occurrence. One simple approach is to treat each pattern as an individual feature. For a given noun pair np_i , the feature value of a feature f_k is the frequency of occurrence between the pattern pat_k and the given noun pair, i.e. $val(f_k) = freq(np_i, pat_k)$. In this representation, the feature values are *continuous*, as they are based on pattern frequencies of a given noun pair.

Snow *et al.* (2005) achieved their best result by discretising the feature values into a vector of binary values based on thresholding with an increasing power series of degree 2. A given pattern pat_k with a maximum noun pair frequency of $max(pat_k)$ is mapped into L unique feature vectors $\{f_k^0, f_k^1, f_k^2, \dots, f_k^L\}$, where $L = \lceil \log_2(max(pat_k)) \rceil$. For a noun pair np_i , the values for these vectors are:

$$f_k^l = \begin{cases} 1 & \text{if } freq(np_i, pat_k) \geq 2^l \\ 0 & \text{otherwise} \end{cases}$$

Table 4.5 presents the final form of feature vector representations of the examples in Table 4.1.

We follow Snow *et al.* (2005)’s feature discretisation procedure when forming our feature sets for the learners. Because every pattern appearing more than once in corpora is mapped into multiple feature vectors, the size of the entire feature vectors for our machine learner is larger than the number of patterns listed in Table 4.4. Finally, we train our system with a SVM.

4.2 Comparison with Snow *et al.* (2005)

As the purpose of this experiment is to investigate the feasibility of our RE system, it is meaningful to compare our results with those of Snow *et al.* (2005). Strictly speaking, a direct comparison between our system and theirs is not possible due to the differences in the respective experimental details: we used almost 15 times (GIGAWORD) and 7 times (WIKIPEDIA) the amount of the data they used in their experiment, different pattern representations, relation labelling procedure, and machine learner. However, as the underlying architecture of our system closely resembles theirs, a performance comparison can provide an estimate of our success in reproducing their method.

The performance comparison can only be applied to hypernymy, as that is the only relation Snow *et al.* (2005) experimented on. For performance comparison with Snow *et al.* (2005), we use the same parameters as in their experiment, i.e. $\Theta n_i, n_j \geq 5$, $|\mathbf{N}|/|\mathbf{P}| = 50$, and $\Lambda pat_k \geq 5$ and apply 10-fold stratified cross validation to our data. We present a comparison of our best result over the 10-fold with those of Snow *et al.* (2005) over the aforementioned configurations in Table 4.6.

As we can observe in Table 4.6, both our GIGAWORD-trained and WIKIPEDIA-trained systems are more accurate than Snow *et al.* (2005)’s system. Further, it is encouraging to see the reasonably high precision scores in our results, implying that our method is quite successful at identifying patterns indicating hypernymy. This is in accordance with Snow *et al.* (2005)’s conclusion, although they did not report their precision score. Based on these comparisons, we conclude that our RE system

HYPER	Recall	Precision	F-score
Our system (GIGAWORD)	0.632	0.545	0.585
Our system (WIKIPEDIA)	0.465	0.484	0.474
Snow <i>et al.</i> (2005)	?	?	0.348

Table 4.6: Result comparisons for hypernymy between our systems and Snow *et al.* (2005). Snow *et al.* (2005) did not report their recall and precision

is competitive for hypernymy.

We note that the accuracy of our system differs when trained on different corpora. The f-score of the system trained on GIGAWORD is higher than the system trained on WIKIPEDIA. We discuss the effect of corpora on our system further in Section 4.4.2.

4.3 Extending Relation Sets

Having established that our methodology is competitive for hypernymy RE (see Section 4.2), we proceed to investigate its generalisability by applying it to the other two relations: antonymy and synonymy.

First, we build baseline classifiers for all three relations to allow a relative comparison between our supervised methodology to the traditional approach. The baseline classifiers classify a noun pair as having a given relation iff that noun pair occurs at least once in any one of the hand-crafted patterns for that specific relation. The hand-crafted patterns used in the baseline classifiers are listed in Table 4.7.

The patterns used for hypernymy are high-precision patterns discovered by Hearst (1992) in her seminal work. On the other hand, to our knowledge, there does not exist a set of high-precision patterns for antonymy and synonymy. Their baseline patterns listed in Table 4.7 are derived from work related to semantic similarity where two entities are considered semantically closer if they appear in the baseline synonymy patterns (Widdows and Dorow 2002), and conversely, semantically incompatible if they appear in the baseline antonymy patterns (Lin *et al.* 2003).

Relation	Patterns
hypernymy (Hearst 1992)	n_i and other n_j n_i or other n_j n_j such as n_i Such n_j as n_i n_j including n_i n_j , especially n_i
synonymy (Widdows and Dorow 2002)	n_i and n_j n_i or n_j
antonymy (Lin <i>et al.</i> 2003)	from n_i to n_j either n_i or n_j

Table 4.7: Patterns normally associated with the given relation (NB: for hypernymy, n_j is a hypernym of n_i)

Systems	GIGAWORD			WIKIPEDIA		
	R	P	F	R	P	F
Our system (ANTO)	0.886	0.355	0.507	0.816	0.747	0.780
Baseline (ANTO)	0.266	0.127	0.172	0.322	0.128	0.183
Our system (SYNO)	0.871	0.539	0.666	0.860	0.581	0.693
Baseline (SYNO)	0.503	0.031	0.058	0.480	0.028	0.053
Our system (HYPER)	0.632	0.545	0.585	0.465	0.484	0.474
Baseline (HYPER)	0.210	0.132	0.162	0.205	0.154	0.176

Table 4.8: Comparison between our systems and the baselines' accuracy over GIGAWORD and WIKIPEDIA

Table 4.8 shows the accuracy of our systems trained on two different corpora over the three relations, using the same parameter settings as used in Snow *et al.* (2005) experiment, i.e. $\Theta n_i, n_j \geq 5$, $|\mathbf{N}|/|\mathbf{P}| = 50$, and $\Lambda pat_k \geq 5$. Note that we apply the $|\mathbf{N}|/|\mathbf{P}| = 50$ setting here for both antonymy and synonymy although we have no evidence suggesting that it is their natural class distribution.

As we can observe, our systems are comfortably better than the baseline classifiers. It is also encouraging that our reported results are better than that of Mohammad *et al.* (2008) (antonymy) and Turney (2008) (both antonymy and synonymy), although a direct result comparison is not reliable due to the differences in experimental setup. In all three relations, our systems perform reasonably well over both the baseline and benchmark results, demonstrating the generalisability of our method.

4.4 Experiments on Supervised Relation Learning

In addition to demonstrating our method’s generalisability, our other research objective in this chapter is to investigate the effect of different system parameters and corpora used in building our RE system. This section describes the impact of different experiment configurations on the accuracy of our method. Figures 4.3 and 4.4 show the systems’ accuracy in three coloured clusters, each representing a specific relation. Within each cluster, the figures show system accuracy over different system parameters and corpora settings. The f-scores reported in those figures are macro-averaged across the different fold of the 10-fold stratified cross validation.

4.4.1 Effects of Using Different System Parameters

In this section, we investigate the effect of three system parameters which control the instance sampling¹ (Section 4.1.4) and feature representation (Section 4.1.5) to RE accuracy:

¹We use the term “sampling” throughout this thesis in its statistical sense, that is selecting a population for our study, as compared to its other common use in statistical inference (e.g. Gibbs sampling) as commonly found in NLP applications

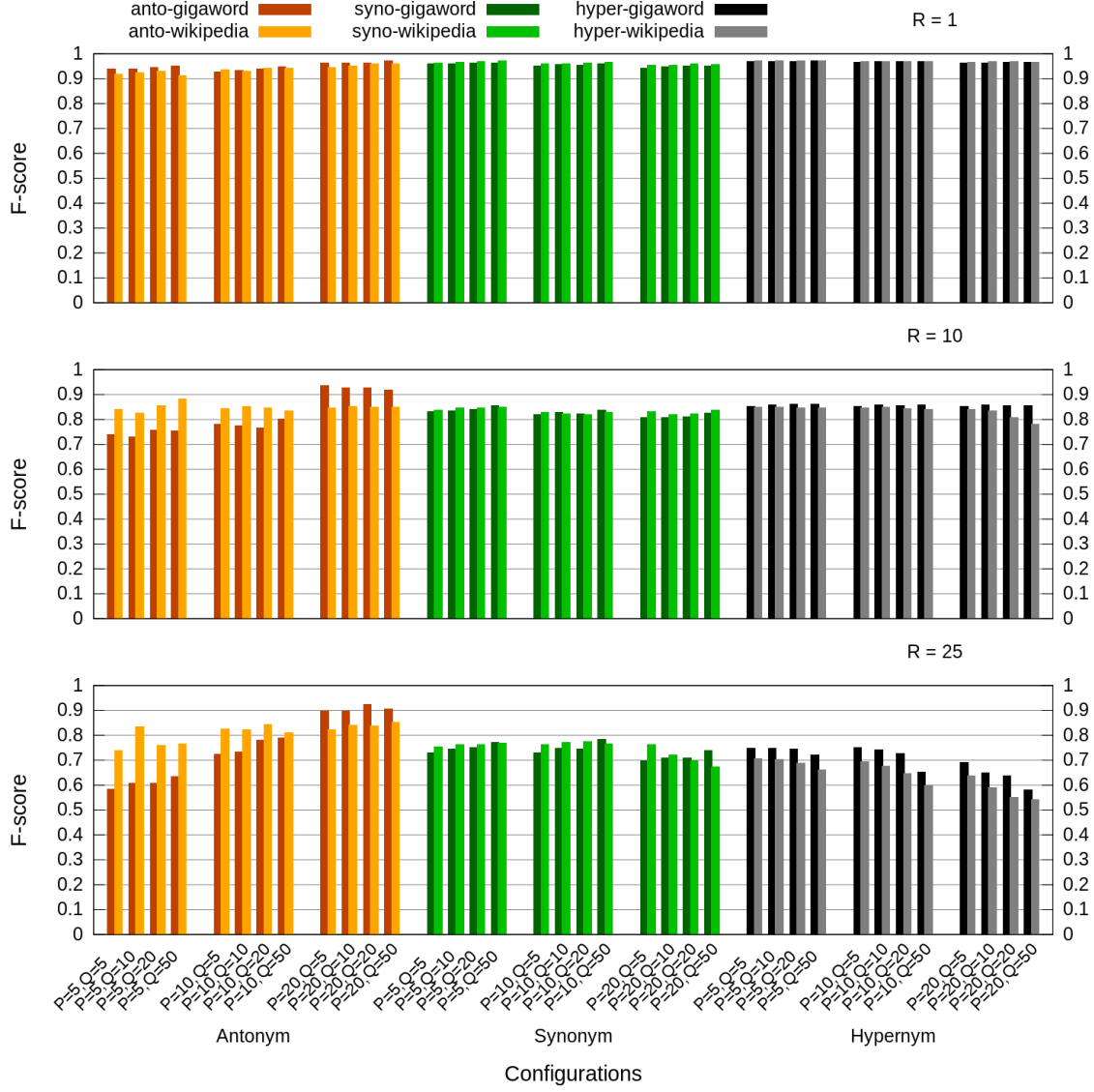
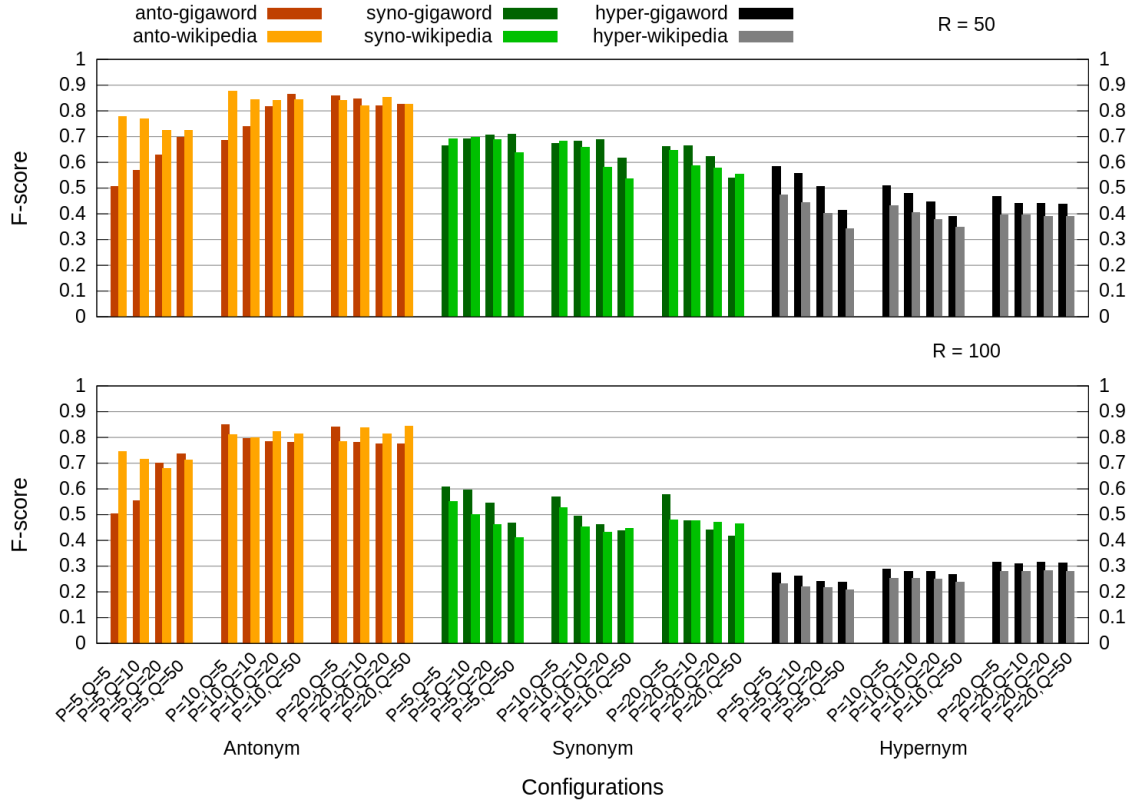


Figure 4.3: System accuracy on the three relations, with $R \in \{1, 10, 25\}$

1. $\Theta n_i, n_j \geq P$, where P is the minimum number of unique patterns a noun pair (n_i, n_j) must occur in to be included in the dataset. We experiment with $P \in \{5, 10, 20\}$.
2. $\Lambda pat_k \geq Q$, where Q is the minimum number of unique noun pairs a pattern pat_k must occur in to be considered a feature in the dataset. We experiment with $Q \in \{5, 10, 20, 50\}$.

Figure 4.4: System accuracy on the three relations, with $R \in \{50, 100\}$

3. $|\mathbf{N}|/|\mathbf{P}| = R$, where R is the ratio of the number of positive ($|\mathbf{P}|$) and negative instances ($|\mathbf{N}|$) in the dataset satisfying the requirement that $|\mathbf{N}| = R \times |\mathbf{P}|$. We experiment with $R \in \{1, 10, 25, 50, 100\}$.

When investigating the effect of a parameter, we compare the system accuracy over different values for that particular parameter, while fixing the values for the other two parameters.

We test three settings of P for the $\Theta n_i, n_j \geq P$ parameter in an attempt to ascertain whether noun pairs which occur across less patterns are harder to classify. To this end, we expect that the systems' accuracy to increase as the threshold value increases, as intuitively noun pairs which occur across more patterns are more informative, and hence easier to predict.

In practice, however, we tend to run out of positive instances for threshold values higher than 20 ($P > 20$), and risking evaluating over a small dataset which may not provide an accurate indication of the system performance. On the other hand, for lower values, we have the facility to classify more data and hence increase our coverage, but there is a risk of misclassifying more often due to the data being sparser.

Looking at the Figures 4.3 and 4.4, we observed that there is no clear pattern in the effect of $\Theta n_i, n_j \geq P$, where there are occasions where higher values of P result in better accuracy, and vice versa. The immediate conclusion of this observation does not conform with our hypothesis that noun pairs cooccurring across less patterns are harder to classify. However, we note that the evaluation procedure may run into problems with higher P values as the number of positive instances drops significantly (see Table 4.3). Therefore, it is even harder to conclude the effect of this parameter on RE, as the difference between the number of training instances (both positive and negative instances) may have affected the classifier training process.

Next, we experiment with various settings of Q for $\Lambda pat_k \geq Q$ parameter to investigate the effect of pattern frequency on RE performance. As the value of Q rises, the feature space is thinned out.

Again, we refer the reader to Figures 4.3 and 4.4. Similar to the $\Theta n_i, n_j \geq P$ parameter, there is no clear pattern in the effect of $\Lambda pat_k \geq Q$: higher values of Q result in better performance in some instances, and worse performance in others. Hypernymy proves to be an exception as the performance levels tend to be slightly lower as we increase the value of Q . This observation suggests that it is possible to run the systems over a reduced set of patterns while maintaining high accuracy.

The final parameter is the ratio of the negative to positive instances in our data ($|\mathbf{N}|/|\mathbf{P}| = R$), where we experiment with five different values of R . Naturally, we expect there to be an overall drop in system performance as we increase the value of R , as the negative instances progressively overshadow the positive instances.

Unlike the effects of changing the values of the other two parameters, we can observe in the system accuracy figures that the effect of changing the value of the ratio parameter is substantial. Comparing the three relations, hypernymy seems to be the most affected as we increase the R value. We suspect the main reason for this

is its substantially larger number of positive instances (as compared to the other two relations) which leads to negative instances swamping the data during training and evaluation. As a result, the systems' predictions are increasingly biased towards the negative instances for higher R values. We also note that the hypernymy systems' accuracy drops to near the baseline level when we set $R = 100$.

Superficially, these results suggest that we should keep the ratio as low as possible to optimise RE accuracy. However, it needs to be pointed out that a classification task is relatively easier when the number of instances for each class is similarly distributed in the data, as is the case in our experiment with low R values. On the other hand, the task is considerably harder when the class distribution is highly imbalanced, where instances from one class dominate the instances from the other (Tan *et al.* 2006).

As Snow *et al.* (2005) showed, it is unreasonable to expect that the class distribution between positive and negative pairs for any relation to be equal in real-world corpora. In this sense, the result comparisons over different R values do not tell us much about how well the method would perform over raw data. In follow-up experimentation (Chapter 5), we investigate the accuracy of our systems trained over different ratio values on a fixed set of randomly-sampled held-out data. In this way we hope to arrive at a more representative, standardised set of results which reflects the natural class distribution of our three relations.

4.4.2 Effects of Using Different Corpora

When analysing the effect of corpus on RE performance, we directly compare the f-score for the two corpora, noting that the datasets for each corpora are not the same. The datasets difference is due to the dataset formation process, which depends on the corpus and the system experiment settings. Table 4.3 shows the different number of positive instances available for both corpora at various thresholds of $\Theta n_i, n_j$, indicating the different dataset sizes (the number of negative instances are implicitly different as well, as they depend on the number of positive instances via the $|\mathbf{N}|/|\mathbf{P}|$ threshold). Due to the difference in datasets at various $\Theta n_i, n_j$ thresholds, the effect of corpus on the reported system accuracy has to be interpreted carefully.

From Figures 4.3 and 4.4, we observe that hypernymy systems trained on GIGAWORD are consistently slightly more accurate than their equivalent on WIKIPEDIA. Synonymy systems are similar to the hypernymy systems, although there are more settings which show that WIKIPEDIA-trained systems are better than GIGAWORD-trained systems. Antonymy systems show contradicting results, where WIKIPEDIA-trained systems are more accurate than GIGAWORD-trained systems in some settings, but are less accurate in others.

It is interesting to note that despite WIKIPEDIA being less than half the size of GIGAWORD, systems trained on WIKIPEDIA are almost as accurate to those trained on GIGAWORD. Furthermore, the number of positive instances gathered from WIKIPEDIA is always more than for GIGAWORD (see Table 4.3). This is most likely due to WIKIPEDIA being an encyclopedia, and thus being richer in diverse usages, and having greater domain coverage than GIGAWORD (which is comprised of newswire articles).

As direct comparison is not possible (due to the difference in the datasets), we cannot satisfactorily conclude that one corpus is a better training resource than the other. We revisit this issue later in Chapter 5, by evaluating the system performances over a fixed set of held-out data sampled across noun pairs occurring in both GIGAWORD and WIKIPEDIA.

4.4.3 Different Levels of Relations Difficulty

Our experiments involve three different semantic relations: antonymy, synonymy, and hypernymy. Using the results from Figures 4.3 and 4.4, we are interested in observing whether it is possible to conclude that a given relation is relatively more difficult to learn compared to the others.

Based on the reported results, there is little difference between the three relations over the various experiment settings and corpora when $R = 1$ or $R = 10$. However, the gaps for the three relations are clearly noticeable for $R = 25$, and are their most extreme level when $R = 100$. In those settings, it is clear that hypernymy systems are the least accurate of the three relations, while antonymy systems are the best.

These results suggest that hypernymy is the most difficult relation to learn of

the three. As RE task concentrates mostly on identifying sets of patterns that unambiguously define a relation, it can be inferred that based on these results, the relation-identifying patterns for hypernymy are the most difficult to identify. However, this conclusion contradicts previous findings where there exists a set of widely accepted high-precision patterns for hypernymy, but similar pattern sets do not exist for antonymy and synonymy, suggesting that antonymy and synonymy are harder to learn than hypernymy (see Section 4.3).

We suspect that this observation can be explained by the sampling procedures performed in the current experiment. The number of positive instances for antonymy and synonymy are comparatively smaller than hypernymy instances for any experiment settings (see Table 4.3). This leads to greater data imbalance for hypernymy as we increase the R value in the experiments, precipitating a commensurate performance loss, as evidenced by lower system performance for higher R values.

Recall that the negative instances are sampled by picking the top- X frequently appearing noun pairs from corpora. As we increase the ratio, the sampling procedure chooses instances that appear less frequently in the corpora. In addition to increasing the greater data imbalance, these low-frequency negative pairs are also less informative than the previously sampled negative pairs with higher frequency, leading to greater difficulty in relation learning.

As we have some reservations regarding the sampling procedure in this experiment, we cannot yet conclude that hypernymy is harder to learn compare to synonymy and antonymy. We revisit this issue again in Chapter 5 where we attempt to address the issue with the sampling and evaluation procedures from our experiments in this chapter.

4.5 Summary

In this chapter, we described our RE system based on Snow *et al.* (2005)’s methodology. In building our system, we closely followed the details of their implementation. However, we find that there are some simplifying assumptions in their methodology, particularly with the relation labelling process which we find to be unrealistic.

We propose a modification to their heuristic, which addresses their assumption of non-polysemous nouns for RE.

To measure our implementation success, we compared the results from our system to Snow *et al.* (2005)’s system. Although a direct comparison is not feasible due to the differences such as the relation labelling process, the corpora and machine learner, we applied the same parameters settings to our system as the ones used in original experiment. Under this condition, we conclude that our implementation is capable of replicating the success of the original experiment in performing hypernymy RE.

Using this implementation, we investigated our RE method’s generalisability by applying it to two additional relations (antonymy and synonymy). Our results demonstrate that the system is reasonably accurate over the three relations, and competitive against various baseline and benchmark systems.

Next, we experimented with a great number of system parameter settings to investigate their individual impact on RE. This proved to be challenging as a direct comparison between different settings is not always possible. For this reason, we are unable to satisfactorily conclude whether one corpus is better than the other as a training resource for RE. However, we are successful in identifying that it is possible to train RE classifiers over a reduced set of patterns while maintaining high performance levels. We also observed that the ratio between positive to negative instances affects RE accuracy the most.

While the immediate results suggest that we are successful in our RE task, there are two main challenges that need to be addressed. The first challenge is the heuristic we employ in the relation labelling process. Although we consider it to be an improvement over the original methodology, the current heuristic remains somewhat artificial in its assumptions, which simplifies the extraction task. The second challenge is the lack of facility to produce a fixed set of held-out data. This is the reason we are unable to directly compare the effect of several parameters on RE. Further, the fixed held-out data may solve the problem of determining the correct ratio of positive and negative instances for any given relation.

We note that these challenges directly relate to the three aspects prominent in the sampling of noun pairs in our dataset: determining the class label of pairings,

which pairings to include in the dataset, and the class distribution in the dataset. As we have shown, our systems' accuracy fluctuates greatly when applied to different datasets produced by different sampling procedures. This suggests that the current methodology is inadequate in addressing the noun pair sampling for our RE task, which we consider very important in the feasibility of our task and the interpretability of its results.

In Chapter 5, we propose several modifications to the current methodology in order to remove the artificialness of the current RE task. We then use the result of these modifications to further investigate the impact of our various system parameters on RE, which is currently not viable.

Chapter 5

Effects of Sampling

In Chapter 4, we built our relation extraction system based on the Snow *et al.* (2005) methodology. Their methodology was only evaluated over a particular experimental setting: using a single corpus, a single relation (hypernymy), and fixed settings of three system parameters. Using our system, which was built based on this methodology, we performed various experiments involving two corpora, two additional relations, and various system parameters combinations in order to investigate their effect on relation extraction.

When performing our experiments, we noticed that Snow *et al.* (2005)’s methodology was unable to provide a fair and sensible evaluation across different experimental settings. This is mainly due to its inability to provide a common dataset to be evaluated across various settings, especially across multiple corpora. As a result, we cannot draw reasonable conclusions on the effect of different experimental settings on RE.

Snow *et al.* (2005) suggested in their experiment that the distribution for instances with and without the hypernymy relation is 1:50, based on their hand annotation of sample data. However, it is unreasonable to assume that the two additional relations in our experiments (antonymy and synonymy) will have the same distribution. While it is possible to derive the instance distributions for antonymy and synonymy by performing similar data hand-labelling for the two relations, such annotation is resource expensive.

Both these challenges above relate to the issue of instance sampling for our data,

i.e. determining which instances should be in the dataset to satisfy the given constraints. The beginning of this chapter (Section 5.1) focuses on proposing various sampling procedures aimed at adapting the Snow *et al.* (2005) evaluation methodology to compare the effect of various experimental settings on RE. In addition to this, we aim to come up with a realistic sampling procedure that can be applied seamlessly to open and large-scale data without much adaptation effort.

Section 5.2 describes the design and experimentation of our RE task, which we argue to be a more realistic task than the Snow *et al.* (2005) task. We analyse Snow *et al.* (2005)’s RE task setup, concentrating on its simplifying assumptions which are not reflective of the real-world situation. These assumptions include the various system parameters required for training and evaluation processes, and the artificial exclusion of noun pairs during relation labelling. Next, we use the information from our sampling experiments to produce training and test partition data for evaluation purposes.

We perform experiments on our proposed RE task in Section 5.3. Contrary to the promising results shown in Chapter 4, our system performs poorly on all three relations. We analyse its errors, and identify factors that may have contributed to its poor accuracy.

Finally, the low accuracy of our system leads us to investigate the competitiveness of our methodology in Section 5.4. We examine our methodology by applying it to a hypernymy extraction task proposed by Navigli and Velardi (2010). After a few changes to adapt our system to the task, we find that our methodology is very competitive. We conclude by discussing the differences between our task and the Navigli and Velardi (2010) task.

5.1 Experiments on Different Sampling Procedures

Having built our RE system based on Snow *et al.* (2005)’s methodology, one of our research objectives is to investigate the effect of various experimental settings. Snow *et al.* (2005)’s methodology, however, was restricted and evaluated in the context of a particular experimental setting, while our experiments involve multiple experimental

settings with combinations of different corpora, relations, and system parameters. For each experimental setting, the methodology produces different datasets, which makes it difficult to evaluate the impact of an experimental setting to RE because the system is trained and tested on different datasets in each experiment.

One way to solve this problem is to have a common dataset for training and testing purposes across different experimental settings. In this scenario, we can accurately measure the impact of each setting using the same dataset, and draw reasonable conclusions over them. Our objective in this section is to propose several sampling procedures aimed at redressing the methodology’s inability to produce a common dataset under different experimental settings.

5.1.1 Intersecting Corpora Instances

One of our research objectives is to investigate the impact of different corpora on RE, by involving two different corpora: GIGAWORD and WIKIPEDIA. However, as described in Chapter 4, the sampling procedure used in that experiment produced two distinct datasets, one for each corpus. These two datasets are not identical to each other, as the noun pairs in each dataset are sampled based on their occurrences in the corresponding corpora (due to factors such as a noun pair appearing in one corpus and not the other). We can observe the difference between instances sampled from the two corpora in Table 4.3.

After training and testing the system using these two distinct datasets, we reported the system’s f-score on each corpus. The f-score on each corpus, however, can only indicate how well the system performed using the corpus of choice, and cannot reasonably indicate that one corpus is better than the other due to their differences.

To address this issue, we propose to perform a similar experiment to that from Chapter 4, but sampling only from instances that appear in both GIGAWORD and WIKIPEDIA. This enables a reasonable comparison between a system trained exclusively using features from one corpus to the other.

We apply this sampling procedure in our experiment by taking the intersection of instances for GIGAWORD and WIKIPEDIA for all settings of P ($P \in \{5, 10, 20\}$; recall

$\Theta_{n_i, n_j} \geq P$		INTERSECTION		
		ANTO	SYNO	HYPER
$P = 5$	+	74	556	5,482
	−	1,488,610	1,485,228	1,464,484
$P = 10$	+	59	373	3,444
	−	768,930	766,697	753,263
$P = 20$	+	50	231	1,989
	−	363,652	362,307	354,364

Table 5.1: Comparisons of class distribution for instances present in both GIGAWORD and WIKIPEDIA for various thresholds P of Θ_{n_i, n_j}

that P is the minimum number of unique patterns a noun pair must occur in — see Section 4.1.4 for details). Table 5.1 lists the number of such instances, while Table 5.2 lists the fraction of such instances to the total number of instances, in each corpus (as listed in Table 4.3). From here on, we refer to this set of instances as INTERSECTION.

By sampling over instances that appear in both corpora, we lose some of the instances that appear exclusively in one corpus. By nature, these instances are very difficult to evaluate, as such instances do not have any features for systems trained over a corpus in which they do not occur. As our experiments involve cross-corpus comparison, we consider the exclusion of such instances from the dataset both necessary and fair, although it is preferable to not have excluded a large proportion of such instances from the dataset and ended up with a tiny dataset. However, as observed in Table 5.2, the presence of a considerable number of positive instances in INTERSECTION — which contains at least 65% of the individual corpus instances — is a positive finding for our proposed sampling procedure.

As with the previous experiments in Chapter 4, we propose to include all the positive instances in INTERSECTION in the final dataset. We face the same problem with those experiments, where the number of negative instances far exceeds that of posi-

$\Theta_{n_i, n_j} \geq P$		INTERSECTION		
		ANTO	SYNO	HYPER
$P = 5$	+	93.6% G, 85.1% W	83.2% G, 68.2% W	78.9% G, 66.7% W
	–	52.9% G, 63.5% W	54.3% G, 65.2% W	54.0% G, 65.1% W
$P = 10$	+	92.2% G, 88.1% W	82.3% G, 70.4% W	77.0% G, 65.7% W
	–	50.9% G, 65.4% W	50.9% G, 65.3% W	50.6% G, 65.3% W
$P = 20$	+	94.3% G, 83.3% W	79.7% G, 68.1% W	73.9% G, 65.4% W
	–	46.2% G, 64.6% W	46.2% G, 64.6% W	45.8% G, 64.5% W

Table 5.2: Fractions of each corpus instances present in both GIGAWORD and WIKIPEDIA for various thresholds P of Θ_{n_i, n_j}

tive instances in INTERSECTION, and therefore we have to determine which negative instances to include in the final dataset. For the current experiment, we adopt a similar approach in choosing the top- $|\mathbf{N}|$ most frequently appearing instances, with $|\mathbf{N}|$ determined by the value of R (the ratio of negative instances to positive instances — see Section 4.1.4 for detail). However, unlike our experiments in Chapter 4, where the sampling procedure produced two distinct datasets based on the individual frequency in each corpus, the current sampling procedure will only produce one dataset common to the two corpora. Therefore, the top- $|\mathbf{N}|$ most frequently appearing instances should reflect the instances’ aggregate frequency in both corpora.

In order to do this, we rank the negative instances for each corpus individually, from the most frequent to the least frequent. Next, for each instance in INTERSECTION, we aggregate its ranking by summing up its corresponding rank in both GIGAWORD and WIKIPEDIA. We then sort the aggregate rank (from lowest rank to highest rank) and pick the top- $|\mathbf{N}|$ instances as our negative instances. Table 5.3 provides some examples from this process.

We perform this sampling procedure for the three relations, over the different values of the three system parameters of P , Q , and R , and evaluate the system using

Noun Pair	GIGAWORD	WIKIPEDIA	INTERSECTION
	Frequency (Rank)	Frequency (Rank)	Aggregate Rank
(<i>medal, gold</i>)	42,724 (#101)	18,716 (#96)	#197
(<i>rate, interest</i>)	169,925 (#3)	4,089 (#1,182)	#1,185
(<i>years, age</i>)	3,701 (#6,031)	86,555 (#4)	#6,035

Table 5.3: Examples of noun pairs occurring in the data, sorted by their aggregate rank

stratified 10-fold cross validation. As the objective of this experiment is to compare the quality of the two corpora as a learning source, we run the cross validation process twice, using the features exclusively extracted from each corpus in each run. We present the evaluation results in Figures 5.1 and 5.2, with the f-scores macro-averaged across the different folds.

We compare the f-score between the systems trained on different corpora using a paired t-test at confidence level of 95%. For antonymy, there is a statistically significant f-score improvement for WIKIPEDIA-trained systems compared to the GIGAWORD-trained systems across the different system parameters settings (P , Q , and R). In contrast, for synonymy and hypernymy, the paired t-test shows that there is a statistically significant improvement for GIGAWORD-trained systems compared to WIKIPEDIA-trained systems. As such, it is not surprising that over the three relations in this experiment, there is no statistically significant difference between the two systems trained on different corpora. Therefore, from this experiment, it is not possible to decisively conclude that one corpus is better than the other for every relation and system parameterisation.

We do, however, consider the current sampling procedure to be an improvement over the procedure used in Chapter 4, as it allows fairer and more reasonable comparisons over different experimental settings, especially over different corpora. Despite this, it is still unable to provide a common dataset over other experimental settings. This is evidenced by the different datasets produced for different relations and system

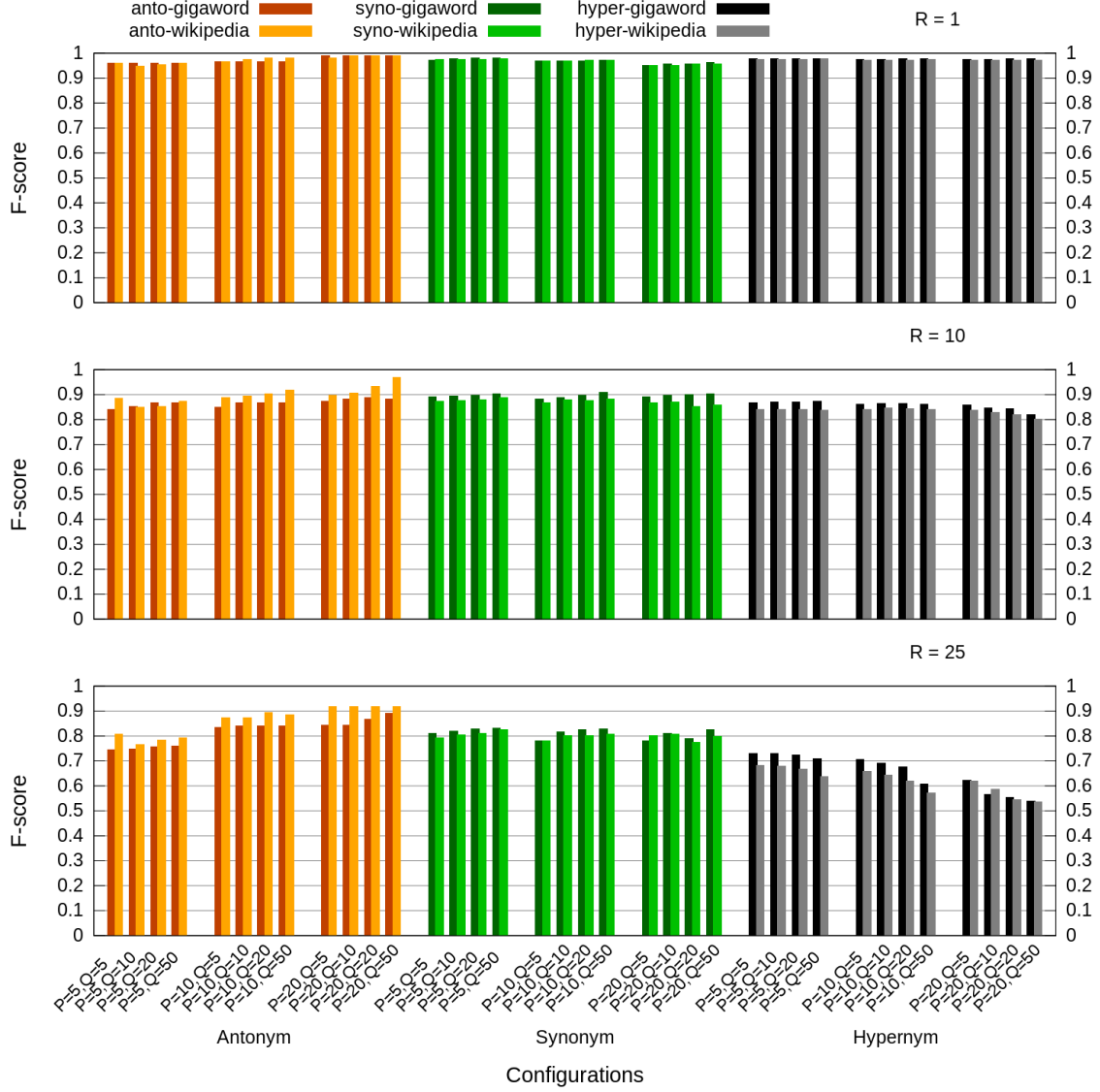


Figure 5.1: System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{1, 10, 25\}$

parameters in the experiments. We aim to improve on this in the next section.

5.1.2 One Million Noun Pairs Experiment

Based on hand annotation, Snow *et al.* (2005) suggested that the true distribution between positive and negative instances for hypernymy is 1:50. However, as we

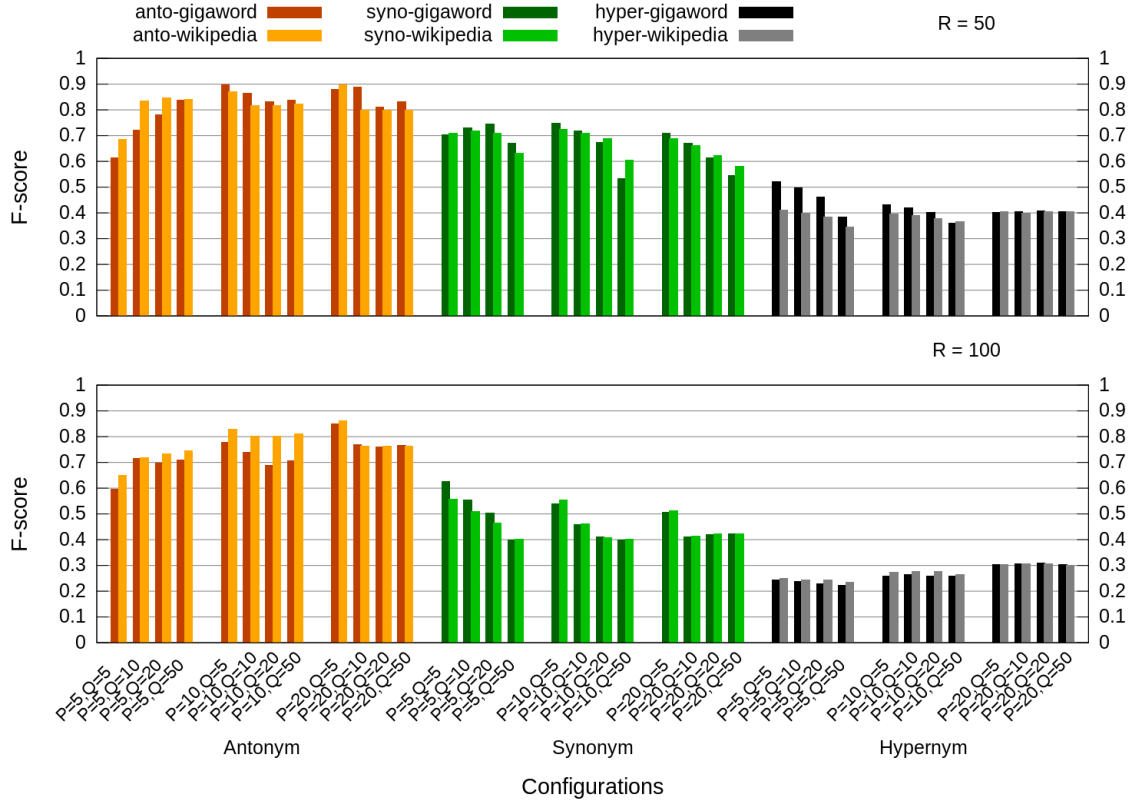


Figure 5.2: System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{50, 100\}$

pointed out in Chapter 4, it is unreasonable to assume that the same distribution holds true for antonymy and synonymy. Following this, we abandon the idea of using the same R values for all the relations during the sampling process, as it does not reflect the real proportion of positive and negative instances of the three relations in the data.

There are two methods to address this problem. The first method is to perform hand annotation a la Snow *et al.* (2005) for the antonym and synonym instances. Assuming the distribution of positive and negative instances for antonymy and synonymy is different to that for hypernymy, we can then set different R values for each relation in our experiment based on the results of the hand annotation. However, this procedure is resource expensive, especially if more relations are involved.

Our alternative method is to produce a common dataset for all the relations by randomly sampling instances from the data, and assuming that the sampled instances encode the natural distribution of the positive and negative instances for each of the relations. In other words, the dataset itself automatically defines the R value for each relation. This is in stark contrast to the first method, where the sampling procedure produces different datasets for each relation, whose composition of positive and negative instances is dictated by the R value. Our method has the dual benefit of avoiding the resource-expensive process of discovering separate R values for each relation, and provides a common dataset for all relations.

We implement this method by extending the sampling procedure of intersecting instances between GIGAWORD and WIKIPEDIA from Section 5.1.1. Instead of performing sampling based on the different relations and R values, we randomly pick one million noun pairs from the INTERSECTION dataset. There is no specific reason why we choose to sample one million noun pairs, other than the hypothesis that an instance set of that size would provide a close estimate of the natural distribution of positive and negative instances for all three relations, and indeed, any other relations we may choose to study. For the current experiment, to avoid choosing low-frequency noun pairs with a small number of features and the associated data-sparsity problem, we require our noun pairs from the INTERSECTION dataset to have appeared with at least 5 different patterns in the data ($P = 5$). Note that one million noun pairs is roughly 70% of the INTERSECTION dataset for $P = 5$ in Table 5.1.

Prior to randomly selecting the final one million noun pair set, we filter out instances that cannot be classified due to missing information or relations not being between their first WORDNET senses. In other words, we only consider noun pairs that are successfully classified as either positive or negative instances by our labelling procedure (see Section 4.1.3). We refer to this final one million noun pair set as 1-MIL dataset from hereon. The breakdown of positive and negative instances for the three relations in 1-MIL is listed in Table 5.4.

A special property of this sampling procedure is that it produces a fixed dataset regardless of our experimental settings: corpora, relations, and system parameters. Apart from fixing the system parameter $P = 5$, the other experimental settings do not

$\Theta n_i, n_j \geq P$		1-MIL		
		ANTO	SYNO	HYPER
$P = 5$	+	48	337	3,711
	−	999,952	999,663	996,289
	$ \mathbf{N} / \mathbf{P} $	20,832.33	2,966.36	268.47

Table 5.4: Comparison of the instances class distributions for all relations in 1-MIL

play any part in our sampling procedure, unlike the previous sampling approaches where each experimental setting played an important role in determining the final instance set.

As we can observe from Table 5.4, the proportion of negative to positive instances set far exceeds the ratio used in our earlier experiments in Chapter 4 and Section 5.1.1, especially for antonymy and synonymy. The current proportions, however, are closer to the ratios observed for all the noun pairs in each corpus (as listed in Table 4.3). This raises the question of whether we have been undersampling the negative instances for our data in earlier experiments.

We believe that the current sampling procedure is more reflective of the corpora than the previous sampling procedures. However, it still remains to be seen whether systems trained on previous sampled data can perform well on the 1-MIL dataset. In order to explore this, we choose to apply models from Section 5.1.1 to the 1-MIL dataset.

We first produce the feature vectors for our 1-MIL dataset. As models from Section 5.1.1 are trained using four Q values ($Q \in \{5, 10, 20, 50\}$; recall that Q is the minimum number of unique noun pairs a pattern must occur in to be considered as features by the machine learner — see Section 4.1.4 for details), we produced the equivalent feature vectors for the 1-MIL dataset using these Q values. Note that the Q value only affects the features — not the instances — used in the experiments.

Next, we pick the most accurate models out of the 10 folds of each experimental setting trained on our INTERSECTION data to classify the 1-MIL dataset. We only

apply models trained on a given Q value to classify the 1-MIL dataset with features from the equivalent Q value.

One caveat of this experiment is that the models from the INTERSECTION data are most likely to have been trained by some instances present in the 1-MIL dataset. This is due to how the previous experiment was conducted, where the training involves the use of all positive instances present in the data in stratified 10-fold cross validation. As a result, there may be overlap in both positive and negative instances used in the previous experiments with the ones sampled in the 1-MIL dataset. Because of possible “data peeking”, the reported accuracy of the INTERSECTION models to the 1-MIL dataset may be overestimated.

To our surprise, the accuracy of these models is very disappointing. Antonymy and synonymy models in particular, are very inaccurate: the majority of them fail to achieve an f-score of 0.1. Hypernymy models trained on lower R values ($R \in \{5, 10, 25\}$) undergo a similar drop in f-score, while only models trained on higher R values show reasonable accuracy. The most accurate models are hypernymy models trained using $R = 100$ (regardless of the other experimental settings), where their f-scores are in the range of 0.444 to 0.795.¹

These results are both disappointing and puzzling. Antonymy and synonymy models are perhaps the biggest disappointment, as these models have consistently scored very well in experiments in Section 5.1.1 (their lowest f-scores are slightly below the 0.4 mark). We also observe that in an equivalent experimental setting, hypernymy models outscore both antonymy and synonymy models, an event which has not happened prior to this experiment. Another novel finding is that models trained with higher R values are more accurate than models trained on lower ratios, as illustrated by the hypernymy models trained on $R = 100$ being the most accurate models on the 1-MIL dataset. Similar events can be observed in antonymy and synonymy models trained on higher R values, although the improvement is less obvious due to the low f-scores of models trained on lower R values.

We observe that the overwhelming number of false positives is the main cause of

¹We choose to summarise the results of this experiment as we consider it to be a preliminary study and also the large size of the full results.

the low system accuracy in this experiment. This indicates that the models, especially those trained on lower R values, are overly confident in predicting positive instances. While high recall scores are obtained, it comes at a great cost of very low precision, hence bringing down the individual f-scores. This is in line with our hypothesis that we have been undersampling negative instances in our earlier experiments.

Related to this, strong f-scores from models trained on higher R values perhaps suggest that a system trained on a dataset with the proportion of positive and negative instances closer to that found in 1-MIL dataset will perform better. This is reflected in the higher accuracy of hypernymy models trained on $R = 100$, where the R value is quite close to the actual ratio of negative to positive instances in the 1-MIL dataset (268.47 — refer to Table 5.4). This explains the poor accuracy of the synonymy and antonymy models, where the maximum R value of 100 used in previous experiments in Chapter 4 and Section 5.1.1 was 30 to 200 times less than the actual class distribution for both synonymy and antonymy instances in the 1-MIL dataset, respectively.

Disconnect between INTERSECTION and 1-MIL

As mentioned earlier, we suspect that undersampling of negative instances is one of the main contributors to the poor accuracy of the models trained on the INTERSECTION dataset when applied to the 1-MIL dataset. To investigate this further, we look at the instances used to train these models (instances from INTERSECTION), and compare them with the instances found in the 1-MIL dataset.

One major difference between the two sampling procedures is how the negative instances are sampled. Negative instances from the INTERSECTION dataset (and also the experiments in Chapter 4) are sampled based on their frequency in the corpora. Models are trained by using the top- $|\mathbf{N}|$ most negative instances, where $|\mathbf{N}|$ is determined by the system parameter R and the number of positive instances used.

In contrast, the 1-MIL dataset is sampled based on the instances from the INTERSECTION dataset that have appeared with at least 5 different patterns in the data ($P = 5$). These instances (both positive and negative) are randomly sampled from

the data without looking at their frequency, with slightly less than 1,500,000 noun pairs available to be sampled (see Table 5.1).

Because of this difference, we suspect that the negative instances chosen to train the models in our earlier experiments in Section 5.1.1 (and also in Chapter 4) are biased towards the more frequently appearing pairs in the data. However, the negative instances sampled in the INTERSECTION dataset, we suspect, do not exhibit the same bias, as they are randomly sampled from the data regardless of their corpus frequency.

To investigate this, we use the information on the aggregate frequency rank for each noun pair sampled in both INTERSECTION and 1-MIL datasets, which is the sum of the frequency rank of the noun pair in GIGAWORD and WIKIPEDIA, as an indication of instances' corpus frequencies (see Section 5.1.1 and Table 5.3, for detail). We then sort all of these noun pairs based on their aggregate frequency rank from the lowest to the highest. Next, we can assign an *aggregate sort index* to the sorted instances, so that the instance with the lowest aggregate frequency rank is assigned an aggregate sort index of 1. In other words, instances with lower aggregate sort index appear more frequently in the data than the instances with higher aggregate sort index.

Following this, we collect the aggregate sort index for both the instances sampled in INTERSECTION and 1-MIL datasets. However, recall that the INTERSECTION dataset is produced for various system parameters. To simplify the comparison between the INTERSECTION and 1-MIL datasets, we use the INTERSECTION dataset for the hypernymy relation, with $P = 5$ (the same P setting used for producing the 1-MIL dataset), on the various R settings.

Figure 5.3 presents the aggregate sort index distribution for the INTERSECTION and 1-MIL datasets at an index interval of 100,000. It shows the biased distribution of the INTERSECTION datasets towards the lower aggregate sort indices, meaning that the datasets consists more high-frequency instances than low-frequency instances. For example, more than 90% of the $R = 10$ INTERSECTION dataset can be found in the lowest interval bucket. This situation is expected due to the procedure used in sampling them, preferring frequently appearing negative instances to be included in the data. In contrast, the index distribution for the 1-MIL dataset shows a uniform

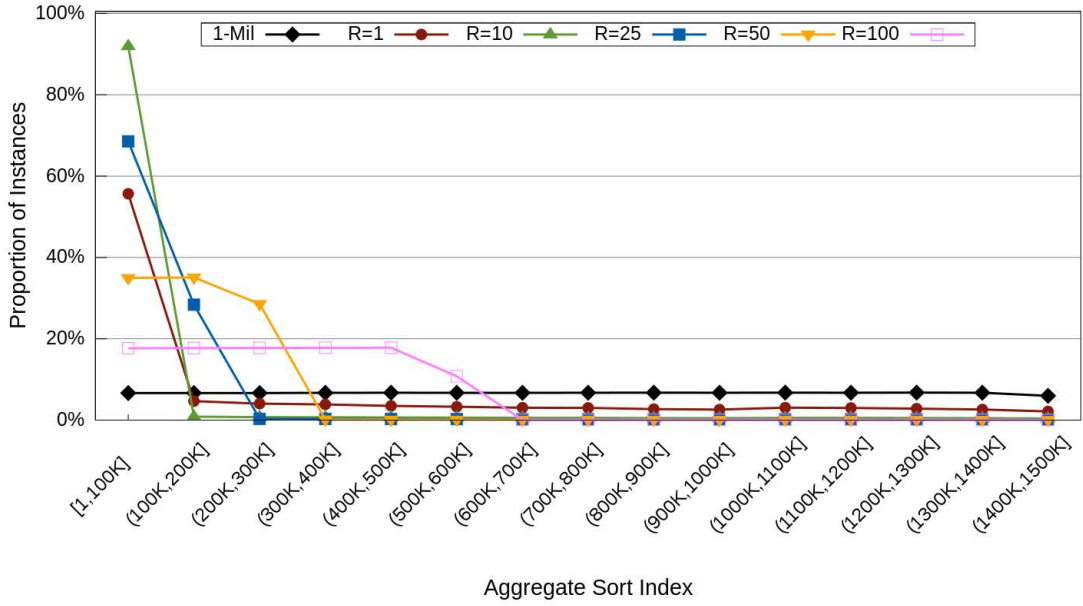


Figure 5.3: Aggregate sort index distribution of 1-MIL instances and hypernymy INTERSECTION instances on various R settings

distribution, implying that the datasets contain an equal distribution of instances based on frequency.

Instances with high aggregate sort indices (i.e. low frequency instances) are consequently underrepresented when training models using the INTERSECTION datasets. Also, although we do not show the distribution for antonymy and synonymy instances, the sampling bias is likely to be more severe for these two relations as they have less instances to train with compared to hypernymy instances.

Because of this sampling bias, the INTERSECTION models only fit very well with the small set of frequently appearing noun pairs in the 1-MIL dataset. These models were not optimised to classify the majority of the low-frequency instances, which have less features and are arguably harder to classify. This may explain the poor accuracy of our INTERSECTION models on the 1-MIL dataset.

Superior results for models trained on higher R values (especially on hypernymy) can also be linked to the more uniformly distributed instances as sampled from the data. As the R value increases and the procedure includes more low-frequency in-

stances in the data, the resulting models are arguably less biased towards frequently appearing instances and more accommodating of lower-frequency instances, leading to improved accuracy. This partly confirms our original hypothesis that undersampling of negative instances is the main reason for the poor accuracy of the INTERSECTION models on the 1-MIL dataset.

Training Using Randomly Picked Negative Instances

The experiment in the previous section demonstrated the disconnect between the aggregate sort index distribution between instances in the 1-MIL and INTERSECTION datasets, due to the major difference in sampling. A straightforward solution to address this problem is to sample the instances (especially the negative instances) in a similar manner to instances in the 1-MIL dataset. This section investigates whether randomly sampling negative instances for the INTERSECTION dataset experiments can produce more accurate models to classify the 1-MIL dataset.

After applying this modification to our sampling approach, we analyse the aggregate sort index distribution of hypernymy instances with $P = 5$ over different R values ($R \in \{1, 10, 25, 50, 100\}$). As expected, the distribution of these instances' aggregate sort index is more uniform and very similar to the distribution seen in the 1-MIL dataset (see Figure 5.4).

Using the modified sampling procedure, we run 10-fold cross validation to the INTERSECTION dataset. We present the macro-averaged f-scores over the 10-folds on the various experimental settings in Figures 5.5 and 5.6.

Overall, the f-scores show a descending trend of system accuracy for larger R values, similar to our observations in Section 5.1.1. However, paired t-tests at a confidence level of 95% show that there is a statistically significant f-score improvement for WIKIPEDIA-trained systems compared to GIGAWORD-trained systems for every relation and across different system parameter settings. This is in contrast to our previous observations where GIGAWORD-trained systems are better for synonymy and hypernymy.

Next, in order to investigate the impact of sampling procedures to our RE ac-

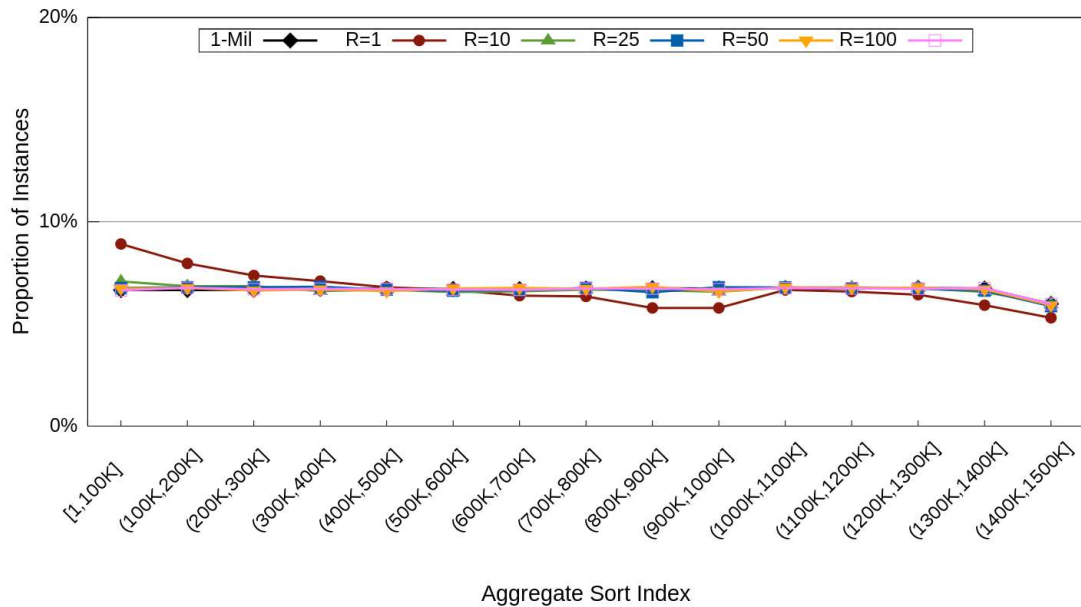


Figure 5.4: Aggregate sort index distribution on hypernymy INTERSECTION instances on various R settings whose negative instances are randomly sampled from data

curacy, we do a side-by-side f-score comparison of the current models (shown in Figures 5.5 and 5.6), to the ones produced in Section 5.1.1 (shown in Figures 5.1 and 5.2). Recall that the only difference between the current experiment and the experiments in Section 5.1.1 is the negative instances in the data (due to the different sampling procedures).

In general, the f-scores reported for models trained on the top- $|\mathbf{N}|$ frequently appearing negative instances are higher compared to the equivalent models trained on randomly picked negative instances. It is also interesting that synonymy is the worst out of the three relations in the current experiment, where previously, hypernymy was the relation that reported the worst results. These contradicting results between the two experiments suggest that the sampling procedure has a huge influence on an experiment’s outcome.

The above comparison between the two sampling procedures is based on the system’s accuracy on the sets of instances that the sampling procedure individually produced. However, this comparison does not indicate how accurate the models are

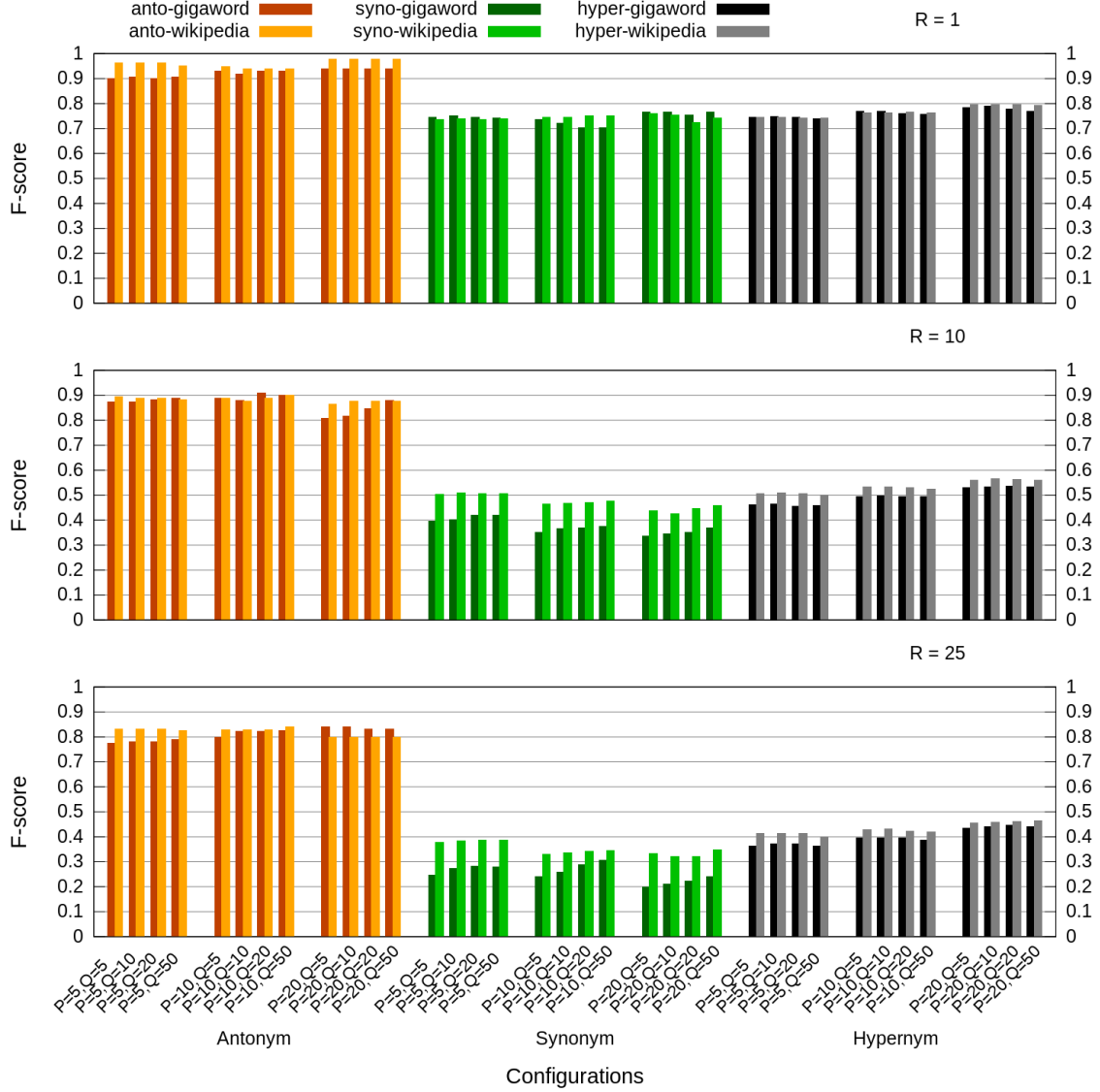


Figure 5.5: System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{1, 10, 25\}$. The negative instances are randomly sampled from data

over the 1-MIL dataset. In order to do this, for both of the sampling procedures, we take the most accurate models out of the 10 folds of each experiment setting, as evaluated within the instances that they produced. We then use these models to classify the 1-MIL test dataset. As with our experiment from previous section, we did not isolate the INTERSECTION and 1-MIL datasets, and subsequently, there may be

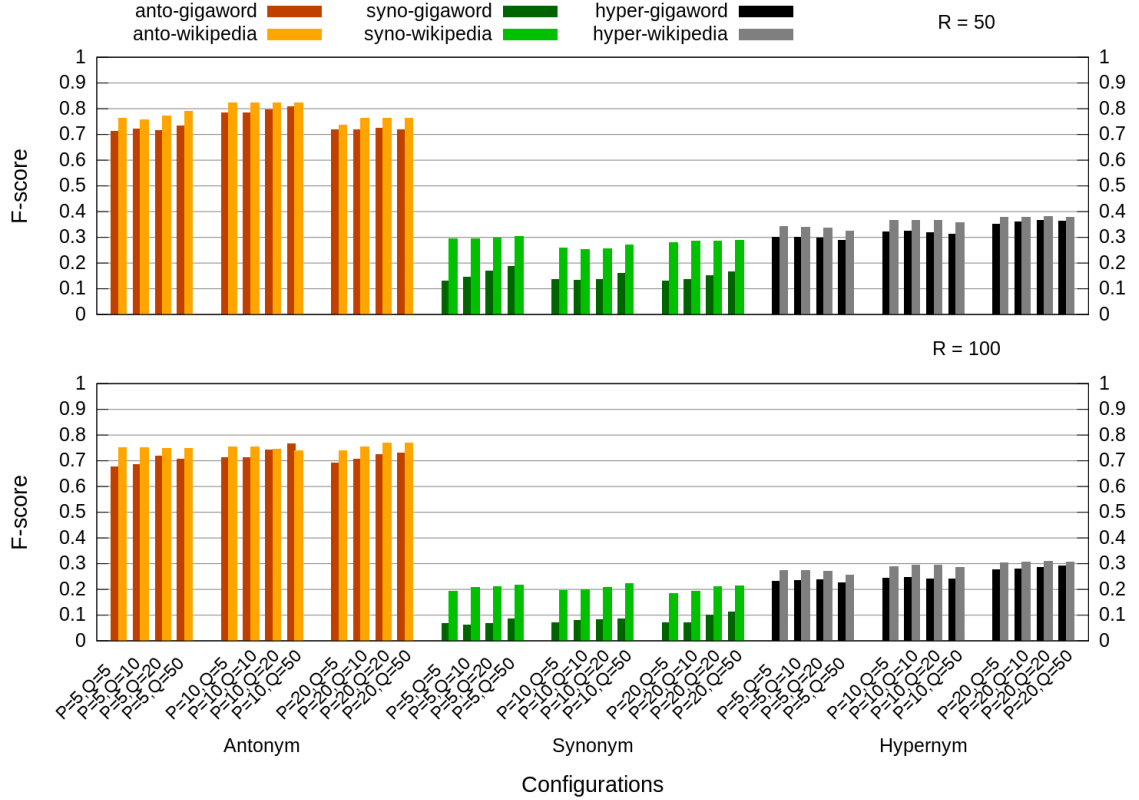


Figure 5.6: System accuracy on the three relations over the INTERSECTION dataset, with $R \in \{50, 100\}$. The negative instances are randomly sampled from data

overlaps between instances from the INTERSECTION dataset used to train the models and the instances in the test 1-MIL dataset. As a result, the accuracy reported from this experiment may be slightly overestimated.

Figures 5.7 and 5.8 show the comparison of WIKIPEDIA models trained on these two different sampling procedures on the 1-MIL dataset.² Note that the comparison inside each cluster (representing a relation) is between the two sampling procedures (as opposed to between the two corpora in previous figures). We can observe that our $R = 1$ models are not reliable, as none of the models from either sampling procedure reaches an f-score of 0.1. However, as the R values increase, the models' f-scores trained on randomly selected negative instances (**-Rand.neg* models), show a much

²We do not show the result for GIGAWORD models as we observe a similar trend to WIKIPEDIA models.

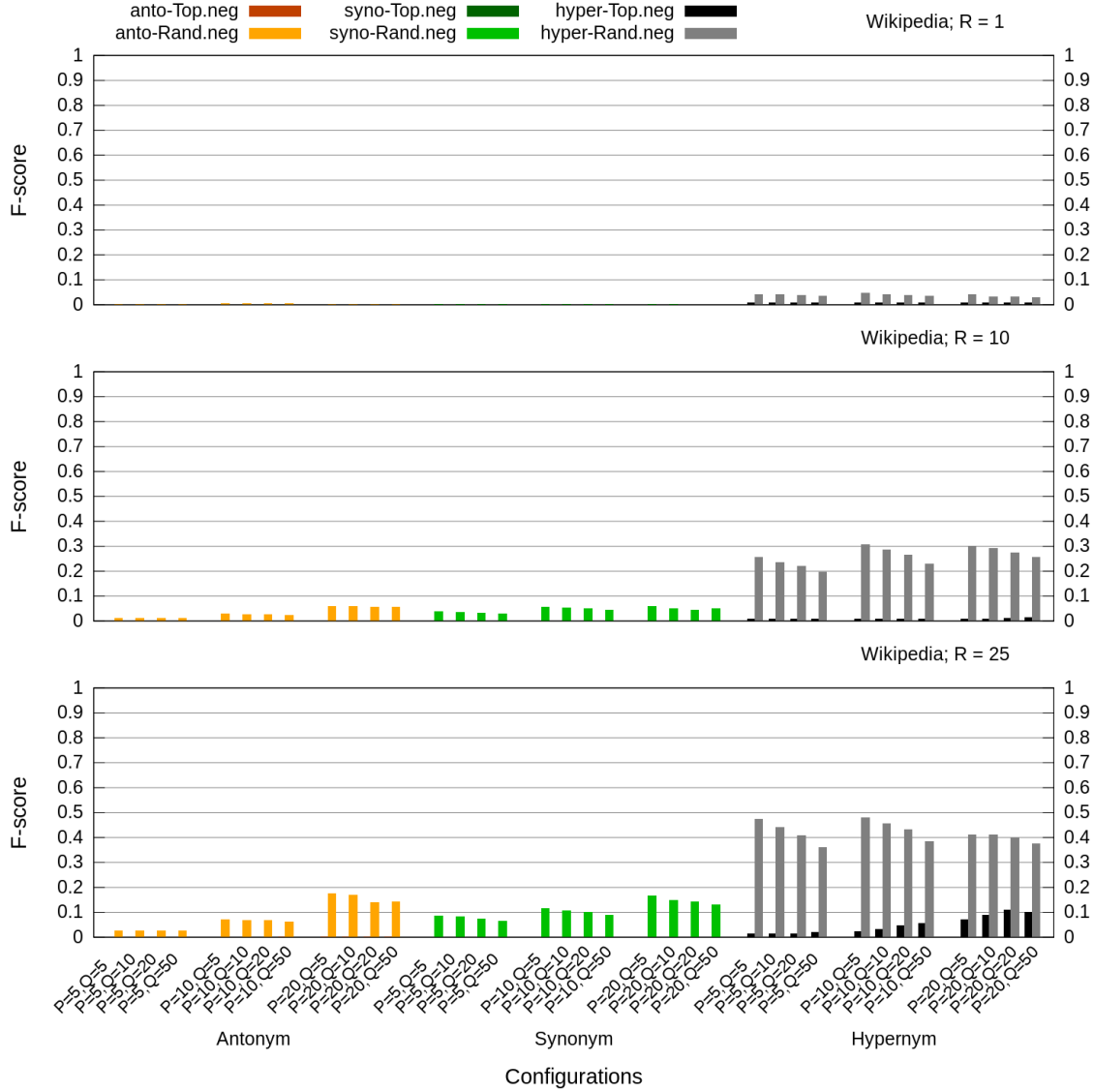


Figure 5.7: The accuracy of various WIKIPEDIA models based on two sampling procedures on the 1-MIL dataset

faster improvement over the equivalent models trained on the top- $|N|$ frequently occurring negative instances ($*-Top.neg$ models). Note also that the majority of the $*-Rand.neg$ models consistently outperform their $*-Top.neg$ counterparts by a large margin. There are a few exceptions to this, where $*-Top.neg$ models are more accurate than $*-Rand.neg$ models, but in this scenario, the f-scores margin between

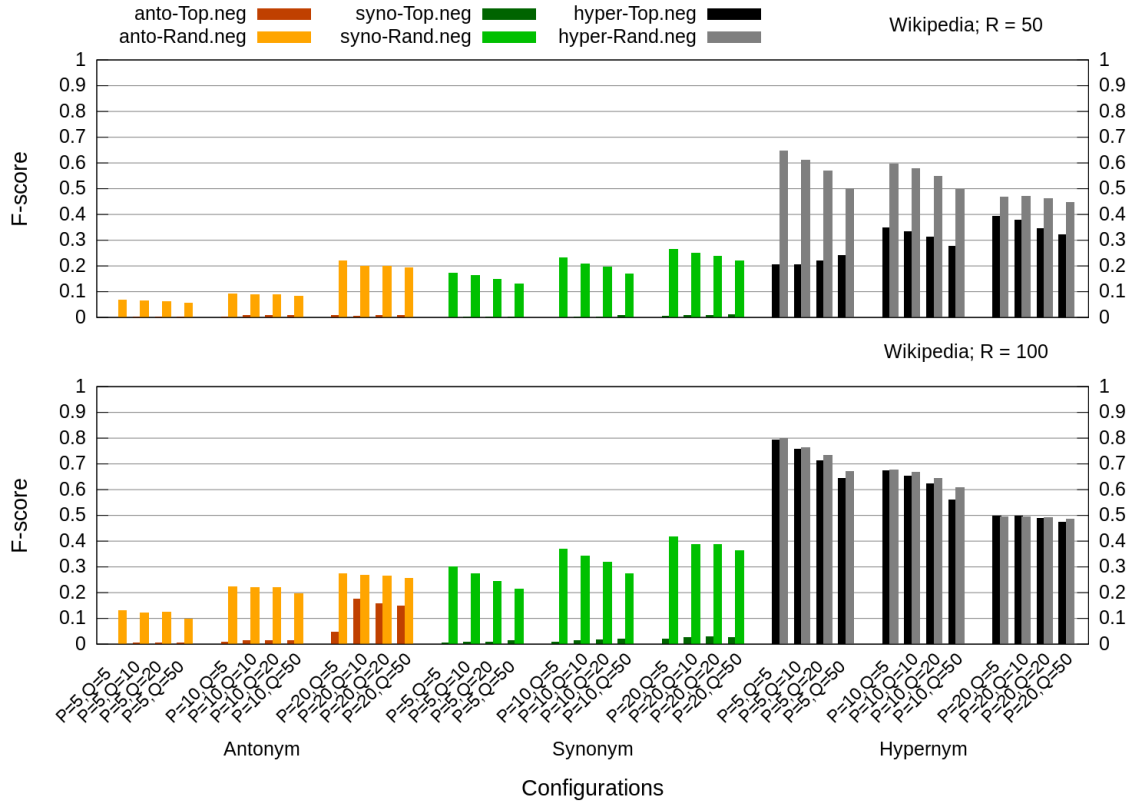


Figure 5.8: The accuracy of various WIKIPEDIA models based on two sampling procedures on the 1-MIL dataset

the two is very small (e.g. some hypernymy models when setting $R = 100$).

The major reason for the result improvement for the $\ast$ -Rand.neg models is due to the reduction in false positive predictions, which leads to higher precision. This suggests that the $\ast$ -Rand.neg models are more capable of identifying and classifying negative instances as compared to the $\ast$ -Top.neg models. This can be confidently attributed to the composition of the training instances whose aggregate sort index distribution closely resembles the distribution in the test data (1-MIL dataset).

Discussion

So far, we have discussed several sampling procedures to produce a common dataset that can be used to compare models trained using different settings. We have

solved the problem of evaluating across different corpora by sampling only from instances that have appeared in both corpora. Furthermore, we have also addressed the challenge of automatically determining the ratio of positive and negative instances for all three relations by randomly producing a large number of instances and assuming that these instances have the natural distribution of positive and negative instances for all of the three relations (and indeed, any other relations). We argue that this sampling procedure is more practical than Snow *et al.* (2005)’s procedure that requires hand annotation to discover the distribution of positive and negative instances for each relation. The various experiments in Section 5.1.2 have shown that our sampling procedure is capable of producing a dataset that allows fair and reasonable comparisons over different experimental settings.

Further, we have demonstrated the disconnect between the Snow *et al.* (2005) training methodology and our current sampling procedure. This can be observed by two important factors: (i) the way the negative instances are sampled, and (ii) the dependence on the R value in the training process. While we have proposed a new sampling procedure to address the first factor, the training procedure is still heavily dependent on the R value in producing the training dataset. This goes against the purpose of our sampling procedure which is to be independent of the influence of R value in generating the dataset. As a result, we have reservations regarding this training process and consider it to be flawed. This is a major reason why we did not fully investigate the impact of the other system parameters, P and Q , on the results produced by this training process.

Our next step is to modify the current training process to be more in line with the purpose of our sampling procedure. This requires a major shift in how our RE task is structured. The modification in the sampling procedure, however, is only a part of the restructuring of the current task to make it more reflective of an open evaluation. We describe the full setup in the next section.

5.2 Revisiting Our RE Task

Up to this point, the core of our RE task has been based on how Snow *et al.* (2005) conducted their experiments. This is in agreement with one of our original objectives, which is to investigate the extendability of Snow *et al.* (2005)’s system. However, in building an RE system and conducting experiments based on Snow *et al.* (2005)’s task, we have identified a number of assumptions which arguably do not reflect the actuality of relation extraction. These assumptions can be found in both the relation labelling process and the sampling procedure in both training and evaluation. In this section, we revisit these assumptions made in the original task. We analyse and address these assumptions in detail, in order to propose a better task that is more reflective of the real-world situation.

5.2.1 Revisiting the Relation Labelling Process

We described our initial relation labelling process in Section 4.1.3. The process is largely based on the Snow *et al.* (2005) heuristic, which was designed for hypernymy instances only. We extend this process to include the two additional relations in our experimentation: antonymy and synonymy.

Reconsidering SEMCOR’s Role

It can be argued that there is no restriction on which nouns can co-occur in real-world data. However, our relation labelling process is dependent on WORDNET relational information, and it is not possible to label a relation between noun pairs using WORDNET information if either one or both nouns do not have a WORDNET entry. To avoid the expensive manual relation labelling between noun pairs that do not appear in WORDNET, we argue that it is fair to only train on noun pairs where both nouns appear in WORDNET.

The absence of WORDNET entries for such nouns does not prevent the possibility from using such nouns in the evaluation process, although this is not one of our research objectives. In fact, by using such nouns in the evaluation process, the system

may predict the existence of relations between a noun in WORDNET entry and a noun not in WORDNET, or between two nouns not in WORDNET. Such predictions may have to be validated by an expert annotator. If the expert annotator agrees with the system's predictions, the discovered nouns and relations may be added into the WORDNET database. An accurate system may thus prove to be valuable for expert annotators for populating, curating, and maintaining a lexical hierarchy or similar resources.

Snow *et al.* (2005) originally required nouns to have exactly one SEMCOR-tagged sense, and excluded nouns with more than one SEMCOR-tagged sense from their experiments. This is to reduce the number of polysemous hypernym pairings in their data, which greatly simplifies their task. This, however, does not mean that they exclude all polysemous nouns in their data, as polysemous nouns may be present in the data, as long as only one of the nouns' multiple senses is SEMCOR-tagged. By definition, this SEMCOR-tagged sense is the most-frequently appearing sense (the dominant sense).

In Section 4.1.3, we considered it to be highly unrealistic to restrict polysemous nouns to only have one SEMCOR-tagged sense, and responded to this by relaxing the SEMCOR requirement: requiring nouns to have at least one SEMCOR-tagged sense. As a result, polysemy is more apparent in our RE task in Chapter 4, and thus the task is more realistic compared to the Snow *et al.* (2005) task.

Arguably, the task would be more realistic if all WORDNET nouns were eligible, regardless of whether they have a SEMCOR-tagged sense or not. This prompts us to question the role of SEMCOR in this section.

Naturally, our labelling heuristic requires the knowledge of the order of the nouns' senses to determine the relation labels (recall that the heuristic considers a noun pair to be a positive instance iff there is a link between both of the nouns' dominant (first) senses, and a noun pair to be a negative instance if there is no link between any pairings of the noun senses). We discussed in Section 3.1 that WORDNET presents the senses of a word in the descending order of their frequency SEMCOR, and that the order for senses with the same SEMCOR frequency is determined by a random tie-breaking process. Senses without any SEMCOR occurrences are presented last in

the ordering.

For polysemous nouns, SEMCOR is absolutely necessary to determine the order of WORDNET senses for our heuristic. By requiring polysemous nouns to have at least one SEMCOR-tagged sense, we ensure the presence of a dominant sense for a particular noun. It is not possible to deal with polysemous nouns without any SEMCOR-tagged sense, as we cannot rely on the order of their WORDNET senses to determine a dominant sense due to the random tie-breaking process in ordering the senses'. As a result, we still have to require polysemous nouns to have at least one SEMCOR-tagged sense.

In contrast, SEMCOR is not required in determining the sense order for monosemous nouns, as by definition, the single sense is the dominant sense. This property of monosemous nouns was not considered by both Snow *et al.* (2005)'s and our previous relation labelling heuristic in Section 4.1.3, which require monosemous nouns to have at least one SEMCOR-tagged sense. For our current task, we remove this requirement, and therefore enabling all monosemous nouns to be eligible.

There are 95,820 monosemous nouns whose single sense is not SEMCOR-tagged. Adding these nouns to the 11,846 nouns eligible in Chapter 4, there are 107,666 nouns eligible for our RE task out of the entire 117,798 WORDNET nouns. We conclude that the modification of the heuristic by reconsidering the role of SEMCOR in relation labelling brings the current RE setup to be more reflective of real-world data.

Revisiting the Relation Labelling Heuristic

The relation labelling heuristic described in Section 4.1.3 maps WORDNET nouns to their WORDNET first senses. For a given relation, the heuristic labels a noun pair to be positive iff the first senses of both nouns have a WORDNET relation link for that relation (there is a slight difference in the labelling process for the antonymy and synonymy relations; see Section 4.1.3). Conversely, for a given relation, the heuristic labels a noun pair to be negative iff there is no such WORDNET relation link between any senses of the two nouns.

In Section 4.1.3, we identified two types of noun pairs that are excluded due

Noun Pairs	ANTO	SYNO	HYPER
(<i>agreement</i> , <i>disagreement</i>)	excluded	—	—
(<i>heir</i> , <i>successor</i>)	—	excluded	—
(<i>chip</i> , <i>food</i>)	—	—	excluded

Table 5.5: Examples of noun pairs excluded from the experiment because the relation link is not between the first senses of both nouns. Notice that the noun pair is excluded only in one relation, but is labelled as a negative pair for the other relations

to this heuristic. The first type is where noun pairs are excluded across all three relations because they do not appear in WORDNET or because the nouns do not have any SEMCOR-tagged senses. Previously in this section, we addressed this issue for monosemous nouns and words not appearing in WORDNET. The second type is noun pairs that are excluded because there exists a relation between the two nouns, but not via their first senses. This exclusion only applies to a specific relation, rather than to all three relations. We reproduce examples for these noun pairs types used in Section 4.1.3 in Table 5.5.

In this section, we revisit the heuristic for excluding noun pairs that have a relation link, but not between their WORDNET first senses. Looking at the examples in Table 5.5, we can argue that our previous heuristic may have missed some potential positive instances for training and evaluation processes, as the examples shown noun pairs whose relations are arguably clearly-attested (at the word-level).

At the same time, our previous heuristic avoids not-so-clearly-attested annotations between two nouns that are linked by their two non-dominant senses. For example, there is a hypernym link between *lap*₅ “movement once around a course” and *travel*₃ “self-propelled movement”. This situation clearly shows the tension of using a word-level corpus and a sense-level gold-standard, which we circumnavigate by mapping every noun to its WORDNET first sense and assuming that the first sense is the correct sense to assign for all nouns.

To differentiate between these different types of instances, we introduce new ter-

minology for each instances type:

1. *white* instances, referring to noun pairs whose relations are between their WORDNET first senses (i.e. the positive instances);
2. *black* instances, referring to noun pairs which do not have a relation between any of their WORDNET senses (i.e. the negative instances);
3. *grey* instances, referring to noun pairs whose relations are between their non-first senses. The grey colour represents the “murky” nature of such instances (which are often not self-apparent at the word-level) as compared to the more clearly-attested white and black instances. They are the focus of this section.

Grey instances regularly appear in the data, and we argue that there is much to be said for factoring them into our RE task, rather than completely ignoring them as we have previously done. We propose four methods for addressing grey instances:

1. In both the training and evaluation processes, treat the grey and black instances as negative instances, and white instances as positive instances (**train:** pos=w, neg=g,b; **eval:** pos=w, neg=g,b)
2. In both the training and evaluation processes, treat the grey and white instances as positive instances, and black instances as negative instances (**train:** pos=w,g, neg=b; **eval:** pos=w,g, neg=b)
3. Exclude the grey instances from the training process (using white instances as positive instances, and black instances as negative instances), but treat them as negative instances during evaluation (**train:** pos=w, neg=b; **eval:** pos=w, neg=g,b)
4. Exclude the grey instances from the training process (using white instances as positive instances, and black instances as negative instances), but treat them as positive instances during evaluation (**train:** pos=w, neg=b; **eval:** pos=w,g, neg=b)

These four methods demonstrate different possible positions on the murky nature of grey instances. The idea in the last two methods is to exclude grey instances, which are not as clearly-attested at word-level compared to the white and black instances, from possibly confusing the models during training.

We consider our main RE task to be extracting clearly-attested (white) instances from the data. Therefore in our main experiment, we assume that the grey instances are negative instances (methods 1 and 3 above). However, we can be flexible with our task setup (methods 2 and 4 above), which enables us to investigate the impact of grey instances on RE. We argue that by not excluding the grey instances in our experiments, we have placed our task closer to the real world actuality than Snow *et al.* (2005) task.

5.2.2 Revisiting the Training and Evaluation Processes

Having labelled the data, the next process is to determine which instances to include in our experiments. In this section, we address several pertinent issues with the sampling procedure, training and evaluation described in Section 5.1.2 in forming our new RE task. Recognising the advantage of randomly sampling from the data, we decide to keep the random sampling procedure for our new task setup. However, we remove the constraint of having to sample from the INTERSECTION dataset with at least 5 different patterns ($P = 5$). Instead, all the instances in the INTERSECTION dataset are eligible for random sampling. This is in response to what we perceive to be an essential requirement for a good supervised RE system, that it is able to produce a prediction for a given noun pair regardless of the (potentially small) number of features associated with that particular noun pair. Furthermore, by removing this requirement, we do not have to worry about training from different datasets produced by various P parameters as we were in previous setups.

With the new relation labelling heuristic proposed in Section 5.2.1, and after excluding noun pairs where at least one of the nouns does not appear in WORDNET, there are slightly less than 8 million instances eligible for sampling (7,905,696 instances to be exact). Table 5.6 lists the distributions of the white, grey, and black

RELATION	CLASS	All Eligible Instances	Training Partition	Dev Partition	Test Partition
ANTO	white	208	64	66	63
	grey	226	75	67	73
	black	7,905,262	2,499,861	2,499,867	2,499,864
SYNO	white	3,566	1,190	1,063	1,141
	grey	6,685	2,136	2,074	2,124
	black	7,895,445	2,496,674	2,496,863	2,496,735
HYPER	white	27,116	8,620	8,523	8,577
	grey	46,157	14,621	14,683	14,517
	black	7,832,423	2,476,759	2,476,794	2,476,906

Table 5.6: The class distributions of all eligible instances and the instances found in the three experimental partitions

instances in the data. As we can observe, black instances are more prominent in the data than both the white and grey instances. Further, there are slightly more grey instances than white instances for all relations assumed, a fact that is ignored in our earlier experiments.

As per our experimental setup, we also present the different ratios of negative to positive instances for the three relations in Table 5.7, depending on the interpretation of the grey instances as reflected in our proposed methods in Section 5.2.1. These ratios may give an idea of the level of class imbalance for each relation under the different experimental setups, which may indicate the difficulty level of the learning task (the higher the ratio, the more difficult the learning task). As with previous experiments, antonyms have the highest ratio, followed by synonyms and hypernyms. The grey instances play a significant role in the data, as seen from the drastic ratio reduction when treating the grey instances as positive instances (i.e. ratios for **pos=w**, **neg=g,b** vs. **pos=w,g**, **neg=b**).

RELATION	CLASS	All Eligible Instances	Training Partition	Dev Partition	Test Partition
ANTO	pos=w neg=g,b	38,007.15	39,061.50	37,877.79	39,681.54
	pos=w,g neg=b	18,214.89	17,984.61	18,795.99	18,381.35
	pos=w neg=b	38,006.07	39,060.33	37,876.77	39,680.38
SYNO	pos=w neg=g,b	2,215.96	2,099.84	2,350.83	2,190.06
	pos=w,g neg=b	770.21	750.65	795.94	764.70
	pos=w neg=b	2,214.09	2,098.05	2,348.88	2,188.20
HYPER	pos=w neg=g,b	290.55	289.02	292.32	290.48
	pos=w,g neg=b	106.89	106.57	106.73	107.25
	pos=w neg=b	288.85	287.33	290.60	288.78

Table 5.7: The ratio of negative to positive of all eligible instances and the instances found in the three experiment partitions

We propose to run a holdout method for both training and evaluation, by splitting the entire data of almost 8 million instances into three equal-sized partitions of 2.5 million instances, which we use as training, development, and test partitions. The

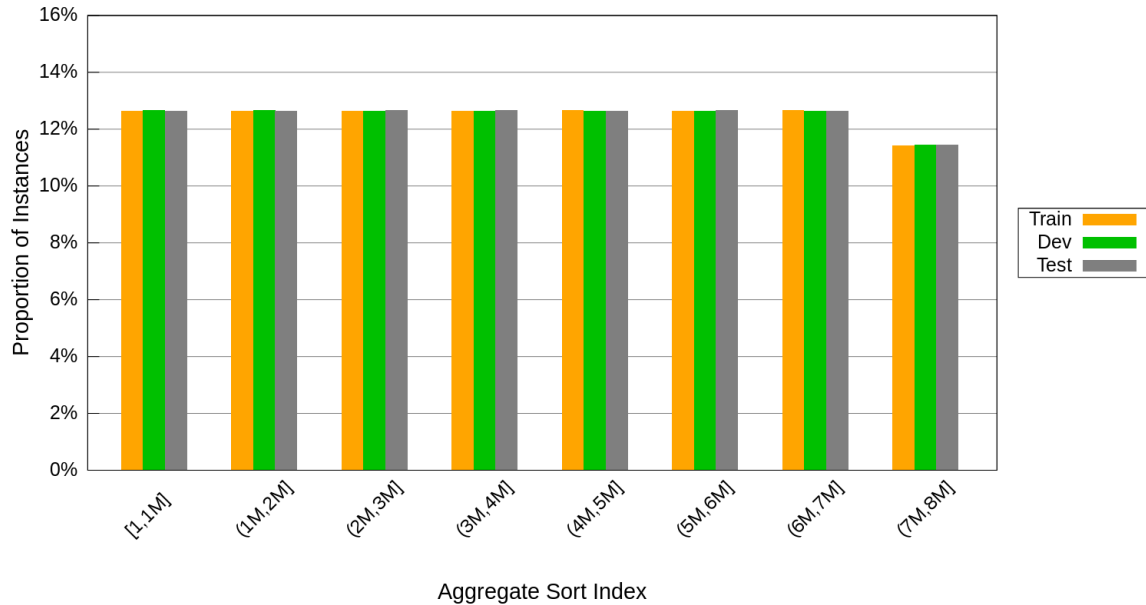


Figure 5.9: Aggregate sort index distribution on the three experiment partitions

training partition can be used in training models, while the development partition can be used to optimise our classifiers before applying them to the test partition. There is no special reason why we decided to split our data into sets of 2.5 million instances apart from the fact that it is the largest round number when we divide 8 million by 3. As a result, there is a small portion (just over 5%) of all eligible instances that are not used in our experiments.

As per the natural distribution of random sampling, we expect the three partitions to have a similar class distribution. This can be confirmed from Tables 5.6 and 5.7. Admittedly, the ratio of antonym instances in different partitions is not as close to the entire dataset as the ratio of synonym and hypernym instances, but this can be justified by the smaller number of white and grey instances and resultant data sparseness.

Furthermore, we check the aggregate sort index of each partition to ensure an equal distribution of instance frequency. To do this, we assign an individual aggregate sort index to each instance in the entire dataset based on their aggregate frequency in both corpora (similar to the previous experiment in Section 5.1.2). We then check

the distribution of these indexes for the instances in the three experiment partitions. As we can observe from Figure 5.9, the three partitions have similar distributions to each other, and within each partition, the aggregate sort index is equally distributed in the data.

5.3 Experiments

To compare the quality of the two corpora as a learning source for an RE task, we generate two sets of features for all noun pairs, each based on the individual instance-pattern co-appearance information in GIGAWORD and WIKIPEDIA separately.

In previous experiments, we have used various settings of the system parameter Q ($Q \in \{5, 10, 20, 50\}$) to select patterns for feature generation based on the minimum number of unique noun pairs (Q) the patterns must occur with. Ideally, we prefer to use all the patterns that have occurred in the corpora as our features, parallel to our consideration that all noun pairs appearing in both corpora are eligible for the experiments. However, we run into a practical problem in that due to the large number of all patterns in the corpora, it is not possible to convert them into a feature vector that can fit into main memory for the machine learners. We address this problem by compensating our requirement to use the lowest Q setting possible so that all the patterns satisfying that setting can be translated to a feature vector that fits into memory ($Q = 2$).

Table 5.8 illustrates the number of patterns found in the corpora. As a reminder, because of the pattern-to-feature conversion process (see Section 4.1.5), the size of the feature sets for any Q settings will be larger than the number of patterns listed in Table 5.8.

Having generated the features, we train our models on the training partition, and perform an initial evaluation on the development partition to discover the best models. In turn, we use these best models to classify on the test partition.

We present our results for each relation in two tables, based on the different treatments of grey instances for evaluation (eval: pos=w, neg=g,b vs. eval: pos=w,g, neg=b). We divide each table into two, where the top half shows the results when

$\Lambda pat_k \geq Q$	GIGAWORD	WIKIPEDIA
$Q = 1$ (all patterns)	136,307,240	128,114,140
$Q = 2$	19,577,846	15,798,492
$Q = 5$	5,120,966	3,935,698
$Q = 10$	2,316,150	1,756,760
$Q = 20$	1,097,524	827,964
$Q = 50$	416,692	314,620

Table 5.8: The number of patterns extracted from GIGAWORD and WIKIPEDIA for various values of Q

the grey instances are used during training (train: **pos=w, neg=g,b** or train: **pos=w,g, neg=b**, depending on the evaluation), and the bottom half shows the results when the grey instances are excluded from training (train: **pos=w, neg=b**).

5.3.1 Results on the Development Partition

Antonymy

We present the system results for antonymy evaluated on the development partition in Tables 5.9 and 5.10. The majority of our reported results are very poor (f-scores below 0.05). This can be attributed to the small number of instances predicted as positive (having an antonymy relation) by the classifiers, with the majority of the models predicting less than 10 positive noun pairs in the development partition. Considering the number of positive noun pairs in the data, the coverage of our system is undoubtedly very low, and this is clearly reflected in our low recall. In contrast, the reported precision is generally higher than recall, although the small set of positive predictions are arguably favourable for the precision calculation.

The exceptions to the poor results are the models that use grey instances as positive instances (**pos=w,g, neg=b**) during both training and evaluation (the top half of Table 5.10). Compared to the other models, they predicted more (correct)

ANTO Models		Evaluated on the Dev Partition					
CLASS	CORPUS, Q	pos=w, neg=g,b					
		TP	FP	FN	R	P	F
pos=w neg=g,b	GIGAWORD, $Q = 2$	1	2	65	0.015	0.333	0.029
	GIGAWORD, $Q = 5$	1	3	65	0.015	0.250	0.028
	GIGAWORD, $Q = 10$	1	3	65	0.015	0.250	0.028
	GIGAWORD, $Q = 20$	1	1	65	0.015	0.500	0.029
	GIGAWORD, $Q = 50$	1	2	65	0.015	0.333	0.029
	WIKIPEDIA, $Q = 2$	0	2	66	0.000	0.000	0
	WIKIPEDIA, $Q = 5$	0	2	66	0.000	0.000	0
	WIKIPEDIA, $Q = 10$	0	2	66	0.000	0.000	0
	WIKIPEDIA, $Q = 20$	0	2	66	0.000	0.000	0
	WIKIPEDIA, $Q = 50$	1	3	65	0.015	0.250	0.028
pos=w neg=b	GIGAWORD, $Q = 2$	1	2	65	0.015	0.333	0.029
	GIGAWORD, $Q = 5$	1	4	65	0.015	0.200	0.028
	GIGAWORD, $Q = 10$	1	4	65	0.015	0.200	0.028
	GIGAWORD, $Q = 20$	1	3	65	0.015	0.250	0.028
	GIGAWORD, $Q = 50$	1	2	65	0.015	0.333	0.029
	WIKIPEDIA, $Q = 2$	0	4	66	0.000	0.000	0
	WIKIPEDIA, $Q = 5$	0	4	66	0.000	0.000	0
	WIKIPEDIA, $Q = 10$	0	4	66	0.000	0.000	0
	WIKIPEDIA, $Q = 20$	0	4	66	0.000	0.000	0
	WIKIPEDIA, $Q = 50$	1	4	65	0.015	0.200	0.028

Table 5.9: Results for antonymy models trained on the training partition and evaluated on the development partition, treating white instances as positive and grey and black instances as negative

ANTO Models		Evaluated on the Dev Partition					
CLASS	CORPUS, Q	pos=w,g, neg=b					
		TP	FP	FN	R	P	F
pos=w,g neg=b	GIGAWORD, $Q = 2$	7	11	126	0.053	0.389	0.093
	GIGAWORD, $Q = 5$	8	6	125	0.060	0.571	0.109
	GIGAWORD, $Q = 10$	10	12	123	0.075	0.455	0.129
	GIGAWORD, $Q = 20$	9	20	124	0.068	0.310	0.112
	GIGAWORD, $Q = 50$	9	17	124	0.068	0.346	0.114
	WIKIPEDIA, $Q = 2$	17	19	116	0.128	0.472	0.201
	WIKIPEDIA, $Q = 5$	19	19	114	0.143	0.500	0.222
	WIKIPEDIA, $Q = 10$	19	20	114	0.143	0.487	0.221
	WIKIPEDIA, $Q = 20$	22	21	111	0.165	0.512	0.250
	WIKIPEDIA, $Q = 50$	22	20	111	0.165	0.524	0.251
pos=w neg=b	GIGAWORD, $Q = 2$	1	2	132	0.008	0.333	0.016
	GIGAWORD, $Q = 5$	2	3	131	0.015	0.400	0.029
	GIGAWORD, $Q = 10$	2	3	131	0.015	0.400	0.029
	GIGAWORD, $Q = 20$	3	1	130	0.023	0.750	0.045
	GIGAWORD, $Q = 50$	2	1	131	0.015	0.667	0.029
	WIKIPEDIA, $Q = 2$	1	3	132	0.008	0.250	0.016
	WIKIPEDIA, $Q = 5$	1	3	132	0.008	0.250	0.016
	WIKIPEDIA, $Q = 10$	1	3	132	0.008	0.250	0.016
	WIKIPEDIA, $Q = 20$	1	3	132	0.008	0.250	0.016
	WIKIPEDIA, $Q = 50$	2	3	131	0.015	0.400	0.029

Table 5.10: Results for antonymy models trained on the training partition and evaluated on the development partition, treating white and grey instances as positive and black instances as negative

positive instances, which resulted in better f-scores.

The best model when treating grey instances as negative during evaluation (**eval: pos=w, neg=g,b**) is trained on patterns extracted from GIGAWORD, $Q = 20$, and treating grey instances as negative (**train: pos=w, neg=g,b**), with an f-score of 0.029. On the other hand, if we treat grey instances as positive during evaluation (**eval: pos=w,g, neg=b**), the best model is trained on patterns from WIKIPEDIA, $Q = 50$, and treating the grey instances as positive (**train: pos=w,g, neg=b**), with an f-score of 0.251.

It is difficult to determine the best corpus for antonymy learning based on the results in Table 5.9 and Table 5.10, as there is no clear indication that one corpus is better than the other. Indeed, a paired t-test at a confidence level of 95% shows that there is no statistically significant improvement in the WIKIPEDIA-trained models over the GIGAWORD-trained models. For the Q setting, we observed that models tend to be more accurate when using higher Q values, although we do not read too much into this observation due to the small number of true positives predicted by the models.

Synonymy

Tables 5.11 and 5.12 show the system's results for synonymy evaluated on the development partition.

Similar to the antonymy models, the reported results for the synonymy models are very poor, with none of the models' f-scores above 0.04. The synonymy models share some similar properties to the antonymy models, with a small number of positive predictions (relative to the total number of positive instances) for every system parameter setting. Overall, the recall scores are always lower than the precision scores.

The best model when treating grey instances as negative during evaluation (**eval: pos=w, neg=g,b**) is trained on patterns from WIKIPEDIA, $Q = 2$, and ignoring grey instances during model training (**train: pos=w, neg=b**), with an f-score of 0.034. The exclusion of grey instances in the best synonymy model during training is different

SYNO Models		Evaluated on the Dev Partition					
CLASS	CORPUS, Q	pos=w, neg=g,b					
		TP	FP	FN	R	P	F
pos=w	GIGAWORD, $Q = 2$	15	20	1,048	0.014	0.429	0.027
	GIGAWORD, $Q = 5$	17	33	1,046	0.016	0.340	0.031
	GIGAWORD, $Q = 10$	16	38	1,047	0.015	0.296	0.029
	GIGAWORD, $Q = 20$	16	38	1,047	0.015	0.296	0.029
	GIGAWORD, $Q = 50$	7	35	1,056	0.007	0.167	0.013
neg=g,b	WIKIPEDIA, $Q = 2$	17	30	1,046	0.016	0.362	0.031
	WIKIPEDIA, $Q = 5$	14	28	1,049	0.013	0.333	0.025
	WIKIPEDIA, $Q = 10$	12	27	1,051	0.011	0.308	0.021
	WIKIPEDIA, $Q = 20$	12	31	1,051	0.011	0.279	0.021
	WIKIPEDIA, $Q = 50$	15	40	1,048	0.014	0.273	0.027
pos=w	GIGAWORD, $Q = 2$	14	21	1,049	0.013	0.400	0.025
	GIGAWORD, $Q = 5$	18	40	1,045	0.017	0.310	0.032
	GIGAWORD, $Q = 10$	17	36	1,046	0.016	0.321	0.030
	GIGAWORD, $Q = 20$	16	39	1,047	0.015	0.291	0.029
	GIGAWORD, $Q = 50$	7	34	1,056	0.007	0.171	0.013
neg=b	WIKIPEDIA, $Q = 2$	19	36	1,044	0.018	0.345	0.034
	WIKIPEDIA, $Q = 5$	14	38	1,049	0.013	0.269	0.025
	WIKIPEDIA, $Q = 10$	14	32	1,049	0.013	0.304	0.025
	WIKIPEDIA, $Q = 20$	15	40	1,048	0.014	0.273	0.027
	WIKIPEDIA, $Q = 50$	16	41	1,047	0.015	0.281	0.028

Table 5.11: Results for synonymy models trained on the training partition and evaluated on the development partition, treating white instances as positive and grey and black instances as negative

SYNO Models		Evaluated on the Dev Partition					
CLASS	CORPUS, Q	pos=w,g, neg=b					
		TP	FP	FN	R	P	F
pos=w,g neg=b	GIGAWORD, $Q = 2$	20	193	3,117	0.006	0.094	0.011
	GIGAWORD, $Q = 5$	25	258	3,112	0.008	0.088	0.015
	GIGAWORD, $Q = 10$	22	235	3,115	0.007	0.086	0.013
	GIGAWORD, $Q = 20$	25	223	3,112	0.008	0.101	0.015
	GIGAWORD, $Q = 50$	23	277	3,114	0.007	0.077	0.013
	WIKIPEDIA, $Q = 2$	37	123	3,100	0.012	0.231	0.023
	WIKIPEDIA, $Q = 5$	35	174	3,102	0.011	0.167	0.021
	WIKIPEDIA, $Q = 10$	38	193	3,099	0.012	0.165	0.022
	WIKIPEDIA, $Q = 20$	50	265	3,087	0.016	0.159	0.029
	WIKIPEDIA, $Q = 50$	48	240	3,089	0.015	0.167	0.028
pos=w neg=b	GIGAWORD, $Q = 2$	14	21	3,123	0.004	0.400	0.008
	GIGAWORD, $Q = 5$	18	40	3,119	0.006	0.310	0.012
	GIGAWORD, $Q = 10$	17	36	3,120	0.005	0.321	0.010
	GIGAWORD, $Q = 20$	16	39	3,121	0.005	0.291	0.010
	GIGAWORD, $Q = 50$	7	34	3,130	0.002	0.171	0.004
	WIKIPEDIA, $Q = 2$	23	32	3,114	0.007	0.418	0.014
	WIKIPEDIA, $Q = 5$	18	34	3,119	0.006	0.346	0.012
	WIKIPEDIA, $Q = 10$	17	29	3,120	0.005	0.370	0.010
	WIKIPEDIA, $Q = 20$	19	36	3,118	0.006	0.345	0.012
	WIKIPEDIA, $Q = 50$	22	35	3,115	0.007	0.386	0.014

Table 5.12: Results for synonymy models trained on the training partition and evaluated on the development partition, treating white and grey instances as positive and black instances as negative

than the best antonymy model, where it uses the grey instances as negative instances during training (**train: pos=w, neg=g,b**). With the other evaluation setting — treating grey instances as positive instances (**eval: pos=w,g, neg=b**) — the best model is trained on patterns from WIKIPEDIA, $Q = 20$, and similar to the evaluation setting, treating grey instances as positive (**eval: pos=w,g, neg=b**), with an f-score of 0.029.

Comparing the two corpora, WIKIPEDIA is a better learning resource than GIGAWORD for synonymy, at a level of statistical significance (paired t-test, $p < 0.05$). There is no clear-cut observation that can be made for the Q setting: some systems perform better on low Q values, while the others perform better at higher Q values.

Hypernymy

We present the system results for hypernymy evaluated on the development partition in Tables 5.13 and 5.14. Compared to both the antonymy and synonymy models, hypernymy models are consistently more accurate (with the exception of select antonymy models evaluated and trained on (**pos=w,g, neg=b**)). However, they still suffer from the small number of positive predictions relative to the total number of positive instances. Moreover, like the antonymy and synonymy models, the recall scores for hypernymy models are always lower than their precision scores.

The best model when assuming grey instances as negative during evaluation (**eval: pos=w, neg=g,b**) is trained on patterns from WIKIPEDIA, $Q = 10$, and ignoring grey instances during model training (**train: pos=w, neg=g,b**), with an f-score of 0.117. When treating grey instances as positive during evaluation (**eval: pos=w,g, neg=b**), the best model is trained on patterns from WIKIPEDIA, $Q = 5$, and treating grey instances as positive (**train: pos=w,g, neg=b**).

We observed in both Tables 5.13 and 5.14 that on an equivalent system parameter setting, WIKIPEDIA-trained models are always more accurate than GIGAWORD-trained models, at a level of statistical significance (paired t-test, $p < 0.05$). While not as obvious as the difference in the corpora, systems trained on mid-range Q values ($Q = 5$ or 10) tend to be more accurate than systems trained on other Q values.

HYPER Models		Evaluated on the Dev Partition					
CLASS	CORPUS, Q	pos=w, neg=g,b					
		TP	FP	FN	R	P	F
pos=w neg=g,b	GIGAWORD, $Q = 2$	357	876	8,166	0.042	0.290	0.073
	GIGAWORD, $Q = 5$	384	1,051	8,139	0.045	0.268	0.077
	GIGAWORD, $Q = 10$	397	1,128	8,126	0.047	0.260	0.080
	GIGAWORD, $Q = 20$	353	960	8,170	0.041	0.269	0.071
	GIGAWORD, $Q = 50$	361	1,036	8,162	0.042	0.258	0.072
	WIKIPEDIA, $Q = 2$	478	637	8,045	0.056	0.429	0.099
	WIKIPEDIA, $Q = 5$	524	897	7,999	0.061	0.369	0.105
	WIKIPEDIA, $Q = 10$	518	942	8,005	0.061	0.355	0.104
	WIKIPEDIA, $Q = 20$	519	995	8,004	0.061	0.343	0.104
	WIKIPEDIA, $Q = 50$	440	769	8,083	0.052	0.364	0.091
pos=w neg=b	GIGAWORD, $Q = 2$	381	963	8,142	0.045	0.283	0.078
	GIGAWORD, $Q = 5$	409	1,166	8,114	0.048	0.260	0.081
	GIGAWORD, $Q = 10$	413	1,190	8,110	0.048	0.258	0.081
	GIGAWORD, $Q = 20$	401	1,219	8,122	0.047	0.248	0.079
	GIGAWORD, $Q = 50$	387	1,123	8,136	0.045	0.256	0.077
	WIKIPEDIA, $Q = 2$	558	848	7,965	0.065	0.397	0.112
	WIKIPEDIA, $Q = 5$	574	925	7,949	0.067	0.383	0.114
	WIKIPEDIA, $Q = 10$	600	1,102	7,923	0.070	0.353	0.117
	WIKIPEDIA, $Q = 20$	556	970	7,967	0.065	0.364	0.110
	WIKIPEDIA, $Q = 50$	512	913	8,011	0.060	0.359	0.103

Table 5.13: Results for hypernymy models trained on the training partition and evaluated on the development partition, treating white instances as positive and grey and black instances as negative

HYPER Models		Evaluated on the Dev Partition					
CLASS	CORPUS, Q	pos=w,g, neg=b					
		TP	FP	FN	R	P	F
pos=w,g neg=b	GIGAWORD, $Q = 2$	1,041	3,415	22,165	0.045	0.234	0.075
	GIGAWORD, $Q = 5$	1,015	3,352	22,191	0.044	0.232	0.074
	GIGAWORD, $Q = 10$	1,131	4,511	22,075	0.049	0.200	0.079
	GIGAWORD, $Q = 20$	1,066	4,372	22,140	0.046	0.196	0.075
	GIGAWORD, $Q = 50$	1,051	4,589	22,155	0.045	0.186	0.072
	WIKIPEDIA, $Q = 2$	1,340	3,135	21,866	0.058	0.299	0.097
	WIKIPEDIA, $Q = 5$	1,471	3,912	21,735	0.063	0.273	0.102
	WIKIPEDIA, $Q = 10$	1,478	4,293	21,728	0.064	0.256	0.102
	WIKIPEDIA, $Q = 20$	1,481	4,471	21,725	0.064	0.249	0.102
	WIKIPEDIA, $Q = 50$	1,320	3,935	21,886	0.057	0.251	0.093
pos=w neg=b	GIGAWORD, $Q = 2$	503	841	22,703	0.022	0.374	0.042
	GIGAWORD, $Q = 5$	540	1,035	22,666	0.023	0.343	0.043
	GIGAWORD, $Q = 10$	540	1,063	22,666	0.023	0.337	0.043
	GIGAWORD, $Q = 20$	535	1,085	22,671	0.023	0.330	0.043
	GIGAWORD, $Q = 50$	512	998	22,694	0.022	0.339	0.041
	WIKIPEDIA, $Q = 2$	742	664	22,464	0.032	0.528	0.060
	WIKIPEDIA, $Q = 5$	769	730	22,437	0.033	0.513	0.062
	WIKIPEDIA, $Q = 10$	802	900	22,404	0.035	0.471	0.065
	WIKIPEDIA, $Q = 20$	741	785	22,465	0.032	0.486	0.060
	WIKIPEDIA, $Q = 50$	694	731	22,512	0.030	0.487	0.057

Table 5.14: Results for hypernymy models trained on the training partition and evaluated on the development partition, treating white and grey instances as positive and black instances as negative

5.3.2 Observations on System Parameter Settings

There are some general observations on the system parameter settings that can be drawn based on the system results on the development partition.

With the exception of antonymy, WIKIPEDIA is a significantly better learning corpus as compared to GIGAWORD for our RE task. This is despite WIKIPEDIA being smaller in size, and containing less sentences.

There is no strong agreement for the best Q setting for the different relations. However, it is interesting to note that models trained on $Q = 2$ are usually not the most competitive, despite learning from the most number of patterns as compared to the other models. This suggests that the low-frequency patterns are not as reliable as the high-frequency patterns. Similarly, models trained on $Q = 50$ tend not to be the most competitive, suggesting that the models may not have learned from reliable patterns that are excluded during training on the basis of not meeting the required threshold.

Given all this, it is not surprising to observe that models trained on mid-range Q values ($Q = 5$ or 10) are consistently competitive as compared to systems trained on other Q values. This suggests that the mid-range Q values are optimal cut-off points for excluding unreliable patterns from learning.

On another topic, the use of grey instances in RE learning had variable impact on system accuracy. For our RE task that considers grey instances as negative during evaluation (eval: pos=w, neg=g,b), the results for synonymy and hypernymy suggest that grey instances are best excluded from learning altogether, i.e. that one should train only using white and black instances (train: pos=w, neg=b). This is somewhat unexpected, as one may argue that when grey instances are considered to be negative, training the system with grey instances as negative may provide a clearer class separation for the models. Arguably, this reflects the nature of the grey instances, i.e. they are not as homogeneous and clear-cut as white and black instances.

On the other task that considers grey instances as positive during evaluation (eval: pos=w,g, neg=b), all the best models for the three relations use the grey instances as positive instances during training (train: pos=w,g, neg=b). These best models

are always more accurate than the models that excluded positive instances during training (train: `pos=w, neg=b`) for the same evaluation procedure (eval: `pos=w,g, neg=b`). This result suggests that models that exclude grey instances as positive during training miss out on reliable (positive) class information provided by the grey instances.

At this point, we do not know how much of an impact the grey instances have in training our RE system in separating the clearly-attested instances (white and black instances), as our evaluation methods always consider grey instances as either positive or negative instances. To address this, we propose to evaluate our training models on data that only contains white and black instances (eval: `pos=w, neg=b`). Based on our earlier observations regarding the optimal settings, we choose to present results from the three models (each for the different methods of treating the grey instances during training) trained using WIKIPEDIA features, and $Q = 10$ after applying them to the subset of the development partition in which the grey instances are excluded from evaluation.

Table 5.15 shows the results of this experiment. We observe that out of the three training methods, the method which takes grey instances as positive for training (train: `pos=w,g, neg=b`) produces the best models for all relations. Such models predict more positive instances compared to models trained using the other two training methods.

For the current evaluation setup, models that use grey instances as positive during training (train: `pos=w,g, neg=b`) are more accurate (have higher f-scores) than models that ignore grey instances during training (train: `pos=w, neg=b`). This observation suggests that for this evaluation setup, the grey instances are better utilised as positive rather than negative examples during training. However, we note that the improvement in f-score is due to a higher recall, but at the expense of precision. Comparing the hypernymy models, the recall of the model trained with grey instances as positive (train: `pos=w,g, neg=b`) is higher than the model that excludes grey instances (train: `pos=w, neg=b`) from training (0.095 vs. 0.070), but its precision is much lower (0.158 vs. 0.400).

This leads us to question whether the f-score improvement is mainly due to the quality of the grey instances' features as positive instances. In this case, apart from

WIKIPEDIA, $Q = 10$ Models		Evaluated on the Dev Partition					
RELATION	Training Method	pos=w, neg=b					
		TP	FP	FN	R	P	F
ANTO	train: pos=w, neg=g,b	0	2	66	0.000	0.000	0.000
	train: pos=w,g, neg=b	8	20	58	0.121	0.286	0.170
	train: pos=w, neg=b	0	3	66	0.000	0.000	0.000
SYNO	train: pos=w, neg=g,b	12	24	1,051	0.011	0.333	0.021
	train: pos=w,g, neg=b	22	193	1,041	0.021	0.102	0.035
	train: pos=w, neg=b	14	29	1,049	0.013	0.326	0.025
HYPER	train: pos=w, neg=g,b	518	778	8,005	0.061	0.400	0.106
	train: pos=w,g, neg=b	806	4,293	7,717	0.095	0.158	0.119
	train: pos=w, neg=b	600	900	7,923	0.070	0.400	0.119

Table 5.15: Results for various models trained on the training partition on WIKIPEDIA features and $Q = 10$, evaluated on the subset of the development partition where grey instances are excluded from evaluation

the expected improvement in recall, it is fair to expect that the precision of the **train: pos=w,g, neg=b** models would be at a comparable level to the **train: pos=w, neg=b** models, which implies that the grey instances possess quality features indicating that they are good positive instances for training. However, the results in Table 5.15 contradict this expectation, with the precision scores for the **train: pos=w,g, neg=b** models below those for the **train: pos=w, neg=b** models (the precision score drops more than 50%).

It is possible that the f-score improvement is due to a side effect of treating the grey instances as positive which reduces the ratio of negative to positive instances during training. As we showed in Table 5.7, the ratio of negative to positive between instances in the data fluctuates greatly between the three training methods, especially when we treat grey instances as positive during training. Due to the reduction of the

ratio of negative and positive instances in the data, the models are more willing to predict positive instances, leading to a higher recall. At the same time, as the models are trained using grey instances (which are not-clearly attested) as positive, one may expect the models to incorrectly classify some negative pairs as positive, leading to a lower precision. These expectations are in line with the results shown in Table 5.15, prompting us to conclude that the main reason for the f-score improvement for the **train: pos=w,g, neg=b** models is due to subsequent reduction of the ratio of negative to positive instances as a result of treating grey instances as positive during training.

The other result that we want to note for this task is that models that exclude grey instances from the data (**train: pos=w, neg=b**) always perform better than models that treat grey instances as negative during training (**train: pos=w, neg=g,b**). We argue that without the non-clearly attested instances (grey instances) as either positive or negative instances in the training data, the learner may achieve a better class separation between the clearly-attested instances, which subsequently leads to an overall improvement in system accuracy. Further, the nature of the improvement between **train: pos=w, neg=b** models to **train: pos=w, neg=g,b** models is mainly due to higher recall, but at a comparable (or better precision), which supports our hypothesis. We also note that the small reduction in the ratio of negative to positive instances (due to the exclusion of grey instances from training data) may have contributed in improving system accuracy.³

Based on the observations above, for the current task of separating clearly-attested instances (white and black instances), we conclude that using grey instances (as either positive or negative examples) for model training is best avoided. Although using grey instances as positive training instance may improve the f-scores, the explanation behind the improvement is dissatisfying, as there is a high possibility the improvement is caused simply by the reduction of the ratio of negative to positive instances.

³Although we only showed the results for models trained on WIKIPEDIA, and $Q = 10$, we observed similar results for systems trained over different system parameter settings.

RELATION	Settings to Classify Test Partition, assuming pos=w, neg=g,b
ANTO	GIGAWORD, $Q = 20$, pos=w, neg=g,b
SYNO	WIKIPEDIA, $Q = 2$, pos=w, neg=b
HYPER	WIKIPEDIA, $Q = 10$, pos=w, neg=b

RELATION	Settings to Classify Test Partition, assuming pos=w,g, neg=b
ANTO	WIKIPEDIA, $Q = 50$, pos=w,g, neg=b
SYNO	WIKIPEDIA, $Q = 20$, pos=w,g, neg=b
HYPER	WIKIPEDIA, $Q = 5$, pos=w,g, neg=b

Table 5.16: The system parameter settings of the models chosen to classify the test partition

5.3.3 Results on the Test Partition

We have evaluated models depending on whether they treat the grey instances as either negative (eval: pos=w, neg=g,b) or positive (eval: pos=w,g, neg=b). The models that we use here are the best models as evaluated on the development partition as observed in Section 5.3.1. Table 5.16 summarises the best parameter settings for two of our evaluation methods.

We present the results of using such models to classify instances in the test partition in Tables 5.17 and 5.18, with each table showing the models' accuracy based on the two different ways of evaluating grey instances. We also include the accuracy of baseline systems, which classify a noun pair as having a specific relation iff they appear in one of the hand-crafted patterns in either GIGAWORD or WIKIPEDIA for that specific relation (as listed in Table 4.7).

In general, the results for the best models on the test partition are comparable to their accuracy on the development partition. With the exception of the hypernymy model in Table 5.18, all of our RE models are more accurate compared to the baseline

Models	Evaluated on the Test Partition					
	pos=w, neg=g,b					
	TP	FP	FN	R	P	F
ANTO-best model	2	1	61	0.032	0.667	0.061
ANTO-baseline (GIGAWORD)	9	3,888	54	0.143	0.002	0.004
ANTO-baseline (WIKIPEDIA)	9	2,958	54	0.143	0.003	0.006
SYNO-best model	25	40	1,116	0.022	0.385	0.042
SYNO-baseline (GIGAWORD)	202	123,713	939	0.177	0.002	0.004
SYNO-baseline (WIKIPEDIA)	234	106,778	907	0.205	0.002	0.004
HYPER-best model	606	1,108	7,971	0.071	0.354	0.118
HYPER-baseline (GIGAWORD)	893	10,485	7,684	0.104	0.078	0.089
HYPER-baseline (WIKIPEDIA)	935	8,781	7,642	0.109	0.096	0.102

Table 5.17: Results for the best models trained on the training partition and evaluated on the test partition, treating white instances as positive and grey and black instances as negative in evaluation

systems. Due to how the baseline systems are implemented, it is not surprising to see the large number of positive predictions, although the majority of them are inaccurate as reflected in their low precision scores.

Compared to the baseline systems, our RE models are very conservative in making positive predictions, and suffer from low recall scores. Although they are better than the baseline systems, this experiment confirmed the overall poor results of our RE approach. None of the systems for any of the relations are able to match the level of accuracy shown in our earlier experiments in Chapter 4 and Section 5.1.1.

5.3.4 Error Analysis

In this section, we analyse the errors of our best models using the evaluation method that treats grey instances as negative (eval: pos=w, neg=g,b).

Models	Evaluated on the Test Partition					
	pos=w,g, neg=b					
	TP	FP	FN	R	P	F
ANTO-best model	21	26	115	0.154	0.447	0.229
ANTO-baseline (GIGAWORD)	29	3,848	107	0.213	0.007	0.014
ANTO-baseline (WIKIPEDIA)	28	2,920	108	0.206	0.009	0.017
SYNO-best model	67	304	3,198	0.021	0.181	0.038
SYNO-baseline (GIGAWORD)	800	122,517	2,465	0.245	0.006	0.012
SYNO-baseline (WIKIPEDIA)	913	105,420	2,352	0.280	0.009	0.017
HYPER-best model	1,509	3,969	21,585	0.065	0.275	0.105
HYPER-baseline (GIGAWORD)	1,612	9,047	21,482	0.070	0.151	0.096
HYPER-baseline (WIKIPEDIA)	1,701	7,249	21,393	0.074	0.190	0.107

Table 5.18: Results for the best models trained on the training partition and evaluated on the test partition, treating white and grey instances as positive and black instances as negative in evaluation

False Negatives

First, we look at the false negative predictions. We suspect that one of the main causes of false negative predictions is the small number of features. To investigate this, we look at the co-occurrence frequency of the noun pairs in the corpus that the best model was trained on, as this provides a rough indication of how many features the noun pairs have for classification. We refer to noun pairs that have appeared 10 times or less in the corpus as low-frequency noun pairs, between 11 to 100 times as mid-frequency noun pairs, and more than 100 times as high-frequency noun pairs. Referring back to our initial hypothesis that most false negatives are caused by a small number of features, we expect most false negatives to be lower frequency noun pairs, which correlates with a smaller chance of those noun pairs appearing in one of the patterns indicating a certain relation due to less frequent corpus appearance.

	ANTO	SYNO	HYPER
Low-frequency noun pairs [1,10]	32 (52.5%)	774 (69.4%)	5,971 (74.9%)
Mid-frequency noun pairs[11,100]	16 (26.2%)	295 (26.4%)	1,765 (22.1%)
High-frequency noun pairs [101, ∞)	13 (21.3%)	47 (4.2%)	235 (3.0%)
All false negatives	61 (100%)	1,116 (100%)	7,971 (100%)

Table 5.19: The number (and its proportion to the entire size) of false negatives, based on their corpus frequency

	ANTO	SYNO	HYPER
Low-frequency noun pairs [1,10]	0 (0.0%)	13 (52.0%)	189 (31.18%)
Mid-frequency noun pairs[11,100]	0 (0.0%)	11 (44.0%)	360 (59.41%)
High-frequency noun pairs [101, ∞)	2 (100%)	1 (4.0%)	57 (9.41%)
All true positives	2 (100%)	25 (100%)	606 (100%)

Table 5.20: The number (and its proportion to the entire size) of true positives, based on their corpus frequency

We present the breakdown of the frequency of false negative noun pairs for the three relations in Table 5.19. As we can observe, the majority (more than half) of the false negatives are low-frequency noun pairs. It is also possible to investigate our hypothesis by looking at the breakdown of the frequency of true positives, where we expect that most of the true positives are higher-frequency noun pairs. The information in Table 5.20 supports our hypothesis: more than half of the true positives are higher-frequency noun pairs (with the synonym relation just missing the threshold). The observations on the frequency of both false negatives and true positives go some way to confirming our hypothesis that low-frequency noun pairs are harder to classify than more frequently appearing noun pairs.

In analysing the errors for hypernym false negatives, we observed that the ma-

Noun Pairs (<i>hyponyms</i> , <i>hypernyms</i>)	# Unique <i>hyponyms</i> for the Given <i>hypernym</i>	# Correctly Predicted Pairs
(<i>hyponyms</i> , <i>entity</i>)	427	10 (2.3%)
(<i>hyponyms</i> , <i>event</i>)	314	17 (5.4%)
(<i>hyponyms</i> , <i>object</i>)	258	15 (5.8%)
(<i>hyponyms</i> , <i>city</i>)	210	112 (53.3%)
(<i>hyponyms</i> , <i>group</i>)	207	5 (2.4%)
(<i>hyponyms</i> , <i>person</i>)	204	8 (3.9%)
(<i>hyponyms</i> , <i>location</i>)	119	13 (11%)
(<i>hyponyms</i> , <i>individual</i>)	115	3 (2.6%)
(<i>hyponyms</i> , <i>activity</i>)	98	7 (7.1%)
(<i>hyponyms</i> , <i>writer</i>)	97	7 (7.2%)
(<i>hyponyms</i> , <i>part</i>)	95	1 (1.1%)
(<i>hyponyms</i> , <i>district</i>)	85	11 (12.9%)
(<i>hyponyms</i> , <i>someone</i>)	81	0 (0%)
(<i>hyponyms</i> , <i>knowledge</i>)	76	0 (0%)
(<i>hyponyms</i> , <i>territory</i>)	74	8 (10.8%)

Table 5.21: 15 most frequently-paired hypernyms in the test partition and the success rate of our best hypernym model in predicting such instances

jority of the positive (*hyponyms*, *hypernyms*) noun pairs involve only a small set of *hypernyms*. In other words, this small set of *hypernyms* is often paired with a large number of hyponym candidates.

Curiously, when analysing the false negatives for the test data, we observed that the model has a low success rate in correctly identifying positive noun pairs involving this set of hypernyms. Table 5.21 lists 15 of the most frequently-paired hypernym nouns (as part of positive noun pairs) in the test partition, and the number of positive instances involving those hypernyms that our model correctly predicts to be positive.

From Table 5.21, we can see that the our model has a very low success rate in

classifying the majority of noun pairs whose hypernyms nouns are frequently-paired in the test data. The exceptions to this observation are predictions involving geospatial hypernyms such as *city*, *location*, *district*, and *territory*. This problem is of grave concern as noun pairs with 60 of the most frequently-paired hypernyms made up more than half of the positive noun pairs in the test data, and although not shown here, we observed that our model has an equally low success rate for predicting such pairs. In Chapter 6, we propose to use a minimally-supervised bootstrapping approach (McIntosh and Curran 2008) to tackle this problem.

False Positives

We now turn our attention to the false positive predictions. We first look at the antonymy relation as there is only one false positive prediction. The only false positive prediction is for (*public*, *private*), which we argue is reasonably clear-cut at the word-level. However, our labelling process annotates this pair as black (negative) instance, as WORDNET does not have a relation between the two noun senses of public (*public*₁ “people in general considered as a whole” and *public*₂ “a body of people sharing some common interest”) and the only noun sense of *private*₁ “an enlisted man of the lowest rank in the Army or Marines”. This prediction perhaps is a better fit for the WORDNET adjective senses of both words, where in fact, there is an antonymy relation between the first adjective senses of both words.

There are 40 false positives for synonymy. Out of those 40 false positives, 5 of them ((*formosa*, *taiwan*), (*cookie*, *biscuit*), (*pennsylvania*, *pa*), (*hood*, *bonnet*), (*earth*, *ground*)) are actually grey instances, which at the word-level are arguably clearly-attested synonym pairs. The remaining 35 false positives are a mix of:

1. incorrect pairs, such as (*sun*, *europe*), (*cyclist*, *pedestrian*), (*obscurity*, *security*), (*tissue*, *organ*), and (*no*, *yes*);
2. close or correct pairs but annotated as negative instances, such as (*ecstasy*, *mdma*), (*pony*, *horse*), (*boot*, *trunk*), (*fr*, *france*), and (*emotion*, *feeling*);
3. near-synonym pairs with high lexical association, such as (*report*, *broadcasting*),

(*fat, lipid*), (*spacecraft, satellite*), (*equator, latitude*), and (*force, torque*).

We similarly look at the false positive predictions for hypernym instances. There are 206 instances out of the total 1,108 false positives that are annotated as grey instances. These include 27 predictions of (*X, country*), where *X* is a valid country such as *gambia*, *belarus*, *libya*, and *thailand*. The reason for such instances being grey is that WORDNET considers the hypernym relation to be between the first sense of *X* and the second sense of *country*. Next, we randomly sample 16 out of the 206 predicted instances that are annotated as grey instances and present them below:

- | | |
|---------------------------------|---------------------------------|
| – (<i>carrier, warship</i>) | – (<i>sweden, country</i>) |
| – (<i>condition, illness</i>) | – (<i>clover, plant</i>) |
| – (<i>soil, material</i>) | – (<i>bible, object</i>) |
| – (<i>elijah, prophet</i>) | – (<i>decade, number</i>) |
| – (<i>button, control</i>) | – (<i>plutonium, element</i>) |
| – (<i>japan, region</i>) | – (<i>wheat, food</i>) |
| – (<i>knife, weapon</i>) | – (<i>punjab, region</i>) |
| – (<i>missiouri, region</i>) | – (<i>bass, fish</i>) |

As we can see, the majority of the instances are clear instances of hypernymy at the word-level. Having said that, there are some near-hypernym instances, such as (*condition, illness*), (*button, control*), and (*decade, number*).

Next, we look at the remaining 902 false positives, i.e. false positives that are annotated as negative (black) instances by WORDNET. We randomly sample 30 such noun pairs. Below are 16 out of the 30 noun pairs that are annotated as negative, which we (subjectively) judge to be clear-cut positive instances:

- | | |
|------------------------------------|---|
| – (<i>james thurber, writer</i>) | – (<i>william tell, opera</i>) |
| – (<i>john ford, director</i>) | – (<i>swansea, town</i>) |
| – (<i>john huston, director</i>) | – (<i>livingston, district</i>) |
| – (<i>theseus, hero</i>) | – (<i>milan, capital</i>) |
| – (<i>hp, company</i>) | – (<i>smallpox, infectious disease</i>) |

- | | |
|-------------------------------|---|
| – (<i>galleon, ship</i>) | – (<i>poland, area</i>) |
| – (<i>corn, commodity</i>) | – (<i>accounting, subject</i>) |
| – (<i>caviar, delicacy</i>) | – (<i>optometrist, ophthalmologist</i>) |

And here are the 14 remaining noun pairs that are not as clear-cut:

- | | |
|-------------------------------------|----------------------------------|
| – (<i>packaging, industry</i>) | – (<i>left wing, party</i>) |
| – (<i>institution, company</i>) | – (<i>criminal, activity</i>) |
| – (<i>vegetable, agriculture</i>) | – (<i>origin, root</i>) |
| – (<i>shop, space</i>) | – (<i>ice, technology</i>) |
| – (<i>bookstore, business</i>) | – (<i>breast, cancer</i>) |
| – (<i>sewer, service</i>) | – (<i>cotton, agriculture</i>) |
| – (<i>athlete, people</i>) | – (<i>world, dignity</i>) |

There are some reasons why the clear-cut instances are mislabelled by our relation labelling procedure. First, WORDNET may not have a noun sense referring to one or both predicted nouns. For example, WORDNET does not have a noun sense of *hp* referring to the company *Hewlett-Packard*, or the noun sense of *william tell* referring to the opera of the same subject by the Italian composer Gioachino Rossini.

Next, WORDNET does not have a relation between the predicted instances. For example, WORDNET defines *galleon*, *milan*, and *theseus* using the words *ship*, *capital*, and *hero* respectively. Yet, it does not contain any hypernym relations between these noun pairs (e.g. between (*galleon, ship*) or (*theseus, hero*)).

Considering the quality of our false positives of all three relations (especially synonymy and hypernymy), we argue that they are worthy of manual inspection as there are significant numbers of accurate predictions that do not exist in WORDNET. Given the precision-centric nature of our RE models, the positive predictions are quite selective. While naturally there are some incorrect predictions, the volume is not overwhelmingly large, which suggests that a more thorough careful examination of the false positives may be warranted. Finding accurate false positives that are not present in the current lexical resource is highly valuable for knowledge base curators.

Assuming that all the annotated true negatives are indeed true negatives (i.e. there are no true positive instances marked as true negatives), and taking the existence of accurate noun pairs in the false positive predictions into account, we can argue that the current reported precision is an underestimate. This is because the current evaluation method penalises our RE models for accurate noun pairs not present in the gold-standard, which lowers the precision.

Similarly, we argue that the reported recall is also underestimated due to the number of true positives being underestimated. As with the precision, the evaluation method penalises our models for identifying accurate noun pairs not present in the gold-standard. If the noun pairs exist in the gold-standard in the first place, then the evaluation would have judged such predictions as true positives, which positively impacts on both the recall and precision.

Further, as the gold-standard is annotated independently of the corpus, there is also a possibility that there is no evidence for the positive annotations in the corpus, violating our task assumption that there is always evidence for positive annotations in the corpus. In this scenario, it is not possible for the models to make an accurate prediction for such instances, as the feature sets of those noun pairs are simply not helpful for the models to make positive predictions. This leads to an unfair penalty for the models, and subsequently lowering of the recall.

Discussion

Our RE models have arguably not been very accurate in predicting correct instances for our task. The antonymy model is undoubtedly very poor, as it can only identify two true positives despite the size of the learning corpora. In this case, a manual effort to list all the antonyms is more efficient as a means of capturing antonyms. In comparison to the antonymy model, both the synonymy and hypernymy models do not fare much better. However, they provide better utility in identifying pairs whose relations are clear-cut, but are not present in the current gold-standard (WORDNET).

Our current RE results are very poor in comparison with the results from earlier experiments in Chapter 4 and Section 5.1.1, although strictly speaking, the current

RE task setup is comparably more difficult. Due to the disproportionately large number of negative to positive instances in the data, the current RE task can be thought of as a task to detect rare positive instances amidst the large collection of the negative instances. This situation, akin to finding a needle in a haystack, is commonly referred to as the class imbalance problem (Weiss 2004; Tan *et al.* 2006). While the degree of imbalance between negative and positive instances is subjective, the magnitude of the class imbalance in our current RE task setup (especially with antonymy and synonymy) is comparable (or even greater) than the examples of class imbalance task discussed in the literature (Akbari *et al.* 2004).

Machine learners tend to perform poorly on imbalanced datasets because they are designed to generalise from the sample data, and based on Occam’s razor, output the simplest hypothesis best suited to the data (Akbari *et al.* 2004). Therefore, machine learners tend to predict all instances as the majority class, as they may consider the instances of the minority class as noise in the data due to their small size compared to the majority instances.

The class imbalance problem of our task makes it difficult to draw any hard conclusions as to whether one relation is harder to learn than another. This is due to the (unknown) impact that class imbalance may have on our classifier. If we are to assume that the data has no class imbalance, and just look at the results, we may argue that hypernymy is easier to learn than antonymy and synonymy. This conclusion seems to be consistent with the fact that there is a lack of commonly accepted sets of lexico-syntactic patterns for antonymy and synonymy as compared with the equivalent Hearst’s patterns (Hearst 1992) for hypernymy.

In conclusion, at this point in time, we do not have concrete answers to the poor accuracy of our RE models. There are questions about whether our task setup is too hard (due to the task assumptions and/or class imbalance), or our RE methodology and approach is just not suited to our proposed RE task. To investigate some of these questions, we apply our RE methodology to an alternative task/dataset, in order to compare the capability of our RE methodology by contrasting our models with other published results in a comparable setting. We discuss this experiment in Section 5.4.

5.4 Experiment on Navigli and Velardi (2010)’s Dataset

Currently, we have a number of hypotheses but no hard facts to explain the poor accuracy of our system. As a sanity check, we decide to apply our RE methodology to an alternative dataset where there are published results.

We choose to apply our methodology to Navigli and Velardi (2010)’s dataset. It is designed for a definition extraction task, that is identifying sentences which provide a definition or a formal explanation of a word of interest within a corpus. A property of the definitional sentences in the data is that they always contain a hypernym relation, between the word that is being defined, and the phrase that the word was being defined with. This property is exploited by Navigli and Velardi (2010) to also adapt their data as an avenue for hypernym extraction.

Due to how the dataset is prepared, there are several preprocessing steps required to adapt their data to our RE methodology. In doing so, we attempt to make as little modification to the dataset as possible. However, there are some assumptions in Navigli and Velardi (2010)’s methodology that cannot be directly applied to our RE setup. In this case, we choose to modify our RE methodology to adapt to their RE task setup, rather than the other way around (modifying their data to fit our RE methodology), to preserve comparability between Navigli and Velardi (2010) and our RE methodology.

5.4.1 Data Description and Hypernym Extraction Task

We use the Wikipedia subset of version 1.1 of the data⁴ (Navigli *et al.* 2010), containing 4,619 sentences, which includes 1,772 definitional and 2,847 non-definitional sentences. All sentences are associated with a word of interest (the target word) that is specifically marked, but only definitional sentences provide a definition for the target word.

Following Storrer and Wellinghoff (2006), Navigli and Velardi (2010) identified

⁴Available from <http://lcl.uniroma1.it/wcl/>

the following four fields in a definition sentence (e.g. *In chemistry, a solution is a homogeneous mixture composed of two or more substances*):

- The *definiendum* field (DF), which includes the word being defined (i.e. the definiendum) and its modifiers (e.g. *In chemistry, a solution*)
- The *definitior* field (VF), which includes the verb phrase used to specify the definition (e.g. *is*)
- The *definiens* field (GF), which includes the genus phrase and usually includes the definiendum’s hypernym(s) (e.g. *a homogeneous mixture*)
- The *rest* field (RF), which includes the additional clauses that further specify the difference between the definiendum with respect to its genus (e.g. *composed of two or more substances*)

Navigli *et al.* (2010) manually annotated all 1,772 definitional sentences with the four fields above. Further, the definiendums’ hypernyms were also specifically marked in the data. All sentences in the dataset were also POS-tagged. Table 5.22 provides some examples of sentences found in the dataset (with the POS tags omitted).

Although the data is constructed mainly for the definition extraction task, it is possible to adapt the data for a hypernym extraction task. The hypernym extraction task proposed by Navigli and Velardi (2010) is to identify pairs of definiendum and its hypernyms from all the definitional sentences. The task naively assumes that there is no hypernym pair in non-definitional sentences, although as we will discover later, this assumption is not true. The training process involves the use of all sentences in the corpus, and its set of hypernym predictions is evaluated against the list of all pairs manually identified by the annotators in the data.

5.4.2 Navigli and Velardi (2010)’s Approach

Definition extraction systems tend to rely heavily on lexico-syntactic patterns to identify definitional sentences, in the same fashion as hypernym extraction systems. However, these systems suffer from both low recall and precision, as definitional

Definitional sentences:

[In geometry, a **quadrilateral**]_{DF} [is]_{VF} [a *polygon*]_{GF} [with four sides or edges and four vertices or corners]_{REST}.

[In chemistry, a **solution**]_{DF} [is]_{VF} [a *homogeneous mixture*]_{GF} [composed of two or more substances]_{REST}.

[In particle physics, a **meson**]_{DF} [is]_{VF} [a strongly *interacting boson*]_{GF} [— that is, a hadron with integer spin]_{REST}.

Non-definitional sentences:

A **clitic** attaches to an adjacent word, known as its host.

A person with **monoplegia** can be referred to as monoplegic.

In **Kumher**, 19% of the population is under 6 years of age.

Table 5.22: Examples of sentences and the annotation markings. **Target** terms are in bold face, while their *hypernyms* are in italics

sentences occur in highly variable syntactic structures, and the most frequently-used pattern — *X is a Y* — is very noisy (Navigli and Velardi 2010).

Navigli and Velardi (2010) proposed to learn Word-Class Lattices (WCLs), as an alternative to the more commonly-used lexico-syntactic patterns, for the definition and hypernym extraction tasks. A lattice is a directed acyclic graph whose structure retains the important differences among distinct sequences, but without redundant information (Navigli and Velardi 2010). Navigli and Velardi (2010) exploited the graphical nature of WCLs which offer greater flexibility over the traditional lexico-syntactic patterns to achieve better results in both the definition and hypernym extraction tasks.

There are three steps in constructing WCLs:

1. Constructing star patterns: each training sentence is preprocessed and gener-

alised to what is termed a *star pattern*.

2. Sentence clustering: the star patterns are clustered
3. Word-Class Lattice construction: each sentence cluster is aligned via a greedy algorithm to form a WCL.

Using the definitional sentence *In geometry, a quadrilateral is a polygon with four sides or edges and four vertices or corners* as an example, we describe each of these three steps in detail below.

Constructing Star Patterns

The initial step in constructing WCLs is to convert each sentence into a star pattern. Let F be the set of frequent words found in all training sentences, whose frequency is above a threshold θ .

Next, we introduce the concept of word class. For a given word in a sentence, its word class is either the word itself or its POS-tag, as determined based on whether that given word is in the set of frequent words F . The word class of a word is left unchanged (i.e. the word itself) if it occurs in F , or replaced with its POS-tag if it does not occur in F . The target word is an exception to this rule, where it is replaced by a special marker (i.e. $\langle \text{TARGET} \rangle$). Further, all the words in the rest field (RF) are discarded.

Following these steps, a training sentence is converted into a generalised sentence. Using our example above, we obtain the generalised sentence *In NN, a $\langle \text{TARGET} \rangle$ is a NN*. We then replace any sequence of POS-tags (i.e. words that are not in F) with $*$ to obtain the star pattern of that sentence: *In *, a $\langle \text{TARGET} \rangle$ is a **.

Sentence Clustering

The next step is to cluster the star patterns of the training sentences. This is a straightforward step which involves detecting star patterns that are similar to each other, recording which training sentences share similar star patterns, and putting them into the same cluster.

For example, all the definitional sentences from Table 5.22 can be converted into the same star patterns *In **, *a* $\langle$ TARGET $\rangle$ *is a ** which can then be placed into the same cluster. The cluster also records that the star pattern is produced from these three sentences.

Word-Class Lattice Construction

The final step is to construct a Word-Class Lattice for each sentence cluster. To begin with, for each cluster, we extract the original sentences and convert them into the generalised sentences (not into star patterns).⁵ For the first generalised sentence in the cluster, we create a directed graph, in which the words in the generalised sentence are vertices, and the edges are between two consecutive vertices (i.e. a directed graph whose first word points to the second word, the second word points to the third word, and so on). This serves as the initial word-class lattice for that cluster.

Next, for the remaining sentences in the cluster, we determine the alignment between the current sentence to the initial sentence, based on the word class of the sentences (recall that the word class can either be the word itself or its POS-tag). This alignment process is a greedy algorithm which minimises the number of word class sequences misalignments between the current sentence and the initial sentence (see Navigli and Velardi (2010) for the alignment details). If there are any misalignments, new vertices or edges are added to the initial word-class lattice to reflect them. This process is repeated until we exhaust all the sentences in the cluster, producing a final word-class lattice representing the cluster. Finally, in the final lattice, the vertices where the hypernym words appear are specially marked to be used in the hypernym extraction process.

For example, consider the three definitional sentences in Table 5.22. All three sentences are placed in the same cluster because they share an identical star pattern *In **, *a* $\langle$ TARGET $\rangle$ *is a **. Following the WCL construction steps, we convert the first sentence into the generalised sentence producing *In NN*, *a* $\langle$ TARGET $\rangle$ *is a NN*. Next, the initial word-class lattice is constructed by connecting the consecutive words.

⁵Note that the star patterns are only used for sentence clustering, but not for constructing the final word-class lattice.

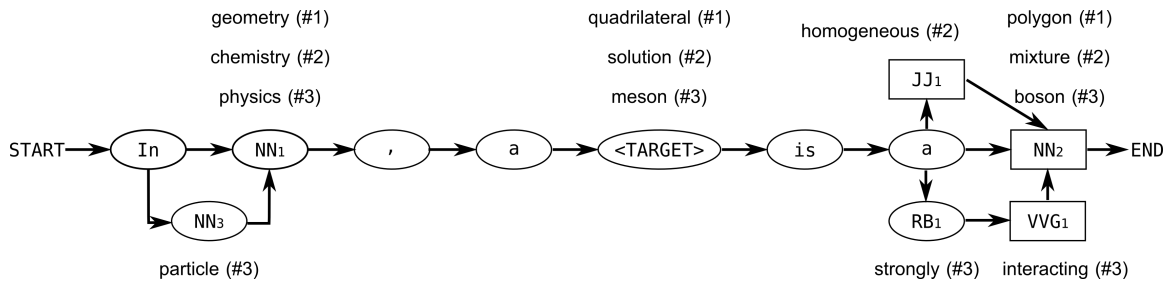


Figure 5.10: The Word-Class Lattice for the three definitional sentences in Table 5.22. Hypernym vertices are indicated with rectangles. The original words (and which sentence they are from) for each word class are also presented

This initial WCL is shown in the middle sequence in Figure 5.10. Also notice that the hypernym vertex $\boxed{\text{NN}_2}$ is presented in a rectangle shape for hypernym extraction purposes.

We next align the second sentence *In chemistry, a solution is a homogeneous mixture ...* to the current lattice. We convert the sentence to the generalised sentence producing *In NN, a <TARGET> is a JJ NN*. Observe that this generalised sentence is almost perfectly aligned with the current lattice, apart from an additional JJ representing the word *homogeneous*. To correct this misalignment, we add in a new vertex $\boxed{\text{JJ}_1}$ and edges connecting the vertex a to $\boxed{\text{JJ}_1}$ and $\boxed{\text{JJ}_1}$ to $\boxed{\text{NN}_2}$ to the word-class lattice. The vertex $\boxed{\text{JJ}_1}$ is in rectangular shape as the word *homogeneous* that it represented is a substring of the tagged hypernyms *homogeneous mixture*.

We apply similar process to the third sentence, resulting in the word-class lattice shown in Figure 5.10.

Variants of the WCL Learning

Navigli and Velardi (2010) proposed two methods for learning WCLs. The first method is to learn the WCL in its entirety, as described above (referred to as the WCL-1 model). The second method is to learn WCLs separately for each of the fields of the definition sentence: definiendum field (DF), definitor field (VF), and definiens field (GF) (recall that the rest field (RF) is not used in learning WCL model). The learning method, however, is practically the same as the WCL-1 model, but only

applied to the part of sentences tagged with each of these three fields. This model is referred to as the WCL-3 model, with the “3” in “WCL-3” referring to the three fields of the definition sentences that the learning method is applied to.⁶

5.4.3 WCLs for Definition and Hypernym Extraction

A set of WCLs is constructed after the learning process. Given a test sentence ts , we can use this set of WCLs to determine whether ts is a definitional sentence or not. For the WCL-1 model, it is a straightforward process to check whether ts fits in one of the lattices.

For the WCL-3 model, the process is not as straightforward. ts is first subjected to all the combinations of WCL for each field. If ts does not fit in any combination, then it is predicted to be non-definitional. If, on the other hand, ts fits in multiple combinations, then the best combination of each field’s WCL is selected based on the following confidence score:

$$score(ts, l_{DF}, l_{VF}, l_{GF}) = coverage \cdot \log(support)$$

where l_{DF} , l_{VF} , l_{GF} are the three WCLs for each definition field, *coverage* is the fraction of words in the input sentence covered by the three lattices, and *support* is the number of training sentences covered by the combination of the three lattices (Navigli and Velardi 2010).

For all predicted definitional sentences, the classifier predicts hypernym pairs by first detecting the words located in the $\langle \text{TARGET} \rangle$ and the hypernym vertices (rectangular-shaped vertices) and using the words in the corresponding location to form *hypernym(words marked as $\langle \text{TARGET} \rangle$, words marked as hypernym)* predictions.

5.4.4 Applying Our RE Methodology to the Navigli and Velardi (2010) Data

We apply our RE methodology to the Wikipedia subset of the Navigli and Velardi (2010) data. There are 4,619 sentences in it, with 1,772 definitional and 2,847 non-

⁶Note that there is no WCL-2 model.

definitional sentences. From here on, we refer to this as the NV data.

There are a few challenges in applying our RE methodology to the NV data. First, the sentences provided in it are annotated with definition field markup and tagged with POS. Since our system takes raw sentences as its input, we need to strip the sentences of information not required by our system.

Second, there is a possible conflict in the training and evaluation processes due to the different tokenisation methods found in the NV data (the tokens are manually identified) and our system (we use **MINIPAR** to identify word tokens). This disconnect may cause problems during training and evaluating our system. If we use tokens produced by **MINIPAR**, our system may not be able to identify some tokens found in the NV data and not able to fully benefit from the hypernym pair (gold-standard) annotation provided. Similarly, **MINIPAR** may produce different sets of tokens for evaluation, making a side-by-side result comparison impossible.

To address this, we force **MINIPAR** to adopt the Navigli and Velardi (2010) tokenisation for the targets and the hypernyms.⁷ By doing so, we are able to train and evaluate our system based on the same sets of hypernym pairs found in the NV data. We successfully identify 1,951 unique hypernym pairs (positive instances) from the 1,772 definitional sentences in which some sentences contain more than one hypernym pair.

Next, we extract features from the resulting **MINIPAR** parse, and construct our feature sets for all the noun pairs in the data (see Section 4.1.5 for details). We then scan the features to get a sense of frequently appearing patterns in the data strongly indicating hypernymy. To our surprise, we are only able to extract features for 21 out of the 1,951 positive instances, providing very little information for the learner. After looking at the data, we find that most of the definitional sentences in the data are of the form **TARGET** *is a/the* **HYPERNYM**. The partial **MINIPAR** output of sentences with this form can be seen in Figure 5.11.

As we can observe in Figure 5.11, there is no direct path connecting the two

⁷We map Navigli and Velardi (2010)’s tokens into unique strings that are tagged as nouns by **MINIPAR**. Then, we substitute these unique strings in place of the tokens in the sentences, and extract features from them. After that, we remap all these unique strings back into their original tokens.

A bat is a mammal in the order of Chiroptera

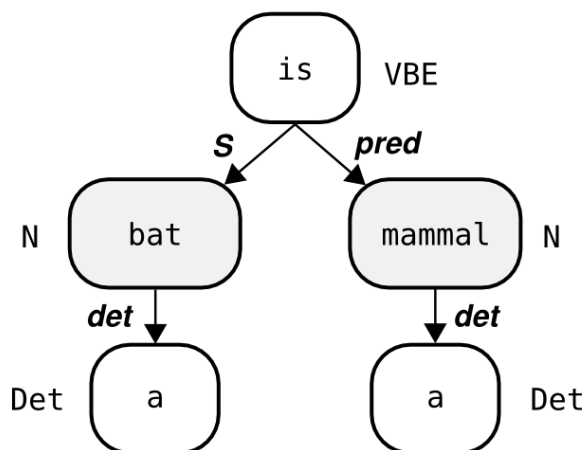


Figure 5.11: A snippet of the MINIPAR output for typical definitional sentences found in the NV data as represented by *A bat is a mammal in the order of Chiroptera*. There is no directed path between the TARGET word *bat* with its hypernym *mammal*

nouns of interest (*bat* and *mammal*). This explains the lack of features for positive instances in the data, since our feature formation process depends on the existence of the directed path between two nouns. In order to capture useful patterns from positive instances, we modify our system to enable extraction of undirected paths, on top of the directed paths (see Section 3.3.1 for details). The reason why we avoid enabling the extraction of undirected paths in previous experiments is due to the large size of our corpora, which contain a large number of undirected paths. After transforming them into the feature sets, our system is unable to load all of the features into main memory. Compared to the corpora used in our earlier experiments, the NV data is relatively small that it is possible to extract features from undirected paths and fit them in main memory.

Having enabled the undirected path extraction, we are now able to extract features for 1,625 out of the 1,951 positive instances. However, we are still unable to extract any features for the remaining 326 positive instances due to the two nouns being linked by more than four edges (recall that we only consider paths between two nouns if they are linked by four edges or less).

Next, we collect negative instances for our models. In this experiment, we consider a noun pair to be a negative instance if we are able to extract features from them (the noun pair is connected by four or less MINIPAR edges), and the noun pair is not one of the 1,951 identified positive instances. Unlike the assumption in the NV data where the negative instances can only be found in the non-definitional sentences, it is possible to extract the negative instances for our experiments from both the definitional and non-definitional sentences. For example, we extracted a negative instance (*geometry, quadrilateral*) from the first definitional sentences of Table 5.22. We collected 64,661 negative instances in the data. And finally, we follow Navigli and Velardi (2010) in training and evaluating our system via 10-fold cross validation.

5.4.5 Result

Next, we compare the precision of our system to Navigli and Velardi (2010)’s two models (WCL-1 and WCL-3). Navigli and Velardi (2010) reported two precision scores for each model: Substring and Full. The Substring precision score refers to a more lenient evaluation process where the model is rewarded for correctly predicting the full string of the hypernym, or failing that, the substring of the hypernym (e.g. only predicting *mixture* out of *homogeneous mixture*). The Full precision score is a stricter process which requires the model to successfully predict the full string of the hypernym.

Our classifier works differently as it is only making a binary decision whether a pre-identified noun pair has a hypernym relation or not. Also, since our system is using the tokenisation manually annotated in the data, the pre-identified noun pairs always include the full strings of the targets’ hypernym. Our task assumption prevents us from identifying the full strings of the hypernym separately, and arguably does not fit in the two evaluation criteria proposed by Navigli and Velardi (2010). In light of this, we list our system result independently from both the Navigli and Velardi (2010)’s evaluation criteria.

We present the macro-averaged precision of Navigli and Velardi (2010)’s WCLs⁸

⁸Navigli and Velardi (2010) did not specify how the precision scores are calculated in their paper.

Models	Full	Substring	Pre-identified word strings
WCL-1	0.428	0.770	?
WCL-3	0.407	0.786	?
Our RE system	?	?	0.673

Table 5.23: Macro-averaged precision scores in hypernym extraction task over the NV data for three different RE models

to our RE system in Table 5.23, with each column reflecting the three different ways of evaluation due to the different assumptions in the RE task.

It is difficult to directly compare our result with Navigli and Velardi (2010)’s results due to the different task assumptions, especially regarding string identification. As we argued previously, our use of the annotated strings is necessary to provide an identical training and evaluation noun pair set, and to maximise the benefit of the gold-standard provided in the NV data to our system. In this sense, the use of Navigli and Velardi (2010)’s tokenisation provides some benefit for our classifiers in identifying hypernym pairs.

On the other hand, there are some factors that disadvantaged our systems. For example, our system is unable to identify features for 326 positive instances. Further, our system is penalised for correctly identifying a few noun pairs which have hypernymy relations (based on a manual inspection) that are not in the gold-standard hypernym pairs due to not occurring in non-definitional sentences, or one of the nouns not being identified as the target word in definitional sentences.⁹

For example, one of our models (out of the 10-folds) extracts *hypernym(d-loc, kingspade)* from the non-definitional sentence *D-loc is a member of Kingspade*. We also identify correct noun pairs such as *(alpenhorn, instrument)* from definitional

⁹We obtained this information via personal communication with the authors.

⁹The reported Navigli and Velardi (2010) precision scores reflect the precision of their models in extracting pre-identified hypernym pairs over the pre-identified definitional sentences. This, however, does not reflect their models’ ability to extract hypernyms over the whole corpus. The evaluation does not take into account the possibility of hypernym pairs occurring in non-definitional sentences, and assumes them to be negative instances.

sentences, but are penalised because *alpenhorn* is not the target word of the sentences, which naturally lowered the precision.¹⁰

We can perhaps look at the reported results for Full and Substring as the lower and upper estimate, respectively, of Navigli and Velardi (2010)’s models in predicting hypernym pairs. Using this as the basis, we argue that our reported precision score is quite competitive. The high precision score reported in this experiment contrasts with our earlier results from Section 5.3.

There are a few possible explanations for this. First, it is worth noting that the corpus used in this experiment is much smaller than the corpora used in Section 5.3. The ratio of the negative to positive instances in this data (64,661 negative instances / 1,951 positive instances = 33.14) is also smaller than the equivalent ratio (more than 100) for the hypernymy learning task in Section 5.3. This is even smaller than the 1:50 ratio that Snow *et al.* (2005) reported. The lower ratio of the negative to positive instances in the data means that there is a lesser chance of the machine learners being affected by class imbalance.

Keep in mind that the NV data is a dataset specifically prepared for a research task. The data composition is perhaps not as reflective of the real-world data as GIGAWORD and WIKIPEDIA. In other words, it is not appropriate to consider the ratio of negative to positive hypernym instances in the NV data as the natural distribution of hypernym instances in the real-world.

Second, the positive instances in the NV data are extracted from definitional sentences. Structurally, the definitional sentences in the NV data are not as varied as the sentences containing hypernym pairs in GIGAWORD and WIKIPEDIA (as mentioned in Section 5.4.4, most of the definitional sentences in the NV data are of the form TARGET *is a/the* HYPERNYM). In turn, these sentences translate to more consistent feature vectors sets for the learner, which may lead to better extraction accuracy.

Finally, we use undirected paths in this experiment which we did not utilise in previous experiments. We have to be content with learning from directed paths only

¹⁰The full sentence is *Alphorn or alpenhorn is a brass instrument, consisting of ...*, where the target is *Alphorn* (not *alpenhorn*) and its hypernym is *instrument*.

in previous experiments as our learner is unable to handle the feature vectors sets (due to the need for our code to load the feature vectors into main memory) produced by both the undirected and directed paths. We believe that undirected paths offer additional helpful information on top of directed paths, as shown in this experiment.

From this result, we hypothesise that our RE methodology is perhaps better suited to a contained smaller-scale task, e.g. the Navigli and Velardi (2010) dataset, or experiments in Chapter 4 where the task setup artificially decides the ratio between negative and positive instances for all the relations for both training and evaluation, rather than using the natural distribution of negative and positive instances found in the data.

This experiment demonstrates that supervised RE systems seem to struggle with class imbalance, which is consistent with our earlier observation in Chapter 4 and Section 5.1.1 that the system accuracy declines as we increase the ratio between negative and positive instances. This, however, does not mean that our RE methodology is not suitable for any RE tasks.

5.5 Summary

This chapter discussed a number of important issues in defining and forming our RE task, which largely involved addressing the problems with the original Snow *et al.* (2005) methodology, particularly with its sampling procedure. To begin with, we looked at the problem of the Snow *et al.* (2005)'s methodology, whereby it did not provide a common dataset for fair and sensible evaluation across various experimental settings, especially across multiple corpora. We came up with a common dataset for cross-corpora evaluation by sampling from only instances that appear in both corpora (intersection-based sampling). By using this common dataset for training and evaluation, it is now possible to evaluate the individual quality of the corpora as a learning resource for RE.

Next, we looked at the issue of determining the number of negative to positive instances for any given relation. In Snow *et al.* (2005), this was determined by manual inspection of the data. As such a process is resource expensive, we proposed that it is

best to have a sampling procedure that can produce a dataset that naturally reflects the distribution. We argued that by randomly sampling a large number of instances (we used 1 million instances) from the common dataset produced earlier, it is possible to get a natural distribution of negative to positive instances without manual data inspection.

Using the 1 million instances, we found that the natural distribution of negative and positive instances is different for the three relations. While this was to be expected, the disproportionate number of negative and positive instances for the three relations proved to be a source of concern, as they are at a much higher scale than what we hypothesised with our earlier experiments with different R values. This was one of the main reasons for the poor accuracy of models trained using Snow *et al.* (2005)’s methodology when applied to the randomly sampled 1 million instances.

Another reason for the poor accuracy of the models trained using Snow *et al.* (2005)’s methodology was the difference in sampling between the training and evaluation datasets. As we believed the 1 million instances to be a better reflection of the real-world task, we modified our training methodology to reflect how the dataset was sampled for evaluation. This modification led to an improvement for our models trained using the modified methodology over their equivalent models trained using Snow *et al.* (2005)’s methodology.

This sampling insight prompted us to do a closer examination of some of the assumptions in our earlier task in Chapter 4. We revisited the relation labelling process, and examined in detail the role of SEMCOR and the heuristics in mapping the word-level nouns found in the corpus to the sense-level WORDNET entries. During this examination, we discovered unrealistic assumptions in our earlier experiment setup, and proposed solutions to develop a more realistic RE task. These solutions largely revolve around enabling a large number of previously excluded nouns to be eligible for sampling, and allowing a higher degree of polysemy in the RE task. While the resulting task setup based on these changes to the relation labelling process is arguably more difficult than the previous setup, it is in line with our research objective to define our RE task to be as close as possible to the real-world situation.

Continuing from the previous effort to define our RE task to better reflect the

real-world situation, we reexamined our training and evaluation processes. The previous instance selection process depends on the number of features the instances have (the P system parameter) and also the ratio between negative and positive instances (the R system parameter). As we have learnt previously from the 1 million instances, it is possible to create a realistic dataset independent of the R parameter. Based on this, we randomly split all noun pairs that are successfully annotated by the relation labelling process into three data partitions, which we used for training, development, and test purposes. This decision rendered our RE task to be independent of both the P and R system parameters.

With these changes, our RE task diverged significantly from the Snow *et al.* (2005) task setup. However, we believe that these changes are necessary to achieve our research objectives in creating an RE task that better reflects the real-world situation.

We experimented with the updated task setup, and found valuable insights into the system parameters, such as the best corpus and features to use for training, and the impact polysemous words may have on RE. The overall results of our models were very disappointing, especially compared to the highly promising results observed in earlier experiments.

In analysing the errors, we identified several factors that explain the models' poor results, and suggest that the results were possibly underestimated. One of the factors was the presence of a large portion of positive instances with sparse features, that are arguably harder to classify. The failure of our models to predict such positive instances was reflected in a very low recall. Next, the evaluation process unfairly penalised the models for accurately predicting a set of instances that are arguably clear-cut for a given relation due to the fact that these instances do not appear in the gold-standard.

We also hypothesised that our learning process may have suffered from class imbalance due to the large disproportionate between negative and positive instances in the data. Class imbalance may have contributed to producing suboptimal relation models, and subsequently poor results. And finally, as the gold-standard is produced independently of the corpus, there is a possibility that the gold-standard annotations were not backed up by evidence in the corpus.

Despite these poor results, we also made some encouraging observations (especially regarding false positives) that suggest our synonymy and hypernymy models are quite selective in predicting positive instances. We suspect that this behaviour will bode well in aiding the knowledge base maintenance task. Further, error analysis of hypernymy false negatives showed that a bootstrapping approach for RE may help to improve our RE system. We pursue this avenue in Chapter 6.

The poor results prompted us to question the credibility of our methodology at RE. To investigate this, we applied our methodology to another hypernymy task which applied a methodology different to ours. This experiment allowed us to compare the results of two methodologies to a common RE task, and to judge the credibility of our methodology. To perform this experiment, we used the dataset from Navigli and Velardi (2010). After some necessary (minimal) modifications to the dataset, we found that our system was competitive with systems trained using a different methodology.

In accordance with this and earlier results in Chapter 4, we concluded that our methodology is perhaps suited to a smaller-scale task which is less likely to suffer from class imbalance and data sparsity, which our system struggles with.

As mentioned earlier, by analysing the false negative hypernym instances, we identified a possible avenue for improving the accuracy of our hypernym model. This avenue concentrates on exploiting semantic category information for hypernymy instances in our data. Chapter 6 further discusses our experiments in exploiting the semantic category information for our hypernymy RE task.

Chapter 6

Semantic Categories for Hypernymy RE

Chapter 5 formalised our RE task, and described the process of building our RE system. However, the system was arguably not very successful at performing the RE task, as evidenced by the disappointing f-score. We were surprised by this result, given that our experiments in Chapter 4 showed promising results.

In analysing the errors of our RE system, we identified a special property of the positive hypernymy instances (i.e. $\text{hypernym}(\text{hyponym}, \text{hypernym})$), namely that there is a small set of frequently-paired *hypernyms* which occur with a large number of different *hyponyms*. In fact, positive instances whose hypernyms are one of the 60 most frequently-paired hypernyms make up more than half of the 8,577 positive noun pairs in the test data. In general, we found that our system has a low success rate in predicting such instances, which hurts its accuracy due to their volume in the test data.

We observed that most of the frequently-paired hypernyms in the data are nouns that are higher up in the WORDNET hierarchy. That is, these nouns are general abstractions of all the more specific nouns below them (their hyponyms) in the WORDNET hierarchy. Due to their location in the WORDNET hierarchy, these hypernyms encapsulate a large number of hyponyms, hence increasing the probability of them being paired with their hyponyms. Some examples of these nouns are *entity*, *object*,

event, and *individual*. The noun *entity* is special, as it is actually the highest node in the WORDNET noun hierarchy. As a result, all WORDNET nouns are hyponyms of *entity*.

We suspect that one of the factors for our poor hypernymy accuracy in classifying such noun pairs is the distance of the WORDNET links between a hyponym and these very abstract hypernyms in WORDNET. As a result, the patterns for such noun pairs are arguably not as clear in the data, as compared to patterns for noun pairs with closer hypernym links. For example, the three most frequently appearing patterns for the noun pair (*cat*, *animal*) in our data are:¹

1. (12) N:punc:U,, — dtd--N:conj:N — N:mod:A,other
2. (10) N:punc:U,and — dtd--N:conj:N — N:mod:A,other
3. (7) — dbu--N:mod:Prep,such as,such as,Prep:pcomp-n:N —

While, the three most frequent patterns for the noun pair (*cat*, *entity*) are:

1. (1) N:det:Det,a — dtd--N:pnmod:A,together,together,A:mod:Prep,into,into,
Prep:pcomp-n:N — N:nn:N
2. (1) N:det:Det,an;;N:mod:A,evil;;N:punc:U,, — dbu--N:appo:N — N:det:Det,a;;N:nn:N
3. (1) N:det:Det,a;;N:mod:A,special;;N:nn:N — dbu--N:rel:C,d_n,d_n,C:i:V,issue,
issue,V:obj1:N — N:det:Det,the

As we can observe, the patterns for (*cat*, *animal*) are examples of high-quality Hearst's patterns (Hearst 1992), and cooccur with the noun pair regularly. In contrast, the patterns for (*cat*, *entity*) are not Hearst's patterns, and arguably are of lower-quality. Also note that they all cooccur with the noun pair just once in the data. This lack of quality patterns for the abstract hypernym pairs, such as (*cat*, *entity*), may explain our system's inability to correctly identify them.

So far, our RE approach has been driven by independent classification of each noun pair in our data. We hypothesise that our RE system may be improved through the

¹The number in parentheses indicates the pair-pattern frequency, and the patterns for (*X*, *Y*) are in the format “satellite to *X* — dependency path between *X* and *Y* — satellite to *Y*”

use of a generalisation method over the hypernymy instances, using its hierarchical nature and semantic property.

In WORDNET, the hypernymy links between a (*hyponymy*, *hypernym*) can be interpreted as *hyponymy* belonging to the semantic category of *hypernym*. For example, given **hypernym**(*cat*, *animal*), **hypernym**(*fish*, *animal*), **hypernym**(*bee*, *animal*), **hypernym**(*snake*, *animal*), and **hypernym**(*buffalo*, *animal*), we can think of *animal* as a semantic category, with *cat*, *fish*, *bee*, *snake*, and *buffalo* as exemplary terms of the semantic category *animal*. In this chapter, we propose to generalise our learning using this semantic category information, in order to improve our RE system’s accuracy.

Experiments in this chapter only concern the hypernymy relation. This is because we did not observe the presence of a small set of the equivalent frequently-paired nouns for both the antonymy and synonymy relations in our data. This may be attributed to the non-hierarchical nature of both these relations. As a result, we suspect that there is little (or no) benefit to be gained from our proposed generalisation method via semantic categories for these non-hierarchical relations.

It is straightforward to automatically learn semantic categories from the training data by simply collecting all the positive hypernymy pairs from the gold-standard. However, we cannot apply this method to the test data, as the gold-standard labelling is withheld in the evaluation. As we intend to generalise our instances in both the training and test data, we require a system that can produce a generalisation to a semantic category for any given noun.

In our experiments, we use **WMEB** (McIntosh and Curran 2008), a bootstrapping-based system capable of extracting instances of semantic categories based on a set of seed terms of those categories and a corpus to collect evidence for bootstrapping. Using **WMEB**’s output, we can then label a semantic category for any given noun. Section 6.1 discusses the **WMEB** architecture, the mutual exclusion assumption used by **WMEB** in reducing semantic drift, and the possible limitations of **WMEB** in handling a large number of semantic categories due to that assumption.

Section 6.2 describes the challenges of applying **WMEB** to predict semantic categories. As we will discover later, there are far more semantic categories found in our data that **WMEB** was originally intended for, which may lead to suboptimal results from

WMEB. To address this, we propose to divide our semantic categories into smaller-sized groups, which are closer to the size of the semantic categories **WMEB** was originally applied to, and run **WMEB** separately over each of these groups. It is not trivial to divide the semantic categories into different groups, as there are certain requirements that need to be met in order to ensure quality outputs from **WMEB**.

We propose three approaches to using the semantic categories extracted by **WMEB** in our RE task. First, we discuss how we use them as the basis for postclassifying predictions made by our hypernymy RE system in Section 6.3. Further, as the quality of **WMEB**'s extractions drops over the subsequent iterations due to semantic drift, we investigate various cut-off points of the ranked **WMEB** predictions in order to optimise our overall RE accuracy.

Second, we examine the possibilities of using **WMEB** semantic categories to generalise over our noun pairs. Section 6.4 describes our proposal to collate separate features of semantically-similar nouns into combined features for these nouns prior to training and evaluation. This is based on our hypothesis that the system is better off learning from a single instance with a high-quality feature set than a large number of instances with low-quality individual features.

We hypothesise that the two approaches above using **WMEB** extractions have a complementary impact on our basic RE system. In Section 6.5, we explore this to form a metaclassifier that combines the individual predictions of these two approaches, confirming our hypothesis.

6.1 **WMEB**

This section discusses the Weighted Mutual Exclusion Bootstrapping (**WMEB**) algorithm, a minimally supervised algorithm for simultaneous extraction of semantic lexicons and their patterns from text (McIntosh and Curran 2008). We start by providing background on automatic bootstrapping approaches developed for information extraction. We look at the motivation for such algorithms, and briefly discuss how the approaches evolved in addressing the challenges of semantic lexicon extraction. Finally, we describe the architecture of **WMEB**, and how we employ **WMEB** in our exper-

iments.

6.1.1 Background

WMEB is a member of the larger set of automatic-minimally supervised bootstrapping approaches, as described in Section 2.3.2. Bootstrapping approaches have been successfully used to automatically construct semantic lexicons based on a small set of seed terms and/or patterns.

The seed terms used for bootstrapping belong to the same target semantic category. The system then finds occurrences of the seed terms in text, and identifies contextual patterns cooccurring with the terms. After selecting high quality patterns (by some ranking or filtering process), patterns are then used to find additional highly-ranked terms. The process continues iteratively using the extracted terms from the previous iteration to find new patterns (and vice versa), until a stopping criteria is met. This approach is based on the hypothesis that terms belonging to the same semantic category appear in similar contexts.

One challenge in bootstrapping approaches is the inevitable *semantic drift* phenomenon, where the system selects ambiguous or erroneous terms in its iterative process (Lita and Carbonell 2004; Curran *et al.* 2007; Xu *et al.* 2009). For example, a system identifying terms relating to *computing devices* may drift into identifying *animals* when the term *mouse* is extracted. One approach to reducing semantic drift is to run the bootstrapping process with multiple semantic categories simultaneously, encouraging terms and patterns of different categories to compete with each other, in order to direct the categories away from each other (McIntosh 2010).

An example of multi-category bootstrapping techniques is Mutual Exclusion Bootstrapping (MEB) (Curran *et al.* 2007). The key assumptions MEB makes are: (i) terms can only belong to a single semantic category; and (ii) that patterns can only extract terms of a single semantic category (terms and patterns of each category are mutually exclusive of each other). The hypothesis in these assumptions is that semantic drift can be reduced by excluding terms and patterns that are extracted by multiple categories within a bootstrapping iteration. In turn, this reduces the

chance of identifying generic patterns that can extract terms for multiple categories, and assigning ambiguous terms to their less frequently appearing sense (McIntosh and Curran 2008). **WMEB** is an extension to **MEB**, which uses an improved weighting scheme for term and pattern ranking and filtering (McIntosh and Curran 2008).

6.1.2 System Architecture

WMEB takes a set of manually labelled seed terms for each target semantic category as input. These terms form the initial semantic lexicon of the corresponding semantic category. The system also requires the setting of three parameters: (i) m , the number of patterns to be added in each iteration; (ii) n , the number of terms to be added in each iteration; and (iii) z , the maximum number of bootstrapping iterations. Next, it proceeds to iterate between two phases: (i) pattern extraction and selection, and (ii) term extraction and selection. The bootstrapping process stops after z iterations. Figure 6.1 shows the overview of **WMEB**'s architecture.

Pattern Extraction and Selection Phase

In this phase, for each category, **WMEB** searches in the corpus for the contexts in which all the terms in the current lexicon appear. For each occurrence in the text, the system forms a pattern constructed from the two tokens either side of the term. The selected candidate patterns for each category are then compared to each other category's patterns. If a pattern is a candidate for multiple categories, it is excluded. The remaining candidate patterns for each category are then ranked according to their *reliability* and *relevance* weight (see page 192).

For each semantic category, **WMEB** builds a cumulative pattern pool. The purpose of this pattern pool is to keep track of all patterns used to identify new terms in previous iterations. McIntosh (2010) noticed that the quality of patterns tends to drop in later iterations, which leads to semantic drift. To delay semantic drift, **WMEB** ranks the candidate patterns in the current iteration, and adds the top- m patterns to the cumulative pattern pool. If the top- m patterns are already in the pool, the next best pattern not present is added, ensuring that at least one new pattern is added

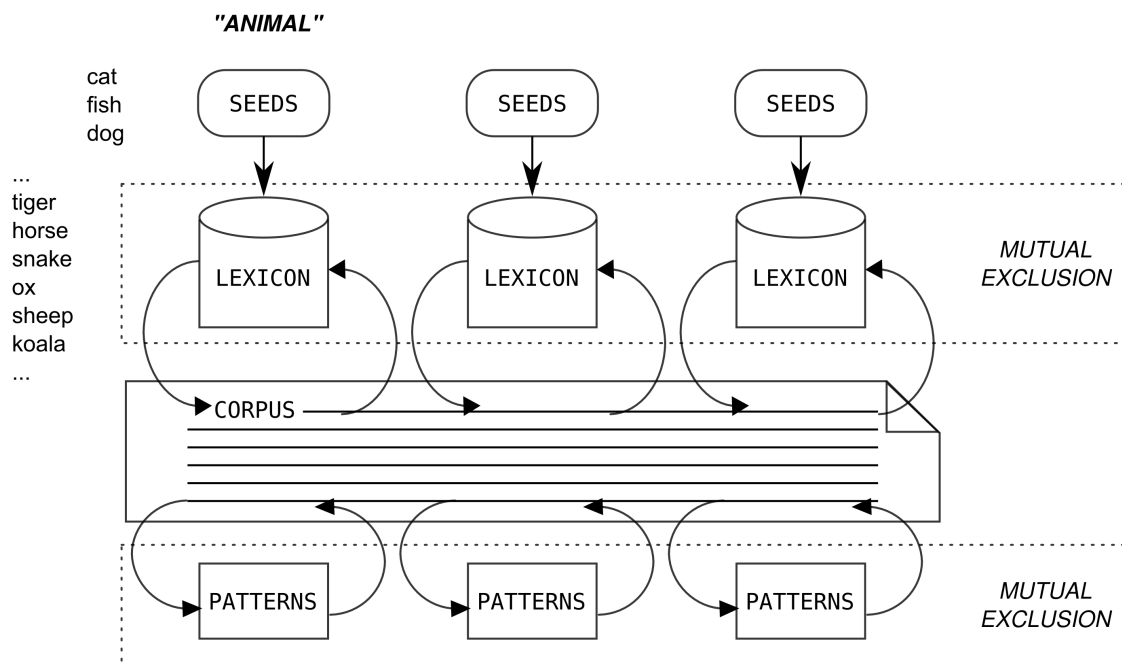


Figure 6.1: Overview of WMEB architecture, showing the expanded ANIMAL semantic lexicon using *cat*, *fish*, and *dog* as the seed terms

into the pool in each iteration.

Term Extraction and Selection Phase

Using the patterns in the cumulative pattern pool, WMEB searches the corpus for new candidate terms for each category. Similar to the patterns, WMEB excludes terms which are extracted by multiple categories in a given iteration. The remaining terms are then ranked according to their *reliability* and *relevance* weight, in which the top- n terms are added to the category's lexicon. The expanded lexicons are then used as the seed terms for the next iteration.

Reliability and Relevance Weight

In WMEB, the *reliability* measure for a term in a particular iteration is defined as the number of patterns that match that term based on corpus evidence. Similarly, the *reliability* measure for a pattern in a particular iteration is defined as the number of

terms that match that pattern in the corpus. Terms or patterns with higher *reliability* scores are ranked higher.

McIntosh (2010) investigated a number of metrics to compute this cooccurrence strength, including three variants of Point-wise Mutual Information (PMI) and the chi-square distribution (χ^2). We choose to use the chi-square distribution in our experiments, as McIntosh (2010) showed it to be superior. The chi-square value for a term w and a pattern p can be calculated by:

$$\chi^2 = \frac{N(F_{wp}F_{\bar{w}\bar{p}} - F_{w\bar{p}}F_{\bar{w}p})^2}{(F_{wp} + F_{w\bar{p}})(F_{wp} + F_{\bar{w}p})(F_{w\bar{p}} + F_{\bar{w}p})(F_{\bar{w}\bar{p}} + F_{\bar{w}p})} \quad (6.1)$$

where:

- F_{wp} : the frequency of term w and pattern p cooccurring in the data
- $F_{w\bar{p}}$: the term w frequency occurring in the data without the pattern p
- $F_{\bar{w}p}$: the pattern p frequency occurring in the data without the term w
- $F_{\bar{w}\bar{p}}$: the frequency of neither the term w and pattern p occurring in the data
- N : the total number of term and pattern pairs in the data.

WMEB automatically excludes terms and patterns with less than 5 appearances, as the χ^2 distribution is sensitive to low frequency counts.

The *relevance* weight of a term w , in a given iteration, is defined as the sum of its chi-square values with each pattern p in the cumulative pattern pool P in that iteration. Similarly, the *relevance* weight of a pattern p in a given iteration is defined as the sum of its chi-square values with each term w in the current lexicon W .

$$relevance(w) = \sum_{p \in P} \chi^2(w, p) \quad (6.2)$$

$$relevance(p) = \sum_{t \in W} \chi^2(p, w) \quad (6.3)$$

WMEB orders both the candidate terms and patterns using their *reliability* scores, and uses the *relevance* weight to break any ties for both terms and patterns. Terms and patterns with higher *relevance* weights are ranked higher.

6.1.3 Stop Categories

WMEB multi-category bootstrapping and mutual exclusion assumption can be exploited further to improve its accuracy by introducing stop categories (McIntosh 2010). Stop categories are categories that are of no interest to the user, but are placed in the system to prevent known semantic drift of categories of interest. For example, McIntosh (2010) noticed that the semantic drift in the categories of CELL, DISEASE, and SYMPTOM tend to start with the identification of terms or patterns relating to BODY PARTS. To prevent this, the BODY PART category is introduced to prevent body part-related terms infecting the CELL, DISEASE, and SYMPTOM categories.

6.2 Using WMEB to Induce Semantic Lexicons

6.2.1 WMEB as a Blackbox

In our experiments, we treat WMEB as a blackbox system: we use the system as it is, without any modification. We use the recommended settings by the authors: extracting a maximum of 5 terms and patterns per iteration (i.e. $m = 5, n = 5$), and use the chi-square distribution to calculate the *relevance* weight.

WMEB was developed using three corpora: the GOOGLE WEB 1T corpus (Brants and Franz 2006), MEDLINE abstracts (NLM 2009), and the TREC GENOMICS TRACK 2007 full-text articles (Hersh *et al.* 2007). We select the GOOGLE WEB 1T corpus as the input corpus for WMEB in our experiments, as it is the closest in nature to our corpora. We refer the reader to McIntosh and Curran (2008) for further information on the GOOGLE WEB 1T corpus.

6.2.2 Forming and Organising Semantic Categories for WMEB

To form semantic categories for our experiments, we first scan for positive (white instances only) hypernym instances in our training partition. As WMEB is unable to use terms that do not appear in its input corpus during bootstrapping, we filter out

pairs whose nouns do not appear in the GOOGLE WEB 1T corpus.

In the remaining positive hypernym instances in our data, (*hypernym(hyponyms, hypernoms)*), there is a large number of hypernoms paired with only a few hyponyms, and conversely, a small number of hypernoms paired with a large number of hyponyms. To illustrate this, there are 1,463 different hypernoms after the initial scanning and filtering. However, there are only 99 hypernoms that appear with at least 10 different hyponyms. The top-5 hypernoms paired with the most hyponyms in our data are *entity* (389), *event* (277), *object* (257), *person* (202), and *city* (175).

During bootstrapping, WMEB requires a healthy number of seed terms to represent a semantic category (McIntosh (2010) used 5 seed terms per category). In our experiments, we consider only hypernoms that appear with at least 10 different hyponyms as semantic categories. As mentioned above, there are 99 such nouns, as listed in Table 6.1.

In her experiments, McIntosh (2010) used around 10 categories per bootstrapping run, as compared to our 99 categories. While WMEB requires multiple categories to perform effectively, we suspect that it may not perform well if it is given a large number of categories. This is due to the combination of its multiple category bootstrapping and mutual exclusivity assumption as will be explained below.

In extracting a large number of categories, we can expect an increase in overlap between terms and patterns identified for all the categories involved. Because of the mutual exclusion assumption, the patterns identified for a certain category may be too specific and not be able to identify new terms on the next iteration. As a result, WMEB may only output a small-sized lexicon.

Related to this, there is a high possibility that we have more semantically-close categories as the number of categories increases. For example, we have a number of categories that are semantically close to each other, such as ANIMAL, BEAST, BIRD, DOG, CREATURE, and INSECT, which are likely to compete for similar patterns and terms in WMEB, and force WMEB to exclude the patterns and terms extracted by multiple categories (due to its mutual exclusivity assumption). This potentially leads to low-coverage extraction results.

We propose to solve this problem by dividing all our semantic categories into

ACTION	ACTIVITY	ANIMAL	AREA	ARTIFACT
ARTIST	AUTHOR	BEAST	BELIEF	BIRD
BREAD	BUILDING	CENTER	CHEMICAL	CITY
CLOTHING	COMPOSER	CONCEPT	CONDITION	CONTAINER
COST	CREATURE	CRIMINAL	DEITY	DEVICE
DISEASE	DISTRICT	DOCUMENT	DOG	DRUG
ENTITY	EQUIPMENT	EVENT	FEELING	FISH
FOOD	FURNITURE	GARMENT	GROUP	GROUPING
HORMONE	IDEA	INDIVIDUAL	INSTRUMENT	ISLAND
KNOWLEDGE	LANGUAGE	LEADER	LOCATION	MAMMAL
MATERIAL	METAL	MIDDLE	MUSIC	MUSICIAN
OBJECT	OCCURRENCE	ORGANISM	ORGANIZATION	PAINTER
PART	PEOPLE	PERFORMER	PERIOD	PERSON
PHILOSOPHER	PLAYER	POET	PORT	PORTION
PROFESSIONAL	PUBLICATION	QUALITY	QUANTITY	REGION
RELATION	RELATIVE	RIVER	ROOM	SCIENCE
SCIENTIST	SOMEBODY	SOMEONE	SPORT	STATE
STATEMENT	STATUS	STRUCTURE	SUBSTANCE	TERRITORY
THOUGHT	TOWN	TRANSPORT	UNIT	VEHICLE
WOMAN	WORK	WORKER	WRITER	

Table 6.1: List of SEMANTIC CATEGORIES in our experiments. Each category appeared with at least 10 different hyponyms in the data

smaller groups, and apply WMEB separately on these smaller groups of categories. To avoid having semantically-close categories in the same runs, we first measure semantic similarity between all pairs of the semantic categories, and place semantically-close categories into different runs.

Measuring Semantic Similarity Between Semantic Categories

To measure the semantic similarity between the categories, we use Resnik (1995)'s WORDNET semantic similarity measure as implemented in version 2.05 of the Perl module `WordNet::Similarity::res` (Pedersen *et al.* 2004).² Resnik (1995)'s approach is based on the *information content* of the terms, where two terms are more similar to each other if they share more information (Resnik 1995). This can be measured by the information content of the terms that subsume them in the taxonomy (WORDNET is our taxonomy in this experiment).

Formally, for any term t in the taxonomy, let $P(t)$ be the probability of encountering an instance of t in that taxonomy. Using the standard definition from information theory, the information content of t , is $-\log P(t)$ (Resnik 1995). Therefore, the similarity of two terms t_1 and t_2 , which share the common subsumer terms T , can be defined as:

$$\text{sim}(t_1, t_2) = \max_{t \in T} [-\log P(t)] \quad (6.4)$$

Resnik (1995) proposed to estimate the probability of a term, $P(t)$ using the BROWN CORPUS (Francis and Kučera 1982). An appearance of the term t as a noun in the corpus is counted as an occurrence of term t and every term that subsumes t in the taxonomy. Formally,

$$P(t) = \frac{\sum_{s \in S(t)} \text{count}(s)}{N} \quad (6.5)$$

where $S(t)$ is the set of terms in the corpus that are subsumed by the term t in taxonomy, and N is the number of noun tokens in the BROWN CORPUS that are present in WORDNET.

The value of P is monotonic as one moves up the taxonomy. That is, if t_1 is subsumed by t_2 , then $P(t_1) \leq P(t_2)$. Therefore, the higher the position of a term subsuming the two terms, the lower the similarity of the two terms based on Equation (6.4) (Budanitsky and Hirst 2006). Applying this logic, the similarity between the

²Available from <http://search.cpan.org/~tpederse/WordNet-Similarity-2.05/lib/WordNet/Similarity/res.pm>

two terms in Equation (6.4) is maximised for the least common subsumer ($lcs(t_1, t_2)$) of the two terms (Resnik 1995). Therefore, we can redefine the similarity equation to:

$$\text{sim}(t_1, t_2) = -\log P(lcs(t_1, t_2)) \quad (6.6)$$

For polysemous terms, Resnik (1995) suggested calculating the similarity between all senses of the terms of interest. The maximum similarity score of all sense-pairings of the two terms is then considered to be their similarity score. In our experiments, we heuristically map the categories to their first WORDNET senses in calculating their semantic similarities, which is akin to our relation labelling heuristic in Section 5.2.1.

The value for Resnik (1995)’s semantic similarity is always greater than or equal to zero, with a value of zero indicating there is no semantic similarity between the two terms. The greater the semantic similarity value, the closer the terms are semantically.

As mentioned earlier, our objective in using this semantic similarity measure is to automatically find semantically-close categories and avoid placing them in the same WMEB run. Using Resnik (1995)’s measure, we observed that category pairs with a semantic similarity value greater than or equal to 1.9 are semantically close and should be placed in different WMEB runs. Table 6.2 presents the groups of semantically-close categories in our experiment whose pairwise Resnik (1995)’s semantic similarity value is greater than or equal to 1.9.

From Table 6.2, we can see that the majority of the groupings are generally agreeable. There are a few questionable groupings such as {DOCUMENT, LANGUAGE, MUSIC, STATEMENT}, which can possibly be hand-corrected. However, in the interest of automating this process as much as possible, we leave the groupings produced by our heuristic in using Resnik (1995)’s measure untouched. Also note a few groups with only one member, such as ENTITY or OBJECT, which were deemed not to be similar to the rest of the categories.

Selecting Competing Categories

Finally, we need to arrange these categories into multiple runs for WMEB by avoiding the placement of semantically-close categories into the same run. We first concatenate

Size	Group Members
21	{ARTIST, AUTHOR, COMPOSER, CRIMINAL, INDIVIDUAL, LEADER, MUSICIAN, PAINTER, PERFORMER, PERSON, PHILOSOPHER, PLAYER, POET, PROFESSIONAL, RELATIVE, SCIENTIST, SOMEBODY, SOMEONE, WOMAN, WORKER, WRITER}
14	{ARTIFACT, BUILDING, CLOTHING, CONTAINER, DEVICE, EQUIPMENT, FURNITURE, GARMENT, INSTRUMENT, PUBLICATION, ROOM, STRUCTURE, TRANSPORT, VEHICLE}
12	{BREAD, CHEMICAL, COST, DRUG, FOOD, HORMONE, MATERIAL, METAL, PART, PORTION, RELATION, SUBSTANCE}
11	{AREA, CENTER, CITY, DISTRICT, LOCATION, MIDDLE, PORT, REGION, STATE, TERRITORY, TOWN}
7	{ANIMAL, BEAST, BIRD, CREATURE, DOG, FISH, MAMMAL}
7	{BELIEF, CONCEPT, DEITY, IDEA, KNOWLEDGE, SCIENCE, THOUGHT}
6	{ACTION, ACTIVITY, EVENT, OCCURRENCE, SPORT, WORK}
5	{CONDITION, DISEASE, FEELING, QUALITY, STATUS}
4	{DOCUMENT, LANGUAGE, MUSIC, STATEMENT}
4	{GROUP, GROUPING, ORGANIZATION, PEOPLE}
3	{PERIOD, QUANTITY, UNIT}
1	{ENTITY}
1	{ISLAND}
1	{OBJECT}
1	{ORGANISM}
1	{RIVER}

Table 6.2: Groups (and their size) of semantically-close categories in our experiment

all members of the groupings based on their appearance order in Table 6.2 into one list. Next, we create as many buckets as the size of the largest groups of semantically-

close categories (21 in this case). Iteratively, we place one category at a time in each of these buckets with categories from the concatenated list. After filling the last bucket (the 21st bucket in this case), we go back to the first bucket and repeat the process until all the categories in the list are assigned. This wrap-around process ensures that all semantically-close categories end up in different category buckets. Finally, we run **WMEB** as many times as there are buckets, with each run focusing only on the categories found in the current bucket. Table 6.3 shows an example of category assignment to a run after this process.

Unlike McIntosh (2010), we do not use stop categories. Although the use of stop categories may help prevent semantic drift during bootstrapping, there is no simple way of analysing which categories to use as stop categories from our data.

In conclusion, our heuristic successfully divides the large number of categories into different **WMEB** runs, adhering to the constraint that semantically-close categories are placed in different runs. As a result, we have divided the 99 categories with a minimum of 10 hyponyms into 21 different runs, with each run focusing on 4 or 5 categories. These numbers of categories per run are close to those used in McIntosh (2010).

Choosing Seed Terms

WMEB requires seed terms representing semantic categories as its input. Adhering to our holdout-method sampling, we seek to train **WMEB** with terms from the training partition. Recall that for **WMEB** purposes, we treat the hypernyms of our positive noun pairs as the semantic categories, and their hyponyms as the seed terms.

There are no guidelines on how to select seed terms to produce the best bootstrapping results (McIntosh and Curran 2009). Bootstrapping systems often rely on the intuitions of domain experts to determine seeds (Curran *et al.* 2007; Thelen and Riloff 2002). Furthermore, it has been argued that different seed terms may produce different extraction results (McIntosh and Curran 2009).

We apply the most common guidelines in picking seed terms. First, we prefer unambiguous seed terms to prevent the early onset of semantic drift. For this reason,

Run	Semantic Categories
1	ARTIST, ARTIFACT, METAL, FISH, LANGUAGE
2	AUTHOR, BUILDING, PART, MAMMAL, MUSIC
3	COMPOSER, CLOTHING, PORTION, BELIEF, STATEMENT
4	CRIMINAL, CONTAINER, RELATION, CONCEPT, GROUP
5	INDIVIDUAL, DEVICE, SUBSTANCE, DEITY, GROUPING
6	LEADER, EQUIPMENT, AREA, IDEA, ORGANIZATION
7	MUSICIAN, FURNITURE, CENTER, KNOWLEDGE, PEOPLE
8	PAINTER, GARMENT, CITY, SCIENCE, PERIOD
9	PERFORMER, INSTRUMENT, DISTRICT, THOUGHT, QUANTITY
10	PERSON, PUBLICATION, LOCATION, ACTION, UNIT
11	PHILOSOPHER, ROOM, MIDDLE, ACTIVITY, RIVER
12	PLAYER, STRUCTURE, PORT, EVENT, ISLAND
13	POET, TRANSPORT, REGION, OCCURRENCE, ENTITY
14	PROFESSIONAL, VEHICLE, STATE, SPORT, OBJECT
15	RELATIVE, BREAD, TERRITORY, WORK, ORGANISM
16	SCIENTIST, CHEMICAL, TOWN, CONDITION
17	SOMEBODY, COST, ANIMAL, DISEASE
18	SOMEONE, DRUG, BEAST, FEELING
19	WOMAN, FOOD, BIRD, QUALITY
20	WORKER, HORMONE, CREATURE, STATUS
21	WRITER, MATERIAL, DOG, DOCUMENT

Table 6.3: Example arrangement of semantic categories for multiple WMEB runs

we exclude grey instances as they are not dominant senses, and more likely to cause semantic drift. We prefer candidate seed terms which appear frequently in the *Google Web 1T* corpus. Furthermore, for categories such as *entity* and *object* with a large number of member terms, we choose to have a group of diverse seed terms, reflecting the general properties of the categories. Finally, WMEB requires mutual exclusivity

between the seed terms of different categories, meaning that we have to ensure that the seed terms chosen are exclusive to one category in each **WMEB** run. Naturally, in choosing the seed terms, we have to balance up the above desiderata. As a result, the seed terms chosen are very subjective and prone to bias.

We use only 10 seed terms for each category. Should there be categories that ended up with less than 10 seed terms because of our requirements above, we use all of the remaining seed terms as seeds. The seed terms chosen for each category are shown in Appendix B.

6.2.3 Results

Having arranged our semantic categories into multiple runs and chosen the seeds for each category, we are now able to effectively use **WMEB**. As mentioned earlier in Section 6.2.1, we set **WMEB** to extract a maximum of 5 terms and patterns per iteration ($m = 5, n = 5$), and use the chi-square distribution as the measure for selecting terms and patterns in each iteration. Also, we specify that the bootstrapping process is to be iterated 2,000 times. Recall that **WMEB** may extract between 0 to m terms per iteration, depending on the terms' relevance weights. As a result, **WMEB** may not extract an equal number of terms for our 99 categories.

After the iterations, **WMEB** produces a lexicon of extracted terms per category. Recall that the iterations were run over the **GOOGLE WEB 1T** corpus, and subsequently **WMEB** may extract words which appear in the **GOOGLE WEB 1T** corpus, but not in our evaluation partition (i.e. development or test partition, depending on the experiment). As our gold-standard data does not have any annotation for such words, we remove them from the extracted category lexicon. After this filtering, we inspect the size of the category lexicons, and find that **WMEB** extracted between 200 and over 4,000 new terms per category. Table 6.4 provides a listing of the number of categories relative to the size of their **WMEB**-induced lexicons, assuming that we use the development partition as a basis for filtering.³

As an estimate of **WMEB** accuracy and a baseline for our upcoming experiments,

³We observe a similar distribution when filtering using the test partition.

# Categories	Size of their WMEB-induced lexicon
6	≤ 500 new terms
20	501 to 1,000 new terms
40	1,001 to 1,500 new terms
14	1,501 to 2,000 new terms
4	2,001 to 2,500 new terms
7	2,501 to 3,000 new terms
2	3,001 to 3,500 new terms
5	3,501 to 4,000 new terms
1	$\geq 4,001$ new terms

Table 6.4: The number of categories relative to the size of their WMEB-induced lexicons

we evaluate the quality of extracted lexicons by classifying instances found in the lexicons as positive instances, and any other instances as negative instances (HYPER-WMEB-BASELINE). Considering only the black, grey, and white instances, and assuming white instances as positive, and grey and black instances as negative (**pos=w**, **neg=g,b**), the baseline system predicted 20,020 white, 6,511 grey, and 142,856 black instances to be positive. This translates to a precision of 0.118.

There are 8,577 white instances in our test partition, of which 1,806 are correctly predicted by the lexicons. This translates to a recall of 0.210. The baseline system (HYPER-WMEB-BASELINE), therefore, has an f-score of 0.151 which is higher than our basic hypernymy model on the test partition from Section 5.3 (R: 0.071; P: 0.354; F: 0.118).

After establishing the baseline, we proceed to exploit these expanded lexicons as a resource to improve our hypernym RE accuracy via two approaches: as the basis of postclassifying the predicted results (Section 6.3) and preprocessing the instances prior to the learning process (Section 6.4). Finally, we investigate the benefit of combining these two approaches via metaclassification to further improve our RE system accuracy (Section 6.5).

6.3 Postclassification with WMEB

This section discusses our first approach of using WMEB-induced lexicons in our RE task and directly builds on top of the results of our hypernymy RE system in Section 5.3. Given the precise nature of WMEB and the low-success rate of our system in predicting instances involved with our semantic categories, we propose to use WMEB-induced lexicons for each semantic category as a positive prediction between the extracted nouns and the semantic category it belongs in. In doing so, we look to boost the number of correct positive predictions over these semantic categories which leads to higher recall.

6.3.1 Methodology

Let SC be the set of semantic categories defined in our data (as described in Section 6.2). Using the original classification as produced in Section 5.3, we override the negative prediction of an instance (X, Y) to a positive prediction, iff Y is one of our semantic categories ($Y \in SC$), and X is in the WMEB lexicon for the semantic category Y . Instances whose Y s are not one of the predefined semantic categories are not affected by this “postclassification” process as we do not have any semantic category information for Y .

Figure 6.2 provides an example of this postclassification process. Consider the two categories, ANIMAL and MAMMAL, and their WMEB-induced lexicons.⁴ We override the original negative predictions of $(snake, animal)$ and $(cat, mammal)$ into positive predictions because *snake* is in the ANIMAL lexicon, and similarly, *cat* is in the MAMMAL lexicon. We do not override the negative prediction for $(parrot, animal)$ because *parrot* is not in the ANIMAL lexicon.

While technically we consider all terms in the lexicon for a given semantic category as a positive hypernymy prediction between the terms and the semantic category

⁴Reader may note that *cat* is found in both the ANIMAL and MAMMAL categories, therefore violating WMEB’s mutual exclusion assumption. While this is correct, the mutual exclusion assumption can only be applied to categories within the same run, but cannot be ensured over different runs. Hence, it is possible to have terms identified in multiple categories in our experiment (provided that the categories are not in the same WMEB run).

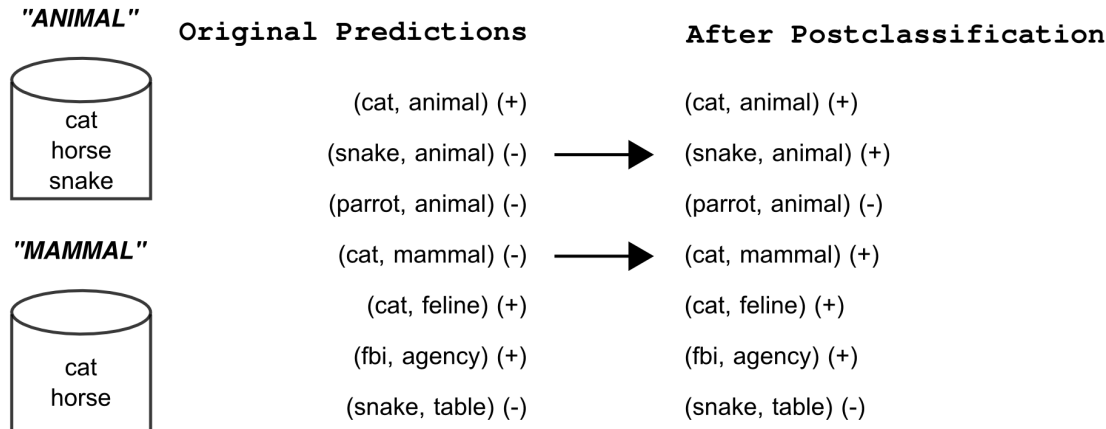


Figure 6.2: An example of postclassification, highlighting the overridden predictions for *(snake, animal)* and *(cat, mammal)*

(i.e. $\text{hypernym}(\text{term}, \text{semantic category})$), not all such possible pairings appear in our evaluation partition (e.g. the possibility of pairings *(horse, animal)* and *(horse, mammal)* in Figure 6.2). We exclude such possible pairings that do not appear in the evaluation partition from our reported results.

There is also a trade-off in using the WMEB lexicons. To boost recall, it is preferable to incorporate as many induced terms as possible into the experiment. However, the quality of the extracted terms in later iterations tends not to be as good as earlier iterations due to semantic drift. Consequently, this may lead to inaccurate predictions, and lower the precision. To address this, we propose to consider only the first- X extracted terms (in increments of 100 terms) from each semantic category lexicon for our postclassification results.

Table 6.4 shows that not all of the lexicons are of the same size, with the largest lexicon (for ARTIFACT) being just over 4,500 terms. We increment X to 4,600, noting that when the X value is larger than the size of a given lexicon, we simply use all terms in the lexicon.

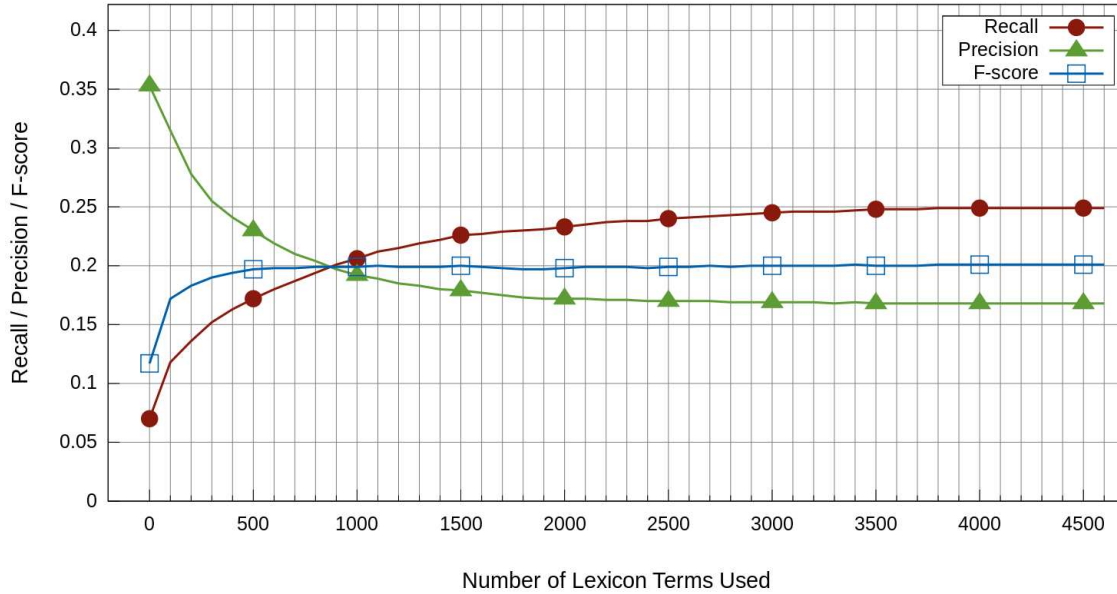


Figure 6.3: The recall, precision, and f-score of model trained on WIKIPEDIA, $\text{pos}=\text{w}$, $\text{neg}=\text{b}$, $Q = 10$, on the development partition after postclassifying using first- X extracted terms of WMEB lexicons

6.3.2 Experiment

For this experiment, we aim to improve the results from our best hypernymy model on the development partition from Section 5.3. It was trained on WIKIPEDIA, $\text{pos}=\text{w}$, $\text{neg}=\text{b}$ (grey instances are not considered), and with $Q = 10$. We first apply this model on our development partition to observe the trade-off between precision and recall for different lexicon size cut-offs. The class setup for evaluation used in this experiment is the same as assumed during our error analysis in Section 5.3.4: $\text{eval: pos}=\text{w}$, $\text{neg}=\text{g,b}$.

Figure 6.3 shows the recall, precision, and f-score, on the development partition of our best hypernymy model after post-classification using the first- X extracted terms of each semantic category’s lexicons. The original result from Section 5.3 is shown where we do not perform post-classification, i.e. $X = 0$. Confirming our initial hypothesis, postclassification using WMEB lexicons improves our system’s f-score, as a consequence of improved recall (at the expense of precision).

Note that before the recall and precision lines crossover between $X = 800$ and $X = 900$, the system's recall improves drastically while the precision undergoes a rapid drop. However, after the crossover, the changes in both recall and precision are considerably less drastic. The minimal change in recall and precision after $X = 2,000$ (relative to the earlier change when $X < 2,000$), can be explained by the small number of categories whose lexicon size are larger than 2,000 (see Table 6.4). The graph also shows that the f-score converges quickly to the 0.2 mark, and undergoes little change after that, which suggests that this value is the upper bound of the system's performance using this postclassification approach.

The f-score line in Figure 6.3 indicates that the system does not suffer from inaccurate predictions when using terms extracted during later iterations. This suggests that using all the extracted terms from WMEB for postclassification is preferable. On the other hand, there is little change in f-score after it converges to around 0.2, suggesting that the optimal accuracy can be achieved by using less terms from the lexicons (e.g. $X = 800$) at the precision/recall crossover point. This also suggests that terms extracted during later iterations actually decrease the system's precision, although the resulting improvement in recall is enough to avoid lowering the overall f-score.

Given these two options, we prefer to extract a smaller number of more accurate terms (precision-oriented), rather than to extract a larger number of less accurate terms (recall-oriented). Part of this reason is that a lot of inaccurate terms (mostly extracted during later iterations) are excluded in the current experiment because there are no such pairings between the terms and the category they belong to in our evaluation partition. Figure 6.4 shows the normalised proportion of the five largest WMEB-induced lexicons found in our evaluation data. As we can observe, the graph shows a downward trend as we include more terms extracted during later iterations, suggesting that these terms are not found as often as the terms extracted during early iterations in the evaluation data. Because we cannot use such pairs for evaluation, our system is not penalised for the possible inaccurate predictions, which arguably are more likely to be terms extracted during later iterations.

Considering the factors above, we decided to use the number of terms at the

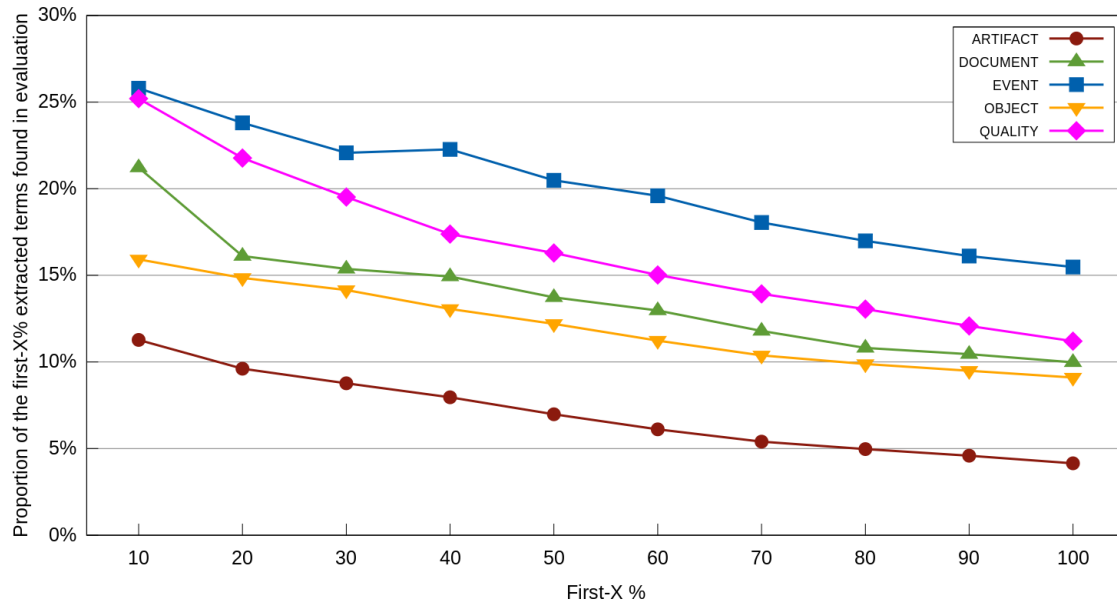


Figure 6.4: Normalised proportion of first- $X\%$ extracted terms of the five largest WMEB-induced semantic lexicons found in our evaluation data

Models	Evaluated on the Test Partition					
	pos=w, neg=g,b					
	TP	FP	FN	R	P	F
HYPER-best basic model	606	1,108	7,971	0.071	0.354	0.118
HYPER-WMEB-POST	1,708	6,914	6,869	0.199	0.198	0.198

Table 6.5: Postclassification improvement to the basic hypernymy model using WMEB semantic category lexicons

precision/recall crossover point (i.e. $X = 800$) as the cut-off point for the number of WMEB terms to be included in our current and future experiments involving the use of WMEB lexicons. We then apply this cut-off point for evaluating our hypernymy model over the test partition. Table 6.5 shows the post-classification improvement over our basic RE system.

Table 6.5 shows that the f-score improvement is largely due to the improvement in

recall outweighing the decline in precision. With post-classification, the system managed to predict 1,102 more true positives compared to our basic system. Considering we have 99 semantic categories, this averages to slightly more than 11 correct predictions per semantic category. At the same time, our false positives have increased by a factor of 6 (1,108 to 6,914 instances). This averages to slightly more than 58 inaccurate predictions per semantic category. As we use 800 terms per semantic category, we can deduce that the majority of the terms (more than 700) in the lexicon are excluded from the evaluation.

We now look at the success rate in classifying frequent semantic categories in the data as compared to the basic system, in order to investigate the individual improvement of each semantic category after postclassification. As a comparison, we use the 15 most frequently appearing semantic categories, as listed in Table 5.21.

Table 6.6 illustrates the improvement in our system's success rate due to the postclassification methodology. The individual improvement varies for each semantic category. For example, in terms of the relative improvement, semantic categories such as INDIVIDUAL, SOMEONE, KNOWLEDGE, DISTRICT, and TERRITORY outperform the other categories. Although the full results are not shown here, we observe that 94 out of our 99 semantic categories show a higher success rate. The five semantic categories that do not show any improvement are STATUS, EQUIPMENT, PORTION, INSTRUMENT, and BELIEF.

Next, we examine the false positives. Due to the postclassification methodology, it is not surprising to see that inaccurate noun pair predictions involving one of our semantic categories are the largest contributors to false positives. We suspect that this is mainly caused by the inevitable semantic drift, causing the extraction of incorrect terms for a semantic category, and subsequently inaccurate hypernymy predictions after post-classification. The top-5 semantic categories that are commonly misclassified in the data are WORK, BUILDING, PERIOD, QUALITY, STATUS.

We noticed that the semantic drift begins in the very early iterations for these semantic categories, with some of them occurring as early as the first iteration. For example, WMEB extracted *war*, *music*, and *work* which have little or no relation at all with the STATUS category. This can be caused by several factors, such as the low-

Semantic Categories	# Positive Instances	# Correct Predictions by Our Basic Model	# Correct Predictions after Post-classification
ENTITY	427	10 (2.3%)	77 (18.03%)
EVENT	314	17 (5.4%)	81 (25.80%)
OBJECT	258	15 (5.8%)	58 (22.48%)
CITY	210	112 (53.3%)	143 (68.10%)
GROUP	207	5 (2.4%)	34 (16.43%)
PERSON	204	8 (3.9%)	38 (18.63%)
LOCATION	119	13 (11%)	32 (26.89%)
INDIVIDUAL	115	3 (2.6%)	57 (49.57%)
ACTIVITY	98	7 (7.1%)	31 (31.63%)
WRITER	97	7 (7.2%)	19 (19.59%)
PART	95	1 (1.1%)	17 (17.89%)
DISTRICT	85	11 (12.9%)	44 (51.76%)
SOMEONE	81	0 (0%)	26 (32.10%)
KNOWLEDGE	76	0 (0%)	27 (35.53%)
TERRITORY	74	8 (10.8%)	47 (63.51%)

Table 6.6: Comparison of our system’s accuracy before and after post-classification on instances involving the 15 most frequently appearing semantic categories in the test partition

quality of the seeds used to represent the semantic category, or it may be that these semantic categories are not suitable for bootstrapping as the patterns they commonly occur in are very general or ambiguous. It is also possible that the GOOGLE WEB 1T corpus may not have adequate coverage of these categories.

Naturally, some semantic categories are harder to bootstrap compared to others. In light of this, we hypothesise that one of the most effective ways of reducing false positives is to use different sized lexicons for each semantic category based on how prone the semantic category is to semantic drift, as opposed to using the first-800

extracted lexicon terms for all semantic categories. We can use more terms from semantic categories that are less prone to semantic drift, and conversely, use less terms from semantic categories that are more prone to semantic drift. By doing so, we can potentially achieve higher precision due to a reduced effect of semantic drift, while at the same time potentially increase the number of true positives from semantic categories less prone to semantic drift (higher recall). This process, however, requires measurement of the semantic drift in each semantic category, which is beyond the scope of this thesis. We refer readers to McIntosh (2010) for techniques to measure semantic drift.

Another possible factor that can be considered when extracting different sized lexicons per each semantic category is the frequency distribution of that semantic category in the data. In this case, to maximise the potential benefit for both recall and precision, it is preferable to extract more terms for frequently-appearing semantic categories as opposed to the rarely-appearing semantic categories.

6.4 Integrating WMEB Lexicons into RE

In this section, we discuss a preprocessing approach to incorporate the WMEB-induced lexicons into our RE learning and classification.

As observed earlier, there is a small set of hypernyms frequently paired with a large number of hyponyms. These hypernyms are often located high up in the WORDNET hierarchy, and are less likely to cooccur in relation-identifying contexts or patterns with the hyponym nouns they are paired with. This results in sparse features for the individual instances. On their own, we suspect that they would not be able to contribute much in training, while similarly, due to their sparse feature sets, they are likely to pose a problem during classification.

We hypothesise that the learning will be more effective if the sparse feature sets of these semantically-similar instances are collated together to form a denser and higher-quality feature set. Therefore, instead of learning from a large number of instances with low-quality features, we are proposing to learn from a smaller number of instances with higher quality features as a result of the instance grouping/collation

process. The most prominent challenge in this experiment is how to effectively collate instances based on their semantic similarity.

6.4.1 Methodology

To collate the instances, we first collect nouns in the data that are a member of a given semantic category. Let SC be the set of 99 semantic categories defined in our data (as described in Section 6.2). Further, let Θ be the set of hypernymy instances $\text{hypernym}(X, Y)$ in our training data whose Y is one of our semantic categories (i.e. $Y \in SC$).

Next, we extract out all the unique nouns X s and the semantic categories they belong to from Θ . We store this information in $M = \{(m_1, sc(m_1)), \dots, (m_{|M|}, sc(m_{|M|}))\}$, where m_i is the term X found in our training data, and $sc(m_i)$ is the set of semantic categories the noun m_i belongs to.⁵

Following this, for every noun pair (A, B) in the data whose A is in M , we map A into all the semantic categories found in $sc(A)$. Unlike noun pairs which consist of a pair of nouns, the resulting mapping is a pairing between a semantic category (which represents all nouns sharing the associated semantic trait) and a noun (i.e. (SEMANTIC CATEGORY, *noun*)), which we refer to as a *semantic pair*. For any arbitrary noun pair (A, B) , the number of the semantic pairs produced by this mapping is equivalent to the size of $sc(A)$. A noun pair (A, B) whose $A \notin M$, cannot be mapped into semantic pairs. We record which noun pairs are mapped into which semantic pairs during this process.

To illustrate this, consider the semantic lexicons and noun pairs in Figure 6.5. Assume that an element of M , m_1 , represents the noun *cat*, which is a member of the categories ANIMAL and MAMMAL (i.e. $sc(m_1) = \{\text{ANIMAL}, \text{MAMMAL}\}$). We then scan for noun pairs (e.g. (A, B)), where $A = \text{cat}$. There are two such pairs in our example: $(\text{cat}, \text{animal})$ and $(\text{cat}, \text{feline})$. We then map each of these noun pairs into

⁵Similar to the postprocessing experiment, a term may belong to multiple categories in our experiment, as we only restrict WMEB's mutual exclusion assumption within the same run, but not over the different WMEB runs.

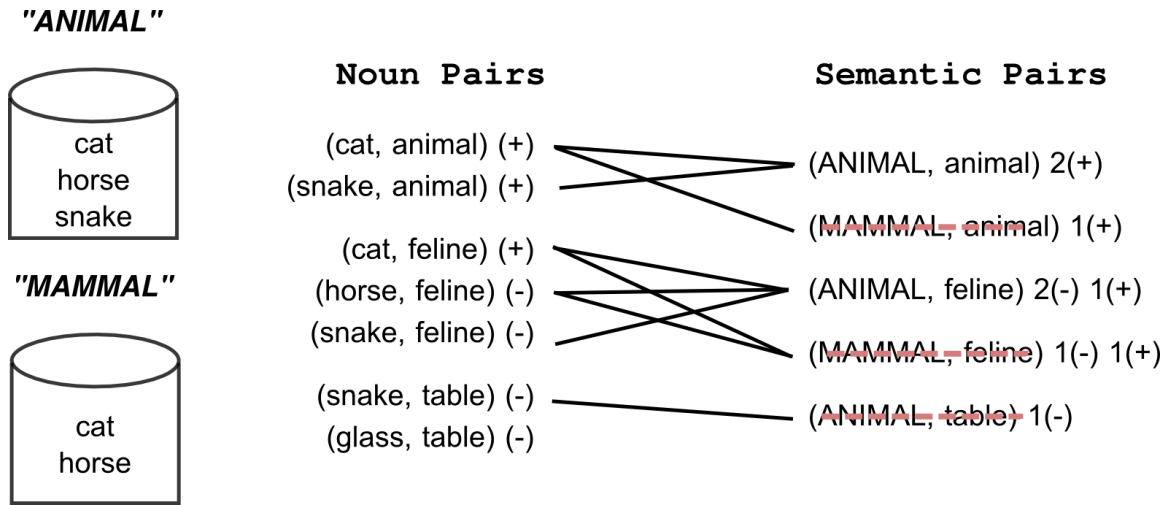


Figure 6.5: An overview for the preprocessing experiment

semantic pairs: (*cat*, *animal*) into (ANIMAL, *animal*)⁶ and (MAMMAL, *animal*); and (*cat*, *feline*) into (ANIMAL, *feline*) and (MAMMAL, *feline*).

At this point, the semantic pairs are not associated with a class (i.e. positive or negative), which is required during training. A simple way to determine the semantic pair's class is to look at the underlying noun pairs' classes. Using the example in Figure 6.5, considering that both (*cat*, *animal*) and (*snake*, *animal*) are annotated as positive by the gold-standard, the semantic pair (ANIMAL, *animal*) can be assigned as positive.

However, determining the semantic pair's class is not as straightforward when the noun pairs mapped into it are from different classes. For example in Figure 6.5, the noun pairs (*cat*, *feline*) (positive), (*horse*, *feline*) (negative) and (*snake*, *feline*) (negative) are mapped into the semantic pair (ANIMAL, *feline*). In this case, (ANIMAL, *feline*) can be assigned either positive or negative, as there are noun pairs supporting each class. To determine the semantic pair class, we make our decision by considering the distribution of the label of the supporting noun pairs.

We observe that there are different distributions of positive and negative instances associated with the semantic pairs. Some of them are clearly biased towards a cer-

⁶Note the different typeface used to represent ANIMAL (signifying that it is one of WMEB categories) and the noun *animal*.

tain class (e.g. 40 negative to 2 positive noun pairs) while others are more equally distributed between the two classes (e.g. 20 negative to 16 positive noun pairs).

We prefer using semantic pairs with a more biased-class distribution to the more balanced-class distribution as they are arguably more informative and discriminating for learning. We label such semantic pairs using the dominating class of the noun pairs associated with them. We ignore semantic pairs with a more uniform-class distribution from training, and in this case, we revert back to training using the individual noun pairs that are mapped into those semantic pairs.

We use the entropy of a semantic pair as the basis to decide whether to include it in the training data. An entropy of a semantic pair $H(sp)$ can be calculated via:

$$H(sp) = -(P(pos) \log_2 P(pos) + P(neg) \log_2 P(neg))$$

where $P(pos)$ and $P(neg)$ are the smoothed probabilities (with simple add-one smoothing) of positive and negative noun pairs associated with the semantic pair sp .⁷ We exclude any semantic pairs with an entropy value above 0.9 from our data. We use this value as it seems to be a clear cut-off point between semantic pairs with a more biased and uniform distribution in our training data. Another reason for using this threshold is to exclude semantic pairs which are mapped by only a single noun pair from our data, as we consider such semantic pairs to be less reliable.

Using the example from Figure 6.5, we refrain from using (MAMMAL, *feline*) in training due to its uniform class distribution. We also refrain from using (MAMMAL, *animal*) as it is only mapped from a single noun pair (i.e. (*cat*, *animal*)). Meanwhile, we label (ANIMAL, *feline*) as negative.

Next, we create features for the semantic pairs. For each noun pair, we divide its feature counts (i.e. pattern frequency counts) equally among the semantic pairs it successfully mapped to (the semantic categories that meet the entropy threshold). The final feature set of a semantic pair is derived by adding up all the features contributed by the individual noun pairs that are mapped into it.

Based on Figure 6.5, (*cat*, *animal*) maps into two semantic pairs: (ANIMAL, *mam-*

⁷The entropy calculation process cannot be applied to the test data as the class labels are withheld.

mal) and (MAMMAL, *mammal*). We divide its feature counts by two. On the other hand, (*snake*, *animal*) only maps into one semantic pair (ANIMAL, *animal*). In this case, we do not divide its feature counts. The feature counts for the semantic pair (ANIMAL, *animal*) are constructed by merging individual contributions from each noun pair, that is, half of the feature counts from (*cat*, *animal*) and full feature counts from (*snake*, *animal*)).

Due to the feature division, we may end up with a pattern with a frequency count less than 1. This poses a problem in our feature representation as we can only represent a feature with a minimum frequency count of 1 (see Section 4.1.5), meaning that such patterns will be excluded from the feature set. As we do not want to lose a potentially valuable feature for learning, we modify our feature representation policy for this experiment by inserting an additional feature to check whether the pattern frequency count (denoted as l below) is greater than 0. Using the notation defined in Section 4.1.5, the feature representation of these vectors can be defined as:

$$\text{for } l = 0, f_k^l = \begin{cases} 1 & \text{if } \text{freq}(np_i, pat_k) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{for } l \geq 1, f_k^l = \begin{cases} 1 & \text{if } \text{freq}(np_i, pat_k) \geq 2^{l-1} \\ 0 & \text{otherwise} \end{cases}$$

Originally, there were 2.5 million noun pairs in our training partition. After the preprocessing process, there are approximately 1.7 million instances, with slightly more than 200,000 semantic pairs. The remaining instances are noun pairs that are not successfully mapped into any semantic pairs either because the semantic pairs that they were mapped into do not meet the entropy threshold requirement, or we do not have any semantic category information for the noun pair.

The majority of commonly-mapped semantic pairs are pairs involving the members of the semantic categories and the semantic categories themselves, e.g. (CITY, *city*). There are a few exceptions, such as (PERSON, *individual*), (WRITER, *author*), and (CITY, *location*), which highlight the semantic-relatedness between the pairs.

The instances for the test partition follow the same preprocessing process as the training instances above, albeit with some modifications. Similar to our postclassi-

fication experiment, we take the **WMEB** lexicon as the resource for labelling semantic categories for our instances. Unlike that experiment, we do not use different sized **WMEB** lexicons, and instead use the same final cut-off point, using the first-800 extracted **WMEB** nouns per semantic category.

Following this, we map each noun pair (A, B) into a semantic pair iff B is one of our predetermined semantic categories, and A is found in the first-800 extracted **WMEB** nouns for the semantic category B (similar to the mapping for the training instances). We prevent the mapping to a semantic pair if the resulting semantic pair is one of the pairs that does not pass our entropy threshold in the training partition. Finally, we generate features for the evaluation instances. As with the training instances, there are two instance types: (i) semantic pairs, and (ii) noun pairs that are not converted into semantic pairs either because none of the associated semantic pairs meet the entropy threshold requirement or they are not in one of **WMEB**'s lexicons, and therefore we do not have any semantic category information on them.

There are just under 1.45 million instances remaining after the preprocessing process, of which approximately 1.1 million are semantic pairs. This means that we have more semantic pairs than noun pairs that are not converted into semantic pairs, which is the opposite of what we encounter in the training partition, where after preprocessing, we have more unconverted noun pairs than semantic pairs. This can be attributed to the size of the **WMEB**-induced lexicons, which enables more possibilities for the noun pairs conversion to semantic pairs.

6.4.2 Experiment

To begin our experiment, we first determine the learning parameters. We use the best hypernymy model setting on our development partition from Section 5.3: the model is trained on **WIKIPEDIA** features, with $Q = 10$. We treat white instances as positive, black instances as negative, and ignore grey instances during training (**train: pos=w, neg=b**). In this case, the grey instances are not considered when calculating the entropy of a semantic pair to determine whether it is a positive or negative instance. Finally, we train a model after we collate the instances and generate their

features.

We similarly run the preprocessing process for the test instances. After generating the features, we apply the model trained earlier to these collated test instances. Following this, we have the class predictions for the test instances. Contrary to our previous experiments which involve only noun pairs, this time the classifier outputs the prediction for the semantic pairs as well. However, recall that our gold-standard has only class information for the noun pairs, and no class information on the semantic pairs.

To evaluate these semantic pairs, we need to map them back to the noun pairs that are associated with them in the first place. The challenge here is that it is very likely that a noun pair is mapped into multiple semantic pairs, and the classifier outputs different predictions for these semantic pairs. This situation is the opposite of what we have in the training partition, where we have to decide the semantic pairs' classes based on the noun pairs that map into them.

For evaluation purposes, we resolve the final noun pair class by looking at the predictions of the semantic classes the noun pair successfully mapped into. Based on our earlier experiments with hypernymy, we hypothesise that the positive predictions are generally accurate. Therefore, we classify a noun pair to be positive if there is at least one positive prediction over the entire set of semantic pairs it successfully mapped to (regardless of the number of negative predictions). In contrast, we classify a noun pair to be negative if all of the semantic pairs predictions are negative. Naturally, we do not apply this procedure for noun pairs that are not mapped into any semantic pairs, and directly use the model's predictions for such noun pairs for evaluation. We evaluate our system using the `eval: pos=w, neg=g,b` setting.

Table 6.7 shows the improvement of our preprocessing model (**HYPER-WMEB-PRE**) over the best basic model from Section 5.3 (**HYPER-best basic model**). Similar to our postclassification experiment in Section 6.3, the improvement in accuracy is gained by improved recall, at the expense of precision.

Next, to investigate the impact of preprocessing process to the individual improvement of each semantic category, we look at the success rate in classifying frequent semantic categories in the data as compared to the basic system. For comparison, we

Models	Evaluated on the Test Partition					
	pos=w, neg=g,b					
	TP	FP	FN	R	P	F
HYPER-best basic model	606	1,108	7,971	0.071	0.354	0.118
HYPER-WMEB-PRE	914	3,538	7,663	0.107	0.205	0.141

Table 6.7: Accuracy of preprocessing model using semantic category information compared to the basic hypernymy model

use the 15 most frequently appearing semantic categories as listed in Table 5.21.

Table 6.8 shows the success rate of our preprocessing model in classifying noun pairs (Xs , Ys), whose Ys are one of the 15 most frequently appearing semantic categories in the data. As we can observe, some semantic categories benefit from the preprocessing process (e.g. ENTITY, OBJECT, CITY and SOMEONE), while others do not benefit (e.g. KNOWLEDGE) or even have lower success rates (e.g. EVENT, GROUP, DISTRICT, and TERRITORY).

This is different from our earlier result with postclassification (shown in Table 6.6), where all semantic categories benefit from the methodology. Looking at the overall 99 categories, we observe that only 22 categories have a higher success rate after the preprocessing process, compared with 94 categories from the postclassification experiment.

The classifier only predicted a small number of positive predictions for the noun pairs belonging to the 77 semantic categories that do not benefit from the preprocessing process. This can be confirmed by the small numbers of false positives as compared with the 22 semantic categories that have higher classification success rates. The 5 hypernyms (that is, the Y of (X, Y)) often found in false positives are *country*, *nation*, *metal*, *item*, and *accessory*, where only *metal* is one of our 99 semantic categories.

We note that more than half of the false positive predictions involving the hypernyms *country* and *nation* are actually clear-cut.⁸ The common mistakes for these two

⁸The hypernymy relation in WORDNET for (*country name*, *country*) is between the first sense

Semantic Categories	# Positive Instances	# Correct Predictions by Our Basic Model	# Correct Predictions after Preprocessing
ENTITY	427	10 (2.3%)	162 (37.94%)
EVENT	314	17 (5.4%)	0 (0%)
OBJECT	258	15 (5.8%)	103 (39.92%)
CITY	210	112 (53.3%)	153 (72.86%)
GROUP	207	5 (2.4%)	1 (0.48%)
PERSON	204	8 (3.9%)	45 (22.06%)
LOCATION	119	13 (11%)	0 (0%)
INDIVIDUAL	115	3 (2.6%)	0 (0%)
ACTIVITY	98	7 (7.1%)	16 (16.33%)
WRITER	97	7 (7.2%)	8 (8.25%)
PART	95	1 (1.1%)	0 (0%)
DISTRICT	85	11 (12.9%)	0 (0%)
SOMEONE	81	0 (0%)	17 (20.99%)
KNOWLEDGE	76	0 (0%)	0 (0%)
TERRITORY	74	8 (10.8%)	0 (0%)

Table 6.8: Comparison of our system’s accuracy before and after the preprocessing process for the 15 most frequently appearing semantic categories in the test partition

hypernyms are positive predictions for pairings between a location name (that is not a country) and *country* or *nation*, which we can attribute to semantic drift.

As with the postclassification experiment, it is reasonable to assume that semantic drift has a negative impact on our current task. As explained earlier, semantic drift causes WMEB to extract incorrect nouns for a semantic category. During the preprocessing process, these incorrect nouns may be translated to a semantic pair that

of *country name* and second sense of *country* (i.e. a grey instance), therefore rendering them to be negative in our current evaluation. WORDNET does not have any hypernymy relation between (*country name*, *nation*) pairs, although we judge that the relations between such pairs are clear-cut.

they may not actually belong to, and subsequently contribute inaccurate features for that semantic pair. Further, given that there are good quality noun pairs that are semantically-related to that semantic pair, and contribute towards the semantic pair's features, the resulting feature sets of such semantic pairs will be a mix of good and bad quality features which do not help the classification process.

The result of this feature mixing can manifest itself in two ways, depending on the severity of the semantic drift. The first case is when the drift is not severe, and results in extractions of semantically-close nouns (but not the same, e.g. drift from city names to state/country names). It is arguable that such semantically-close nouns often share features, and therefore positively contribute towards the semantic pairs that they map to. The classifier may then predict these semantic pairs to be positive instances. Subsequently, this means positive predictions for all the noun pairs that originally mapped into the semantic pairs. The correct noun pairs will then be correctly classified. The downside of this effect is that the classifier is penalised for identifying incorrect noun pairs that are semantically-close to the correct noun pairs (e.g. identifying city names as a country).

The second case is more severe semantic drift where the extracted nouns are semantically remote to the seeds (e.g. drift from specific types of publications to common nouns). In this case, we suspect that the (non-relation identifying) features from the incorrect instances overwhelm the (relation identifying) features from the correct instances for the semantic pairs they map to during preprocessing. As a result, the classifier is likely to predict such semantic pairs to be negative, reducing the likelihood of identifying correct noun pairs in the evaluation.

While technically we use the same set of **WMEB** lexicons for both the preprocessing and postclassification models, we suspect that the semantic drift may have a more severe effect on the results of the preprocessing model. As a reminder, in postclassification, we reclassify every **WMEB** noun in a semantic category lexicon to be a positive hypernymy relation between nouns in the lexicon and that semantic category itself. However, this process can only be performed if there is such a noun pair in the test data. As the majority of the possible pairings are not in the test data, not all of the extracted **WMEB** lexicon contributes towards the postclassification results. In this

sense, the effect of semantic drift in postclassification is perhaps underestimated.

On the other hand, the preprocessing model does not involve an equivalent filtering step during learning. During the preprocessing process, we map a noun to all the semantic categories **WMEB** predicted it to be. This means that all of the **WMEB** predictions contribute towards the preprocessing experiment, and subsequently the effect of semantic drift is perhaps more prominent.

Another notable difference between the preprocessing and the postclassification model is that the preprocessing model may affect the prediction of noun pairs whose nouns are not part of our **WMEB** lexicons, while the postclassification model's effects are limited to noun pairs that are part of our **WMEB** lexicons. This, we suspect, is due the side effect of the condensed feature sets of the collated instances which may provide additional clues for classification for all instances, regardless of whether the nouns in the noun pairs are members of our semantic categories set.

As a matter of fact, the preprocessing model manages to predict a number of true positive noun pairs that are not part of our **WMEB** lexicons and also not predicted by our basic model. Some examples of these noun pairs are **hypernymy**(*ampicillin*, *antibiotic*), **hypernymy**(*curare*, *poison*), and **hypernymy**(*brasilia*, *national capital*), where the nouns *antibiotic*, *poison*, and *national capital* are not one of our **WMEB** semantic categories.

The effect of collating instances and their features can have the opposite impact. Indeed, the preprocessing model failed to correctly identify a number of noun pairs that are not part of our **WMEB** lexicons, but were correctly identified by our postprocessing model.⁹ This situation suggests that there is complementarity between our postprocessing and preprocessing approaches, which we explore further in Section 6.5.

6.5 Metaclassification

In the previous section, we described a preprocessing method to improve system accuracy by exploiting **WMEB** lexicons by collating instances and their features based

⁹Recall that our postprocessing model is basically built on top of our basic hypernymy model by reclassifying the basic model's predictions for noun pairs that are part of our **WMEB** lexicon. Therefore, in the postprocessing model, the predictions for noun pairs that are not part of **WMEB** lexicon are inherited (and therefore identical) from our basic model predictions on those pairs.

Models	Evaluated on the Test Partition					
	pos=w, neg=g,b					
	TP	FP	FN	R	P	F
HYPER-best basic model	606	1,108	7,971	0.071	0.354	0.118
HYPER-WMEB-BASELINE	-	-	-	0.210	0.118	0.151
HYPER-WMEB-POST	1,708	6,914	6,869	0.199	0.198	0.198
HYPER-WMEB-PRE	914	3,538	7,663	0.107	0.205	0.141
HYPER-WMEB-META	1,796	7,381	6,781	0.209	0.196	0.202

Table 6.9: The accuracy of our various RE models using semantic category information compared to the basic hypernymy model

on semantic similarity. Our main objective was to improve our accuracy over noun pairs that are part of our **WMEB** semantic category set. While we have been partly successful in this aspect, we observed that our preprocessing process may have the side effect of classifying noun pairs that we do not initially target: pairs whose nouns are not part of our **WMEB** semantic category set.

In this section, we hope to benefit from improved accuracy on noun pairs that are part of our **WMEB** lexicons, and also noun pairs that are not by combining predictions from postclassification and preprocessing models. To do so, we propose a straightforward disjunction-based metaclassification process, in which we predict a noun pair to be positive if one or both (postclassification and preprocessing) models predict that noun pair as positive. In contrast, we predict a noun pair to be negative iff both models predict that noun pair as negative.

Table 6.9 shows the improvement of our metaclassification model (**HYPER-WMEB-META**) over the individual postclassification and preprocessing models. Due to our metaclassification process, it is not surprising that the **HYPER-WMEB-META** result is similar to the **HYPER-WMEB-POST** result. The predictions from **HYPER-WMEB-PRE** are undoubtedly the reason for the slight improvement of **HYPER-WMEB-META** over **HYPER-WMEB-POST**, which is in line with our hypothesis of the complementary nature

of our preprocessing and postclassification models.

The results in Table 6.9 also demonstrate the superior result of our semantic category model as compared to our basic RE and baseline WMEB models. This confirms our initial hypothesis that semantic category information is beneficial for hypernymy RE. Further, the improvement over baseline WMEB model validates our efforts in integrating the semantic category information to our basic RE system, as compared to be totally dependent on WMEB extraction results for RE.

6.6 Summary

This chapter began with a discussion of the common mistakes made by our initial hypernymy model introduced in Section 5.3. A special property of these erroneous instances is that they belong to a small group of semantic categories. These semantic categories usually appear higher up in the WORDNET hierarchy, and usually do not appear in relation-identifying context with the nouns they are paired with. This situation is not ideal for our initial approach to RE which relies on the nouns in a given noun pair occurring together in a relation-identifying context.

In response to this, we hypothesised that our RE model can benefit from a generalised representation of instances that share a semantic property. To facilitate this, we exploited the semantic categories of our instances as gathered from the training data. As the equivalent information is not available for the evaluation data, we require a system that can predict a noun pair’s semantic category. In our experiments, we use WMEB, a bootstrapping-based system for inducing instances of semantic categories, relying on learning from a set of seed terms of each semantic category and an input corpus.

We presented the WMEB architecture in detail, focusing especially on its approach to delaying semantic drift based on mutual exclusion. We further discussed that while the mutual exclusion assumption is effective for extracting small numbers of semantic categories, it may prevent WMEB from operating well when extracting larger numbers of semantic categories. In response to this, we divided our 99 semantic categories into smaller groups. Related to this, we also need to avoid placing semantically-similar

categories into the same group, to avoid suboptimal **WMEB** results. Following this, we chose the seed terms for each of our semantic categories prior to using **WMEB** to bootstrap over the data.

Having collected the results from **WMEB**, we described three approaches for incorporating them into our RE task. The first approach is to use them as the basis of postclassification. This approach overrides predictions from our basic model into positive predictions based on the **WMEB** results. As semantic drift is more prevalent in later iterations, we used various cut-off points as the basis for postclassification to investigate the optimal point for our overall RE accuracy. The postclassification result showed a remarkable improvement over our basic hypernymy result, as evidenced by our success in addressing the common mistakes made by the basic model.

Our second approach was to use **WMEB**'s results as a means to generalise over noun pairs belonging to a semantic category. This process involved doing some preprocessing prior to learning, by collating individual features of semantically-similar instances into a combined feature for a semantic pair. By doing so, we replaced a number of individual noun pairs with sparse features in the data with smaller numbers of semantic pairs whose features sets are made up from the individual noun pairs, and are comparably more condense (and higher-quality).

Similar to the postclassification experiment, our preprocessing experiment results showed an improvement over our basic hypernymy approach. Although the preprocessing model is not as accurate as the postclassification model, it had the side effect of influencing predictions for noun pairs that are not part of the **WMEB** lexicon. We hypothesised that the two models are complementary, and proposed an approach for merging the two models' predictions via metaclassification. The improved results due to merging confirmed our hypothesis on the complementary nature of our postclassification and preprocessing models.

Our three approaches based on **WMEB** resulted in improved accuracy over our basic hypernymy and baseline **WMEB** models. Based on this observation, we concluded that it is possible to exploit semantic category information and integrate it to our basic RE system to improve its accuracy.

Chapter 7

Conclusion

7.1 Conclusions

In this thesis we explored a supervised approach to relation extraction (RE) on large-scale corpora. There are three main contributions of this thesis. First, we proposed an experimental methodology with the following properties:

1. it is applicable to any relation of interest;
2. it allows direct result comparisons among RE tasks using different resources or on different relations;
3. its design is reflective of the challenges in extracting relations from large-scale corpora.

Second, we investigated the accuracy of a supervised system on the RE task resulting from the proposed experimental methodology. Finally, we proposed three approaches in exploiting semantic categories to improve the accuracy of the system on the hypernym relation.

We summarise the findings and conclusions of our main contributions in the following sections. Section 7.1.1 covers our proposal of experimental methodology, which corresponds to the materials in Chapter 4 and the first half of Chapter 5. Section 7.1.2 summarises the investigation of our supervised RE system accuracy on the proposed

methodology which was covered in the second half of Chapter 5. Materials from Chapter 6 discussing the use of semantic categories to improve hypernymy RE are summarised in Section 7.1.3. Finally, Section 7.2 outlines future research directions that could be taken.

7.1.1 Our RE Methodology

Previous supervised RE approaches have tended to concentrate on a specific relation. The majority of them are inflexible, and much effort is required to repurpose them to other relations. Further, different methodologies assume different RE techniques and resources. It is therefore very difficult to directly compare the reported RE results due to their different experimental settings.

We addressed these pertinent issues by designing a standardised extraction process that is equally applicable to any relation of interest with minimum adaptation effort. Moreover, our standardised pipeline process reduces the possibility of introducing different assumptions about the data and task, allowing better RE comparability across different relations and resources.

Our proposed methodology also looked to reflect challenges in performing RE on large-scale corpora, over a representative set of noun pairs. These challenges include predicting relation labels for difficult pairs with polysemous nouns and/or a small number of features. While our initial task design was largely based on that of Snow *et al.* (2005)'s, we discovered that its key assumptions sidestepped the above challenges by excluding such difficult pairs.

These key assumptions relate to the sampling procedure which directly determines which pairs to include in training and evaluation. As the relative difficulty of an RE task can be judged based on the characteristics of the pairs that are included in the experiment, the sampling procedure plays an important role in shaping the interpretability of the RE task and its results. Therefore, by excluding difficult pairs, Snow *et al.* (2005)'s task is arguably greatly simplified. Chapter 5 discussed this in great detail.

One of our experiments was to investigate the accuracy of a classifier based on

Snow *et al.* (2005)’s methodology on a more realistic task which contains pairs with small numbers of features. We discovered that the resultant classifier was very inaccurate, and the results are far removed from the promising and competitive results reported in Snow *et al.* (2005). This confirmed our initial hypothesis that Snow *et al.* (2005)’s methodology is not suited to addressing challenges in this more realistic task.

Next, we built on our previous experiment by proposing a task design which reflects the challenges above. Our task design uses a simple sampling procedure, in which all pairs found in the corpora are included in the experiment, regardless of whether the pairs include polysemous nouns and/or of the number of features they have. The resulting task is considerably more difficult than that of Snow *et al.* (2005)’s, and most importantly, reflective of the challenges in performing RE on large-scale corpora.

7.1.2 Accuracy of Our Supervised RE System

Using the proposed methodology, we investigated the capabilities of a supervised RE system to predict relation labels of noun pairs extracted from large-scale corpora. We applied the system to learn three relations: antonymy, synonymy, and hypernymy.

The system learns from a set of pre-labelled (positive and negative) pairs of the three relations based on WORDNET annotations. The main learning feature is dependency patterns between a given pair in the corpora.

In our evaluation, we found that the accuracy of our system was very poor for all three relations. We first analysed the reasons why our system failed to predict positive (correct) pairs. Our investigation showed that the majority of the positive pairs in the data appeared less than 10 times in our data, and hence did not have a lot of useful features for learning and classification. Further, the positive pairs did not always appear in patterns that strongly indicate the relation of interest.

Next, we analysed the system’s false positive predictions. Based on a subset of such predictions randomly sampled from data, we identified that the majority (more than 50%) are actually valid (relation-expressing) pairs (especially for the synonymy and hypernymy relations).

The evaluation method penalises the system for finding valid pairs that do not exist in the gold-standard data, instead of rewarding it. In our view, this is unfair, as such pairs are very valuable in identifying the shortcomings of a knowledge database for that relation of interest. Taking the findings of such pairs into account, we also argue that our reported system’s accuracy is underestimated.

The identification of valid pairs not in the gold-standard data is a challenge, as the current evaluation method is unable to distinguish between “true” false positives and false positives that are due to missing data. A manual screening of these predictions is necessary to extract the valid pairs, which could be expensive if there is a large number of potential candidates. This is not the case for our system, as it only predicted a relatively small number of false positives. Such a property is suited to the manual screening process, considering the cost of screening small numbers of candidates and the potential benefit in finding these valid pairs for knowledge base maintenance task.

The majority of the pairs in our data are negative pairs, which outnumber the positive pairs by a large proportion. This leads to class imbalance during learning, and the learner tending to generalise all pairs to be in the majority class (negative), interpreting the small number of pairs in the minority class (positive) to be outliers. We note that the severity of the class imbalance in our experiments is on par or even greater than the levels discussed in the class imbalance literature. We suspect that this is the reason why our system only made a relatively small number of positive predictions, which directly contributed to its poor accuracy.

Currently, we are unable to determine the relative impact of class imbalance. Further, as the three relations themselves have different levels of class imbalance, it is not possible to conclude if one relation is harder to learn than the other based purely on our system accuracy.

The poor accuracy of our system in our proposed task led us to question the methodology’s validity to RE. To investigate this, we applied our methodology to a different RE task for the hypernymy relation proposed by Navigli and Velardi (2010). This experiment allowed us to directly compare our method with that of Navigli and Velardi (2010). The results suggested that our system was very competitive with Navigli and Velardi (2010)’s systems trained using a different methodology.

We noted that the dataset used in Navigli and Velardi (2010) experiment is of much smaller scale compared to ours. As a result, class imbalance and data sparsity are not as apparent in this task as compared to our proposed large-scale task. Given our system’s superior accuracy in this experiment, we conclude that our RE methodology is perhaps better suited to smaller-scale tasks that are less susceptible to both class imbalance and data sparsity.

7.1.3 Using Semantic Categories to Aid Hypernymy RE

So far, our RE system has been learning and predicting relation labels for noun pairs based on their individual features. As we have discussed, a large number of these noun pairs have very sparse features. In Chapter 6, we proposed various methods for exploiting nouns’ semantic categories to generalise over the individual (sparse) features, enabling the system to simultaneously learn and predict over a group of pairs with a similar semantic category. We concentrated on the hypernymy relation, using a set of frequently-paired hypernyms as our semantic categories. We used WMEB (McIntosh and Curran 2008), a bootstrapping-based system, to determine which of the pre-identified semantic categories a noun belongs to based on a set of seeds extracted from our training partition for each category.

The idea of our first approach is to make a generalised prediction for all noun pairs belonging to a semantic category, based on WMEB output. First, a WMEB prediction of a noun belonging to a certain semantic category can be interpreted as the existence of a hypernym relation between the noun and that semantic category. Following our idea to generalise prediction for noun pairs of a semantic category, we translate all WMEB predictions to be positive hypernymy predictions between the noun and the semantic categories WMEB predicted them to be in. Using these positive predictions, we then scan for their occurrences in our evaluation data, and override our initial RE system’s predictions of these pairs to be positive. The result of this experiment was very encouraging, improving our initial RE system’s f-score of 0.118 to 0.198.

Our second proposed approach focused on grouping semantically-similar noun pairs into a single semantic category-based noun pair, based on the WMEB predictions.

The resulting pairs are then used in both training and testing as proxies for the base noun pairs. As a result of this process, instead of dealing with large numbers of individual pairs with sparse features, the system now deals with smaller numbers of proxy pairs with denser features. This effectively reduces the severity of data sparsity during both learning and classification. The result of this experiment was also very encouraging: an improvement in f-score of 0.118 to 0.141.

Analysis of the results of both the first and second approaches revealed that the predictions are complementary in nature (both methods made different correct predictions). We then proposed to combine the results from the first and second approaches via a metaclassification process, which further improved our system’s f-score to 0.202. This result is superior to the results of our basic system, and the results reported from the first and second approaches. Our success in exploiting semantic category information via three different approaches demonstrated the usefulness of semantic information in hypernymy RE.

7.2 Future Work

There are many open issues that are not yet addressed in this thesis. In this section, we identify several potential research avenues that could be taken in building off our research.

Task Setup

There are several problems in using WORDNET as our gold-standard labelling source. First, the limited coverage of WORDNET (especially with proper nouns) prevents us from learning and evaluating nouns that are not in WORDNET inventory. Second, WORDNET relation links are defined at the sense-level, and yet we are performing a word-level task. As such, we need to develop heuristics to resolve the mapping between senses and words which may produce incorrect labellings or exclude valid pairs.

A possible research avenue is to use other knowledge bases whose relation links are

defined at the word-level as our gold-standard, such as FREEBASE (Bollacker *et al.* 2008), YAGO (Suchanek *et al.* 2007), and BABELNET (Navigli and Ponzetto 2010). These resources also have greater coverage than WORDNET, and contain many other possible relations of interest.

A further study on the effects of corpora to RE accuracy can be explored by looking at other resources, such as UKWAC (Ferraresi *et al.* 2008) or BRITISH NATIONAL CORPUS (BNC) (BNC Consortium 2007). Both these resources contain personal and informal texts (such as blogs, forum postings, or transcripts of informal conversations on radio shows), which contrast with the formal journalistic newswire articles in GIGAWORD, or encyclopaedia articles in WIKIPEDIA.

Our system relies greatly on MINIPAR to identify noun tokens and produce patterns between noun pairs as its features. Our current pattern representation procedure is tailored to MINIPAR’s output and does not extend to other parsers. A future experiment may involve using different parsers, or interpreting MINIPAR output using different representations which have been shown to impact RE performance (Miyao *et al.* 2009; Stevenson and Greenwood 2009).

Supervised RE System

Our RE task presents two main challenges to our supervised system: data sparsity and class imbalance. These two problems are widely-researched, and several techniques have been shown to lessen their impact on classification task. We suspect that these two problems greatly affected our system’s accuracy as our system does not incorporate any specific techniques to deal with data sparsity and class imbalance. An obvious area of future work is to incorporate such techniques to our learning process to improve the system’s accuracy.

It is also possible to use additional lexical, syntactic, and semantic features in addition to patterns. These can be part-of-speech, word n-grams, capitalisation, or other features that have been shown to improve RE accuracy (Culotta and Sorensen 2004; Kambhatla 2004; Zhou *et al.* 2005; McNamee *et al.* 2008; Ritter *et al.* 2009; Hachey 2009a).

Currently, our system does not take the symmetrical properties of antonymy and synonymy into account. As such, we treat features from **antonym**(*brother, sister*) separately from **antonym**(*sister, brother*), while it is arguably more beneficial to combine them when predicting relation labels for either (*brother, sister*) or (*sister, brother*).

Similarly, for hierarchical relations, inference or transitive closure may improve the robustness/accuracy of our system. For example, if our system predicted **hypernym**(*cat, animal*) and **hypernym**(*animal, entity*) to be positive, **hypernym**(*cat, entity*) to be negative, we can override the negative prediction of **hypernym**(*cat, entity*) to be positive based on the transitive closure property of the hypernymy relation.

Semantic Categories

One of the problems in using **WMEB** to predict semantic categories is the inevitable manifestation of semantic drift. To deal with semantic drift, we proposed to only use the first- X extracted terms in the **WMEB** lexicons for all categories. However, semantic drift occurred in different iterative phases of different categories. Based on this observation, future research may consider using different cut-off points for different categories taking into account when semantic drift occurs for each category.

Our preprocessing approach currently uses the same first- x cut-off point used in the postclassification approach. However, we observed that semantic drift has a different impact on these two approaches, suggesting that the best cut-off point for preprocessing approach may be different to that for the postclassification approach. Future work may involve investigating the optimal cut-off point for the preprocessing approach.

Bibliography

- AGICHTEIN, EUGENE, and LUIS GRAVANO. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, 85–94, San Antonio, USA.
- AKBANI, REHAN, STEPHEN KWEK, and NATHALIE JAPKOWICZ. 2004. Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, 39–50, Pisa, Italy.
- BANKO, MICHELE, MICHAEL CAFARELLA, STEPHEN SODERLAND, MATT BROADHEAD, and OREN ETZIONI. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 2670–2676, Hyderabad, India.
- , and OREN ETZIONI. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the 46th Annual Meeting of the ACL: HLT*, 28–36, Columbus, USA.
- BERLAND, MATTHEW, and EUGENE CHARNIAK. 1999. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the ACL*, 57–64, College Park, USA.
- BLITZER, JOHN, RYAN McDONALD, and FERNANDO PEREIRA. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 120–128, Sydney, Australia.
- BLOHM, SEBASTIAN, PHILIPP CIMIANO, and EGON STEMLE. 2007. Harvesting relations from the web: quantifying the impact of filtering functions. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, 1316–1321, Vancouver, Canada.
- BNC CONSORTIUM, 2007. *The British National Corpus, version 3 (BNC XML Edition)*. Oxford University Computing Services. <http://www.natcorp.ox.ac.uk/>.

- BOLLACKER, KURT, COLIN EVANS, PRAVEEN PARITOSH, TIM STURGE, and JAMIE TAYLOR. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1247–1250, Vancouver, Canada.
- BRANTS, THORSTEN, and ALEX FRANZ, 2006. *Web 1T 5-gram Version 1*. <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13>.
- BRIN, SERGEY. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology (EDBT'98)*, 172–183, Valencia, Spain.
- BUDANITSKY, ALEXANDER, and GRAEME HIRST. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics* 32.
- BUITELAAR, PAUL, and PHILIPP CIMIANO (eds.) 2008. *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*. Amsterdam, The Netherlands: IOS Press.
- BUNESCU, RAZVAN C., and RAYMOND J. MOONEY. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the ACL*, 576–583, Prague, Czech Republic.
- CARABALLO, SHARON A. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the ACL*, 120–126, College Park, USA.
- CARDIE, CLAIRE. 1997. Empirical methods in information extraction. *AI Magazine* 18.65–79.
- CARLSON, ANDREW, JUSTIN BETTERIDGE, BRYAN KISIEL, BURR SETTLES, ESTEVAM R. HRUSCHKA JR., and TOM M. MITCHELL. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI-10)*, 1306–1313, Atlanta, USA.
- CEDERBERG, SCOTT, and DOMINIC WIDDOWS. 2003. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, 111–118, Edmonton, Canada.
- CHEN, JINXIU, DONGHONG JI, CHEW LIM TAN, and ZHENGYU NIU. 2005a. Automatic relation extraction with model order selection and discriminative label identification. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005)*, 390–401, Jeju Island, Republic of Korea.

- , ——, ——, and —— . 2005b. Unsupervised feature selection for relation extraction. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005)*, 262–267, Jeju Island, Republic of Korea.
- CHINCHOR, NANCY. 1998. Overview of MUC-7. In *Proceedings of the 7th Message Understanding Conference*, Fairfax, USA. Accessed 4 July 2011 from http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html.
- , LYNETTE HIRSCHMAN, and DAVID D. LEWIS. 1993. Evaluating message understanding systems: An analysis of the third message understanding conference (MUC-3). *Computational Linguistics* 19.
- CHKLOVSKI, TIMOTHY, and PATRICK PANTEL. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, 33–40, Barcelona, Spain.
- COHEN, K. BRETONNEL, DINA DEMNER-FUSHMAN, SOPHIA ANANIADOU, JOHN PESTIAN, JUN'ICHI TSUJII, and BONNIE WEBBER (eds.) 2009. *Proceedings of the BioNLP 2009 Workshop*, Boulder, USA.
- CRAVEN, MARK, DAN DIPASQUO, DAYNE FREITAG, ANDREW MCCALLUM, TOM MITCHELL, KAMAL NIGAM, and SEÁN SLATTERY. 1998. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, 509–516, Madison, USA.
- CRISTIANINI, NELLO, and JOHN SHAWE-TAYLOR. 2000. *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press.
- CULOTTA, ARON, ANDREW MCCALLUM, and JONATHAN BETZ. 2006. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2006 (HLT-NAACL 2006)*, 296–303, New York City, USA.
- , and JEFFREY SORENSEN. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the ACL*, 423–429, Barcelona, Spain.
- CURRAN, JAMES R., TARA MURPHY, and BERNHARD SCHOLZ. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, 172–180, Melbourne, Australia.

- DAUME III, HAL. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the ACL*, 256–263, Prague, Czech Republic.
- DAVIDOV, DMITRY, and ARI RAPPOPORT. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proceedings of the 46th Annual Meeting of the ACL: HLT*, 692–700, Columbus, USA.
- DEERWESTER, SCOTT, SUSAN T. DUMAIS, GEORGE W. FURNAS, THOMAS K. LANDAUER, and RICHARD HARSHMAN. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41.391–407.
- DODDINGTON, GEORGE, ALEXIS MITCHELL, MARK PRZYBOCKI, LANCE RAMSHAW, STEPHANIE STRASSEL, and RALPH WEISCHEDEL. 2004. The Automatic Content Extraction (ACE) program — tasks, data, and evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 837–840, Lisbon, Portugal.
- DOWNEY, DOUG, OREN ETZIONI, and STEPHEN SODERLAND. 2010. Analysis of a probabilistic model of redundancy in unsupervised information extraction. *Artificial Intelligence* 174.726–748.
- , ———, ———, and DANIEL S. WELD. 2004. Learning text patterns for web information extraction and assessment. In *Proceedings of the AAAI-2004 Workshop on Adaptive Text Extraction and Mining*, 50–55, San Jose, USA.
- DOWNING, PAMELA. 1977. On the creation and use of English compound nouns. *Language* 53.810–842.
- DUNNING, TED. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19.61–74.
- ETZIONI, OREN. 2011. Search needs a shake-up. *Nature* 476.25–26.
- , MICHELE BANKO, and MICHAEL J. CAFARELLA. 2006. Machine reading. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, 1517–1519, Boston, USA.
- , MICHAEL CAFARELLA, DOUG DOWNEY, ANN-MARIA POPESU, TAL SHAKED, STEPHEN SODERLAND, DANIEL S. WELD, and ALEXANDER YATES. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence* 165.91–134.

- , ANTHONY FADER, JANARA CHRISTENSEN, STEPHEN SODERLAND, and MAUSAM. 2011. Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, 3–10, Barcelona, Spain.
- FADER, ANTHONY, STEPHEN SODERLAND, and OREN ETZIONI. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, 1535–1545, Edinburgh, UK.
- FELLBAUM, CHRISTIANE (ed.) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, USA: MIT Press.
- FERRARESI, ADRIANO, EROS ZANCHETTA, MARCO BARONI, and SILVIA BERNARDINI. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google?*, 47–54, Marrakech, Morocco.
- FRANCIS, WINTHROP NELSON, and HENRY KUČERA. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Boston, USA: Houghton Mifflin.
- FREITAG, DAYNE. 2000. Machine learning for information extraction in informal domains. *Machine Learning* 39.169–202.
- GABRILOVICH, EVGENIY, and SHAUL MARKOVITCH. 2006. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, 1301–1306, Boston, USA.
- , and ——. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 1606–1611, Hyderabad, India.
- GIRJU, ROXANA, ADRIANA BADULESCU, and DAN MOLDOVAN. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2003 (HLT-NAACL 2003)*, 80–87, Edmonton, Canada.
- , —, and —. 2006. Automatic discovery of part-whole relations. *Computational Linguistics* 32.
- , PRESILAV NAKOV, VIVI NASTASE, STAN SZPAKOWICZ, PETER TURNEY, and DENIZ YURET. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 13–18, Prague, Czech Republic.

- GLIOZZO, ALFIO MASSIMILIANO, MARCO PENNACCHIOTTI, and PATRICK PANTEL. 2007. The domain restriction hypothesis: Relating term similarity and semantic consistency. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2007 (HLT-NAACL 2007)*, 131–138, Rochester, USA.
- GREFENSTETTE, GREGORY. 1994. *Explorations in Automatic Thesaurus Discovery*. Boston, USA: Kluwer Academic Publishers.
- GRISHMAN, RALPH, and BETH SUNDHEIM. 1996. Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING 1996)*, 466–471, Copenhagen, Denmark.
- GROLIER. 1990. *Academic American Encyclopedia*. Danbury, USA: Grolier Electronic Publishing.
- HACHEY, BEN. 2009a. Generic relation identification: Models and evaluation. In *Proceedings of the Australasian Language Technology Association (ALTA) Workshop 2009*, 2–10, Sydney, Australia.
- . 2009b. Multi-document summarisation using generic relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, 420–429, Singapore.
- HARRIS, ZELLIG SABBETTAI. 1985. Distributional structure. In *The Philosophy of Linguistics*, ed. by Jerrold J. Katz. New York, USA: Oxford University Press.
- HASEGAWA, TAKAAKI, SATOSHI SEKINE, and RALPH GRISHMAN. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the ACL*, 415–422, Barcelona, Spain.
- HEARST, MARTI A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992)*, 539–545, Nantes, France.
- HENDRICKX, IRIS, SU NAM KIM, ZORNITSA KOZAREVA, PRES LAV NAKOV, DIARMUID Ó SÉAGHDHA, SEBASTIAN PADÓ, MARCO PENNACCHIOTTI, LORENZA ROMANO, and STAN SZPAKOWICZ. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 33–38, Uppsala, Sweden.
- HERSH, WILLIAM, AARON COHEN, LYNN RUSLEN, and PHOEBE ROBERTS. 2007. TREC 2007 Genomics Track Overview. In *Proceedings of the 16th Text REtrieval Conference (TREC 2007)*, 37–50, Maryland, USA.

- HOBBS, JERRY R., and ELLEN RILOFF. 2010. Information extraction. In *Handbook of Natural Language Processing*, ed. by Nitin Indurkha and Fred J. Damerau. Boca Raton, USA: CRC Press, 2nd edition.
- HOFFMANN, RAPHAEL, CONGLE ZHANG, and DANIEL S. WELD. 2010. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the ACL*, 286–295, Uppsala, Sweden.
- HOVY, EDUARD, ZORNITSA KOZAREVA, and ELLEN RILOFF. 2009. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, 948–957, Singapore.
- HOVY, EDUARD H., and CHIN-YEW LIN. 1998. Automated text summarization in SUMMARIST. In *Advances in Automatic Text Summarization*, ed. by Indeerjeet Mani and Mark T. Maybury. Cambridge, USA: MIT Press.
- HSU, CHIH-WEI, and CHIH-JEN LIN. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* 13.415–425.
- JACKSON, HOWARD. 2002. *Lexicography: An Introduction*. London, UK: Routledge.
- JACKSON, PETER, and ISABELLE MOULINIER. 2002. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. Amsterdam, The Netherlands: John Benjamins Publishing Co.
- JIANG, JING. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 1012–1020, Suntec, Singapore.
- JIJKOUN, VALENTIN, JORI MUR, and MAARTEN DE RIJKE. 2004. Information extraction for question answering: Improving recall through syntactic patterns. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, 1284–1290, Geneva, Switzerland.
- JING, HONGYAN, NANDA KAMBHATLA, and SALIM ROUKOS. 2007. Extracting social networks and biographical facts from conversational speech transcripts. In *Proceedings of the 45th Annual Meeting of the ACL*, 1040–1047, Prague, Czech Republic.
- KAMBHATLA, NANDA. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the 42nd Annual Meeting of the ACL*, 178–181, Barcelona, Spain.

- KOELING, ROB, DIANA MCCARTHY, and JOHN CARROLL. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the Joint Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 419–426, Vancouver, Canada.
- KOZAREVA, ZORNITSA, ELLEN RILOFF, and EDUARD HOVY. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the ACL: HLT*, 1048–1056, Columbus, USA.
- LANDES, SHARI, CLAUDIA LEACOCK, and RANDEE I. TENGI. 1998. Building semantic concordances. In *WordNet: An Electronic Lexical Database*, ed. by Christiane Fellbaum. Cambridge, USA: MIT Press.
- LDC, 2005. *The ACE 2005 (ACE05) Evaluation Plan*. Linguistic Data Consortium. Accessed 20 September 2011 from <http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v3.pdf>.
- , 2007. *The ACE 2007 (ACE07) Evaluation Plan*. Linguistic Data Consortium. Accessed 20 September 2011 from <http://www.itl.nist.gov/iad/mig/tests/ace/2007/doc/ace07-evalplan.v1.3a.pdf>.
- , 2008. *The ACE 2008 (ACE08) Evaluation Plan*. Linguistic Data Consortium. Accessed 20 September 2011 from <http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf>.
- LENAT, DOUGLAS B. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38.33–38.
- LI, HAIBO, DANUSHKA BOLLEGALA, YUTAKA MATSUO, and MITSURU ISHIZUKA. 2011. Using graph based method to improve bootstrapping relation extraction. In *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2011)*, 127–138, Tokyo, Japan.
- LIN, DEKANG. 1998. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems at LREC 1998*, 317–330, Granada, Spain.
- , SHAOJUN ZHAO, LIJUAN QIN, and MING ZHOU. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*, 1492–1493, Acapulco, Mexico.
- LIN, THOMAS, MAUSAM, and OREN ETZIONI. 2010. Identifying functional relations in web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, 1266–1276, Cambridge, USA.

- LITA, LUCIAN VLAD, and JAIME CARBONELL. 2004. Unsupervised question answering data acquisition from local corpora. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM 2004)*, 608–614, Washington, D.C., USA.
- MAGNINI, BERNARDO, CARLO STRAPPARAVA, GIOVANNI PEZZULO, and ALFIO GLIOZZO. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering* 8.359–373.
- MANI,INDERJEET, MARC VERHAGEN, BEN WELLNER, CHONG MIN LEE, and JAMES PUSTEJOVSKY. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 753–760, Sydney, Australia.
- MCCRAE, JOHN, and NIGEL COLLIER. 2008. Synonym set extraction from the biomedical literature by lexical pattern discovery. *BMC Bioinformatics* 9.159.
- MCDONALD, RYAN, FERNANDO PEREIRA, SETH KULICK, SCOTT WINTERS, YANG JIN, and PETE WHITE. 2005. Simple algorithms for complex relation extraction with applications to biomedical IE. In *Proceedings of the 43rd Annual Meeting of the ACL*, 491–498, Ann Arbor, USA.
- MCINTOSH, TARA, 2010. *Reducing Semantic Drift in Biomedical Lexicon Bootstrapping*. University of Sydney dissertation.
- , and JAMES R. CURRAN. 2008. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association (ALTA) Workshop 2008*, 97–105, Hobart, Australia.
- , and JAMES R. CURRAN. 2009. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 396–404, Suntec, Singapore.
- , LARS YENCKEN, JAMES R. CURRAN, and TIMOTHY BALDWIN. 2011. Relation guided bootstrapping of semantic lexicons. In *Proceedings of the 49th Annual Meeting of the ACL: HLT*, 266–270, Portland, USA.
- MCNAMEE, PAUL, and HOA TRANG DANG. 2009. Overview of the TAC 2009 knowledge base population track. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, Gaithersburg, USA.
- , RION SNOW, PATRICK SCHONE, and JAMES MAYFIELD. 2008. Learning named entity hyponyms for question answering. In *Proceedings of the 3rd International*

- Joint Conference on Natural Language Processing (IJCNLP 2008)*, 799–804, Hyderabad, India.
- MIN, BONAN, SHUMING SHI, RALPH GRISHMAN, and CHIN-YEW LIN. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-ConLL 2012)*, 1027–1037, Jeju Island, Korea.
- MINTZ, MIKE, STEVEN BILLS, RION SNOW, and DANIEL JURAFSKY. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 1003–1011, Suntec, Singapore.
- MITCHELL, ALEXIS, STEPHANIE STRASSEL, MARK PRZYBOCKI, JK DAVIS, GEORGE DODDINGTON, RALPH GRISHMAN, ADAM MEYERS, ADA BRUNSTEIN, LISA FERRO, and BETH SUNDHEIM, 2003. *ACE-2 Version 1.0*. <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T11>.
- MITKOV, RUSLAN, and CATALINA HALLETT. 2007. Comparing pronoun resolution algorithms. *Computational Intelligence* 23.262–297.
- MIYAO, YUSUKE, KENJI SAGAE, RUNE SÆTRE, TAKUYA MATSUZAKI, and JUN’ICHI TSUJII. 2009. Evaluating contributions of natural language parsers to protein-protein interaction extraction. *Bioinformatics* 25.394–400.
- MOHAMMAD, SAIF, BONNIE DORR, and GRAEME HIRST. 2008. Computing word-pair antonymy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, 982–991, Honolulu, USA.
- MOLDOVAN, DAN, and ROXANA GIRJU. 2001. An interactive tool for the rapid development of knowledge bases. *International Journal on Artificial Intelligence Tools* 10.65–86.
- , SANDA HARABAGIU, ROXANA GIRJU, PAUL MORARESCU, FINLEY LACATUSU, ADRIAN NOVISCHI, ADRIANA BADULESCU, and OREST BOLOHAN. 2002. LCC tools for question answering. In *Proceedings of the 11th Text REtrieval Conference (TREC 2002)*, 388–397, Gaithersburg, USA.
- NAVIGLI, ROBERTO. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)* 41.10:1–10:69.
- , and SIMONE PAOLO PONZETTO. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the ACL*, 216–225, Uppsala, Sweden.

- , and PAOLA VELARDI. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the ACL*, 1318–1327, Uppsala, Sweden.
- , ———, and JUANA MARIA RUIZ-MARTNEZ. 2010. An annotated dataset for extracting definitions and hypernyms from the web. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, 3716–3722, Valletta, Malta.
- NGUYEN, TRUC VIEN T., and ALESSANDRO MOSCHITTI. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the ACL: HLT*, 277–282, Portland, USA.
- NLM, 2009. *National Library of Medicine: MEDLINE Fact Sheet*.
- Ó SÉAGHDHA, DIARMUID, 2008. *Learning compound noun semantics*. Computer Laboratory, University of Cambridge dissertation.
- ÖZGÜR, ARZUCAN, THUY VU, GÜNEŞ ERKAN, and DRAGOMIR R. RADEV. 2008. Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics* 24.i277–285.
- PAŞCA, MARIUS, and SANDA M. HARABAGIU. 2001. The informative role of WordNet in open-domain question answering. In *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, 138–143, Pittsburgh, USA.
- , DEKANG LIN, JEFFREY BIGHAM, ANDREI LIFCHITS, and ALPA JAIN. 2006. Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 809–816, Sydney, Australia.
- PANTEL, PATRICK, and DEKANG LIN. 2002. Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 613–619, Edmonton, Canada.
- , and MARCO PENNACCHIOTTI. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 113–120, Sydney, Australia.
- , and DEEPAK RAVICHANDRAN. 2004. Automatically labeling semantic classes. In *Proceedings of the Human Language Technology Conference of the North*

- American Chapter of the Association for Computational Linguistics 2004 (HLT-NAACL 2004)*, 321–328, Boston, USA.
- , —, and EDUARD HOVY. 2004. Towards terascale knowledge acquisition. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, 771–777, Geneva, Switzerland.
- PEDERSEN, TED, SIDDHARTH PATWARDHAN, and JASON MICHELIZZI. 2004. WordNet::Similarity - measuring the relatedness of concepts. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2004 (HLT-NAACL 2004)*, 38–41, Boston, USA.
- POON, HOIFUNG, JANARA CHRISTENSEN, PEDRO DOMINGOS, OREN ETZIONI, RAPHAEL HOFFMANN, CHLOE KIDDON, THOMAS LIN, XIAO LING, MAUSAM, ALAN RITTER, STEFAN SCHOENMACKERS, STEPHEN SODERLAND, DAN WELD, FEI WU, and CONGLE ZHANG. 2010. Machine reading at the university of washington. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, 87–95, Los Angeles, USA.
- PRADHAN, SAMEER, LANCE RAMSHAW, MITCHELL MARCUS, MARTHA PALMER, RALPH WEISCHEDEL, and NIANWEN XUE. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, 1–27, Portland, USA.
- PUSTEJOVSKY, JAMES, PATRICK HANKS, ROSER SAURÍ, ANDREW SEE, ROBERT GAIZAUSKAS, ANDREA SETZER, DRAGOMIR RADEV, BETH SUNDHEIM, DAVID DAY, LISA FERRO, and MARCIA LAZO. 2003. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, 647–656, Lancaster, UK.
- PYYSALO, SAMPO, RUNE SÆTRE, JUN'ICHI TSUJII, and TAPIO SALAKOSKI. 2008. Why biomedical relation extraction results are incomparable and what to do about it. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine*, 149–152, Turku, Finland.
- RAVICHANDRAN, DEEPAK, and EDUARD HOVY. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the ACL*, 41–47, Philadelphia, USA.
- RAVIN, YAEL, and CLAUDIA LEACOCK. 2000. Polysemy: An overview. In *Polysemy*, ed. by Yael Ravin and Claudia Leacock. Oxford, UK: Oxford University Press.

- RESNIK, PHILIP. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-1995)*, 448–453, Montreal, Canada.
- RILOFF, ELLEN. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93)*, 811–816, Washington, D.C., USA.
- . 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, 1044–1049, Portland, USA.
- , and ROSIE JONES. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference*, 474–479, Orlando, USA.
- , and JESSICA SHEPHERD. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of the second Conference on Empirical Methods in Natural Language Processing (EMNLP 1997)*, 117–124, Providence, USA.
- RITTER, ALAN, STEPHEN SODERLAND, DOUG DOWNEY, and OREN ETZIONI. 2008. It's a contradiction – no, it's not: A case study using functional relations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, 11–20, Honolulu, USA.
- , STEPHEN SODERLAND, and OREN ETZIONI. 2009. What is this, anyway: Automatic hypernym discovery. In *Proceedings of the 2009 AAAI Spring Symposium on Learning by Reading and Learning to Read*, 88–93, Stanford, USA.
- ROARK, BRIAN, and EUGENE CHARNIAK. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, 1110–1116, Montreal, Canada.
- ROSENFELD, BENJAMIN, and RONEN FELDMAN. 2006. URES : an unsupervised web relation extraction system. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 667–674, Sydney, Australia.
- , and ———. 2007. Clustering for unsupervised relation identification. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*, 411–418, Lisbon, Portugal.

- SCHOENMACKERS, STEFAN, JESSE DAVIS, OREN ETZIONI, and DANIEL WELD. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, 1088–1098, Cambridge, USA.
- SCHÜTZE, HINRICH. 1998. Automatic word sense discrimination. *Computational Linguistics* 24.97–124.
- SEKINE, SATOSHI. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 731–738, Sydney, Australia.
- SHINYAMA, YUSUKE, and SATOSHI SEKINE. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2006 (HLT-NAACL 2006)*, 304–311, New York City, USA.
- SNOW, RION, DAN JURAFSKY, and ANDREW Y. NG. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17*, 1297–1304, Vancouver, Canada.
- , DANIEL JURAFSKY, and ANDREW Y. NG. 2006. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 801–808, Sydney, Australia.
- STEVENSON, MARK. 2007. Fact distribution in information extraction. *Language Resources and Evaluation* 40.183–201.
- , and MARK A. GREENWOOD. 2005. A semantic approach to IE pattern induction. In *Proceedings of the 43rd Annual Meeting of the ACL*, 379–386, Ann Arbor, USA.
- , and ——. 2006. Learning information extraction patterns using wordnet. In *Third International WordNet Conference (GWC-2006)*, 95–102, Jeju Island, Korea.
- , and ——. 2009. Dependency pattern models for information extraction. *Research on Language and Computation* 7.13–39.
- STORRER, ANGELIKA, and SANDRA WELLINGHOFF. 2006. Automated detection and annotation of term definitions in German text corpora. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2373–2376, Genoa, Italy.

- SUCHANEK, FABIAN M., GJERGJI KASNECI, and GERHARD WEIKUM. 2007. YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of the 16th International World Wide Web Conference (WWW 2007)*, 697–706, Banff, Canada.
- SUDO, KIYOSHI, SATOSHI SEKINE, and RALPH GRISHMAN. 2003. An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of the 41st Annual Meeting of the ACL*, 224–231, Sapporo, Japan.
- TAC, 2011. *Proposed Task Description for Knowledge-Base Population at TAC 2011*. Text Analysis Conference. Accessed 27 September 2011 from http://nlp.cs.qc.cuny.edu/kbp/2011/KBP2011_TaskDefinition.pdf.
- TAN, PANG-NING, MICHAEL STEINBACH, and VIPIN KUMAR. 2006. *Introduction to Data Mining*. Boston, USA: Addison-Wesley.
- TENGI, RANDEE I. 1998. Design and implementation of the WordNet lexical database and searching software. In *WordNet: An Electronic Lexical Database*, ed. by Christiane Fellbaum. Cambridge, USA: MIT Press.
- THELEN, MICHAEL, and ELLEN RILOFF. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 214–221, Philadelphia, USA.
- TURNEY, PETER. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22th International Conference on Computational Linguistics (COLING 2008)*, 905–912, Manchester, UK.
- TURNEY, PETER D. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning*, 491–502, Freiburg, Germany.
- VAPNIK, VLADIMIR N. 1995. *The Nature of Statistical Learning Theory*. New York, USA: Springer-Verlag.
- WEISS, GARY M. 2004. Mining with rarity: A unifying framework. *SIGKDD Explorations Newsletter* 6.7–19.
- WIDDOWS, DOMINIC, and BEATE DOROW. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, 1093–1099, Taipei, Taiwan.
- WILKS, YORICK A., BRIAN M. SLATOR, and LOUISE M. GUTHRIE. 1996. *Electric Words: Dictionaries, Computers, and Meanings*. Cambridge, USA: MIT Press.

- WITTEN, IAN H., and EIBE FRANK. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann series in data management systems. San Francisco, USA: Morgan Kaufmann Publishers, 2nd edition.
- WU, FEI, and DANIEL S. WELD. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007)*, 41–50, Lisbon, Portugal.
- , and ———. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the ACL*, 118–127, Uppsala, Sweden.
- XU, FEIYU, HANS USZKOREIT, and HONG LI. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of the 45th Annual Meeting of the ACL*, 584–591, Prague, Czech Republic.
- XU, RONG, ALEXANDER A. MORGAN, AMAR DAS, and ALAN GARBER. 2009. Investigation of unsupervised pattern learning techniques for bootstrap construction of a medical treatment lexicon. In *Proceedings of the BioNLP 2009 Workshop*, 63–70, Colorado, USA.
- YAKUSHIJI, AKANE, YUSUKE MIYAO, TOMOKO OHTA, YUKA TATEISI, and JUN'ICHI TSUJII. 2006. Automatic construction of predicate-argument structure patterns for biomedical information extraction. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 284–292, Sydney, Australia.
- YAMADA, ICHIRO, TIMOTHY BALDWIN, HIDEKI SUMIYOSHI, and NOBUYUKI YAGI. 2007. Automatic acquisition of qualia structure from corpus data. *IEICE Transactions on Information and Systems* E90-D.1534–1541.
- , KENTARO TORISAWA, JUN'ICHI KAZAMA, KOW KURODA, MASAKI MURATA, STIJN DE SAEGER, FRANCIS BOND, and ASUKA SUMIDA. 2009. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, 929–937, Singapore.
- YAN, YULAN, NAOAKI OKAZAKI, YUTAKA MATSUO, ZHENGLU YANG, and MITSURU ISHIZUKA. 2009. Unsupervised relation extraction by mining Wikipedia texts using information from the web. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 1021–1029, Suntec, Singapore.
- YANGARBER, ROMAN. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the ACL*, 343–350, Sapporo, Japan.

- YAO, LIMIN, SEBASTIAN RIEDEL, and ANDREW MCCALLUM. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, 1013–1023, Cambridge, USA.
- YAP, WILLY, and TIMOTHY BALDWIN. 2009. Experiments on pattern-based relation learning. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, 1657–1660, Hong Kong, China.
- YATES, ALEXANDER, and OREN ETZIONI. 2007. Unsupervised resolution of objects and relations on the web. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2007 (HLT-NAACL 2007)*, 121–130, Rochester, USA.
- ZELENKO, DMITRY, CHINATSU AONE, and ANTHONY RICHARDELLA. 2003. Kernel methods for relation extraction. *Machine Learning Research* 3.1083–1106.
- ZHANG, MIN, JIAN SU, DANMEI WANG, GUODONG ZHOU, and CHEW LIM TAN. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005)*, 378–389, Jeju Island, Republic of Korea.
- ZHOU, GUODONG, JIAN SU, JIE ZHANG, and MIN ZHANG. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the ACL*, 427–434, Ann Arbor, USA.
- ZHU, JUN, ZAIQING NIE, XIAOJIANG LIU, BO ZHANG, and JI-RONG WEN. 2009. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th International World Wide Web Conference (WWW 2009)*, 101–110, Madrid, Spain.

Appendix A

Examples of patterns extracted from MINIPAR output

<i>noun</i> ₁	<i>noun</i> ₂	Dependency path between <i>noun</i> ₁ and <i>noun</i> ₂	
		Satellites for <i>noun</i> ₁	Satellites for <i>noun</i> ₂
<i>composer</i>	<i>franz schubert</i>	dtd--N:mod:Prep,such as,such as,Prep:pcomp-n:N N:mod:A,prominent	N:lex-mod:U,franz
<i>composer</i>	<i>frederic chopin</i>	dtd--N:mod:Prep,such as,such as,Prep:pcomp-n:N N:mod:A,prominent	N:lex-mod:U,frederic;;N:punc:U,,,;N:punc:U,and
<i>composer</i>	<i>ludwig van beethoven</i>	dtd--N:mod:Prep,such as,such as,Prep:pcomp-n:N N:mod:A,prominent	N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,,
<i>composer</i>	<i>skill</i>	ud--N:d.n:U,d.n,d.n,U:d.n:A,renowned,renowned,A:mod:Prep,for,for,Prep:pcomp-n:N N:mod:A,prominent	N:gen:N
<i>composer</i>	<i>wolfgang amadeus mozart</i>	dtd--N:mod:Prep,such as,such as,Prep:pcomp-n:N N:mod:A,prominent	N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,,

continued on next page ...

<i>franz schubert</i> <i>composer</i>	dbu--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,franz N:mod:A,prominent
<i>franz schubert</i> <i>composer</i>	ud--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,franz N:mod:A,prominent
<i>franz schubert</i> <i>frederic chopin</i>	dbu--N:conj:N N:lex-mod:U,franz N:lex-mod:U,frederic;;N:punc:U,,,;N:punc:U,and
<i>franz schubert</i> <i>frederic chopin</i>	ud--N:conj:N N:lex-mod:U,franz N:lex-mod:U,frederic;;N:punc:U,,,;N:punc:U,and
<i>franz schubert</i> <i>ludwig van beethoven</i>	dbu--N:conj:N N:lex-mod:U,franz N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,,
<i>franz schubert</i> <i>ludwig van beethoven</i>	ud--N:conj:N N:lex-mod:U,franz N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,,

continued on next page ...

<i>franz schubert</i> <i>wolfgang amadeus mozart</i>	dbu--N:conj:N N:lex-mod:U,franz N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,,
<i>franz schubert</i> <i>wolfgang amadeus mozart</i>	ud--N:conj:N N:lex-mod:U,franz N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,,
<i>frederic chopin</i> <i>composer</i>	dbu--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,frederic;;N:punc:U,;;N:punc:U,and N:mod:A,prominent
<i>frederic chopin</i> <i>composer</i>	ud--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,frederic;;N:punc:U,;;N:punc:U,and N:mod:A,prominent
<i>frederic chopin</i> <i>franz schubert</i>	dtd--N:conj:N N:lex-mod:U,frederic;;N:punc:U,;;N:punc:U,and N:lex-mod:U,franz
<i>frederic chopin</i> <i>ludwig van beethoven</i>	dbu--N:conj:N N:lex-mod:U,frederic;;N:punc:U,;;N:punc:U,and N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,,

continued on next page ...

<i>frederic chopin</i> <i>ludwig van beethoven</i>	ud--N:conj:N N:lex-mod:U,frederic;;N:punc:U,;;N:punc:U,and N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,,
<i>frederic chopin</i> <i>wolfgang amadeus mozart</i>	dbu--N:conj:N N:lex-mod:U,frederic;;N:punc:U,;;N:punc:U,and N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,,
<i>frederic chopin</i> <i>wolfgang amadeus mozart</i>	ud--N:conj:N N:lex-mod:U,frederic;;N:punc:U,;;N:punc:U,and N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,,
<i>keyboard</i> <i>skill</i>	dbu--N:pcomp-n:Prep,on,on,Prep:mod:N N:gen:N
<i>keyboard</i> <i>skill</i>	ud--N:pcomp-n:Prep,on,on,Prep:mod:N N:gen:N
<i>keyboard</i> <i>their</i>	ud--N:pcomp-n:Prep,on,on,Prep:mod:N,N:gen:N

continued on next page ...

<i>ludwig van beethoven</i> <i>composer</i>	dbu--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,, N:mod:A,prominent
<i>ludwig van beethoven</i> <i>composer</i>	ud--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,, N:mod:A,prominent
<i>ludwig van beethoven</i> <i>franz schubert</i>	dtd--N:conj:N N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,, N:lex-mod:U,franz
<i>ludwig van beethoven</i> <i>frederic chopin</i>	dtd--N:conj:N N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,, N:lex-mod:U,frederic;;N:punc:U,,,;N:punc:U,and
<i>ludwig van beethoven</i> <i>wolfgang amadeus mozart</i>	dtd--N:conj:N N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,, N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,,
<i>skill</i> <i>composer</i>	ud--N:pcomp-n:Prep,for,for,Prep:mod:A,renowned,renowned,A:d.n:U,d.n,d.n,U:d.n:N N:gen:N N:mod:A,prominent

continued on next page ...

<i>skill</i>	<i>keyboard</i>	N:gen:N dtd--N:mod:Prep,on,on,Prep:pcomp-n:N
<i>skill</i>	<i>their</i>	dtd--N:gen:N
<i>their</i>	<i>keyboard</i>	ud--N:gen:N,N:mod:Prep,on,on,Prep:pcomp-n:N
<i>their</i>	<i>skill</i>	dbu--N:gen:N
<i>their</i>	<i>skill</i>	ud--N:gen:N
<i>wolfgang amadeus mozart</i> <i>composer</i>		dbu--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,, N:mod:A,prominent

continued on next page ...

<i>wolfgang amadeus mozart</i> <i>composer</i>	ud--N:pcomp-n:Prep,such as,such as,Prep:mod:N N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,, N:mod:A,prominent
<i>wolfgang amadeus mozart</i> <i>franz schubert</i>	dtd--N:conj:N N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,, N:lex-mod:U,franz
<i>wolfgang amadeus mozart</i> <i>frederic chopin</i>	dtd--N:conj:N N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,, N:lex-mod:U,frederic;;N:punc:U,,;N:punc:U,and
<i>wolfgang amadeus mozart</i> <i>ludwig van beethoven</i>	dbu--N:conj:N N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,, N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,,
<i>wolfgang amadeus mozart</i> <i>ludwig van beethoven</i>	ud--N:conj:N N:lex-mod:U,wolfgang;;N:lex-mod:U,amadeus;;N:punc:U,, N:lex-mod:U,ludwig;;N:lex-mod:U,van;;N:punc:U,,

Table A.1: List of patterns extracted from a MINIPAR output in Figure 3.4. Note the case-folding of the nouns

Appendix B

List of the Seed Terms for each Category used in Bootstrapping

Categories	Seed Terms
ACTION	<i>installation, violence, rehabilitation, exploration, renewal, interaction, completion, destruction, incorporation, degradation</i>
ACTIVITY	<i>search, planning, survey, monitoring, gaming, hunting, swimming, fraud, carpentry, campaigning</i>
ANIMAL	<i>cat, fox, wolf, bull, deer, frog, lamb, panther, camel, koala</i>
AREA	<i>atlanta, denver, berlin, edinburgh, prague, beijing, baghdad, stockholm, belfast, lisbon</i>
ARTIFACT	<i>magazine, ball, flag, banner, knife, shield, mask, hammer, sword, bullet</i>
ARTIST	<i>gogh, picasso, rivera, dali, rembrandt, albers, giotto, magritte, mondrian, cohan</i>
AUTHOR	<i>wells, shakespeare, poet, kaufman, austen, columnist, screenwriter, salinger, lyricist, speechwriter</i>

continued on next page

Categories	Seed Terms
BEAST	<i>bear, wolf, bull, snake, elephant, dinosaur, mammoth, condor, gorilla, rhinoceros</i>
BELIEF	<i>god, magic, religion, judgment, catholicism, protestantism, methodism, monotheism, rationalism, egalitarianism</i>
BIRD	<i>falcon, goose, pigeon, hen, pelican, finch, mockingbird, songbird, cockatoo, shorebird</i>
BREAD	<i>toast, cracker, loaf, baguette, rusk, scone, croissant, brioche, matzo, chapati</i>
BUILDING	<i>theatre, bungalow, farmhouse, clubhouse, gazebo, schoolhouse, skyscraper, morgue, dormitory, boardinghouse</i>
CENTER	<i>london, washington, atlanta, moscow, prague, beijing, bangkok, bucharest, kabul, zagreb</i>
CHEMICAL	<i>polymer, bayer, carbohydrate, sulfate, explosive, serotonin, hydrochloride, manure, epinephrine, freon</i>
CITY	<i>arlington, leeds, baghdad, milan, brisbane, honolulu, warsaw, oslo, sofia, porto</i>
CLOTHING	<i>hat, bra, diaper, vest, pantyhose, outerwear, swimwear, hosiery, shawl, turban</i>
COMPOSER	<i>schubert, verdi, purcell, dvorak, stravinsky, shostakovich, bruckner, schoenberg, rachmaninoff, palestrina</i>
CONCEPT	<i>point, value, rule, feature, abstract, parameter, aspect, limitation</i>
CONDITION	<i>injury, poverty, allergy, hepatitis, tuberculosis, deficiency, absence, fibrosis, dyslexia, eczema</i>
CONTAINER	<i>kit, basket, tank, bin, luggage, vase, crate, jug, urn, ladle</i>
COST	<i>premium, fee, salary, benefit, tuition, refund, expense, annuity, deduction, surcharge</i>

continued on next page

Categories	Seed Terms
CREATURE	<i>swan, mosquito, dragonfly, squirrel, feline, clam, mammal, snail, armadillo, cockroach</i>
CRIMINAL	<i>thief, burglar, gangster, murderer, traitor, hoodlum, rapist, rustler, mobster, looter</i>
DEITY	<i>goddess, rama, minerva, ganesh, odin, kami, maitreya, tyche, avalokiteshvara, amaterasu</i>
DEVICE	<i>telescope, compass, simulator, compressor, thermometer, keypad, sprinkler, microprocessor, odometer, magnetometer</i>
DISEASE	<i>cold, diabetes, aids, asthma, glaucoma, measles, cirrhosis, bronchitis, dengue, pertussis</i>
DISTRICT	<i>america, texas, china, downtown, malaysia, indianapolis, township, athens, springfield, beijing</i>
DOCUMENT	<i>bill, resolution, draft, charter, brief, authorization, statute, cheque, pact, writ</i>
DOG	<i>terrier, hound, bulldog, spaniel, husky, beagle, papillon, doberman, kuvasz, schipperke</i>
DRUG	<i>lipitor, diazepam, fosamax, narcotic, painkiller, methamphetamine, salve, prednisone, penicillin, phenobarbital</i>
ENTITY	<i>board, system, technology, collection, executive, resource, congress, parliament, writer, pharmacy</i>
EQUIPMENT	<i>telephone, modem, peripheral, spacecraft, audiovisual, vcr, circuitry, oscilloscope, paraphernalia</i>
EVENT	<i>marketing, visit, survey, musical, walk, concert, tournament, deal, lecture, catering</i>
FEELING	<i>passion, hate, rage, shame, ego, nostalgia, sorrow, appetite, angst, sentiment</i>
FISH	<i>salmon, barracuda, marlin, minnow, chub, loach, goby, sculpin, parrotfish, scorpionfish</i>

continued on next page

Categories	Seed Terms
FOOD	<i>rice, alcohol, milk, honey, butter, potato, pickle, muffin, meat-ball, porridge</i>
FURNITURE	<i>bed, desk, cabinet, counter, sofa, bench, booth, crib, wardrobe, pew</i>
GARMENT	<i>shirt, jacket, coat, underwear, sweater, scarf, veil, overcoat, burqa, niqab</i>
GROUP	<i>business, company, public, community, institute, team, union, congress, convention, orchestra</i>
GROUPING	<i>network, party, church, category, grade, population, settlement, clan, formation, profession</i>
HORMONE	<i>testosterone, estrogen, progesterone, estradiol, melatonin, androgen, oxytocin, glucocorticoid, vasopressin, estriol</i>
IDEA	<i>policy, strategy, rule, composite, attribute, compound, intrigue, entitlement, regimen, suspicion</i>
INDIVIDUAL	<i>member, european, king, baby, professional, german, artist, marine, designer, lawyer</i>
INSTRUMENT	<i>watch, telescope, goggles, thermometer, barometer, sundial, forceps, speedometer, machete, hydrometer</i>
ISLAND	<i>timor, aruba, oahu, borneo, formosa, sulawesi, luzon, mindanao, dodecanese, hispaniola</i>
KNOWLEDGE	<i>psychology, mathematics, christianity, interpretation, confirmation, suggestion, appreciation, consideration, epistemology, memorization</i>
LANGUAGE	<i>english, french, chinese, greek, celtic, finnish, khmer, yiddish, afrikaans, pidgin</i>
LEADER	<i>president, executive, prince, chairman, administrator, governor, duke, castro, dayan, gaddafi</i>

continued on next page

Categories	Seed Terms
LOCATION	<i>state, city, west, australia, area, russia, zone, interior, water-front, birthplace</i>
MAMMAL	<i>dog, mouse, monkey, deer, rabbit, sheep, rat, zebra, rhinoceros, porpoise</i>
MATERIAL	<i>oil, clay, gasoline, gravel, cellulose, styrofoam, dna, monomer, rna, plasticizer</i>
METAL	<i>iron, platinum, aluminum, copper, zinc, cobalt, uranium, chromium, cadmium, molybdenum</i>
MIDDLE	<i>boston, dublin, tokyo, beijing, vienna, providence, warsaw, riga, freetown, midfield</i>
MUSIC	<i>opera, boogie, choral, medley, ballad, fantasia, modulation, refrain, polyphony, serialism</i>
MUSICIAN	<i>singer, starr, crosby, soprano, vocalist, soloist, mccartney, percussionist, flutist, violist</i>
OBJECT	<i>house, system, road, computer, sun, website, japan, museum, jewelry, teacher</i>
OCCURRENCE	<i>time, case, start, death, advance, cause, factor, accident, crash, spread</i>
ORGANISM	<i>human, man, fish, mouse, virus, worm, primate, parasite, giardia, microorganism</i>
ORGANIZATION	<i>team, bureau, nation, nasa, nato, ngo, opec, unicef, imf, faa</i>
PAINTER	<i>stella, gauguin, goya, cezanne, giotto, delacroix, seurat, colorist, brueghel, poussin</i>
PART	<i>water, iron, chocolate, waste, copper, cleaner, milk, dust, ivory, fur</i>
PEOPLE	<i>labour, population, aged, blind, deaf, wounded, gentry, bikers, aristocracy, intelligentsia</i>

continued on next page

Categories	Seed Terms
PERFORMER	<i>actor, singer, actress, harlequin, guitarist, minstrel, soloist, bassist, flutist</i>
PERIOD	<i>february, friday, week, winter, season, festival, birthday, weekend, evening, easter</i>
PERSON	<i>japanese, master, senior, douglas, sarah, duke, dealer, citizen, cowboy, surgeon</i>
PHILOSOPHER	<i>socrates, machiavelli, hegel, sartre, derrida, wittgenstein, thales, kierkegaard, diderot, origen</i>
PLAYER	<i>paige, ashe, batter, quarterback, grandmaster, hitter, shortstop, infielder, mvp, halfback</i>
POET	<i>shakespeare, milton, brooke, horace, goethe, plath, auden, rilke, pindar, yevtushenko</i>
PORT	<i>seattle, harbor, liverpool, jacksonville, helsinki, cannes, seville, bellingham, acapulco, palermo</i>
PORTION	<i>paper, steel, word, term, salt, paint, fiber, pad, context, antibody</i>
PROFESSIONAL	<i>attorney, doctor, physician, educator, intern, pharmacist, pathologist, midwife, clinician, pediatrician</i>
PUBLICATION	<i>magazine, catalog, handbook, encyclopedia, quarterly, anthology, booklet, pictorial, guidebook, bestseller</i>
QUALITY	<i>justice, authority, charm, difference, ethnicity, loyalty, fairness, competency, disadvantage, disparity</i>
QUANTITY	<i>standard, days, fall, four, cash, third, month, morning, scale, temperature</i>
REGION	<i>county, front, interior, kent, paradise, outdoors, shoreline, transylvania, troposphere, bridgehead</i>
RELATION	<i>insurance, water, gold, mortgage, payment, connection, competition, brotherhood, kinship, bribe</i>

continued on next page

Categories	Seed Terms
RELATIVE	<i>mother, wife, daughter, spouse, aunt, grandfather, grandmother, sibling, ancestor, descendant</i>
RIVER	<i>thames, loire, klamath, yangtze, susquehanna, zambezi, orinoco, elbe, rappahannock, amur</i>
ROOM	<i>bar, classroom, cabin, lobby, kitchenette, boudoir, restroom, playroom, workroom, sickbay</i>
SCIENCE	<i>biology, mathematics, geography, psychiatry, physiology, pathology, dentistry, pediatrics, neuroscience, ophthalmology</i>
SCIENTIST	<i>newton, researcher, chemist, oppenheimer, biologist, physicist, statistician, ecologist, epidemiologist, toxicologist</i>
SOMEBODY	<i>man, child, player, friend, son, doctor, superior, producer, mama, journalist</i>
SOMEONE	<i>general, president, child, author, adult, master, commissioner, mayor, employee, soldier</i>
SPORT	<i>golf, rugby, cycling, skating, gymnastics, badminton, skateboarding, sumo, croquet, hurling</i>
STATE	<i>texas, florida, ohio, michigan, kansas, hawaii, queensland, bavaria, karnataka, hebei</i>
STATEMENT	<i>notice, commentary, announcement, claim, provision, manifesto, excuse, remark, restatement, misinterpretation</i>
STATUS	<i>marriage, championship, bishop, standing, fame, cardinal, reputation, emphasis, anonymity, notoriety</i>
STRUCTURE	<i>hotel, auditorium, tent, shelter, fence, greenhouse, coliseum, masonry, furnace, supermarket</i>
SUBSTANCE	<i>carbonate, nitrate, contaminant, histamine, glutamate, hemoglobin, polystyrene, prostaglandin, chloroform, ricin</i>
TERRITORY	<i>state, florida, iraq, indonesia, malaysia, ukraine, ghetto, vanuatu, macao, outskirt</i>

continued on next page

Categories	Seed Terms
THOUGHT	<i>component, syndrome, concept, inspiration, suggestion, constant, compound, impression, ethic, preoccupation</i>
TOWN	<i>canterbury, blackpool, redding, hometown, lawton, jonesboro, ayr, salina, pittsfield, sitka</i>
TRANSPORT	<i>truck, ship, boat, motorcycle, jeep, helicopter, carriage, tanker, spaceship, rickshaw</i>
UNIT	<i>inch, euro, mile, meter, yard, cent, acre, franc, horsepower, kilogram</i>
VEHICLE	<i>car, truck, yacht, wagon, tractor, submarine, destroyer, spaceship, hovercraft, warship</i>
WOMAN	<i>miss, rachel, belle, rebecca, madam, bimbo, lulu, lass, crone, courtesan</i>
WORKER	<i>potter, maid, fisherman, miner, seaman, bartender, machinist, blacksmith, engraver, reservist</i>
WORK	<i>medical, survey, therapy, census, inspection, scrutiny, physiotherapy, endeavor, chore, housework</i>
WRITER	<i>byron, emerson, poe, austen, melville, journalist, voltaire, chekhov, pushkin, dramatist</i>

Table B.1: List of the seed terms for the SEMANTIC CATEGORIES as sorted by their frequency in the GOOGLE WEB 1T corpus