Towards the Utilisation of IS to Predict Word Order in Tagalog

A thesis presented
by
Meladel Mistica

to
The Department of Computer Science and Software Engineering
in partial fulfillment of the requirements
for the degree of
Postgraduate Diploma

University of Melbourne
Melbourne, Australia
October 2008
Towards the Utilisation of IS to Predict Word Order in Tagalog

Abstract

The aim of this research is to identify the meaningful units in a sentence as a basis for further work in textual analysis in Tagalog. The study is a precursar to utilising discourse-based contextual information to predict word order in Tagalog, a language exhibiting free word order, native to the Philippines.
# Contents

Title Page ................................................................. i
Abstract ................................................................. ii
Table of Contents .................................................... iii
List of Tables ........................................................ vi
List of Figures ........................................................ vi

1 Introduction .......................................................... 1
   1.1 Information Structure ........................................ 1
   1.2 Towards Analysing Information Structure ................. 2
   1.3 Outline of Thesis ............................................. 3

2 Background .......................................................... 4
   2.1 Constituency ................................................... 4
   2.2 Tagalog Voice Marking ...................................... 5
   2.3 Word Order ................................................... 6
   2.4 Tagalog Information Structure ............................. 7

3 Review of Literature ............................................... 9
   3.1 Introduction ................................................... 9
   3.2 Shallow Parsing and SRL .................................. 9
   3.3 Natural Language Generation - Word Order ............. 11

4 The Data ............................................................. 13
   4.1 Part-of-speech Mark-up .................................... 13
   4.2 Predicate Mark-up .......................................... 14
   4.3 Satellite Mark-up .......................................... 16
   4.4 Mark-up Language ......................................... 17
   4.5 Annotation .................................................. 18

5 Experiments ........................................................ 19
   5.1 Introduction ................................................ 19
   5.2 Architecture ................................................. 19
   5.3 The Tools ................................................... 20
5.3.1 Xfst and lexc .................................................. 20
5.3.2 Crf++ ............................................................ 21
5.4 Preprocessing: The Morphological Analyser .................... 22
5.5 Evaluation Metrics ............................................... 24
5.6 The Three Steps .................................................. 24
  5.6.1 POS Tagger .................................................. 24
  5.6.2 Predicate Predictor ......................................... 25
  5.6.3 Satellite Identifier .......................................... 28
5.7 Discussion ......................................................... 30

6 Final Remarks ....................................................... 32
  6.1 Discussion ...................................................... 32
  6.2 Conclusion ...................................................... 33
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Part of speech tags used in this research</td>
<td>13</td>
</tr>
<tr>
<td>4.2</td>
<td>Satellite counts in our corpus</td>
<td>18</td>
</tr>
<tr>
<td>5.1</td>
<td>Morphological Features</td>
<td>23</td>
</tr>
<tr>
<td>5.2</td>
<td>POS tagger results – unigram and bigram with ‘windowing’</td>
<td>25</td>
</tr>
<tr>
<td>5.3</td>
<td>Pred Picker - leave-one-out using CRFs</td>
<td>26</td>
</tr>
<tr>
<td>5.4</td>
<td>Results of predicate prediction with variable window size</td>
<td>26</td>
</tr>
<tr>
<td>5.5</td>
<td>Results for predicate predictor by evaluating as tokens rather than chunks</td>
<td>27</td>
</tr>
<tr>
<td>5.6</td>
<td>Conflating the main and subordinate predicate distinction</td>
<td>27</td>
</tr>
<tr>
<td>5.7</td>
<td>Predicate predictor results using morphological features - Window = 5</td>
<td>28</td>
</tr>
<tr>
<td>5.8</td>
<td>Predicate predictor results using morphological features - Window = 3</td>
<td>29</td>
</tr>
<tr>
<td>5.9</td>
<td>Satellite identification results using morphological features - Window = 3</td>
<td>29</td>
</tr>
<tr>
<td>5.10</td>
<td>Satellite identification results using morphological features - Window = 5</td>
<td>29</td>
</tr>
<tr>
<td>5.11</td>
<td>Satellite identification results using predicate primers and part-of-speech</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>tags as features - Window = 3 &amp; 5</td>
<td>30</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Segmenting the sentence ....................................................... 2
4.1 Predicate and subordinate predicate example ............................. 14
4.2 An example of the mark-up of a main clause predicate (PRD) and a subordinate clause predicate (PRD-SUB) .......................... 15
4.3 Avoiding disjointed PRD mark-up .......................................... 15
4.4 An example of predicate mark-up which includes negation .......... 16
4.5 An example of a mark-up of a quotative verb complement .......... 17
4.6 Multiple predicates ............................................................. 18
5.1 The pipeline architecture of our system ................................. 19
5.2 A simple finite state transducer allowing singular and plural nouns 20
Chapter 1
Introduction

Natural language generation aims to produce natural language output from some underlying representation that encodes the intended meaning. One of the challenges faced in this area is generating output that is coherent and structured in a meaningful way within the text, for the given context.

This study is a precursor to achieving this end. The aim of this research is to identify the steps required in order to utilise, discourse-based contextual information in predicting word order in Tagalog, a language exhibiting free word order, native to the Philippines.

1.1 Information Structure

When we deviate from the expected word order a language, this usually conveys information beyond the literal meaning of the sentence. This can encode information such as the attitude of the speaker and the assumptions made about the world or situational knowledge of the hearer, or as means of emphasising some particular information in the sentence.

The way in which we package or present this information is referred to as Information Structure, also known as Information Packaging (Halliday 1967; Engdahl and Vallduvi ). It is a means of explaining the motivation behind the deviation from the standard or unmarked way of expressing something. It also explains the form, or the way we express certain things in the sentence. For example, information that is already known to the speaker need not be so explicitly encoded.

\begin{enumerate}
\item \textbf{a.} I met Mary at the party.
\item \textbf{b.} Mary, I met at the party.
\item \textbf{c.} It was Mary I met at the party.
\end{enumerate}

In (1.1a.), the sentence is does not focus on any one element in the sentence, but rather focuses on the proposition as a whole. In (1.1b.), \textit{Mary} is the focus and has been
preposed. We can imagine this sentence being an answer to the question: *Who did you meet at the party?*. The third construction is an example of contrastive or identificational focus and could be an answer to: *You met Jane at the party, right?* (Halliday 1967).

Tagalog encodes the notion of *focus* morpho-syntactically (see Section 2). In addition to having syntactic means of highlighting important information in the sentence, like the shuffling of sentence components in English, Tagalog has ways of marking important information on the phrases themselves. This morpho-syntactic phenomenon is called voice or focus marking.

1.2 Towards Analysing Information Structure

In order to model information structure to assist in predicting word order, we need to define and identify the required building blocks. This study aims to define these blocks or constituents in Tagalog, and determine the required steps and necessary tools to extract them.

The ultimate goal is to utilise information beyond the sentence, that is discourse level information, however, we aim to see how this affects sentence level phenomena, such as word order. Therefore we need to be able to identify the parts of the sentence, or its *constituency*.

![Figure 1.1: Segmenting the sentence](image)

By *constituency*, we refer to the phrasal chunks in a sentence, such as the participants in the sentence or the things that are affected by some action encoded in an event. For example, Figure 1.1 shows which parts of the sentence we would like to identify. In bold, there are the predicates of the sentence, off which the components
‘hang’, such as the subject and the complements of the verb. Not only do we want to identify these components, but we would also like to be able to identify how certain elements are related in the sentence, if at all.

1.3 Outline of Thesis

The thesis is structured as follows: Chapter 2, entitled Constituency, we present the syntax of Tagalog as is relevant to the analysis of information structure. Specifically, we outline Tagalog verbal morphology and word order and see that discourse notions such as focus are encoded morphologically in the language. We also see the extent to which Tagalog exhibits free word order, and the restrictions placed on this ‘freedom of order’.

In Chapter 3, we present work conducted on Tagalog (Filipino) in Language Technology, such as part-of-speech tagging (dela Vega et al. 2002). In our experiments, we apply methods relevant to the fields of semantic role labelling and shallow parsing (or chunking) and present information on this area of research.

Chapter 4 outlines the data used in this study, and how it is developed. In particular we detail the mark up scheme developed for Tagalog.

The design and implementation of the experiments to extract discourse entities are presented in Chapter 5. We also introduce the tools required to build the system and present our initial evaluation. Finally, Chapter 6 concludes this study and discusses future work in this area.
Chapter 2

Background

Tagalog is an Austronesian language of the Malayo-Polynesian branch. It is the basis of the national language of the Philippines, Filipino (also Pilipino), the only difference being that Filipino incorporates terms from other regional languages (Gordon 2005).

Tagalog is a verb initial language, with relatively free word order of the verbal arguments. There are some restrictions on this ‘free word order’, the details of which are outlined in the Section 2.3. In the following sections, we define the notion of constituent in Tagalog, and then see how highlighting of elements in a sentence is achieved morpho-syntactically.

2.1 Constituency

There are three case markers in Tagalog, which are by convention written as prepositions. These markers normally prepose phrasal elements that ‘hang off’ or are dependent on the verb.

• ANG
• NG
• SA

According to Kroeger (1993), SA, is ordinarily used for goals, recipients, locations and definite objects, while NG marks possessors, actors, instruments and indefinite objects.

There is also a system of verbal affixation that enables coreferentiality with a nominal that has special status in the sentence. The function of the ANG-marked nominal is best explained in terms of Tagalog’s voice-marking system. Tagalog has a rich array of verbal morphology, which marks a ‘particular dependent as special’ (Baldridge 2002); the ‘special dependent’ in the sentence is the ANG-marked verb. This system of marking on the verb is called voice or focus marking.
2.2 Tagalog Voice Marking

According to Kroeger (1993), there are 5 major voice types in Tagalog:

- **Actor Voice (av)**
- **Patient/Object Voice (ov)**
- **Dative/Locative Voice (dv)**
- **Instrumental Voice (iv)**
- **Benefactive Voice (bv)**

Voice is marked on the verb with affixes. This focus or voice marking exhibited in the verb reflects the role of the **ang**-marked constituent, as seen in the sentences below from Kroeger (1993). The infix *-um-* in the first sentence below, indicates that we have an actor voice verb, which means that the agent, person or thing responsible for the action is marked with an **ang**.

\[(av)\] Bumili ang lalake ng isda sa tindahan.  
AV.buy NOM man GEN fish DAT store  
'The man bought the fish at the store'

\[(ov)\] Binili ng lalake ang isda sa tindahan.  
OV.buy GEN man NOM fish DAT store  
'The man bought fish at the store.'

\[(dv)\] Binalihan ng lalake ng isda ang tindahan.  
DV.buy GEN man GEN fish NOM store  
'The man bought fish at the store.'

\[(iv)\] Ipinambili ng lalake ng isda ang pera.  
IV.buy GEN man GEN fish NOM money  
'The man bought fish with the money.'

\[(bv)\] Bibili ng lalake ng isda ang bata.  
BV.buy GEN man GEN fish NOM child  
'The man bought fish for the child.'

An ov-marked verb indicates that the object or thing affected by the action must have the special status of being marked with **ang**, while verbs that have DV, IV and BV verbal markings, as seen above, must have the direct object, instrument, or recipient, bearing the special case marking.
Chapter 2: Background

2.3 Word Order

The relative order of nominal phrases in Tagalog is considered free, as exemplified in the following sentences (Schachter, 1972 as cited in Kroeger, 1993).

(2.1) *Nagbigay ng libro sa babae ang lalaki*

gave GEN book DAT woman ANG lalaki

“The man gave the woman a book”

*Nagbigay ng libro ang lalaki sa babae.

Nagbigay sa babae ng libro ang lalaki.

Nagbigay sa babae ang lalaki ng libro.

Nagbigay ang lalaki sa babae ng libro.

Nagbigay ang lalaki ng librosa babae.

Although there is a lot of flexibility in word order, Kroeger claims that this is not completely arbitrary. He provides three principles that interact in order to determine a preferred ordering for non-pronominal arguments (Kroeger 1993):

(2.2) a. the Actor phrase tends to precede all other arguments.

b. the N which bears nominative case tends to follow all other arguments

c. ‘heavier’ NPs tend to follow ‘lighter’ NPs

In addition, Rackowski (2002) gives examples of there being a preference for nominative marked, or ANG-marked arguments to be placed after all other phrases, as seen below. There is not an agreed canonical word order in Tagalog.

Clitic Restrictions and Word Order

Clitics are reduced word forms that are usually dependent on adjacent words, like the possessive ‘s in English. In Tagalog, clitics are restricted to occupy a place in the sentence, namely the second position. There are two types of clitics in Tagalog which affect word order: pronominal clitics and adverbial clitics.

Example 2.3, from Schachter (1972), is an example of how pronominal clitics affect word order.

(2.3) a. *Hindi siya masaya ngayon.*

   NEG 3.SG.NOM happy now

   ‘He isn’t happy today’

b. *Hindi masaya si Ben ngayon.*

   NEG happy NOM Ben now

   ‘Ben isn’t happy today.’
The full noun phrase *si Ben* in the second sentence comes after the verb *masaya*. In 2.3a., where the full noun phrase is replaced with the pronominal clitic *siya*, it can be seen that the verb *masaya* is displaced to allow for the pronominal clitic to reside in the second place position.

The ungrammaticality of the second sentence in the example below shows that this not merely a word order preference, but a constraint in the grammar.

(2.4) a. *Nakita* *siya* *ni* Pedro.
    see 3.SG.NOM GEN Pedro
    ‘Pedro saw him’

b. *Nakita* *ni* Pedro *siya*.
    see GEN Pedro 3.SG.NOM
    *‘Pedro saw him’

2.4 Tagalog Information Structure

In addition to highlighting or giving focus to a constituent in the sentence by way of verbal marking, Tagalog also has a means of preposing a constituent in the sentence to draw our attention to it. This is done via the use of the *ay* inversion. This has also been called left-dislocation in the literature (Nagaya 2006).

(2.5) a. *Binili* ko *ang* damit
    buy-OV 1sg.NG ANG dress
    “I bought the dress”

b. *Ang* damit *ay* binili *ko*
    ANG dress INV buy-OV 1sg.NG
    “I bought the dress”

In 2.5, the expected word order is shown in a., and b. exhibits the inverted form, with the ANG-marked constituent fronted. Constituents that do not exhibit the NG marking may be fronted in this manner. It is noted by Fox (1987) that the *ay*-inversion construction is more commonly used when initiating or introducing a new event or new theme to the discourse. After the initial introduction of this new event, then the *AY*-inversion does not appear as often (Fox 1987).

The notion of events or entities being given or new in the discourse also determines the way that we encode them. If we were to introduce a new idea or thing into the discourse we may be more explicit or descriptive about it. When we want to make mention of the already introduced entity then we have more choice about how we may refer to it, because it is now a known specific thing in the discourse. Therefore
we can refer to this entity with as a definite article or pronoun, such as this x that y or it.

In Tagalog, encoding definiteness has restrictions on the case marking a phrase may exhibit in the sentence (Adams and Manaster-Ramer 1988). For instance, the case marker NG cannot prepose definite objects. Definite elements can only be marked with ANG and SA. This in turn would mean that indefinite elements cannot be highlighted in the same way as other non-NG constituents, in that they cannot undergo ay-inversion.

In addition to voice marking encoding focus, and the addition of word order shuffling to highlight particular objects or participants in the discourse, there is also an interplay in Tagalog with the encoding of definiteness and the ability to highlight these elements syntactically, and morpho-syntactically. These interesting grammatical features of Tagalog make it a challenging case study for our current application.
Chapter 3

Review of Literature

3.1 Introduction

In order to predict the order in which the constituents in a Tagalog sentence may be generated, we must first identify and delimit these units in the sentences. The task of identifying predicates and their dependent phrases in Tagalog text is treated similarly to a semantic role labelling task. Semantic role labelling defines the two-fold task of identifying elements related to a single predicate in a clause and their categorisation, or the labelling of their relationship to the identified predicate. The labels assigned to the dependent of the predicate characterise the event depicted by identifying the WHO, WHAT, to WHOM, WHEN, and HOW of the clause or sentence (Marquez et al. 2008). In essence, these labels map out how to interpret the event and outline how the participants described are involved.

Identifying relevant elements in a sentence can be performed in a number of ways including parsing and shallow parsing (also known as chunking). Tagalog is relatively under-resourced in terms of tools and corpora for language technology related research, and so there are limitations on the techniques available for this task.

Research on parsing Tagalog is limited, and has tended to form part of a larger system such as a machine translation engine (Roxas and Borra 2000; Roxas et al. 2008). Part-of-speech taggers in Tagalog has gained some attention with the development of a template based tagger and a Hidden Markov Model (Roxas and Borra 2000), as well as a Visible Markov Model (dela Vega et al. 2002).

3.2 Shallow Parsing and SRL

The task of automatically labeling semantic roles has a number of uses in natural language processing and natural language understanding. As Gildea and Jurafsky (2002) point out, information about semantic roles can help disambiguate word sense, or alternatively assist in statistical machine translation and automatic text summa-
Chapter 3: Review of Literature

rization, by way of acting as an intermediate representation (Gildea and Jurafsky 2002).

SRL tasks involve two sub-processes, which can be conducted concurrently or sequentially: one of labelling and one of identifying the segments to label, or delimiting contiguous elements that are of interest. Identifying a sequence as a meaningful unit in a sentence can be done via chunking or shallow parsing. This can be regarded as a structured learning task where the ‘parsed’ sequence exhibits some basic internal structure.

The issues that surround the task of labelling is that of discovering the optimal and most meaningful set of labels which can be employed. There seems to be a trade-off between the descriptiveness or specificity of the label set and the comparability of the labels employed, which would result in having a more generalised or coarse set of semantic role labels.

This describes the dichotomy in the semantic role types employed in SRL: labels that are highly tuned to a particular verb or a group of verbs that describe the same event type; and labels that abstract from specific verb types and employ labels that can apply to a broad range of verbs (Gildea and Jurafsky 2002). Examples of abstract or generalised labels are AGENT and PATIENT, applicable to most verbs, and labels specific to only a verb or an event type are roles labels such as EATER and EATEN, being only relevant to verbs that involve the consumption of food.

Márquez et al. (2008) provide a summary of the three major computational lexicons of predicate-argument relations, which populate the cline between the abstract comparable labels and descriptive verb-specific labels. These resources, which are extensively utilised by the SRL community in their research and shared task experiments, are PropBank, FrameNet, and VerbNet.

PropBank overlays its semantic annotation scheme onto the parse structures of the Penn Treebank (Palmer et al. 2005). The semantic roles are developed on a verb-by-verb basis and are assigned numbered labels starting with Arg0, where Arg0 typically denotes an agent and Arg1 typical denotes a theme. Although having a more general labelling schema allows for some parallelism with other verbs, it was found that argument labels beyond Arg1 do not generalise well across other verbs (Zapirain et al. 2008)

The following example from PropBank (Palmer et al. 2005) illustrates the semantic roles defined for the verb love. It shows the boundaries of the arguments and their semantics labels, where Arg0 indicates the AGENT or ACTOR and Arg1 the PATIENT, It also indicates boundaries of the predicate.

\[(3.1) \begin{array}{c}
\text{Arg0}\, Paris \, ] \, \text{Pred} \, loved \, ] \, \text{Arg1}\, her \, ] \, \text{ArgM-TMP} \, at \, first \, sight]. \\
\text{But} \, \text{Arg0}\, I \, ] \, \text{Pred} \, loved \, ] \, \text{Arg1}\, turbans].
\]

FrameNet assigns semantic roles to arguments based on the event depicted. The assignment of the semantic roles is based on Fillmore’s characterisation of events
called case frames (Petruck 1996). The argument-predicate labels assigned are not directly comparable to verbs unless they describe a semantically-related event, in which case they would involve similar participants. For example, in a transaction event, such the case evoked by the verb *buy*, the situation involves a *BUYER*, *SELLER*, *GOODS*, and so on, as does the verb *sell*.

VerbNet is based largely on the Levin’s classification of verbs based on syntactic alternations (Kipper:2000). This method also clusters like verbs, but, unlike FrameNet, grouped verbs are not necessarily connected together because they are related events. The predicate-argument labels found in VerbNet reflect the types of participants involved and are not abstracted to more generic labels, like those used in PropBank.

### 3.3 Natural Language Generation - Word Order

The Nitrogen natural language generator (Langkilde and Knight 1998) for English combines symbolic and statistical knowledge. It constructs word lattices from its meaning representation and then allows the statistical post-processor to generate grammatical forms according to corpus-based statistical knowledge.

The system relies on its grammar representation which encodes semantic relations of the proposition. In order to account for the different ways of expressing the one concept, Nitrogen allows a *recasting* step that keeps semantic content the same but represents it in different ways in order to allow for variation in expression.

In generating the output string, Nitrogen relies on basic unigram and bigram frequencies. Unigrams give the generator information about what linguistic form a particular concept is normally realised while bigrams provide collocational information. (?) point out the limitations in the ngram approach to generation in that dependencies between non-contiguous words are not able to be captured.

Generating natural sounding text in a free word order language poses different issues to those encountered in a more structured language such as English. However, there are areas of English grammar where the units in the phrase are not as syntagmatically constrained, namely the order in which prenominal adjectives can be generated (Malouf 2000). The issue of generating a series of prenominal adjectives is not one of ensuring the grammaticality of the output, but generating a natural sounding one.

(Malouf 2000) uses a number of statistical machine learning techniques to predict the relative order of adjectives in the phrase. One technique used is the *direct evidence* method, which trains on attested adjective sequences, or adjective pairs. For a given pair of tokens, a and b, the order ⟨a,b⟩ or ⟨b,a⟩ is chosen depending on how often this relative sequence occurs in the text. The biggest problem encountered in his study was that often a sequence or pair adjective pairs could only be attested once in the text. A technique Malouf (2000) applies in overcoming this data sparseness problem was by developing the *positional probabilities* method. This technique does not rely
on finding a particular adjective pair in the training data, but looks at where they are most likely to occur independently of which other adjectives accompany it. In essence, he constructs conceptual adjective slots, in the case of adjective pairs, there are two, and this method gathers data on whether a particular adjective is more likely to appear in the first slot or second slot.
Chapter 4

The Data

For this study, we developed our own corpus. We marked up 2 short chapters from a narrative obtained from the Gutenberg Project\(^1\) called *Hiwaga ng Pagibig* (Nanong 1922), the unfortunate translation of which is *The Mystery of Love*.

The purpose of annotating the text is to identify the various components of each sentence that are considered important for the task. In this scheme, we ensure the predicates are identified along with their relevant dependent constituents, which we refer to as the predicates’ *satellites*.

### 4.1 Part-of-speech Mark-up

We develop a set of 5 high level part-of-speech (POS) tags for the task, with an additional tag for sundries, such as punctuation. The tags are outlined in Table 4.1

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>proper name</td>
<td>names of people and cities</td>
</tr>
<tr>
<td>p</td>
<td>pronoun</td>
<td>personal pronouns; demonstratives</td>
</tr>
<tr>
<td>o</td>
<td>open class</td>
<td>verbs; nouns; adjectives</td>
</tr>
<tr>
<td>c</td>
<td>closed class</td>
<td>adverbial clitics; conjunctions</td>
</tr>
<tr>
<td>f</td>
<td>functional category</td>
<td>case marking; the invertor ‘ay’</td>
</tr>
<tr>
<td>d</td>
<td>other</td>
<td>punctuation</td>
</tr>
</tbody>
</table>

Table 4.1: Part of speech tags used in this research

These high level tags aim to assist in the identification of chunks. Our aim in choosing this set was to keep the number of tags to a minimum and still gain benefits from these as a feature. The main idea of the design of the tags was to differentiate between words that had semantic content and those that performed a grammatical

\(^1\)www.gutenberg.org
function, with the idea that words that are marked f, such as case marking, would often, but not always, mark the start of a noun phrase, whilst words that are marked o would often occur within the phrase that we are interested in. Often words belonging to the class c, such as sentence conjunct, for example ‘datapuwa’t’ “however” would be found outside a phrase.

The advantages of having a coarse-grained set of tags is that there is less margin for error and disagreement on how a word can be tagged. For future work, we would like to compare a finer-grained set of tags, such as those employed by dela Vega (2002), with our tags to see if a more detailed distinction results in significant benefits.

4.2 Predicate Mark-up

We employ the standard iob convention in marking up the predicates in the text. There are two different types of predicates we mark-up: prd and prd-sub. The former refers to predicates that belong to main clauses, whilst the latter refers to predicates that occur in subordinate or dependent clauses. Figure 4.1 illustrates this distinction in English.

The man who knew too much could not worry enough.

<table>
<thead>
<tr>
<th>PRD-SUB</th>
<th>PRD</th>
</tr>
</thead>
</table>

Figure 4.1: Predicate and subordinate predicate example

In this example, we have a noun phrase with the modifying relative clause who knew too much. This is a dependent or subordinate clause, with the predicate knew. The main clause predicate is labelled prd.

Figure 4.2 is an example of how subordinate predicates are marked up in the Tagalog text, where 4.1 gives us the word-for-word gloss and translation of this clause.

(4.1) tumagos sa puso ng binibini ang pangungusap na binigkas ng binata

“The words that the man uttered pierced the woman’s heart.”

2Apologies for the ridiculous example, but unfortunately the silly novela is strewn with these unavoidable passages
Chapter 4: The Data

We ensure that all the tagged elements in the chunks are contiguous. Given that there is a second-place clitic restriction, there are instances where the predicate elements are not all adjacent, as seen in 4.2, with the second person pronoun separating the negation from the rest of the predicating elements.

(4.2) Huwag mo-ng pahirapan ang aking puso.

NEG 2.sg.NG-LNK make.difficult ANG 1.sg.POSS heart

“Don’t trouble my heart.”

Figure 4.3: Avoiding disjointed PRD mark-up

In these instances we allow the negation to be omitted from the predicate chunk ensuring all related labels are contiguous. For the sentence in 4.2, we opt to mark this up like Figure 4.3 a. rather than b.

Can it get worse?
Chapter 4: The Data

Figure 4.4: An example of predicate mark-up which includes negation

We obligatorily only label contiguous elements for implementation reasons, because when automatically identifying predicates in the sentence, we attempt to extract all predicates in a sentence simultaneously (see Section 5.6.2 for more details).

For instances where we do not have pronominal clitics restricting word order, we include the negation within the predicate chunk, as seen in 4.3 and Figure 4.4. In this example, we isolate the way we label and delimit predicate elements in the sentence.

(4.3) Si Eduardo ay di nagbabago ang pagkakatitig ng binibini.
        ANG Eduardo INV NEG to.tire.of ANG sound of young.woman
        “Eduardo did not tire of the woman’s sound.”

4.3 Satellite Mark-up

The satellites refer to the dependents of the predicate. There are 4 types of satellites that are marked up, which are ANG, NG, SA and NG-COMP. The first three mark dependent phrases, while the last marks sentential complements (e.g. the object of quotative verbs, such as “remember” or “say”).

4.3 of this and Figure 4.4 show how this is marked up in the text. As with the predicate example shown in 4.4, presented here is the satellite mark-up in isolation for illustrative purposes.

\[4\]Vomit.

\[5\]Note that in Figure 4.4 the previous ‘ng’ attached to the pronominal clitic actually marks the start of the the NG-CMP.
### 4.4 Mark-up Language

The previous two sections illustrate how various components in a sentence labelled. The examples shown were partially marked up to highlight certain aspects of the sentence. In this section we illustrate the full mark-up ensemble.

We take a predicate-by-predicate approach in our mark-up, where each predicate is marked up along with its corresponding satellites. We organise the data in columns: the first column is populated by the tokens in the sentence, followed by the part-of-speech mark-up. Thereafter, each column is populated by one predicate and its dependents, and so the number of columns after the part-of-speech mark-up, is representative of the number of predicates in the sentence.

#### (4.5) Alagaan mo sanang lagi at ako ay iyong alalahanin.

<table>
<thead>
<tr>
<th>take.care</th>
<th>2.sg.NG</th>
<th>hopefully always and 1.sg.MKR INV your alalahanin. remembering</th>
</tr>
</thead>
</table>
| “Take care always and I will remember you.”

<table>
<thead>
<tr>
<th>Alalahin</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>mong</td>
<td>0</td>
</tr>
<tr>
<td>ang</td>
<td>B-NG-CMP</td>
</tr>
<tr>
<td>puso</td>
<td>I-NG-CMP</td>
</tr>
<tr>
<td>ko</td>
<td>I-NG-CMP</td>
</tr>
<tr>
<td>ay</td>
<td>I-NG-CMP</td>
</tr>
<tr>
<td>katulad</td>
<td>I-NG-CMP</td>
</tr>
<tr>
<td>ng</td>
<td>I-NG-CMP</td>
</tr>
<tr>
<td>isang</td>
<td>I-NG-CMP</td>
</tr>
<tr>
<td>bulaklak</td>
<td>I-NG-CMP</td>
</tr>
</tbody>
</table>

Figure 4.5: An example of a mark-up of a quotative verb complement.

#### (4.4) Alalahin mo-ng ang puso ko ay katulad ng isang bulaklak.

<table>
<thead>
<tr>
<th>remember 2.NG-LNK ANG heart my INV like NG one flower</th>
</tr>
</thead>
</table>
| “Remember my heart is like a flower.”

6Oh the bucket! Where is the bucket?!!

7This is the kind of prose is best represented as statistical data in a table.
4.5 Annotation

The first two chapters of the novela, which comprised of a little under 2,500 tokens, were marked up. These two chapters contained 259 predicates, 47 of which were identified as subordinate predicates (PRD-SUB). There were in total 435 dependent phrases. Table 4.5 shows the breakdown of these.

The following chapter examines how we can identify these elements in the Tagalog sentence automatically. In particular, we investigate which combination of features are best predictors of predicates and satellites.
Chapter 5

Experiments

5.1 Introduction

The aim of the experiments is to discover which of the features developed, in what combination, are more amenable to each given sub-task.

The first section of this chapter describes the design of our system. We then give a brief introduction of the tools used. The sections thereafter describe the function of each of the modules in the system.

5.2 Architecture

The overall configuration of the experiments is a pipeline sequence, where the input of one step relies on the output of the previous, as illustrated in Figure 5.1.

This architecture can pose some disadvantages in that possible errors are com-

![Figure 5.1: The pipeline architecture of our system](image-url)
Chapter 5: Experiments

Figure 5.2: A simple finite state transducer allowing singular and plural nouns pounded in the traversal of the pipeline. For example, if the classifier responsible for identifying predicates fails to pick one out, then it would not be possible to identify the satellites of that predicate as the satellite predictor will not be provided with a predicate to attach satellites to. For this reason, the gold standard benchmark is always tested against at every stage of the pipeline, in order to gauge how much error upstream processing is introducing. Using the gold standard data in the experiments aids in the evaluation of the classifier in that it gives us an upper bound or upper limit of expected performance for the given experiment.

We have labelled two steps Additional Features along the pipeline in Figure 5.1. These are linguistic features that we have developed based on the morphology of the words in the sentence. The details of the morphological analyser can be found in Section 5.4.

5.3 The Tools

The following presents the tools that are utilised in the pipeline development. Xfst and lexc are employed to create the morphological analyser to extract morphological features. Crf++ is the toolkit used to build the POS-tagger, Predicate Predictor, and Satellite Identifier.

5.3.1 Xfst and lexc

Xfst is a tool that provides access to basic Finite-State Calculus algorithms to create networks for morphological analysis (Beesley and Karttunen 2003). A morphological analyser identifies the smallest meaningful units that make up a word. For example, we can break down the word horses into two components: the noun horse, which we call the stem, and the suffix s, which indicates plurality.

This tool is used to create a two-sided network called a transducer. This means that there is an acceptor or lower language component, which verifies that the input is well-formed, and an output or upper language. The network does not simply tell
us a word is valid for a language — this kind of network is referred to as a finite state automaton. What we build is a transducer, which also generates an output, as shown in Figure 5.2.

The finite state transducer defined in Figure 5.2 is a language that has two lexical items _horse_ and _cat_, which can either be singular or plural. The input _horses_ would produce the output _horse+plural_ and the input _horse_ would produce _horse+singular_.

The advantages of employing a finite state transducer, as the name suggests, is that transduction is bi-directional, meaning that we could ‘query’ our morphological analyser by giving it a string from the upper language, such as _cat+plural_ and it would produce the surface word _cats_.

LEXC is a declarative language that is used to define the lexicon called by xfst. This avoids defining each word within xfst, thereby separating the definition of the stem lexicon from the way in which define the morphology of the language.

### 5.3.2 Crf++

The tool used in the experiments is an implementation of the conditional random field model. In particular, we use crf++, which is a freely available open source distribution.\(^1\)

Conditional random fields are graphical models for structured learning tasks, where the observed relative sequence of items is of significance (Lafferty _et al._ 2001). CRFs can be regarded as an undirected graph, much like a Markov network, in which the nodes represent a random variable. However, the nodes in a CRF are conditioned on their context and differ from the Markov model by relaxing the independence assumption (Wallach 2004).

The conditional random fields model has been used in many applications such as part-of-speech tagging (Lafferty _et al._ 2001), noun phrase chunking (Sha and Pereira 2003), and named entity recognition (McCallum and Li 2003). The model is well suited to modelling sequential data especially where the contiguous elements exhibit some internal structure, even if as simple as recognising the beginning of a sequence or phrase in contrast to being internal to it.

**Templates**

CRF++ requires the use of templates as instructions on how to combine the features defined for each token for a given input sequence. Given the input below, if the task was to identify all noun phrases (NPs) in the sentence, at the token _a_ (article), we may want to prime the learner on additional information, such as the previous token’s part-of-speech being a _verb_ and the following token’s part-of-speech being an _adjective_.

\(^1\)This tool is available from SourceForge.net (http://sourceforge.net/projects/crfpp/).
In addition to specifying how features of the input sequences should be interpreted, within the template, there is also a bigram and unigram option. This means that in addition to the features specified for each input token, the features for the previous token are also taken into consideration.

Windowing

Within the CRF++ templates, there is a great deal of flexibility in combining features. One way that we combine features is by looking at a variable window size of tokens and their part-of-speech to discover whether this additional information will increase the performance of the system. We look at various window sizes, from 1 to 6. A window size of 3 means that at a given token, we utilise the word and part-of-speech features of, not only the current token, but also the preceding and following token. A window of 5 would also include features from the previous 2 tokens, as well as the 2 following.

5.4 Preprocessing: The Morphological Analyser

Knowledge about how a token in a sentence is composed can provide a lot of information. For Tagalog, information such as focus (voice marking), aspect, and sometimes manner can be encoded on the verb itself. By conducting morphological analysis, we can dissect the word into smaller units and capture the information each of the elements in the word contributes to the token as a whole.

The morphological analyser is encoded using XFST and LEXC. We define the morphemes within XFST, and the lexicon (an inventory of word stems) is ordinarily listed in LEXC. However, for this project instead of creating a list of stem words to define within LEXC, we define permissible syllables based on the work conducted by French 1988. The decision to encode the ‘lexicon’ in this way was based purely on

2Thanks very much to Steven Bird during his Human Language Technology class for helping me with the encoding of the Tagalog morphological analyser, especially with infixation and defining permissible syllables in LEXC
available resources: we did not readily have an extensive list of stems in Tagalog.

The input to the morphological analyser is each token in the sentence, whether or not it should be regarded as a verb. If the morphological analyser can interpret a word as exhibiting verbal morphology then it will extract relevant features based on the morphemes that are recognised.

The aim of the morphological analyser is to extract features such as voice type, and also some features pertaining to aspect and modality (as outlined in Table 5.1). The information returned by the network we build is a set of eight binary features. Without a lexicon, encoding non-concatenative morphology, such as reduplication, is nigh on impossible. Therefore we built a post-processor to recognise full reduplication. Having said this, we were able to encode partial reduplication in xfst, based on solutions by Beesley and Karttunen (Beesley and Karttunen 2003), because partial reduplicated forms in Tagalog are restricted to only being allowed a simple onset, even if the onset of the reduplicated syllable is complex.

From the morphological analyser, including some postprocessing, there are twelve binary features, as presented in Table 5.1.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV</td>
<td>Verb highlights agents or actors</td>
</tr>
<tr>
<td>BV</td>
<td>Recipients or beneficiary of an action is highlighted</td>
</tr>
<tr>
<td>CV</td>
<td>This encodes some causation on the verb</td>
</tr>
<tr>
<td>DV</td>
<td>Direction or location is highlighted</td>
</tr>
<tr>
<td>IV</td>
<td>Instruments are highlighted</td>
</tr>
<tr>
<td>OV</td>
<td>The affect thing or person of an action is highlighted</td>
</tr>
<tr>
<td>CONTEMPL</td>
<td>This encodes whether the action has occurred or may occur</td>
</tr>
<tr>
<td>PART-REDP</td>
<td>Partial reduplication can encode ongoing action</td>
</tr>
<tr>
<td>FULL-REDP</td>
<td>Full reduplication can encode concepts such as randomness or intensification</td>
</tr>
<tr>
<td>IN</td>
<td>Infix which encodes object voice (OV) and non-contemplative action</td>
</tr>
<tr>
<td>UM</td>
<td>Infix encoding actor voice (AV) and non-contemplative action</td>
</tr>
</tbody>
</table>

Table 5.1: Morphological Features

In addition to these binary features, we do a very basic preliminary identification of prefixes and suffixes by taking the first three characters from the start of a token, as well as the last two.

The reason we chose a prefix of 3 and suffix of 2 is that the two most common verbal suffixes in Tagalog are -in and -an, and many of the prefixes, verbal or otherwise, defined in Schachter and Otanes 1972 are 3 characters long, for example nag-, mag-, pag-. This has obvious limitations in that it does not account for prefixes in series.
In future research, we plan to take a variable length of characters, an ngram approach, in identifying stems.

5.5 Evaluation Metrics

There are two basic components types in our pipeline: (1) components which label single-word units (i.e. the POS tagger), and (2) components which label (potentially) multi-word units (i.e. the predicate and satellite predictors). We use different evaluation metrics for the two component types.

To evaluate our part-of-speech tagger we use basic accuracy measures: word accuracy and sentence accuracy. Word accuracy simply measures what proportion of words were tagged with the correct SC POS tag. Sentence accuracy tells us what percentage of sentences had no tagging errors (i.e. all words were correct).

We rely on precision, recall and f-score to evaluate the multiword unit labelling components (i.e. the Predicate Predictor and the Satellite identifier). For the identified chunks, precision tells us how well our classifier correctly labels them. Recall gives us an indication of how well the classifier can identify the chunks in the sentence we are interested in. For example, a high recall for the Predicate Predictor would mean that our classifier does not leave many predicates unidentified and a high precision would indicate that of the predicates identified by the classifier were indeed correct.

F-score is a composite measure of precision and recall. It is the harmonic mean of these two measures, which gives us an indication of the overall performance of the system.

The method by which we evaluate the chunk and labelling task is more exacting than evaluating single tokens, because the measure is taken over the sequence and internal labels of the chunks. Therefore, given this stringent method of evaluating, we would expect an overall lower score for the predicate and satellite prediction tasks.

5.6 The Three Steps

The three steps refer to each stage in the pipeline: the part-of-speech (POS tagger); the predicate chunker (Predicate Predictor); and the chunker and labeller for the dependents of the predicate (Satellite identifier). The purpose of these steps are explained below.

5.6.1 POS Tagger

Part-of-speech tagging is a means of abstracting away from the surface word in a sentence via its syntactic function. These assigned tags function as an extra set of features to assist in the tasks further along the pipeline, as outlined in Figure 5.1.
We examine two ways that we can run the learner over the data: unigram and bigram sequences; and the application of ‘windowing’ as described in Section 5.3.2. The word windows we examine are a window of 1, 2, 3, 5 or 7 words.

Evaluation of the POS tagger is based on two accuracy measures: word accuracy and sentence accuracy.

<table>
<thead>
<tr>
<th>WINDOW</th>
<th>WORD ACCURACY</th>
<th>SENTENCE ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UNI</td>
<td>BI</td>
</tr>
<tr>
<td>baseline</td>
<td>0.811</td>
<td>0.158</td>
</tr>
<tr>
<td>1</td>
<td>0.811</td>
<td>0.914</td>
</tr>
<tr>
<td>2</td>
<td>0.901</td>
<td>0.909</td>
</tr>
<tr>
<td>3</td>
<td>0.914</td>
<td>0.905</td>
</tr>
<tr>
<td>5</td>
<td>0.904</td>
<td>0.897</td>
</tr>
<tr>
<td>7</td>
<td>0.899</td>
<td>0.891</td>
</tr>
</tbody>
</table>

Table 5.2: POS tagger results – unigram and bigram with ‘windowing’

(UNI = unigram; BI = bigram)

The columns labelled UNI and BI in Table 5.2 refer to unigram and bigram results and the rows represent the variable window size. We took the comparison baseline to the classifier based on a unigram window of 1.

The system that gave us the highest sentence accuracy was the classifier trained on bigrams with a window of 1, producing a sentence accuracy of 0.368, followed by the unigram system with a window of 3, which had 0.363.

We take these best two performing POS taggers and test them against the gold standard POS tags in the following task of predicting predicate chunks.

5.6.2 Predicate Predictor

The ‘Predicate Predictor’ is the step in the pipeline that identifies the predicate. The identification of the predicates in one sentence is performed in a single step mirroring the predicate mark-up (as outlined in Section 4.2).

POS Feature Utility

Table 5.3 shows the results from testing the utility of part-of-speech as a feature. These experiments are performed with the two best performing POS taggers, which we have labelled auto1 (bigram system with a window of 3), and auto2 (bigram system with a window of 1). The label no pos indicates that part-of-speech was not used as a feature for those tests and gold is our oracle system, which uses the gold standard POS annotation.
Chapter 5: Experiments

Table 5.3: Pred Picker - leave-one-out using CRFs

<table>
<thead>
<tr>
<th>TAGS</th>
<th>UNIGRAM</th>
<th></th>
<th></th>
<th>BIGRAM</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>no pos</td>
<td>.416</td>
<td>.194</td>
<td>.274</td>
<td>.491</td>
<td>.204</td>
<td>.278</td>
</tr>
<tr>
<td>auto1</td>
<td>.408</td>
<td>.287</td>
<td>.337</td>
<td>.494</td>
<td>.274</td>
<td>.351</td>
</tr>
<tr>
<td>auto2</td>
<td>.432</td>
<td>.345</td>
<td>.384</td>
<td>.558</td>
<td>.349</td>
<td>.429</td>
</tr>
<tr>
<td>gold</td>
<td>.439</td>
<td>.349</td>
<td>.430</td>
<td>.571</td>
<td>.360</td>
<td>.442</td>
</tr>
</tbody>
</table>

Table 5.3: Pred Picker - leave-one-out using CRFs

Results show that bigrams consistently outperform unigrams in this task, based on the f-scores. Also, the systems that utilise part-of-speech features perform better. These results also show that recall pulls down the overall performance of the system. With the best performing system, auto2, when the classifier predicts a predicate, it gets it right 56% of the time, but it does not identify 75% of the predicates. This has as adverse affect on our overall performance, which is indicated in the f-score.

An Optimal Window

The next test we perform aims to determine the optimal window size for predicate prediction. We again perform the experiments using no pos, auto1, auto2 and gold. Even though crf++ optimises over the input token sequence, we can draw attention to words and other features in context. For these experiments, attention is drawn to adjacent words and their POS, when used, at various window widths.

Table 5.4: Results of predicate prediction with variable window size

(P = precision; R = recall; F = f-score)

For four part-of-speech variations, shown in Table 5.4, it is conclusively shown
that a window of 5 is optimal, with a steady increase in performance as the window approaches 5, and then a dip at window 6.

\[
\begin{array}{ccc|ccc|ccc|ccc}
\hline
\text{No Pos} & \text{Auto1} & \text{Auto2} & \text{Gold} \\
\hline
1 & .704 & .268 & .388 & .619 & .321 & .422 & .635 & .313 & .419 & .643 & .314 & .422 \\
\hline
\end{array}
\]

Table 5.5: Results for predicate predictor by evaluating as tokens rather than chunks

\((P = \text{precision}; R = \text{recall}; F = \text{f-score})\)

The results shown in Table 5.5 are evaluated as though this were a token-based task. In other words, we disregard the internal structure of the chunks (i.e. the \text{B}, begin and \text{I}, inside, tags) and simply evaluate on the label type each token is assigned. Results measured in this way gives us an indication of how much more exacting evaluating over phrases is in comparison to single tokens with almost a 100% increase in f-score, especially for the \text{no pos} tests.

**Consolidation of PRD**

Of the 257 predicates marked up in the text, only 47 were identified as subordinate (PRD-SUB). Learning on only 47 instances may have been one of the factors that diminished the performance of the predicate identification classifiers. We conflate the the main and subordinate predicate distinction to test whether the consolidation of these two types of predicate be perform better on this identification task.

\[
\begin{array}{ccc|ccc|ccc}
\hline
\text{TAGS} & \text{PRD & PRD-SUB} & \text{NO PRD-SUB} \\
\hline
\text{No pos} & .491 & .204 & .278 & .500 & .227 & .313 \\
\text{Auto1} & .494 & .274 & .351 & .434 & .330 & .396 \\
\text{Auto2} & .558 & .349 & .429 & .533 & .337 & .413 \\
\text{Gold} & .571 & .360 & .442 & .564 & .424 & .484 \\
\hline
\end{array}
\]

Table 5.6: Conflating the main and subordinate predicate distinction

\((P = \text{precision}; R = \text{recall}; F = \text{f-score})\)
Comparing the tests that conflated subordinate and main predicates with the ones that did not, there was no dramatic increase in precision, but enough of an increase on recall scores to aid the ailing f-scores. It also shows that in extracting predicates there is insufficient evidence to treat subordinate predicates differently to main predicates. However, information on whether the predicate that is extracted is subordinate or main may be significant when experiments are conducted to predict the order of elements in the sentence, for future tests.

A method that could improve predicate identification is by initially extracting all predicates, regardless of their status, in the sentence, and then identifying their status as PRD or PRD-SUB as an additional post-step.

**Morphologically Featured**

These tests compare the two types of morphological features we develop, based on grammatical function and form, as well as the combination of the two. These are referred to in Table 5.7 as **Morph Analyser**, 32Gram, and Combined. Morph Analyser features are a set of binary features that pertain to what the identified morphemes encode, such as the type of voice marking, information about aspect, and manner. The feature set 32Gram simply takes characters from the start and end of the token as possible affixes, namely a prefix of 3 characters and a suffix of 2.

<table>
<thead>
<tr>
<th>Tags</th>
<th>Morph Analyser</th>
<th>32Gram</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>no pos</td>
<td>.447</td>
<td>.197</td>
<td>.274</td>
</tr>
<tr>
<td>auto1</td>
<td>.518</td>
<td>.315</td>
<td>.392</td>
</tr>
<tr>
<td>auto2</td>
<td>.519</td>
<td>.298</td>
<td>.378</td>
</tr>
<tr>
<td>gold</td>
<td>.573</td>
<td>.366</td>
<td>.447</td>
</tr>
</tbody>
</table>

Table 5.7: Predicate predictor results using morphological features - Window = 5 (P = precision; R = recall; F = f-score)

In extracting the predicate, the 32Gram features seems to perform better than the group of features defined by the Morph Analyser. There is not much of an appreciable difference in combining the 32Gram and Morph Analyser features, with the f-scores of the Combined system exhibiting a similar outcome to 32Gram. Similarly with the previous predicate extraction experiments, the recall has an adverse affect on overall performance of the classifiers.

### 5.6.3 Satellite Identifier

For these experiments we test how well our system can identify the constituents that ‘hang off’ the predicate. In addition to the part-of-speech features and morpho-
logical features, we also provide for each sentence a predicate which the system must identify the constituents of. Each predicate is ‘fed’ one at a time for each sentence, and we measure how well our system can identify satellites based on this primed predicate.

In this step we examine how morphological features assist in this sub-task. Again, we test the two types of morphological features that are developed, plus a combination of both. We do not use the gold standard (annotated) predicates in these experiments: we test on the best performing Predicate Predictor results, which were based on the output of auto2 pos tagging with a window of 5.

We performed these test in 2 lots: (1) using a window of 3 and (2) using a window of 5.

<table>
<thead>
<tr>
<th>TAGS</th>
<th>Morph Analyser</th>
<th>32Gram</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F</td>
</tr>
<tr>
<td>no pos</td>
<td>.454</td>
<td>.212</td>
<td>.289</td>
</tr>
<tr>
<td>auto1</td>
<td>.500</td>
<td>.294</td>
<td>.370</td>
</tr>
<tr>
<td>auto2</td>
<td>.476</td>
<td>.280</td>
<td>.352</td>
</tr>
<tr>
<td>gold</td>
<td>.542</td>
<td>.344</td>
<td>.421</td>
</tr>
</tbody>
</table>

Table 5.8: Predicate predictor results using morphological features - Window = 3

P = precision; R = recall; F = f-score

Similarly with the Predicate Predictor results, Table 5.10 shows that for the auto2 and gold systems, the feature set Morph Analyser, which extracts features such as voice, and aspectual information, does not perform as well as 32Gram in identifying and labelling the multi-word units. Having said this, in the Combined system the Morph Analyser features assist in propping up the performance over using the sole feature set 32Gram, except when using the auto2 POS tags.
Table 5.10: Satellite identification results using morphological features - Window = 5

\[
\begin{array}{c|ccc|ccc|ccc}
\text{TAGS} & \text{Morph Analyser} & & \text{32Gram} & & \text{Combined} & \\
 & P & R & F & P & R & F & P & R & F \\
\hline
\text{no pos} & .362 & .137 & .199 & .570 & .274 & .370 & .551 & .290 & .380 \\
\text{gold} & .420 & .207 & .278 & .625 & .380 & .472 & .617 & .389 & .477 \\
\end{array}
\]

\(p = \text{precision}; \ r = \text{recall}; \ f = \text{f-score}\)

Table 5.9 shows that a window of 3 is a better window size for predicting satellites than a window of 5, unlike the POS tagger, which performed better with a window of 5. Surprisingly, with the 5 window results no pos outperformed auto2, for 32Gram and Combined, which ordinarily perform better than Morph Analyser. All other things being equal, this shows that we are better off providing no part-of-speech tags, rather than providing part-of-speech tags from our best POS tagger. However, the results show that this window size, window = 5, is sub-optimal.

The next test we perform does not utilise any morphological features, and simply relies on the part-of-speech tags and the primed predicate. Given the minimal features, this system performs rather well with the result being comparable to the 5 window results and in some cases matching or even outperforming it.

Table 5.11: Satellite identification results using predicate primers and part-of-speech tags as features - Window = 3 & 5

\[
\begin{array}{c|ccc|ccc}
\text{TAGS} & \text{Window 3} & & \text{Window 5} & & \\
 & P & R & F & P & R & F \\
\hline
\text{no pos} & .607 & .266 & .370 & .624 & .278 & .385 \\
\text{auto2} & .649 & .371 & .472 & .611 & .354 & .448 \\
\text{gold} & .633 & .374 & .470 & .608 & .369 & .459 \\
\end{array}
\]

\(p = \text{precision}; \ r = \text{recall}; \ f = \text{f-score}\)

5.7 Discussion

In this chapter we presented the architecture with the tools required to build our system. We tested the POS features and morphological features we developed
and assessed their utility for the task of extracting predicates in the sentence and identifying their related constituents or satellites.

We found that the best parameters in building the POS tagger was window of 3 when using unigrams and a window of 1 for bigrams. For the Predicate Predictor bigrams in general outperformed unigrams. The optimal window size for these experiments was 5 and in general the 32Gram morphological features assisted more in this task than Morph Analyser.

For the Satellite Identifier, we also found that there we more gains from the feature set 32Gram than Morph Analyser and that a 3 window outperformed a 5 window.
Chapter 6

Final Remarks

6.1 Discussion

In this study we have presented a system that extracts predicates and identifies their corresponding dependent phrases. Being able to segment the sentence and identify elements in this way is a necessary step in doing further research involving how these elements interact in a wider context.

For future work, we would like to be able to improve on the system currently presented by feature re-engineering and researching other models and learners that are amenable to structured learning tasks. In particular, we would like to develop a finer-grained part-of-speech tag set to test if this may result in significant improvements further along the pipeline in the identification of the multi-word units (predicates and satellites). For the morphological features, we aim to extend the coverage of the morphological analyser.

The output of the research presented in this thesis are the necessary building blocks in examining the discourse-syntax interface in Tagalog. We would like to see how the relative order of the units in the sentence are affected by what is marked morpho-syntactically as the focus, and whether position of the sentence in the discourse can also affect this ordering.

In addition, we would like to test features related to the way constituents are encoded, that is the way that they are referred to in the text. Relevant features would include constituent weight, testing if longer phrases tend to be located at certain positions for ease of parsing, as well as whether the participants are expressed as full definite or indefinite noun phrases, or as pronouns. It may also be useful to look at semantics properties, such as animacy and agency, to test whether animate or human participants in the discourse tend to occupy certain positions relative to inanimate objects in the sentence.

Lastly, we would like investigate the two types focus strategies in Tagalog, morphological and syntactic, to examine how this may affect the word order of the non-focused elements.
6.2 Conclusion

This study has proposed a way to identify the building blocks to conduct further text analysis in Tagalog. We have presented the architecture and tools to build a system that can identify the predicate chunks and their dependent phrases, or satellites.
Bibliography


ENGDAHL, ELISABET, and ENRIC VALLDUVI. Information packaging and grammar architecture: A constraint-based approach. In Integrating information structure into constraint-based and categorial approaches, ed. by Elisabet Engdahl.


Minerva Access is the Institutional Repository of The University of Melbourne

Author/s: 
MISTICA, MELADEL

Title: 
Towards the utilisation of IS to predict word order in Tagalog

Date: 
2008

Citation: 

Publication Status: 
Unpublished

Persistent Link: 
http://hdl.handle.net/11343/35119

File Description: 
Towards the utilisation of IS to predict word order in Tagalog

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.