The Report Committee for Jason David Mielens
certifies that this is the approved version of the following report:

Unknown Word Sequences in HPSG

APPROVED BY

SUPERVISING COMMITTEE:

Supervisor: ______________________________
Jason Baldridge

____
Katrin Erk

Unknown Word Sequences in HPSG

by

Jason David Mielens, B.S.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF ARTS

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2014

Unknown Word Sequences in HPSG

by

Jason David Mielens, M.A.
The University of Texas at Austin, 2014

Supervisor: Jason Baldridge

This work consists of an investigation into the properties of unknown words in HPSG, and in particular into the phenomenon of multi-word unknown expressions consisting of multiple unknown words in a sequence. The work presented consists first of a study determining the relative frequency of multi-word unknown expressions, and then a survey of the efficacy of a variety of techniques for handling these expressions. The techniques presented consist of modified versions of techniques from the existing unknown-word prediction literature as well as novel techniques, and they are evaluated with a specific concern for how they fare in the context of sentences with many unknown words and long unknown sequences.

Table of Contents

Chapter 1

Introduction

The way in which unknown words are handled during the evaluation of syntactic parsing has a large impact on just how usable the final parser ends up being. If a parser is unable to handle, or handles incorrectly, one or more words in the input sentence, the output could potentially end up being entirely unintelligible or no longer useful. As a result, the topic of unknown word handling is an important one, particularly for those who wish to run systems on low-resource languages or in new domains from which they were developed or trained in. Both of those tasks introduce elements which are likely to cause novel words to appear in evaluation data., either because the training data simply did not include the type of data now being evaluated on, or because there simply was not enough training data overall.

Additionally, unknown word models perhaps become increasingly important when applied to 'real-world', online settings; where the system is constantly being fed a stream of novel 'evaluation' data, although no actual evaluation is done in this context.

Although the general topic of unknown word handling has received a substantial amount of prior work (See Section 1.2), this work will focus on a subset of this problem that, in comparison to the general problem, has received a relatively small amount of attention – the parsing of sentences containing multiple unknown words.

Sentences containing multiple unknown words come in a variety of forms that will be distinguished throughout this work. In particular, it is important to draw a distinction between sentences containing multiple unknown words and sentences containing sequences of unknown words. When I refer to sequences of unknown words, I am referring specifically to the situation where multiple unknown words occur immediately adjacent to each other. However, it is also possible for sentences, particularly longer sentences, to contain a large number of unknown words and not contain any unknown sequences. I also consider these highly degraded sentences with many unknowns, as they many also pose an issue for previously investigated unknown word handlers.

This study is intended to make two main contributions. I will first present the results of a corpus study calculating basic statistics regarding the prevalence of unknown word sequences. This will attempt to be as unbiased as possible in the sense that I make use of a corpus that was not used in the development of the Head-driven Phrase Structure Grammar (HPSG) [26] being used in this work. This is intended to give an accurate sense of how common unknown word sequences might be in a real-world setting making use of highly lexicalized grammar such as HPSG or Combinatory Categorial Grammar (CCG) [27].

Additionally, I present the evaluation of two types of unknown word handlers (described in Chapter 3) that have been modified with the goal of working specifically on unknown word sequences. These are evaluated on three separate metrics in order to determine which major class of handlers could be best for dealing with long unknown sequences. This is the main evaluation task that I focus on, where the task is predicting the HPSG lexical type for unknown words in the corpus, which allow the parser to make correct decisions about the unknown words even though they have not been previously seen.

1.1 HPSG Parsing

In this section, I present an overview of the current state of the art for various aspects of HPSG parsing. HPSG is a highly lexicalized grammar formalism in the sense that the lexical entries for the grammar include a wide variety of information about different features of a lexical item, for instance gender of nouns, subcategorization frames for verbs, etc. An example of a basic HPSG-style lexical entry for a simple pronoun is in Figure 1.

$$\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \left\langle \text{`it'} \right\rangle \\ \textsc{synsym} & \begin{bmatrix} \textit{synsym} & \\ \textsc{cat} & \begin{bmatrix} \textsc{head} & \textit{noun} \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textit{ref-index} & \\ \textsc{pers} & \textit{3rd} \\ \textsc{num} & \textit{sing.} \\ \textsc{gend} & \textit{neut.} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 1: Example HPSG Lexical Entry

HPSG is the parsing framework which I use to evaluate the different unknown word handlers, but it is important to note that the techniques used here are not specific to HPSG and could be used in any lexically rich syntactic framework.

1.1.1 ERG/Redwoods

For performing the actual parsing work needed by this project, I make use of the PET Parser [6] and the English Resource Grammar (ERG) [12]. The ERG is a wide-coverage HPSG grammar for English, and was chosen specifically because of this large coverage. The PET Parser is a development of the LKB Grammar Development Environment, which on its own serves as a grammar engineering setup; the PET Parser is perhaps best viewed as a batch-processing setup on top of the LKB. The ERG is considered to be the state of the art for HPSG parsing in English and has a long history of use in many HPSG-based parsing tasks.

From the ERG, a corpus known as Redwoods was created. This corpus is built directly from the predictions that the ERG makes for the sentences contained in the corpus. The ERG provides all possible parses for a given sentence in the corpus, and then a human annotator manually confirms that the top-ranked parse matches with what the true parse should be. In this way, Redwoods is unique in that it provides a dynamic treebank; the analysis of sentences in the corpus is allowed to evolve over time as the ERG is developed as well.

The success of the Redwoods corpus has led to the development of similarly structured corpora for both other languages and other domains of English. One of these is Wikiwoods, a Redwoods-style corpus of HPSG annotations for the English version of Wikipedia [17]. This is the corpus that I use for this project, and a more detailed discussion of the reasons for its selection may be found in Chapter 2.

1.1.2 Corpus Conversion

An alternative to the ERG/Redwoods approach is that of Miyao et al. [23], who opt to learn HPSG grammars from converted versions of standard Context-Free Grammar (CFG) corpora. In this approach, the Penn Treebank [21] is converted into an HPSG-based corpus, from which a grammar is induced. One major difference of this method of treebanking as opposed to the Redwoods method is that the resulting HPSG corpus is static due to the static nature of the original Penn Treebank. This means that the corpus suffers from many of the problems Redwoods was intended to solve, most notably the lack of the ability to adapt the analysis as the grammar develops.

Corpus conversion does have a major benefit over the Redwoods style of Treebanking when it comes to beginning development of an HPSG grammar. The method of Miyao et al. [23] allows an HPSG grammar to be induced effectively from no annotated HPSG trees. This perhaps makes it more suited for initial creation of grammars, rather than the continuing development of those grammars.

1.1.3 Deep Semantics

One of the major benefits to parsing with a lexically-rich framework such as HPSG is the ability to more easily parse semantics simultaneously with the syntax. Today, most state of the art HPSG parsing that includes semantic analysis makes use of Minimal Recursion Semantics (MRS) [11]. MRS is a very useful representation which factors semantics into elementary predications, and also crucially allows for underspecification of scope ambiguities. An example of an MRS representation of the sentence "Every dog chases some white cat" is shown in Figure 2, and is taken from Copestake et al. (2005)

$$\text{some}(y,\text{white}(y) \wedge \text{cat}(y), \text{ every}(x, \text{ dog}(x), \text{ chase}(x, y)))$$

Figure 2: Example MRS Representation

While MRS is not specifically HPSG-based, it is most widely used by HPSG parsers. A variant of MRS called Robust Minimal Recursion Semantics (RMRS) [10] is also used, and due to its ability to underspecify relational information in addition to scope, RMRS is able to be used in shallower techniques like part-of-speech tagging or noun phrase chunking.

In this project, I evaluate HPSG parses in terms of their MRS representations. This choice was made so as to privilege the semantics, rather than the syntax, since many times the choice to use HPSG is motivated by a desire to include deep semantics in the parse. See Section 4.3 for further discussion of the MRS evaluation setup.

1.1.4 Supertagging

The task that I will be adopting for this work is primarily that of predicting lexical types for words that the grammar has not seen previously. In this case, a lexical type is an abbreviated version of the lexical entry for an HPSG lexical item. For instance, a typical noun may have the lexical type ‘n_-_c-pl_le’ – indicating that it is a countable, plural noun such as ‘cattle’. This task is commonly known as supertagging, after Bangalore and Joshi [2].

Supertagging is essentially identical to the more well-known task of part-of-speech tagging, but supertags of any variety (HPSG, CCG, etc.) are much more detailed and numerous than part-of-speech tags, which makes supertagging a much harder task in general. Bangalore and Joshi initially worked with Tree-Adjoining Grammars, although others have extended the idea to other lexically rich frameworks like HPSG and CCG [8][30][1].

A variety of approaches to supertagging exist in the literature, and for the work at hand (predicting HPSG types for unknown words) I will be using a method described by Blunsom [5] for supertagging for Deep Lexical Acquisition using a Conditional Random Field (CRF) classifier [19]. This method has the advantage of being able to take arbitrary features from the input sequence, making it highly adaptable. See Section 3.1 for details on the features chosen for the present work.

1.2 Previous Work on Unknown Words

As a basic component of HPSG parsing, the topic of unknown-word prediction has been previously tackled by a large number of authors, who have formulated a variety of different solutions to the problem. The majority of approaches found in previous work can be classified into one of two major areas: either direct sequence-based classifiers, or some kind of generic-type instantiation.

1.2.1 Direct Sequence-Based Classifiers

Direct sequence-based classifiers are likely the most common unknown-word prediction solution of the last few years. Solutions of this variety predict the type for an unknown word by extracting features from the surrounding context and building a model to predict the most likely type for the unknown word. Usually this prediction takes the form of a Maximum-Entropy (MaxEnt) model, with the extracted features as inputs. Zhang et al. lay out the basic form of this type of classifier in the context of unknown-word prediction, as well as some of the more common features that are used [31]. The features themselves vary between authors, but often include the types of surrounding words, morphological features of the unknown word or surrounding words [7], or syntactic features derived from partial parsing of the sentence in question [31]. For example, consider the sentence shown in Figure 3: ‘the dogs bark’. The HPSG types for the first two words are listed below the lexical item, but the third type is being considered unknown. Under the direct, sequence classification approach, features from the surrounding types and lexical items are used to make a best prediction for the unknown type. These features (here in abstract) are represented by the arrows in the figure, showing that the model uses features from them to infer a type for the unknown word. For more details about the exact lexical features used in this current work, see Section 3.1.

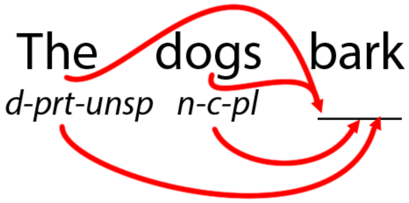

Figure 3: Direct unknown type prediction model

1.2.2 Classifiers with Generics

The generic type instantiation techniques make use of a mapping between more classic Part-of-Speech (POS) Tags, and the more rich set of HPSG types. These approaches essentially bypass the difficult task of actually predicting HPSG types, and instead transform the sequence into POS tags. As POS-tagging is a problem for which high-precision solutions exist, and for which such solutions can be constructed quite rapidly even for previously unworked languages [18], this task is typically much easier than straight HPSG type prediction. Once the POS tags for the sequence are found, the mapping from POS tags to HPSG types is relatively straightforward, with individual POS tags mapping onto underspecified, generic versions of HPSG types. Alternatively, some versions of this technique have the mapping of POS tag to HPSG type map not to a generic version of an HPSG type, but rather some particular fully specified HPSG type – usually the most common HPSG type for a particular POS tag. This is the model I adopt for this work, particularly because it makes the comparison of type accuracy results more meaningful; there are no gold generic types, so any unknown word handler using generic types has no real way of calculating type accuracy.

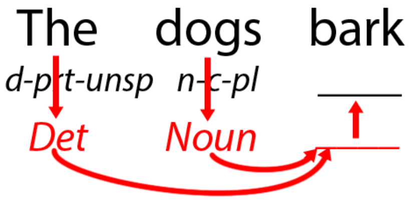

Figure 4: Indirect unknown type prediction model

An example of this type of model is shown in Figure 4, where the same sentence is being considered as in Figure 3. Notice that here, first the POS tags for all the lexical items are determined, leading to the third line in the figure. This process relies primarily on the lexical item itself, rather than the HPSG types, which make it possible to run on words for which the HPSG type is still unknown. Once the POS tag for the unknown word is determined, a final mapping from the POS tag to HPSG type is made – this is shown by the arrow from the third line to the second. As shown, this method allows us to bypass the HPSG types as much as possible in favor of the radically simpler POS tags.

1.2.3 Lexical Acquisition Approaches

A third area of unknown-word prediction found in previous work is automated Lexical Acquisition. While all forms of unknown-word prediction can, in some sense, be thought of as lexical acquisition, this class of techniques aims to perform this learning by modeling the human lexical acquisition process to greater or lesser degrees. For instance, Barg describes a system which can gradually learn more specific representations for a given unknown lexical item by considering the full range of contexts in which that unknown word occurs [3]. The system learns all the information it can from a particular context, and makes use of other contexts to fill in additional information or refine existing information.

For instance, a verb may be used intransitively in one context, and later could also be used transitively; the system would update the lexical entry for this verb to reflect the fact that it may optionally take an object. As an unknown word handler, lexical acquisition based models have become less popular in recent years, perhaps due to the rise of simpler, more straightforward statistical methods such as Maximum-Entropy models that perform just as well, if not better, while also being less dependent on the frequency of the unknown word itself. Lexical acquisition techniques are dependent on the actual unknown word being relatively common. This makes them useful for things like grammar learning, where one might expect common words to still be unknown, but in most applications of unknown word handling the unknown words are low-frequency items.

Chapter 2

Datasets

For this project, multiple data sources were used, corresponding to the multiple aims of the project. Because a major concern of this work was determining the frequency of contiguous unknown word sequences and sentences containing multiple unknown words in general, it was determined that a data source not directly involved in the development of the grammar being used for parsing should be considered. As a result, the Penn Treebank (PTB) [21] was selected as the primary corpus used for collecting these corpus statistics. The PTB was not explicitly used during the development of the HPSG grammar used in this project – the English Resource Grammar (ERG)[12]. By selecting a corpus that the grammar was not specifically developed against, we are able to collect more relevant statistics that are more likely to match the statistics we would find if using this system in a real-world context against a potentially live data stream that would be constantly generating novel data.

In particular, the Wall-Street Journal (WSJ) [21] section of the PTB was used for collecting the corpus statistics. The fact that this data is biased in the sense that it is all from a single source with a single dominant genre (namely financial news) was considered, particularly with regard to the proliferation of proper names. Section 2.1 contains the details on how the analysis took these particular biases into consideration during the corpus study.

However, as the PTB is not annotated for HPSG types and trees, evaluating the accuracy of the parser on that corpus is impossible; an additional corpus was needed for the analysis of the benefits of the various unknown-word prediction methods. There are a limited number of HPSG corpora available, with perhaps the most common being Redwoods [25]. However, once again, being a de-facto standard for HPSG evaluation was not ideal because the grammar has been developed against this corpus. While the grammar could be altered to introduce more unknowns, in essence simulating an earlier state of development by removing random lexical entries, this was considered a poor alternative. If at all possible, a corpus with more naturally occurring unknowns would be preferred over one with entirely artificially constructed unknowns. For this reason, the WikiWoods corpus [17] was chosen as the corpus for evaluation. WikiWoods is a similar corpus to Redwoods, but is newer and has been developed less than Redwoods. As a result, WikiWoods is likely to be closer to the ideal of having never seen the corpus before that would more accurately simulate a real-word use case of this grammar. That is to say, WikiWoods has more naturally occurring unknown words than Redwoods, which makes for more representative data even though eventually synthetic unknowns will need to be introduced in the course of experimentation.

An alternative to picking a corpus already annotated for HPSG would be to utilize a conversion process on a differently annotated treebank. Miyao

Corpus	Sentences	Word Types	Word Tokens
Wall-Street Journal	47k	50k	1253k
WikiWoods	54723k	1954k	769535k

Table 1: Raw count data for corpora

et al. [23] describe a conversion process for transforming the WSJ into an HPSG annotated treebank. This type of conversion could potentially allow HPSG grammar development to take place much earlier in a languages annotation effort. For instance, if a more standard treebank had already been developed, this conversion could provide a method of jump-starting HPSG grammar building. However, for languages that lack any substantial treebank this method provides little help. These languages would benefit much more from a jump-start that doesn't rely on heavy statistical inference, perhaps making use of linguistic universals to inform the early stages of grammar engineering instead. This is the goal of an alternative approach to grammar engineering that has been developed by Bender et al. called the Linguistic Grammars Online (LinGO) Grammar Matrix [4]. Because the goal of this project is not inducing or building grammars, but rather simply exploring the properties of unknown words in existing grammars, neither the treebank conversion nor the LinGO Matrix was used during this project.

The version of WikiWoods used in this project was Version 1212[1], dated October 23, 2013. This version of the corpus contains roughly 1.3 million Wikipedia articles annotated for HPSG types. See Table 1 for details on the

[1] Available at http://moin.delph-in.net/WikiWoods

exact sizes of the two corpora.

The training of the parser for the determination of parse accuracy was done on a 70/15/15 percentage split of the WikiWoods corpus, making use of a development set during the course of the project, and with the final evaluation numbers reported in this paper being obtained from a completely held out test set.

The type accuracy of the individual unknown word handling strategies was evaluated on the type sequences extracted from the WikiWoods corpus, for precisely the same reasons as above, and was also subject to a 70/15/15 percentage-based split into training, development, and test sets.

2.1 Corpus Statistics

In order to determine the prevalence of unknown word sequences of varying lengths, a study was conducted on the Penn Treebank Wall Street Journal (WSJ) corpus, as described above. To accomplish this, the PET parser [6] using the ERG grammar [12] was run on the raw text version of the corpus, with no unknown word handling enabled. This mode thus represents the baseline coverage of the ERG grammar, and was intended to determine how common it is to encounter both sequences of multiple unknown words, and sentences containing multiple unknown words in general.

As the data in Figure 5 show, the occurrence of unknown words in the WSJ data was very frequent. In fact, over the entire corpus, a total of 67.5%

of sentences contained at least one unknown word. This result is an excellent demonstration of the fact that unknown word handling is a vital part of any parser that hopes to handle a wide variety of sentences. The WSJ data is presumably somewhat (intentionally for the purposes of this study) out-of-domain for the ERG grammar, as most of its development was likely done in contexts where the evaluation of progress was made on corpora featuring existing HPSG annotations. This out-of-domain effect is likely to blame for the rather high percentage of sentences containing unknown words. Additionally, the specific genre of the WSJ (namely, financial news) may be playing a role, as named entities such as people or business names may be somewhat inflating these numbers.

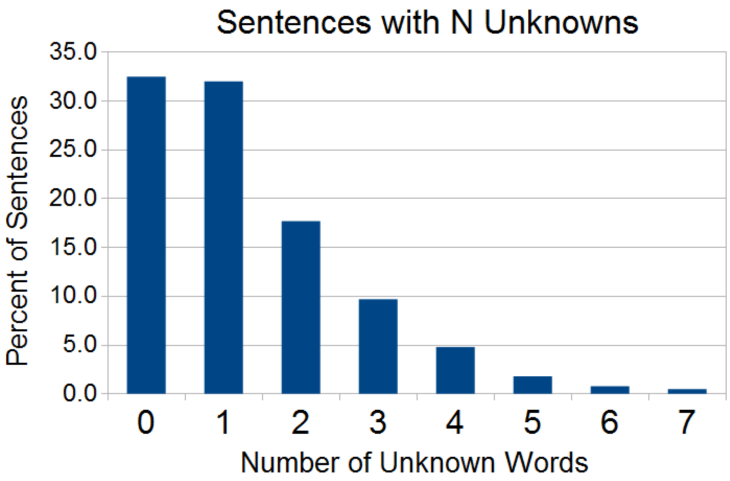

Figure 5: Percentage of sentences in WSJ with a given number of unknown words.

N	0	1	2	3	4	5	6
Sentences with N Unknowns	15213	15494	8238	4629	2289	931	285

Table 2: Absolute counts of sentences in WSJ with a given number of unknown words.

Notice also in Figure 5 that there are still a significant number of sentences that contain up to perhaps four unknown words. After this, the numbers start to tail off rapidly; absolute numbers are available in Table 2. In total, the percentage of sentences containing more than one unknown word is 34.4%. This is essentially equivalent to the percentages of sentences containing both zero unknowns (32.5%) and one unknown (33.1%). The existence of these sentences featuring multiple unknowns implies that the possibility for sequences of consecutive unknown words, at least of a few words long, should be quite substantial. Indeed, this is what we find in the WSJ data; consider the results in Table 3 below.

Longest Unknown Sequence	Percentage of Sentences
No Unknowns	32.5
Single Unknown	55.3
Double Unknown	10.2
Triple Unknown	1.9
Quadruple Unknown	0.1
Double or More Unknown	12.2
Single or No Unknown	87.8

Table 3: Unknown sequence lengths in WSJ.

The data in Table 3 shows the existence of sequences containing up to four consecutive unknown words, with a total of 12.2% of sentences containing an unknown sequence with a length greater than one. This is perhaps more than we might expect, given the number of sentences containing two or more unknown words.

With 34.4% of sentences containing at least two unknowns (Figure 5) and 12.2% of sentences containing a sequence of two or more unknowns, this means that 35.6% of sentences containing two or more unknown words contain a sequence of two or more unknowns. With the average sentence length of the WSJ section of the Penn Treebank around 23 words, it seems likely that there is some effect driving the occurrence of unknown sequences other than pure chance. For instance, perhaps multiword expressions where multiple portions of the expression are all unknown; proper names would be a likely candidate for this type of error, with both first and last names.

Unfortunately, examination of these sequences showed very few consistent patterns. For instance, no single sequence of two unknowns occurred more than three times, making analysis of 'common' unknown sequences impossible at this time. Some of the sequences that did occur were things like 'grassroots newsprint' and 'the Keenan affidavit'. Obviously there are some proper names that do occur, and acronyms also seemed to be troublesome, but there was little indication of systemic failures. Perhaps evaluating on a larger corpus, to allow for more instances of natural unknown sequences, could reveal more significant patterning.

The data above demonstrate just how significant a problem the issue of unknown words and unknown word sequences can be, particularly when switching domains or corpora. The problems inherent in switching corpora between training and testing are well-known and previously described in the literature of domain-adaptation. Relevant to this study are prior results showing that unknown words are one of the most challenging issues in this domain-shift. For instance, Daumé III and Jagarlamudi found that unknown words accounted for the majority of the errors (roughly 50%) they encountered when changing domains in a machine translation task [13]. Perhaps the most interesting result from their analysis was the very common words that failed to occur in their source corpus of European Parliament proceedings; words like 'behavior', 'favorite', and 'boring'. While it is certainly possible to imagine reasons why these words fail to occur (alternative spellings, for instance), it shows that words need not be rare in the general language to be unknown in certain domains or contexts.

McClosky et al., in developing a model to predict syntactic parser performance on a particular corpus, found that the number of unknown words in the corpus was one of the most useful features [22] for predicting the difficulty of that corpus. Of course there are other issues at work in these studies of domain adaptation that are unrelated to the unknown words (register, fluency, sentence lengths/complexities, etc.) but it seems clear that the design of any system that will be encountering unknown words on a frequent basis should heavily prioritize the handling of those unknowns.

Chapter 3

Experiments

Having established the existence of sequential unknowns, and the relative prevalence of sentences with multiple unknowns in general, the question of how best to handle these obstacles to parsing could rightly be raised. Much of the previous literature on unknown-word handling evaluates either on constructed corpora featuring a single unknown word per sentence, or are unclear about the extent of the unknowns handled in their evaluation set. Refer back to Section 1.2 for details of this previous work.

Accordingly, this work sets out to evaluate unknown-word handlers with a particular focus on their performance during the parsing of both sequential unknowns and sentences featuring multiple unknowns. To do this, the Wikiwoods corpus (See Chapter 2 for details) was used due to the fact that it already contains the required HPSG type annotations; this allowed for easy evaluation. However, because the Wikiwoods corpus was used during the development of the ERG grammar being used here, the occurrence of unknown words is far too low. Additionally, the unknown words that are present are likely to be non-representative of the types and distributions of unknown words in a neutral, previously unseen corpus like the WSJ.

To better simulate the conditions found in the neutral WSJ corpus, the following procedure was followed to modify the WikiWoods corpus: First, all the sentences with true unknown words were removed, leaving a corpus that was able to be fully parsed by the PET parser with no active unknown word handling; this did not significantly reduce the size of the corpus. Next, random words were selected for being marked as 'unknown'. This was done in a weighted fashion, by type; this means that the more uncommon words were more likely to be marked as unknown than common words. The unknown word selection process was run several times in an effort to match the basic statistics of the unmodified WSJ Corpus. These basic statistics are shown in Table 4, in comparison with the statistics for the unmodified WSJ corpus found in Chapter 2.1.

Longest Unknown Sequence	Modified WikiWoods	WSJ
No Unknowns	32.3	32.5
Single Unknown	55.4	55.3
Double Unknown	11.1	10.2
Triple Unknown	1.0	1.9
Quadruple Unknown	0.2	0.1

Table 4: Unknown Sequence Length Comparison

As seen in Table 4, the modified Wikiwoods corpus matches the basic statistics of the neutral WSJ corpus quite well, although there are some discrepancies in the percentage of double and triple unknowns. The fact that Wikiwoods is many times larger than the WSJ means that even with slightly lower percentages, the total number of sentences with double and triple unknowns is still quite high. Accordingly, there will still be ample opportunity to evaluate the performance of the various unknown-word handlers on sequential unknown data. Table 5 shows the same comparison, although in terms of unknowns per sentence rather than the length of the unknown sequence. Again, the two corpora show a relatively high amount of similarity.

N	Modified WikiWoods	WSJ
0	32.3	32.5
1	34.0	33.1
2	16.8	17.3
3	10.1	9.5
4	5.7	5.2
5	0.9	1.1

Table 5: Percent of Sentences with N Unknowns – Cross-Corpus Comparison

This modified WikiWoods corpus is used as the input for the evaluation of two separate unknown-word handlers. Both of these techniques are modifications of techniques previously explored in the literature, but I use them here (in slightly modified forms) with the intent to evaluate their performance on unknown sequences.

3.1 One-Sided Classification Model – CRF

The first unknown word handler I evaluate is essentially a restricted version of a typical sequence classification model. Previous versions of direct HPSG type sequence prediction typically make use of features from a fairly wide context; that is, they use features from words on both sides of the unknown word in question, and in general features from anywhere in the sentence. These include features such as those syntactic features derived from partial parsing results, which depend on being able to parse the rest (or at least a large portion) of the sentence. This is the Conditional Random Field classifier described in Section 1.1, used by Blunsom for doing supertagging in the Deep Lexical Acquisition task [5].

The type sequence classifier created for this work makes use of features that are derived entirely from words on the left-hand side of the unknown word in question. In particular, the two words immediately preceding the unknown word are potential sources of features, which are summarized in Table 6. The benefit of such a restriction is that this classifier can in effect be moved from left to right over a sentence, and no matter how degraded the sentence was originally, all of the classifications will be done with all of the features available. Although at the start an unknown word may have another unknown to its left, potentially limiting the available features, this sliding method ensures that the unknowns fill in left to right, making those originally unknown types known by the time they are needed.

While this restriction on the directionality of features almost certainly does reduce performance slightly, and is primarily an artificial limitation, similar real-world tasks and situations do exist in real-time streaming processing type systems. In these instances, the end of the sentence being processed is still unknown or uncertain, and so reasoning using only features from the preceding words is both useful and required in many circumstances. For instance, Lison and Kruijff describe just such a system for speech processing in a CCG framework during human-robot interaction [20], where predictions about the

Classification Features	Example
Lexical Word	*cancel*
HPSG Type	v_np*_le
POS Tag	V
Subcategorization Frame	np*

Table 6: Classification Model Features

word currently being processed relies only on those features from prior words in the sentence.

Additionally, this may be especially useful in the context of sequential unknowns. For instance, in a sequence of three unknowns, the classifier first considers the left-most unknown, for which all of the features are known (from the two known types to the left). Next, the classifier moves to the second unknown in the sequence, for which all of the features are now known (from the known type two to the left, and the predicted type immediately preceding). Finally, the third unknown is predicted using the features from both of the predicted types to its left. For unknown words at the beginning of a sentence, dummy type-values signaling the beginning of a sentence are inserted as preceding material.

The features extracted from the preceding words are used as the input to a Conditional Random Field (CRF) classifier, which produces a prediction for the type of the unknown word under consideration[19].

3.2 Alternative Sequence Modeling

This unknown word handling strategy is very similar to the commonly used approach involving ‘generic’ lexical entries. Under these types of handlers, a mapping is established between part-of-speech tags and specially created HPSG types which is are generic as possible for an entry of that part-of-speech. For instance, if a given unknown word is found to have a Noun part-of-speech, a generic lexical entry for ‘noun-ness’ is selected as the HPSG type for that word. I refer to this type of approach as an alternative sequence model because the heavy lifting is done in an entirely separate domain (parts-of-speech) from the target (HPSG types), while the transformation from part-of-speech to HPSG type is trivial once established.

However, the instantiation of an alternative sequence model used here differs from the generic strategy in an important way. The alternative sequence model used in this work selects the most common HPSG type for a given POS tag rather than selecting a specially created generic entry. This approach has been used before, although usually the mapping from POS to HPSG tag is specified by hand since the set of POS tags is small enough to allow for this [31].

The mapping of POS tag to the most common HPSG type for that POS tag was created by POS tagging the training section of the Wikiwoods corpus and considering the HPSG types that occurred most commonly for a given POS tag. For instance, the POS tag ‘NNS’ occurred most frequently with the HPSG type ‘n_-_c_le’, corresponding to a countable noun. Other possible

HPSG types that occur with 'NNS' include things like 'n_-_m_le', which denotes a mass noun.

This change was intended to potentially increase the overall parse accuracy of the trees built on the predicted types, as generics sometimes suffer from issues related to parse accuracy. Additionally, evaluating type accuracy (percentage of unknown types correctly predicted) is essentially meaningless in a generic setup, because no gold HPSG type is annotated with a generic entry. Using the 'most common' rather than the generic allows for somewhat meaningful comparison with the other unknown word handlers in the context of type accuracy.

I use the Stanford POS tagger of Toutanova et al. [28] to provide the POS tags for the input sentences, making use of the provided model for English rather than re-training.

Chapter 4

Results

This section contains the results of the experiments conducted on the two constructed unknown word handlers described in Chapter 3. The handlers were evaluated under three separate metrics, designed to determine their suitability for handling data featuring large numbers of unknown words and potentially lengthy unknown sequences. These metrics included coverage, type accuracy, and parse accuracy; each will be addressed individually.

4.1 Parse Coverage

As can be seen clearly in Table 7, both methods are quite successful in terms of overall coverage for parsing, where coverage is defined at the sentence level. To be considered in the coverage, the PET parser must produce at least one parse tree for that sentence. Recall that the modified Wikiwoods corpus was pruned prior to inducing the artificial unknowns such that every sentence was originally parsable by the ERG grammar; thus, the theoretical maximum for the coverage percentage is 100%, since the grammar is guaranteed to contain the higher level rules need to combine the true types. In other words, the failure of the parser to produce a tree can be attributed to the newly tagged

Unknown Word Handler	Coverage (at least one valid parse)
None	34.9
One-Sided CRF	91.3
Alternative Sequence	**93.9**

Table 7: Sentence Coverage Statistics – Modified Wikiwoods Corpus

unknown words rather than some other part of the parser. Note that the baseline is really simply a measure of the percentage of sentences containing no unknowns, as described in previous Chapters, rather than an actual tagging technique.

Since both of the unknown word handlers never fail to produce some HPSG type for every unknown word presented to them, their failure to reach 100% is indicative of their failure to produce some type that allows at least one analysis of the parse tree to be produced. There is no guarantee (and it is in fact often not the case) that the parse produced is the correct one, but even in these cases it is often the case that the grammar is able to find some interpretation for the sentence. See Section 4.3 for a discussion of the parse accuracy, which seeks to specifically measure this.

The fact that the alternative sequence model outperforms the one-sided CRF may be demonstrating, at least in part, the ability of the alternative sequence model to handle longer unknown sequences and the fact that it consistently produces very common types. The one-sided CRF has the potential to produce rare types, whereas the alternative sequence model is limited to the most common type for each POS tag; the grammar is more likely to be able to produce some interpretation for a common type (even the wrong one)

Unknown Word Handler	1st Position	2nd Position	3rd Position
One-Sided CRF	**41.6**	**34.3**	11.8
Alternative Sequence	21.0	20.6	**20.6**

Table 8: Type Accuracy

than a rare, specialized type.

4.2 HPSG Type Accuracy

HPSG type accuracy was determined by calculating the percentage of the induced unknown HPSG types that the two unknown word handlers were able to accurately reproduce. Note that this was over the full ERG type dictionary, which contains 1100 types, and not any reduced type set. In Table 8, the 'positions' indicate the position in an unknown word sequence, and these percentages are calculated over just the unknown words rather than all of the tokens. For instance, 34.3% of unknown words in the 2nd position of an unknown word sequence had their types correctly predicted. The results in Table 8 show several important facts.

First, in general the alternative sequence model performs worse than the one-sided CRF, and neither perform amazingly well. However, the performance of the CRF is in line with previous results for type accuracy on the full type set using a sequence classifier [14][5]; other previous work limits the type set in various ways, which make them poor comparisons.

One interesting result is the sharp falloff of the CRF accuracies on longer unknown sequences. This can likely be attributed to the fact that at the end of longer sequences (3rd position), the classifier is totally relying on features extracted from previously predicted types. Basing predictions on predictions is rarely a successful strategy, and this is further evidence of that. It seems likely that a CRF making use of additional features, perhaps a bi-direction model using features from both sides of an unknown word for instance, may do slightly better.

On the other hand, the alternative sequence model suffers essentially no degradation over the longer spans, although it is never that great to begin with. Both of these effects can be predicted from the fact that the determination of the HPSG type in this model is essentially a local operation, dependent only on the predicted POS tag for that word. With the prediction of POS tags for English a very high precision operation, we are almost always assured an accurate POS tag, from which to map into an HPSG type; it is not dependent on previous HPSG type predictions for prior words.

The fact that the alternative sequence model eventually overtakes the one-sided CRF is particularly interesting, and may mean that the determination of which type of unknown word handling to use could be dependent on the corpus being used and the type of data expected to appear. Alternatively, it may be the case that both methods or some combination of them could be used to take advantage of their relative strengths at the appropriate times through the use of an ensemble type classifier.

4.3 Parse Accuracy

The evaluation of the final parses associated with HPSG trees is slightly complicated by the question of whether to evaluate the syntax or the semantics primarily. Here, I follow the precedent set by Dridan and Oepen [16] by using the metric of Elementary Dependency Matching (EDM). This metric is essentially equivalent to the PARSEVAL metric, except it is defined over the Minimal Recursion Semantics (MRS) [11] representation of the semantics of the sentence rather than the constituents of a syntactic tree. As with PARSEVAL, a perfect match scores a 100.0.

These metrics operate by breaking down the gold standard parse into small, self-contained units. Then, the scoring metric is simply defined as the percentage of these smaller units which the system output correctly predicts. In the case of PARSEVAL, each node of a syntactic parse tree covers a certain set of words from the sentence, and these nodes serve as the small units. A system output tree that also contains a node with the same constituent words will score as a match. In the case of EDM, the small units are elementary pieces of semantic structure, for instance if the sentence contains 'she', one piece of elementary semantics will be the property that 'she' has a GEND (gender) feature of *f*.

As described previously, the actual parsing was done with the PET parser [6]. In the cases where multiple possible parses for a given input sentence were returned, the top ranked parse as determined by PET was chosen as the representative parse.

Unknown Word Handler	1-2 UNK	3-4 UNK	5+ UNK	Overall
One-Sided CRF	**81.5**	**78.2**	72.1	**78.5**
Alternative Sequence	79.1	76.1	**73.1**	77.3

Table 9: Parse Accuracy

The data in Table 9 shows parse accuracies that are slightly below that of previous work, although this is probably to be expected given that the predicted types themselves were slightly less accurate. Note that the table here is showing accuracies over varying total numbers of unknown words in a sentence, and not sequences. This choice was made to avoid the fact that simply selecting sentences with a double unknown sequence doesn't control for the total number of unknown words in the sentence. Since parse accuracy is a whole-sentence metric, controlling for the total number of unknowns is more appropriate. Also note that although there were a relatively small percentage of sentences in the 5+ category, the massive scale of the Wikiwoods corpus means that there are still almost one million sentences in that category.

As in the type accuracy numbers, the alternative sequence model eventually overtakes the one-sided CRF as the number of unknowns increases. Unlike the type accuracy numbers however, there is not as sharp of a falloff, although it is clear that the one-sided CRF begins to struggle on more highly degraded sentences.

Chapter 5

Conclusion

The primary goals of this work were to answer a number of questions related to the prediction of detailed syntactic types for unknown words, and the behavior of unknown words in general, in HPSG and other lexically rich syntactic representations.

It was determined that, in a neutral corpus for which the grammar being used was not specifically developed against, over 12% of sentences contain sequences of two or more unknown words, and many sentences were found to contain a high number of unknowns. However, the percentage of sentences containing an unknown sequence is perhaps higher than would be predicted by random chance, given the statistics related to the total number of unknown words in a sentence. Thus, it seems likely that there is some other factor making it more likely for unknown words to occur in pairs or triples. On inspection, it is not immediately apparent that these sequences fall into one particular category such as proper names. Together, these facts suggest that evaluation of unknown word handling should explicitly deal with such highly degraded sentences and that evaluation setups featuring sentences with just a single unknown may not be accurately modelling the situations encountered by parsers in real-world environments.

With a substantial portion of sentences containing either sequences of unknowns or a large total number of unknowns, the question of what type of unknown word handler performs best on the these highly degraded sentences was investigated. Two strategies were considered that were intended to be representative of the two major classes of unknown word handlers previously proposed in the literature; these strategies included a direct sequence classifier in the form of a CRF that used features from only one side of the unknown word, along with an alternative sequence model that was intended to mimic the behavior of generic lexical entries.

It was determined that, in general, the direct sequence classifier using a CRF achieved higher scores in both type and parse accuracy, while the alternative sequence model achieved slightly higher coverage. The performance of the direct sequence classifier fell off sharply in the context of longer unknown sequences, while the alternative sequence model was able to provide consistent (though slightly lower performance) results even in these long unknown sequences.

With the results being variable on the total number of unknowns and unknown sequences, it seems plausible that a setup might make use of both strategies; the direct one-sided CRF for isolated unknowns, or unknowns in shorter sequences, and the alternative sequence model for unknowns in longer sequences. Alternatively, an examination of the corpus or an analysis of the type of data the parser expects to encounter should drive the selection of unknown word handler.

Bibliography

[1] Jason Baldridge. Weakly supervised supertagging with grammar-informed initialization. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 57–64. Association for Computational Linguistics, 2008.

[2] Srinivas Bangalore and Aravind K Joshi. Supertagging: An approach to almost parsing. *Computational linguistics*, 25(2):237–265, 1999.

[3] Petra Barg and Markus Walther. Processing unknown words in HPSG. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 91–95. Association for Computational Linguistics, 1998.

[4] Emily M Bender, Dan Flickinger, and Stephan Oepen. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation-Volume 15*, pages 1–7. Association for Computational Linguistics, 2002.

[5] Philip Blunsom. *Structured classification for multilingual natural language processing*. PhD thesis, Citeseer, 2007.

[6] Ulrich Callmeier. PET-a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–107, 2000.

[7] Kostadin Cholakov. Towards morphologically enhanced automated lexical acquisition. *Journal Of the European Summer School for Logic, Language, and Information*, page 117, 2009.

[8] Stephen Clark and James R Curran. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of the 20th international conference on Computational Linguistics*, page 282. Association for Computational Linguistics, 2004.

[9] Ann Copestake. *Implementing typed feature structure grammars*, volume 110. CSLI publications Stanford, 2002.

[10] Ann Copestake. Report on the design of RMRS. *DeepThought project deliverable*, 2003.

[11] Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332, 2005.

[12] Ann A Copestake and Dan Flickinger. An Open Source Grammar Development Environment and Broad-coverage English Grammar Using HPSG. In *Proceedings of the 2nd Language Resource and Evaluation Conference*, 2000.

[13] Hal Daumé III and Jagadeesh Jagarlamudi. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 407–412. Association for Computational Linguistics, 2011.

[14] Rebecca Dridan. *Using Lexical Statistics to Improve HPSG Parsing.* PhD thesis, Saarland University, 2009.

[15] Rebecca Dridan, Valia Kordoni, and Jeremy Nicholson. Enhancing Performance of Lexicalised Grammars. In *Proceedings of the 46th Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 613–621, 2008.

[16] Rebecca Dridan and Stephan Oepen. Parser Evaluation Using Elementary Dependency Matching. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 225–230, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[17] Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. WikiWoods: Syntacto-Semantic Annotation for English Wikipedia. In *Proceedings of the 7th Language Resource and Evaluation Conference*, 2010.

[18] Dan Garrette, Jason Mielens, and Jason Baldridge. Real-World Semi-Supervised Learning of POS-Taggers for Low-Resource Languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 2013.

[19] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*, 2001.

[20] Pierre Lison and GM Kruijff. Efficient parsing of spoken inputs for human-robot interaction. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 885–890. IEEE, 2009.

[21] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

[22] McClosky, David and Charniak, Eugene and Johnson, Mark. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 28–36, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[23] Yusuke Miyao, Takashi Ninomiya, and Junichi Tsujii. Corpus-oriented grammar development for acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pages 684–693. Springer, 2005.

[24] Jeremy Nicholson, Valia Kordoni, Yi Zhang, Timothy Baldwin, and Rebecca Dridan. Evaluating and Extending the Coverage of HPSG Gram-

mars: A Case Study for German. In *Proceedings of the 6th Language Resources and Evaluation Conference*, 2008.

[25] Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. The LinGO Redwoods Treebank Motivation and Preliminary Applications. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 2*, COLING '02, pages 1–5, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[26] Carl Pollard and Ivan Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

[27] Mark Steedman and Jason Baldridge. Combinatory categorial grammar. *Non-Transformational Syntax Oxford: Blackwell*, pages 181–224, 2011.

[28] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[29] Naoki Yoshinaga and Yusuke Miyao. Grammar conversion from LTAG to HPSG. In *Proceedings of the sixth European Summer School in Logic, Language, and Information Student Session*, pages 309–324, 2001.

[30] Yao-zhong Zhang, Takuya Matsuzaki, and Jun'ichi Tsujii. HPSG supertagging: A sequence labeling view. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 210–213. Association for Computational Linguistics, 2009.

[31] Yi Zhang and Valia Kordoni. A statistical approach towards unknown word type prediction for deep grammars. In *Proceedings of the Australasian Language Technology Workshop*, pages 24–31, 2005.