Analysis of Unknown Lexical Items using Morphological and Syntactic Information with the TIMIT Corpus

Scott M. Thede and Mary Harper
Purdue University
{thede,harper}@ecn.purdue.edu

Abstract

The importance of dealing with unknown words in Natural Language Processing (NLP) is growing as NLP systems are used in more and more applications. One aid in predicting the lexical class of words that do not appear in the lexicon (referred to as *unknown words*) is the use of syntactic parsing rules. The distinction between closed-class and open-class words together with morphological recognition appears to be pivotal in increasing the ability of the system to predict the lexical categories of unknown words. An experiment is performed to investigate the ability of a parser to parse unknown words using morphology and syntactic parsing rules without human intervention. This experiment shows that the performance of the parser is enhanced greatly when morphological recognition is used in conjunction with syntactic rules to parse sentences containing unknown words from the TIMIT corpus.

1 Introduction

One of the problems facing natural language parsing (NLP) systems is the appearance of unknown words; words that appear in sentences, but are not contained within the lexicon for the system. This problem is one that will only get worse as NLP systems are used for more on-line computer applications. New words are continually added to the language, and people will often use words that a parsing system may not expect.

This paper will empirically investigate how well a dictionary of closed-class words, syntactic parsing rules, and a morphological recognizer can parse sentences containing unknown words in natural language processing tasks. Syntactic knowledge can be used to aid in the analysis of unknown words—sentence structure can be a strong clue as to the possible part of speech of an unknown word. The distinction between closed-class and open-class words should help to refine the possibilities for an unknown word and enhance the information provided by the syntactic knowledge. Morphological recognition can also be helpful in predicting possible parts of speech for many unknown words. We expect that these three knowledge sources will greatly improve our parser's ability to process and cope with words that are not in the system lexicon.

2 Problem Description

A major problem that can occur when parsing sentences is the appearance of unknown words—words that are not contained in the lexicon of the system. As mentioned in Weischedel, et al. [15], the best-performing system at the Second Message Understanding Conference (MUC-2) simply halted parsing when an unknown word was encountered. Clearly, for a parser to be considered robust it must have mechanisms to process unknown words. The need to cope with unknown words will continue to grow as new words are coined, and words associated with sub-cultures leak into the main-stream vocabulary. In the case of a large corpus, especially one with no specific domain, a comprehensive lexicon is prohibitive. Thus, it is important that parsers have the ability to cope with new words.

There are two broad approaches to handling unknown words. The first approach is to attempt to construct a complete lexicon, then deal with unknown words in a rudimentary way—for example, rejecting the input or interacting with the user to obtain the needed information about the unknown word. The second way is to attempt to analyze the word at the time of encounter with as little human interaction as possible. This would allow the parser to parse sentences containing unknown words in a robust and autonomous fashion. Unknown words could be learned by discovering their part of speech and feature information during parsing and storing that information in the lexicon. Thus, if the word is encountered again later, it now would be in the lexicon.

Before examining the problem more fully, it is useful to consider work that has already been done on the problem. There have been several attempts to study the problem of learning unknown words. These attempts have followed several different methodologies and have focused on various aspects of the unknown words.

Statistical methods are most commonly used in part-of-speech tagging. Charniak's paper [5] outlines the use of statistical equations in part-of-speech tagging. Tagging systems make only limited use of the syntactic knowledge inherent in the sentence, in contrast to parsers. An n-gram tagger concentrates on the n neighbors of a word (where n tends to be 2 or 3), ignoring the global sentence structure. Also, many part-of-speech tagging systems are only concerned with resolving ambiguity, not dealing with unknown words.

Kupiec [8] and Brill [3] make use of morphology to handle unknown words during part-of-speech tagging. Brill's tagger begins by tagging unknown words as proper nouns if capitalized, common nouns if not. Then the tagger learns various transformational rules by training on a tagged corpus. It applies these rules to unknown words to tag them with the appropriate part-of-speech information. Kupiec's hidden Markov model uses a set of suffixes to assign probabilities and state transformations to unknown words. Both these methods work well, but they ignore the global syntactic content of the sentence. We will examine the effects of combining morphology and syntax, while using a deterministic system to perform parsing.

Weischedel, et al [15] study the effectiveness of probablistic methods for part-of-speech tagging with unknown words. They show that using morphological information can increase the accuracy of their tagger on unknown words by a factor of five. They also briefly address the effect of unknown words in parsing with their part-of-speech tagger. However, all of their results assume that only one unknown word is present in each sentence or is within the tri-tag range of the tagger. They also assume that automatic disambiguation will eliminate extraneous parses. These assumptions will not always hold while parsing many corpora.

In his thesis, Tomita [13] mentions the impact of syntax on determining the part of speech of unknown words during parsing. He suggests that his system can handle unknown words by simply assigning them all possible parts of speech (without using any morphological analysis of the words). He performs no experiment to assess his method's viability, but we will demonstrate that this is not a good approach. The use of all possible parts of speech will cause an exponential increase in the number of parses for a sentence as the number of unknown words increases.

The FOUL-UP system [6], by Granger, is an example of a method that focuses on the use of context. This method assumes that most words are known, and that all sentences lie in a common semantic domain. In FOUL-UP, a strong top-down context in the form of a script is needed to provide the expected attributes of unknown words. The required use of a script to provide information limits the applicability of this method to situations where scripts are available.

Jacobs and Zernik use a combination of methods in the SCISOR system [7]. Though this system performs morphological and syntactic analysis, SCISOR was designed to be used in a single domain. When new words are discovered, they tend to be given specialized meanings that are related to the semantic domain, limiting the system to that specific domain.

There is a need for a parsing system that can act over less precisely defined domains and still efficiently cope with unknown words. We feel that the use of morphological recognition, a small lexicon of closed-class words, and a dictionary of known open-class words can be used to help our parser to determine the parts of speech for unknown words as they occur. For this research, we define a word by its parts of speech and a small set of features. These features include :

- Number (singular / plural)
- Person (first / second / third)
- Verb form (infinitive / present / past / past participle / *-ing* participle)
- Noun type (common / proper)
- Noun case (subjective / objective / possessive / reflexive)
- Verb sub-categorization (12 different categories)

The parser in the experiment outlined in this paper has been developed with two specific goals:

- It will be able to parse sentences with an incomplete lexicon with as few errors as possible.
- It will operate without any human interaction.

3 Techniques for Word Analysis

This section will cover the various concepts used by our parser to help in the processing of unknown words. First, the importance of closed-class words will be discussed. Second, the morphological recognition system is detailed. Third, the usefulness of syntactic knowledge is explained. Finally, the post-mortem method, which integrates these concepts into a parsing system, is described.

3.1 Closed-Class versus Open-Class Words

An important source of information that is used in this experiment is the distinction between closed-class and open-class words. This distinction can be used to develop a small core dictionary of closed-class words that can greatly ease the task of processing unknown words in a sentence.

Closed-class parts of speech are those parts of speech that may not normally be assigned to new words. Closed-class words are words with a closed-class part of speech. For example, pronouns and determiners are members of the closed-class set of words; it is very rare that a new determiner or pronoun is added to the language. Closed-class parts of speech are such that all the words with that part of speech can be enumerated completely. In addition, these closed-class words are not generally used as other parts of speech. This research does not consider the meta-linguistic use of a word, as in this sentence: *"The" is a determiner.*

There are a number of closed-class parts of speech, including determiners, prepositions, conjunctions, predeterminers, and quantifiers. Pronouns and auxiliary verbs (*be*, *have*, and *do*) are also closed-class parts of speech. In addition, we designate irregular verbs as closed-class words for this research since they are a static part of the language. The irregular verbs are enumerated by Quirk, et al [12]. Typically new verbs in a language are not coined as irregular verbs. The enumeration of irregular verbs allows the recognition of unknown verb forms to be rule-based. For example, all past tense regular verbs end in *-ed*, third person singular regular verbs end in *-s*, and so on. For similar reasons, irregular noun plurals are included in the set of closed-class words.

Open-class parts of speech are those parts of speech that accept the addition of new items with little difficulty. Open-class words are comprised of words with the following parts of speech: *nouns*, *verbs*, *adjectives*, and *adverbs*. *Noun modifier* is a fifth part of speech that is used in this research to indicate those words that can be used to modify nouns; this will eliminate extraneous parses that occur when a word defined as both a noun and adjective is used to modify a head noun. This part of speech is also used by Cardie [4] in her experiments.

The existence of a set of closed-class words allows the construction of a dictionary in such a way as to facilitate the detection and analysis of unknown words. By creating a dictionary containing all the closed-class words, some words in any sentence will very likely be known. A limit is also placed on the possible parts of speech of unknown words. They cannot have a closed-class part of speech, since the closed-class words are enumerated in the dictionary. So unknown words are assumed to be open class, restricting them to noun, verb, adjective, adverb, or noun modifier.

3.2 Morphological Recognition

For convenience, we split the field of morphology into three different areas—morphological generation, morphological reconstruction, and morphological recognition. Morphological generation research examines the ability of morphological affixation rules to generate new words from a lexicon of base roots. Theoretical linguists and psychologists are interested in morphological generation for its use in linguistic theory or in understanding how people learn a language. For example, Muysken [11] studied the classification of affixes according to their order of application for a theoretical discussion on morphology. Badecker and Caramazza [2] discussed the distinction between inflectional and derivational morphology as it applies to acquired language deficit disorder, and, in general, to the theory of language learning. Baayen and Lieber [1] studied the productivity of certain English affixes in the CELEX lexical database, in an effort to study the differences between frequency of appearance and productivity. Viegas, et al [14] show that the use of lexical rules and morphological generation can greatly aid in the task of lexical acquisition. However, morphological generation involves the construction of new word forms by applying rules of affixation to base forms, and so it is only indirectly helpful in the analysis of unknown words.

Morphological reconstruction researchers process an unknown word by using knowledge of the root stem and affixes of that word. For example, Milne [10] makes use of morphological reconstruction to resolve lexical ambiguity while parsing. Light [9] uses morphological cues to determine semantic features of words by using various restrictions and knowledge sources. Jacobs and Zernik [7] make use of morphology in their case study of lexical acquisition, in which they attempt to augment their lexicon using a variety of knowledge sources. However, they assume a large dictionary of general roots is available, and that the unknown words tend to have specialized meanings. Morphological reconstruction research relies on the presence of stem information, making morphological reconstruction of less value for coping with unknown words. If an unknown word is encountered, the root of that word is likely to also be unknown. A method to cope with unknown words cannot be based on knowledge of the root if the root is also unknown.

Morphological recognition uses knowledge about affixes to determine the possible parts of speech and other features of a word, without utilizing any direct information about the word's stem. There has been little research done in this area. As noted above, many of the uses of morphology for analysis of lexical items require more knowledge than may be possible for some applications, especially if the system will be using a limited lexicon but will be expected to cope with words in the language but not found in the lexicon.

There is one caveat concerning the use of only affix information in a morphological recognizer. Since the root of an unknown word is assumed to be unknown, the recognizer can only consider whether an affix matches the word. This can lead to an interesting type of error. For example, while checking the word *butterfly*, the affix *-ly* matches as a suffix. Typically, the *-ly* affix is attached to adjectives to form adverbs (e.g., *happy* → *happily*), or to nouns to form adjectives (e.g., *beast* → *beastly*); however, the word *butterfly* was not formed by this process, but rather by compounding the words *butter* and *fly*. Without the knowledge that **butterf* is not an acceptable root word, and without some notion of legal word structure, there is no way to determine that *butterfly* was not formed by applying *-ly*. So in this case, *butterfly* is mistakenly assumed to have the *-ly* suffix. By using additional information, a morphological recognizer could circumvent this problem. However, we still believe that a simple morphological recognition module can be useful. The question is: how effective can it be?

For the morphological recognition module in this system, we constructed a list of suffixes and prefixes by hand, using lists found in Quirk, et al [12]. Of the two, suffixes general offer more effective constraints on the possible parts of speech of a word. For each of these affixes, we constructed two lists of parts of speech. The *first-choice list* contains those parts of speech that are most likely to be found in words ending (suffix) or beginning (prefix) with that affix. The *second-choice list* contains those parts of speech that are fairly likely to be found in words ending or beginning with that affix. The first-choice list is a subset of the second-choice list. The two lists for each affix were created by hand, using rules described in Quirk, et al [12]. The creation of two separate lists is used by the post-mortem parsing approach of our experimental system, and the use of the two lists will be detailed in that section.

3.3 Syntactic Knowledge

Syntactic knowledge is used implicitly in this research by the parser. When an unknown word is encountered in a sentence, its context in that sentence can be of great importance when predicting its part of speech. For example, a word directly following a determiner is typically a noun or noun modifier. Consider the unknown word *smolked* used in this sentence:

> The cat *smolked* the dog.

Assuming that all the other words in the sentence are in the lexicon, then based on purely syntactic knowledge the unknown word must be a finite tense verb, either past or present tense. However, the use of morphological recognition can refine this information. The *-ed* ending on *smolked* indicates either a past tense verb or a past participle. By combining the syntactic and morphological information, the word *smolked* is identified as a past tense verb. Further, it is a verb which takes a noun phrase as its object. Thus, the syntactic information can augment the morphological information, and vice versa. Obviously, the more words in a sentence that are defined in the lexicon, the more the syntactic knowledge can limit possible parts of speech of unknown words.

3.4 Sentence Parsing with Post-Mortem Analysis

The information sources in the preceding three sections are combined in our experimental post-mortem parsing system. The following approach is used to allow our parser to cope with unknown words in a sentence:

1. COMBINED LEXICON AND FIRST-CHOICE MORPHOLOGICAL RECOGNIZER

   The lexicon is consulted for each word in the sentence. If the word is defined in the lexicon, its definition—consisting of the word, its part of speech, and various features affecting its use—is used to parse the sentence. If it is not in the lexicon, we assume that it can only be an open-class part of speech, and it could possibly be any of them. In this pass, we use the morphological recognizer to reduce the number of possible parts of speech for the word. Consulting a list of affixes, the recognizer determines which affix, if any, are present in the word. Then the recognizer assigns all the parts of speech from the first-choice list to that word. For example, an unknown word ending in *-ly* is assumed to be an adverb. A parse forest for the sentence is generated. If the parse fails, the parser moves to pass two.

2. COMBINED LEXICON AND SECOND-CHOICE MORPHOLOGICAL RECOGNIZER

   If the sentence fails its first attempt to parse, then it is reparsed using the second-choice list for the affix, instead of the first-choice list. This will assign a more liberal set of parts of speech to the word based on its affix. For example, an unknown word ending in *-ly* is now assumed to be an adverb, adjective, and a modifier. Again, the parse forest for the sentence is returned. If the sentence fails to parse, the parser goes on to pass three.

3. ALL OPEN-CLASS VARIANTS FOR UNKNOWN WORDS

   If the sentence fails its second parse attempt, it is reparsed assigning all open-class lexical categories to every unknown word. For example, an unknown word ending in *-ly* is now assumed to be a noun, verb, adverb, adjective, and modifier. The parse forest for the sentence is returned. If the sentence fails to parse in this pass, the parser moves on to pass four.

4. ALL MORPHOLOGICAL VARIANTS FOR ALL OPEN-CLASS WORDS

   If the sentence fails its third parse attempt, it is reparsed using the morphological recognizer on all open-class words. It assigns a set of parts of speech to each word, again based on the second-choice list for that word's affixes. Note that only the closed-class lexicon is consulted during this attempt to parse. The morphological recognizer is used to determine definitions for all open-class words. This approach allows the parsing system to find new definitions for words that are already in the dictionary. For example, *any* word ending in *-ly* is now assumed to be an adverb, an adjective, and a modifier. The parse forest for the sentence is returned. If the sentence fails to parse in this pass, the parser moves on to pass five.

5. ALL OPEN-CLASS VARIANTS

   If the sentence still fails to parse, all possible open-class parts of speech are given to all open-class words in the sentence. Again, only the closed-class lexicon is consulted. For example, any word ending in *-ly* is now assumed to be a noun, verb, modifier, adjective, and adverb. The parse forest is returned, or the sentence fails completely.

This system is a *post-mortem* error handling technique—if (and only if) the sentence fails to parse, the parser tries again, using a more liberal interpretation in its word look-up algorithm. The idea is to limit the possibilities in the beginning to those that are most likely, and broaden the search space later if the first methods fail. The use of this approach will limit the possible parts of speech for many unknown words.

4 Description of Experiment

The parsing system used in this experiment is based on a Tomita [13] style parser. It is an LR-type parser, using a shift-reduce algorithm with packed parse forests. It is implemented in Common Lisp. The test corpus is a set of 356 sentences from the Timit corpus [16], a corpus of sentences that has no underlying semantic theme. It was originally designed as a corpus for training phoneme recognition for speech processing. This corpus was specifically chosen for our tests because it has no theme, and would thus offer a wider range of sentence types.

The rule set was designed first to properly parse the test corpus, and second to be as general as possible. It is hoped that these rules could deal with a wide variety of English sentences, though they have not been tested on other corpora. The issue of the size of the grammar is not addressed in this experiment. Also, the possible failure of the parser due to insufficient rule coverage is not considered. There are many applications that use grammars that do not cover an extensive range of English sentences, and these applications would benefit from our mechanisms for dealing with unknown words.

There are four separate files that constitute the lexicon for this parser and corpus:

- nouns - contains all irregular noun plurals.
- verbs - contains all forms of all irregular verbs.
- closed - contains all other closed-class words.
- dict - actually a set of files, named dict1 to dict10. Dict10 contains all the words from the corpus that are not contained in the other three files and are thus all open-class words. Dict9 contains 90% of the words from dict10, dict8 contains 80%, and so on down to dict1, which contains 10% of these words. All these files were created randomly and independently using the words in dict10. These percentages are based on a word count, not on a definition count—many words have more than one definition. For example, *acts* is one word, but it has two definitions—as a noun and a verb.

For our experiment, we perform two data runs. The first run uses a variation Tomita's method; it assigns all possible open-class parts of speech (noun, verb, adjective, adverb, and modifier) to unknown words. This data run is called the **baseline** run. The second run assigns parts of speech using the post-mortem approach described in Section 3.4. This data run is called the **experimental** run.

For each test run, the sentences in the corpus are parsed by the system eleven times. Each time through the corpus, all the closed-class words are loaded into the lexicon. In each separate run, a different open-class dictionary is used. For the first run, the full dictionary found in dict10 is used. This run is used as a control, since all words in the corpus are defined in the lexicon. For each successive run, the next dictionary is used, from dict9 down to dict1. Finally, the eleventh run is done without loading any extra dictionary files—only the three closed-class files are used.

After all of the parse trees have been generated, each run is compared to the control run, and three numbers are calculated for each sentence in each run—the number of *matches*, *deletions*, and *insertions*. A *match* occurs when the sentence has generated a parse that occurs within the control parse forest for that sentence. A *deletion* occurs when the sentence has failed to produce a parse that occurs in the control parse forest for that sentence. An *insertion* occurs when the sentence produces a parse that does not occur in the control parse forest for that sentence. For example, assume sentence #16 produces parses A, B, C, and D in the first, or control, run. In a later run, sentence #16 produces parses A, C, E, F, and G. Then, for sentence #16, we have two matches (A and C), two deletions (B and D), and three insertions (E, F, and G). By using these measurements, we can determine the precision and recall of the parsing system when parsing sentences with unknown words.

The issue of disambiguation is not explicitly dealt with in this experiment. We wish to see how well the morphological recognizer can replicate the performance of a parser with a full dictionary. This is demonstrated in our use of the *match*, *insertion*, and *deletion* numbers above. The number of matches is important, since ideally we want the recognizer to return all possible parses that occur when the full dictionary is used. The issue of which of these parses is the correct one would require that we utilize semantic, pragmatic, and contextual information to select the correct parse, a topic beyond the scope of our experiment.

5 Results

Two sets of data were collected in this experiment, one for the baseline run and one for the experimental run. Figure 1 compares the deletion rates, Figure 2 shows the insertion rates, and Figure 3 shows the match rates. Table 1 shows the total number of deletions and insertions, as well as each as a percentage of the total number of parses for all the sentences in the corpus (1137).

In Figure 1, we see that the deletion rate for the baseline data is zero, as expected. Since every possible part of speech is assigned to each unknown word, all the original parses should be generated. The deletion rate for the experimental run shows that some parses are being deleted when there is one or more unknown words. With 10% of the open-class dictionary missing, there are 22 deletions out of 1137 total possible parse matches. This is an average of only 0.045 deletions per sentence, or 1.9% of the total parses, as shown in Figure 1 and Table 1. With 100% of the open-class dictionary missing—in other words, using only closed-class words—there are 225 deletions, an average of 0.63 per sentence, or 19.8% of the total parses. In other words, over 80% of the original parses are produced. The deletion rate is in part due to the fact that many words in the complete dictionary are lexically ambiguous; whereas, many times the morphological recognizer assigns a smaller set of parts of speech, which can result in a correct parse being generated, but not the entire parse forest.

The true value of the experimental approach can be seen when comparing the insertion rates. Figure 2 shows that the insertion rate in the baseline data is enormous. Data are only available up to 30% of the open-class dictionary missing, because after this point the program runs out of memory for storing all of the spurious parses. Since the baseline parser assigns every open-class part of speech to an unknown word, a combinatorial explosion of parses occurs. In the baseline case with 30% of the open-class words missing, there are a total of 110,942 insertions, or 311.6

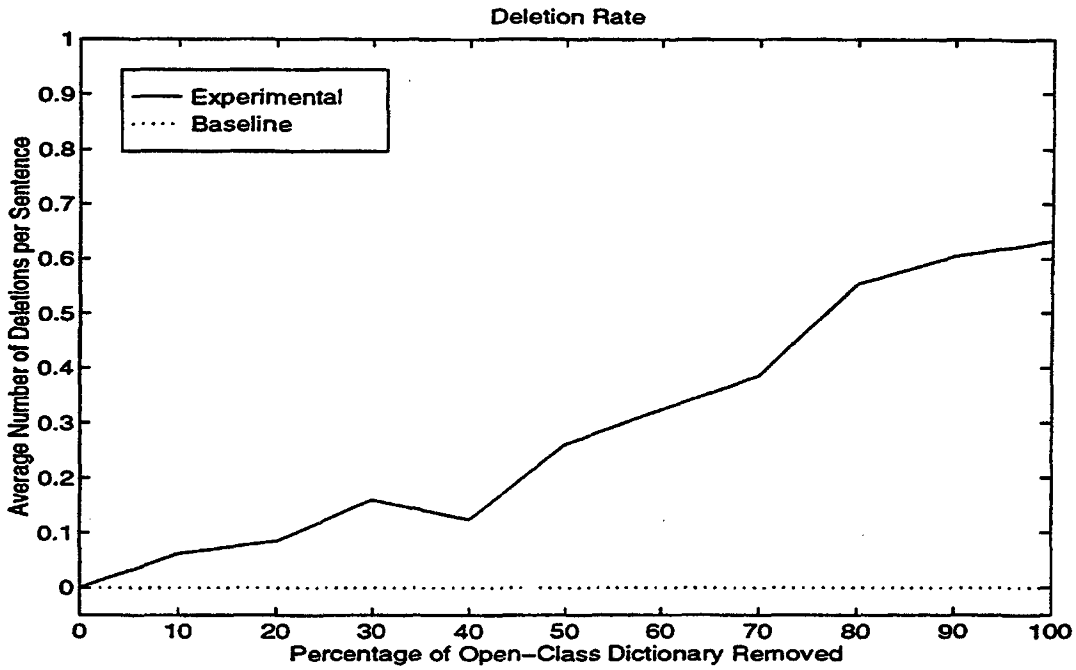

Figure 1: Average Number of Deletions

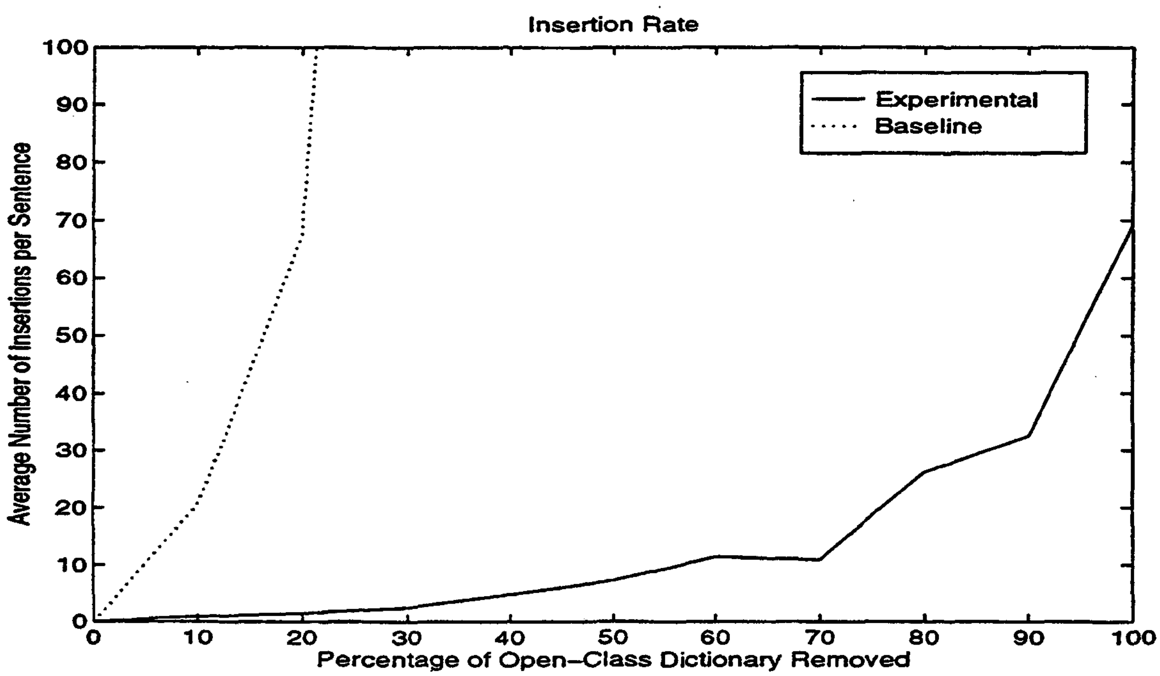

Figure 2: Average Number of Insertions

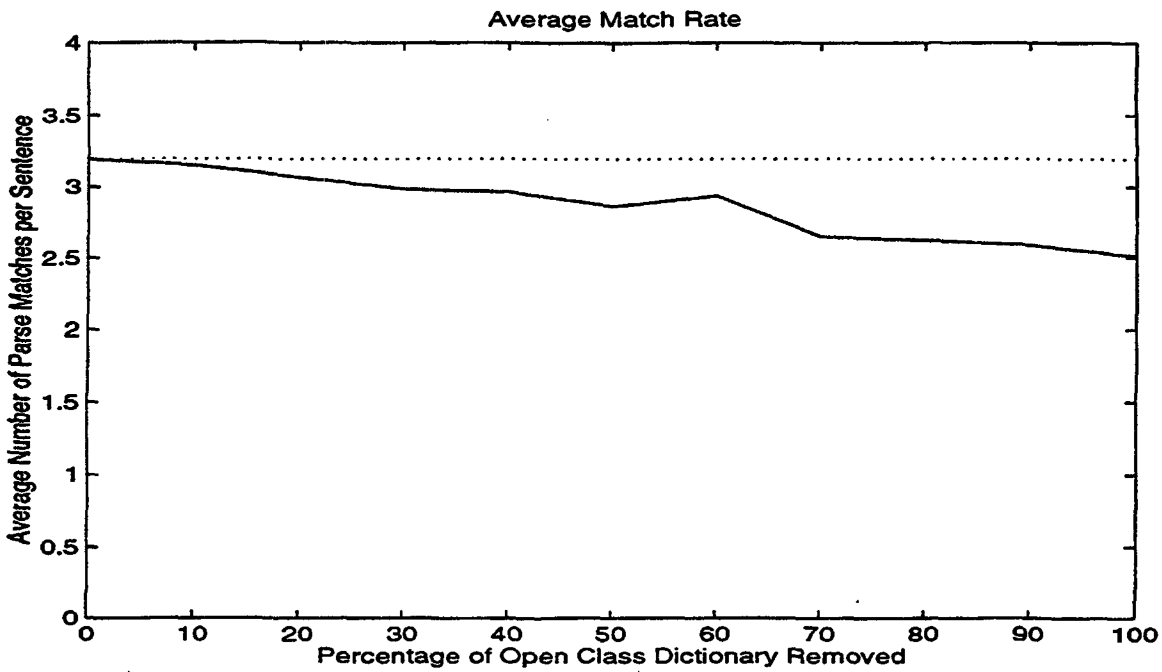

Figure 3: Average Number of Matches

per sentence. At the same point in the experimental run (30% missing), there are only 2.5 insertions per sentence, less than one-hundredth of the baseline value. With 100% of the open-class dictionary missing, there are 69.3 insertions per sentence when using the morphological recognizer in the experimental run, as compared to the 311.6 insertions per sentence when using the baseline method with only 30% of the dictionary missing. Performance of the experimental system in terms of insertions with 100% of the open-class dictionary missing is comparable to the baseline performance with 20% of the open-class dictionary removed. Obviously, the experimental data shows that the insertion rate has been cut drastically from the baseline performance.

6 Conclusions

We have shown that morphological recognition, the distinction between closed-class and open-class words, and syntactic knowledge are powerful tools in handling unknown words, especially when we use a post-mortem method of determining the probable lexical classes of words. These knowledge sources allow us to determine parts of speech for unknown words without using domain-specific knowledge in the TIMIT corpus. The insertion rate can be drastically reduced with only a moderate increase in the deletion rate. Obviously, there is a trade-off between the deletion rate and the insertion rate. This tradeoff can be manipulated by altering the morphological rules to place more importance on a low deletion rate or a low insertion rate, by modifying our post-mortem approach to obtain finer control over the process of handling unknown words, or by considering additional knowledge sources. This issue should be of interest for any researcher developing a parsing system that will need to deal with unknown words.

Future work will investigate the effectiveness of the morphological recognizer. We would like to compare a computer-generated morphological recognition module with the hand-generated

Deletion and Insertion Rates								
% of Open	Experimental Data				Baseline Data			
Class Dict	Deletions		Insertions		Deletions		Insertions	
Removed	Total	Percent	Total	Percent	Total	Percent	Total	Percent
0	0	0%	0	0%	0	0%	0	0%
10	22	1.9%	315	27.7%	0	0%	7248	636.8%
20	30	2.6%	533	46.9%	0	0%	24095	2118.6%
30	57	5.0%	892	78.5%	0	0%	110942	9757.4%
40	44	3.9%	1718	151.1%	—	—	—	—
50	93	8.2%	2598	228.5%	—	—	—	—
60	115	10.1%	4065	357.5%	—	—	—	—
70	137	12.0%	3866	340.0%	—	—	—	—
80	197	17.3%	9340	821.5%	—	—	—	—
90	215	18.9%	11545	1015.4%	—	—	—	—
100	225	19.8%	24663	2169.1%	—	—	—	—

Table 1: Deletion and Insertion Data

module used in this work. We would also like to test these morphological recognizers on other corpora. Finally, we would like to refine the post-mortem approach by offering a more elegant solution than the combination of first-choice and second-choice lists. There is information in the parse forest of failed parses that may allow single words to be identified as "problem words". This would allow the parser to reparse the sentence changing only a few words' definitions, providing better all-around performance.

References

[1] Harald Baayen and Rochelle Lieber. Productivity and English derivation: A corpus-based study. *Linguistics*, 29:801–843, 1991.

[2] William Badecker and Alfonso Caramazza. A lexical distinction between inflection and derivation. *Linguistic Inquiry*, 20(1):108–116, 1989.

[3] Eric Brill. A report of recent progress in transformation-based error-driven learning. *Proceedings of the Twelfth National Conference on Artifical Intelligence*, pages 722–727, 1994.

[4] Claire Cardie. A case-based approach to knowledge acquisition for domain-specific sentence analysis. *Proceedings on the Eleventh National Conference on Artificial Intelligence*, pages 798–803, 1993.

[5] Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. Equations for part-of-speech tagging. *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 784–789, 1993.

[6] Henry Granger, Jr. FOUL-UP : A program that figures out meanings of words from context. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 172–178, 1977.

[7] Paul Jacobs and Uri Zernik. Acquiring lexical knowledge from text : A case study. *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 739–744, 1988.

[8] Julian Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech and Language*, 6(3):225–242, 1992.

[9] Marc Light. Morphological cues for lexical semantics. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 25–31, 1996.

[10] Robert Milne. Resolving lexical ambiguity in a deterministic parser. *Computational Linguistics*, 12(1):1–12, 1986.

[11] Pieter Muysken. Approaches to affix order. *Linguistics*, 24:629–643, 1986.

[12] Randolph Quirk, Sidney Greenbaum, Geoffry Leech, and Jan Svartnik. *A Grammar of Contemporary English, Twelfth Impression*, pages 109–122, 976–1008. Longman Group, Ltd., 1986.

[13] Masaru Tomita. *Efficient Parsing for Natural Language.* PhD thesis, Carnegie-Mellon University, 1986.

[14] Evelyne Viegas, Boyan Onyshkevych, Victor Raskin, and Sergei Nirenburg. From *Submit* to *Submitted* via *Submission* : On lexical rules in large-scale lexical acquisition. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 32–39, 1996.

[15] Ralph Weischedel, Marie Meeter, Richard Schwartz, Lance Ramshaw, and Jeff Palmucci. Coping with ambiguity and unknown words through probabilitic models. *Computational Linguistics*, 19:359–382, 1993.

[16] Victor Zue, Stephanie Seneff, and James Glass. Speech database development at MIT : TIMIT and beyond. *Speech Communication*, 9(4):351–356, 1990.