Classifying New Words for Robust Parsing

Alexander Franz
Center for Machine Translation and Computational Linguistics Program
Carnegie Mellon University
Pittsburgh, PA 15213
amf@cs.cmu.edu

Abstract

Robust natural language parsing systems must be able to handle words that are not in their lexicons. This paper describes a statistical classifier that determines the most likely parts of speech of new words. The classifier uses a loglinear model to obtain smoothed conditional probabilities that take into account the interactions between different features. We show accuracy results for this model, and compare it to some simpler methods.

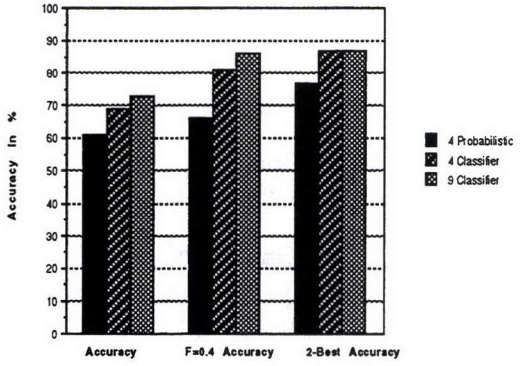

Figure 1: Performance of Different Models

1 Introduction

Current natural language parsing systems typically assume a closed vocabulary. For example, the most successful data extraction system at the second Message Understanding Conference would "simply halt processing when a new word was encountered" [Weischedel et al., 1993].

As natural language analysis systems move out of the realm of small, experimental domains with limited vocabulary and toward applications with open-ended vocabulary, robust methods to handle new words become necessary.

This paper describes a statistical classifier that determines the most likely parts of speech (POS) for unknown words. The classifier is used to supply the lexical probabilities for unknown words for a stochastic part of speech tagger. In this way, the most likely part of speech given the unknown word and the context is found, and the new word becomes amenable to further analysis.

We compare the results of the classifier with a number of simpler procedures for finding the parts of speech of unknown words, and show that the classifier obtains higher accuracy.

2 Constructing the Classifier

The classifier was constructed in the following way. First, features that could be used to guess the part of speech of a word, such as a prefix or suffix, were determined by examining a portion of an online text corpus.

Exploratory data analysis was performed in order to determine relevant features, their possible values, and approximate how they interact.

The training data were turned into feature vectors $\vec{v}$, and the feature vectors were cross-classified in a contingency table. The contingency table was smoothed using a loglinear model.

Let $P(i)$ the prior probability distribution over the possible parts of speech, and $P(\vec{v})$ the prior distribution for the features. From the training data that are annotated with the correct part of speech we can estimate $P(\vec{v}|i)$, the conditional probability distribution for $\vec{v}$, given the part of speech i.

Bayes rule can then be used to derive the conditional probability distribution for the correct interpretation, given the feature vector:

$$(1) \quad P(i|\vec{v}) = \frac{P(\vec{v}|i)P(i)}{P(\vec{v})}$$

As will be shown in Section 3.4, the conditional probabilities $P(i|\vec{v})$ can be obtained directly from the smoothed contingency table. Maximizing this conditional probability leads to *minimum error rate* classification [Duda and Hart, 1973].

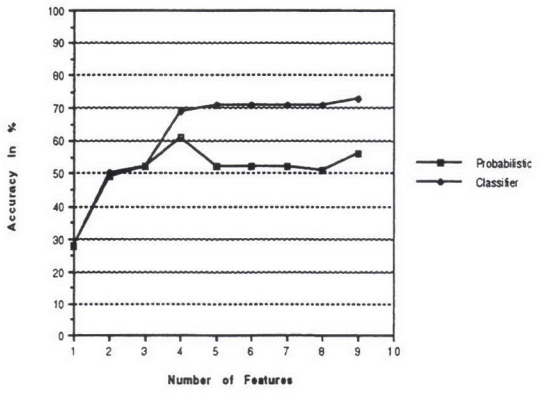

Figure 2: Effect of Number of Features on Accuracy

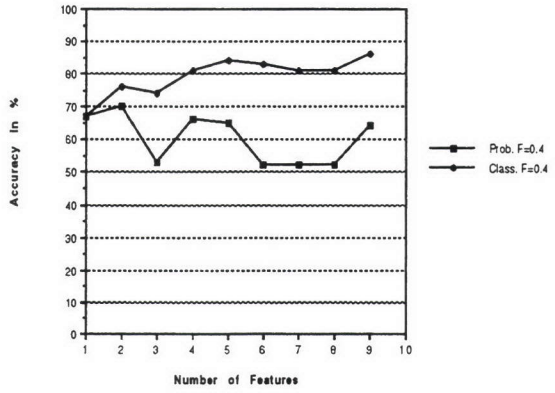

Figure 3: Effect of Number of Features on Cutoff-factor Accuracy

3 Estimating the Conditional Probabilities

This section describes how we used a loglinear model to estimate the smoothed conditional probabilities that are used for classification.

3.1 The Contingency Table

A contingency table was used to count the observed features. A contingency table is an array with one dimension for each feature; the size of the table is the product of the number of possible values for the features. Each cell in the contingency table records the frequency of data with the appropriate feature values.

Cells in the contingency table are addressed using the appropriate subscripts. In a table with four features, x_{ijkl} is the cell at level i of the first feature, level j of the second feature, and so on. Given a contingency table, the cell entries can be summed up to form *marginal totals*. A marginal total is represented by replacing the subscripts for the dimensions that are summed over by a plus sign. For example, the sum of all cell counts where the first feature is at level 1 is denoted by x_{1+++}.

3.2 Smoothing the Conditional Probabilities

We used a loglinear model as a "smoothing device, used to obtain cell estimates for every cell in a sparse array, even if the observed count is zero" [Bishop et al., 1975].

A loglinear model is a statistical model of the effect of the statistical features and their combinations on the cell counts in a contingency table. Marginal totals of the observed counts are used to estimate the parameters of the loglinear model; the model in turn delivers estimated expected cell counts, which are smoother than the original cell counts. Let m_{ijkl} be the expected cell count for cell (i, j, k, l). The values for the expected cell counts that are estimated by the model are represented by the symbol $\hat{m}_{ijkl}$. The general form of a loglinear model is shown in Equation 2:

$$(2) \quad \log m_{ijk\ldots} = u + u_{1(i)} + u_{2(j)} + u_{3(k)} + u_{\ldots} + ldots$$

u is the mean of the logarithms of all the expected counts, $u + u_{1(i)}$ is the mean of the logarithms of the expected counts at level i of the first feature, $u + u_{2(j)}$ is the mean of the logarithms of the expected counts at level j of the second feature, and so on. In other words, the term $u_{1(i)}$ represents the deviation of the expected cell counts at level i of the first feature from the grand mean u.

A loglinear model provides a way to estimate expected cell counts that depend not only on the main effects of the features, but also on the interactions between features. This is achieved by adding "interaction terms" to the model. For further details, see [Agresti, 1990].

3.3 The Iterative Estimation Procedure

For some loglinear models, it is possible to obtain closed forms for the expected cell counts. For more complicated models, an iterative estimation procedure is used. The *iterative proportional fitting* algorithm for hierarchical loglinear models was first presented by [Deming and Stephan, 1940]. Briefly, this procedure works as follows.

The interaction terms in the loglinear models represent constraints on the estimated expected marginal totals. Each of these marginal constraints translates into an adjustment scaling factor for the cell entries. The iterative procedure has the following steps:

1. Start with initial estimates for the estimated expected cell counts. For example, set all $\hat{m}_{ijkl} = 1.0$.
2. Adjust each cell entry by multiplying it with the scaling factors. This moves the cell entries towards satisfaction of the marginal constraints specified by the model.
3. Iterate through the adjustment steps until the maximum difference ϵ between the marginal totals observed in the sample and the estimated marginal totals becomes small enough, e.g. $\epsilon = 0.1$.

3.4 Obtaining Smoothed Conditional Probabilities

Recall that the smoothed cell counts in the contingency table are denoted by $\hat{m}_{ijkl}$, that $P(i_i)$ is the prior probability of the **Interpretation** feature having level i, and

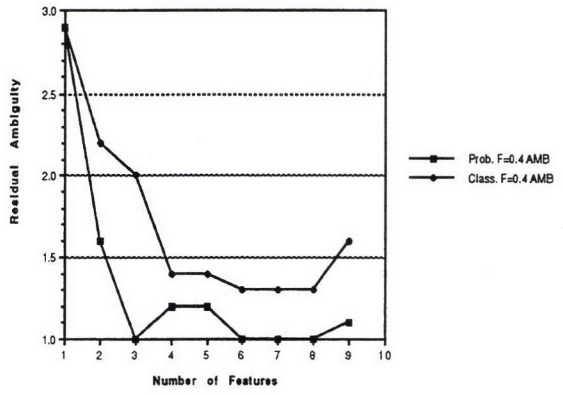

Figure 4: Effect of Number of Features on Residual Ambiguity

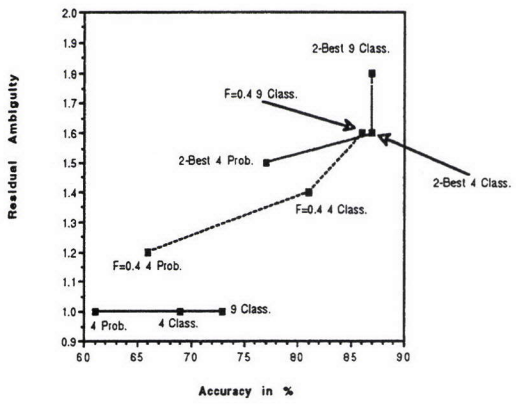

Figure 5: Ambiguity versus Accuracy in Different Models

that $P(\vec{v}_{jkl})$ is the probability of the remaining features in $\vec{v}$ having levels j, k, l.

The contingency table is used to obtain the probability distribution for i_i given values for the features in $\vec{v}$ in the following way:

$$P(i_i) = \frac{\hat{m}_{i+++}}{\hat{m}_{++++}}$$

$$P(\vec{v}_{jkl}) = \frac{\hat{m}_{+jkl}}{\hat{m}_{++++}}$$

$$P(\vec{v}_{jkl}|i_i) = \frac{\hat{m}_{ijkl}}{\hat{m}_{i+++}}$$

Substituting this into Bayes theorem yields the following:

$$P(i_i|\vec{v}_{jkl}) = \frac{P(\vec{v}_{jkl}|i_i)P(i_i)}{P(\vec{v}_{jkl})} = \frac{\hat{m}_{ijkl}}{\hat{m}_{+jkl}}$$

Thus, the prior probabilities $P(i_i)$ and $P(\vec{v}_{jkl})$ correspond to the marginal totals $\hat{m}_{i+++}$ and $\hat{m}_{+jkl}$, and the conditional probability $P(i_a|\vec{v})$ can be calculated directly from the cell entry $\hat{m}_{ijkl}$ and the marginal total $\hat{m}_{+jkl}$.

4 Experimental Results

This section describes the results of evaluating the classifier using online natural language corpora on the following problem: Given an unknown word in isolation or in context, find its most likely open class part(s) of speech. (The open class parts of speech, into which an unknown word must fall, are summarized in Table 1.)

4.1 Data

To obtain training and evaluation data, the Penn Treebank Brown corpus [Marcus et al., 1993] was divided at random, on a file-by-file basis, into three sets:

1. A set S_1 containing 400,000 words was selected to represent the "known words".
2. A set S_2 containing 300,000 words was selected to represent the new text. Every word in S_1 but not in S_2 is considered an unknown word. These unknown words make up the training data.
3. A third set S_3 containing 300,000 words was selected to evaluate the model. Every word in S_2 but not in S_3 was used as an unknown word for evaluation.

There were 17,302 training words, and 21,375 evaluation words.

4.2 Determining the Feature Set

A set of data does not come with instructions for its analysis. Which aspects of the data should be modeled? Which features should be chosen? What should be chosen as the levels (possible values) for the features? For the loglinear model, which features appear independent, and which features interact?

There are no exact recipes for answering these questions. We explored the data to look for variables that are good discriminators, tried out different combinations of variables, and compared the performance of different models.

Based on this exploration, some features were added and deleted, and the number of levels was changed for others. These changes have the effect of changing the shape of the contingency table. By deleting features or some levels of features we merge parts of the table, or "coarsen" it. By adding features or introducing finer distinctions in the levels of a variable we introduce more cells, or "refine" the table.

The initial set of features is shown in Appendix A. This set of features, prefixes, and suffixes would have lead to an impossibly large contingency table. There is both a theoretical and a practical reason why the table size has to be kept small. First, if the table had many more cells than the number of training instances, the smoothing method would break down, and many cell counts would remain zero even after smoothing. Second, there are inherent limitations in the available software and hardware, and

Tag	Part of Speech	Example
CD	Cardinal (number)	500,000
FW	Foreign word	Fahrvergnügen
JJ	Adjective	yellow, large
JJR	Comparative Adjective	larger, nicer
JJS	Superlative adjective	largest, nicest
LS	List item marker	1. . . . , a) . . .
NN	Singular or mass noun	water, rock
NNS	Plural noun	rocks, cars
NP	Singular proper noun	English, March
NPS	Plural proper noun	The English
RB	Adverb	quickly, quite
RBR	Comparative Adverb	wiser, deeper
RBS	Superlative adverb	nearest, best
SYM	Symbol	%, *
UH	Interjection	uh, hmpf
VB	Base form verb	do, go
VBD	Past tense verb	did, went
VBG	Present participle verb	doing, going
VBN	Past participle verb	gone, flown
VBP	Non-3sg present verb	do, go
VBZ	3sg present verb	does, goes

Table 1: Open Class Wordtags Used In The Treebank

we found it difficult to handle tables with more than a few hundred thousand cells.

For these reasons it was necessary to reduce the dimensionality of the data. There are two principal methods for this: Reducing the number of features, and reducing the number of possible values for the features. The number of features can be reduced by merging multiple features into a single feature, and by discarding features that do not provide good discrimination for the response variable.

The number of values for the **Prefix** and **Suffix** features was reduced until only those affixes that occurred 100 times or more were used; this resulted in 26 prefixes and 37 suffixes. Even so, the affixes had to go into a separate contingency table, so an **Inflection** feature was added to the main table:

- **Inflection.** Does the word carry one of the following inflectional suffixes? *-ed, -er, -est, -ing, -ly, -s*

During data exploration the ability of different features to discriminate between different POSs was investigated, with the aim of excluding features entirely from the model. For each feature, we plotted the percentage of tags covered by that feature. For example, the **Number** feature picks out over 95% of the words tagged CD (number), and very few words with other tags. On the other hand, the feature **Includes-period** (one of the characters in the word is a period) only picked out 12% of the symbols (SYM) and less than 5% of the interjections (UH), so it was dropped from the model.

Answer Set	Accuracy	Res. Amb.	Set Size
Overall	73%	3.4	
2-best	87%	1.8	
3-best	93%	2.3	
4-best	96%	2.7	
5-best	98%	2.9	
0.7-factor	77%	1.1	1.1
0.4-factor	86%	1.6	1.8
0.1-factor	94%	2.3	2.9
0.07-factor	96%	2.6	3.7
0.04-factor	97%	2.8	4.3

Table 2: Performance of the Statistical Classifier with Nine Features

4.3 Measuring Performance

To measure the performance of the classifier, we can either look at the most likely POS tag, or we can consider answer sets that contain some of the most likely POS tags. The tags in such an answer set can be the n most likely tags returned by the model, or all tags within a certain "cutoff" factor of the most probable tag [de Marcken, 1990]. To describe the performance of the models with answer sets, we use a number of different measures:

1. **n-best Accuracy.** Accuracy of the answer set, which consists of the the n most likely POS tags.
2. **Cutoff Factor Accuracy.** Accuracy of the answer set, which consists of all POS tags whose probability lies within a cutoff factor F of the most likely POS.
3. **Residual Ambiguity.** Mean residual ambiguity for the POS tags in the answer set, measured by the perplexity of the answer set [Jelinek et al., 1977]. (The perplexity corresponds to the number of equiprobable members in the set.)
4. **Cutoff Factor Answer Set Size.** Mean number of tags in the answer sets derived using a cutoff factor.

4.4 Accuracy Results

[Weischedel et al., 1993] describe a model for unknown words that uses four features, which are shown in Appendix B. The features were treated as independent. The probabilities for these features were estimated directly from tagged training data. We reimplemented this model by using four features: **POS**, **Inflection**, **Capitalized**, and **Hyphenated**. In Figures 2–5, the results for this model are labeled **4 Prob**(abilistic).

For comparison, we also created a model with the same four features, but using a contingency table that was smoothed with a loglinear model. The results for this model are labeled **4 Class**(ifier).

The performance of the best model is summarized in Table 2. This model consists of two contingency tables; the features in these two tables are described in Appendix C. The results for this model are labeled **9 Class**(ifier).

The accuracy of the different models in assigning the most likely parts of speech to the word in isolation is summarized in Figure 1. The three sets of bars show three different accuracy measures: Percent correct (**Accuracy**), percent correct within the F=0.4 cutoff factor answer set (**F=0.4 Accuracy**), and percent correct within the two most likely answers (**2-Best Accuracy**).

In each case, the statistical classifier with four features shows better performance than the model assuming independence between the four features. The statistical classifier with nine features further improves this score, except for the case of 2-Best accuracy, where it ties with the classifier with four features.

4.5 Effect of Number of Features on Accuracy

The previous section showed that the performance of the statistical classifier can be improved by adding more features. Is this also possible with the approach assuming independent features?

In order to answer this question, the performance of the two types of models was measured with feature sets that ranged from a single feature to nine features. The performance for this series of models is shown in Figure 2.

This diagram shows two trends. First, the statistical classifier shows higher accuracy than the simple model. Second, the accuracy of the classifier increases as more features are added, but does not decrease as nuisance features are added. For example, the performance of the models with five, six, and seven features are the same; this suggests that these features do not contain any new information. These features do not have a negative effect on the classifier, however, and when the ninth feature is added, the classifier is able to take advantage of the information contributed by that feature to increase accuracy.

The simple probabilistic model, on the other hand, peaks a four features, and then degrades as features with little or no new information are added. When the ninth feature is added the simple model improves somewhat, but not enough to even recover to its accuracy with four features.

Similar trends can be observed in the graph of feature set size versus model cutoff factor accuracy for the cutoff factor F=0.4. This graph is shown in Figure 3.

4.6 Tradeoff between Accuracy and Ambiguity

Clearly, answer set classifications obtain higher accuracy than individual classifications at the expense of some residual ambiguity. The effect of feature set size on F=0.4 cutoff factor residual ambiguity is shown in Figure 4. Residual ambiguity decreases as more features are added, and the statistical classifier shows slightly higher residual ambiguity than the simple probabilistic model.

What is the relation between accuracy and residual ambiguity? Figure 5 shows a scatter plot of residual ambiguity versus accuracy. Each point in this diagram corresponds to one particular model. For example, the point labeled "F=0.4 9 Class." refers to the statistical classifier using nine features, where the answer set is derived using the F=0.4 cutoff factor. The connected series of points correspond to a single accuracy measure with a series of different models.

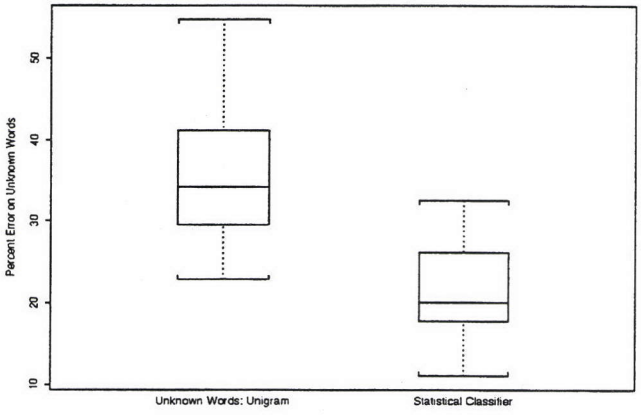

Figure 6: Error Rate on Unknown Words

The highest accuracy (87%) is obtained by the 2-best set of the four-feature statistical classifier, with a residual ambiguity of 1.6. The best accuracy-ambiguity tradeoff seems to be the F=0.4 set of the four-feature classifier.

5 Adding Context to the Model

So far, we have examined unknown words in isolation. But the surrounding context often provides important clues about the part of speech of a new word. To take this into account, we integrated our classifier with a stochastic part of speech tagger.

5.1 Part of Speech Tagging

The part of speech tagger assigns part of speech tags to words in a sentence. It uses two types of parameters:

- **Lexical Probabilities:** $P(t|w)$ — the probability of observing tag t given that we have observed word w.
- **Contextual Probabilities:** $P(t_i|t_{i-1}, t_{i-2})$ — the probability of observing tag t_i given that we have observed the 2 previous tags.

The tagger maximizes the probability of the tag sequence given the word sequence, which is approximated as follows:

$$P(T|W) \equiv \prod_{i=1}^{n} P(w_i|t_i)P(t_i| < t_{i-1}, t_{i-2} >)$$

5.2 Evaluating Combined Accuracy

We evaluated the accuracy of the combined local and global method by comparing three methods of handling unknown words:

- **Unigram**: Using the prior probability distribution $P(i)$ of the part of speech tags for rare words.
- **Probabilistic UWM**: Using the probabilistic model that assumes independence between the features.

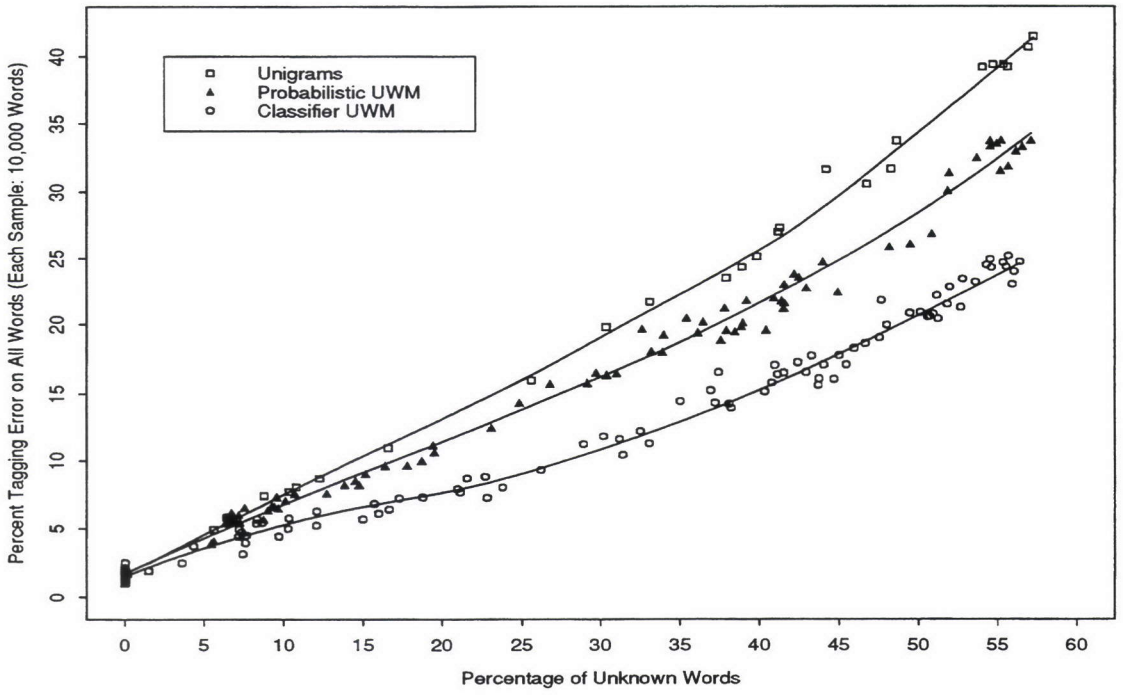

Figure 7: Percentage of Unknown Words versus Accuracy for Different Unknown Word Models

- **Classifier UWM**: Using the statistical Classifier.

The unknown word model was trained on text from the Brown Corpus. We evaluated different configurations of the system on 30–40 different samples of 4,000 words of text from the Wall Street Journal. The tagger displays considerable variance in its accuracy in assigning part of speech to unknown words in context. Figure 6 compares the tagging error rate on unknown words for the unigram method (left) and the classifier with nine features (right). The classifier lowers the error rate considerable, and eliminates all samples with error rates over 32%.

5.3 Effect of Proportion of Unknown Words

Another question related to unknown words is this: How does the overall tagging accuracy depend on unknown words? How does it vary when there are different amounts of unknown words in the text to be tagged?

To answer this question, we tagged samples of text that contained different proportions of unknown words. We found that the overall tagging error rate increases significantly as the proportion of new words increases. Figure 7 shows a graph of overall tagging accuracy versus percentage of unknown words in the text. The graph compares the three different methods of handling unknown words.

This diagram shows that the statistical classifier leads to better overall tagging performance than the simpler methods, with a clear separation of all samples whose proportion of new words is above approximately 9%.

6 Conclusions

We have demonstrated a simple statistical classification technique to help natural language analysis systems handle words that have never been encountered before.

Our results show that the statistical classification method is better than a probabilistic method that assumes independence between the features. First, the statistical classifier achieves higher accuracy. Second, the classifier handles larger feature sets, which may contain nuisance features, while the performance of simpler feature combination method degrades as more features are added.

In the future we are going to apply this method to other problems in robust natural language analysis. We believe that the framework of categorial data analysis and statistical classification holds the key to solving the pervasive problem of ambiguity.

7 Acknowledgments

I would like to thank Jaime Carbonell, Teddy Seidenfeld, Michael Mauldin, Ted Gibson, and Akira Ushioda for helpful discussions and comments on this work.

References

[Agresti, 1990] Agresti, A. (1990). *Categorical Data Analysis*. John Wiley & Sons, New York.

[Bishop et al., 1975] Bishop, Y. M., Fienberg, S. E., and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice*. MIT Press, Cambridge, MA.

[de Marcken, 1990] de Marcken, C. G. (1990). Parsing the LOB corpus. In *ACL-90*, pages 243–251.

[Deming and Stephan, 1940] Deming, W. E. and Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Ann. Math. Statis*, (11):427–444.

[Duda and Hart, 1973] Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

[Fienberg, 1977] Fienberg, S. E. (1977). *The Analysis of Cross-classified Categorial Data*. MIT Press, Cambridge, MA, second edition edition.

[Jelinek et al., 1977] Jelinek, F., Mercer, R. L., Bahl, L. R., and J, K. B. (1977). Perplexity — a measure of difficulty of speech recognition tasks. In *94th Meeting of the Acoustical Society of America*, Miami Beach, FL.

[Marcus et al., 1993] Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

[Weischedel et al., 1993] Weischedel, R., Meteer, M., Schwartz, R., Ramshaw, L., and Palmucci, J. (1993). Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19(2):359–382.

Appendices

A Initial Feature Set

- **Includes-Number.** Does the word include a number? Positive example: *836-901*. Negative example: *Absent-minded.*
- **Capitalized.** Is the first character of the word a capitalized letter? Positive example: *Abnormal.* Negative example: *catch, 500,000.*
- **Includes-Period.** Does the word include a period? Positive example: *B.C., 4.2, U.N.* Negative example: *Union.*
- **Includes-Comma.** Does the word include a comma? Positive example: *500,000.* Negative example: *amazement.*
- **Final-Period.** Is the last character of the word a period? Positive example: *B.C., Co.* Negative example: *U.N, Command.*
- **Includes-Hyphen.** Does the word include a hyphen? Positive example: *Poynting-Robertson, anti-party.* Negative example: *answer.*
- **Sentence-initial.** Is the word the first word in the sentence? The value of this feature is determined by looking at the context of the word in the original Treebank file.
- **All-upper-case.** Is the word in all upper case? "All upper case" is defined as the absence of any lower case letters. Thus, words without any letters at all are also in "all upper case". Positive example: *CTCA, 1532.* Negative example: *Fred, accomplish.*
- **Short.** Is the length of the word three characters or less? Positive example: *W., Yes, bar, Eta.* Negative example: *Heaven, 100,000.*
- **Prefix.** Does the word carry one of a list of known prefixes?
- **Suffix.** Does the word carry one of a list of known suffixes?

B Meteer et al. 's Four Features

- **Inflectional ending.** Possible values: "ed", "ing", "s".
- **Derivational ending.** 32 possible values, including "ion", "al", "ive", "ly".
- **Capitalization.** Four possible values: "+sentenceinitial+capital", "-sentenceinitial+capitalized", etc.
- **Hyphenation.** "true"/"false".

C Set of Nine Features

The first table contains the following seven features:

- **POS.** "CD", "FW", "JJ", "JJR", "JJS", "LS", "NN", "NNS", "NP", "NPS", "RB", "RBR", "RBS", "SYM", "UH", "VB", "VBD", "VBG", "VBN", "VBP", "VBZ".
- **All-upper-case.** "true"/"false".
- **Hyphenated.** "true"/"false".
- **Includes-number.** "true"/"false".
- **Capitalized.** Three values: "capitalized in sentence initial position", "capitalized in the middle of the sentence", and "lower case".
- **Inflection.** "ed", "er", "est", "ing", "ly", "s", "no-inflection".
- **Short.** "true"/"false".

The second table handles the affixes with the following features:

- **POS.**
- **Prefix.** The 26 prefixes that occurred 100 or more times in the training data, plus a "no-prefix" value:
- **Suffix.** The 37 suffixes that occurred 100 or more times in the training data, plus a "no-suffix" value: