Predicting Part-of-Speech Information about Unknown Words using Statistical Methods

Scott M. Thede
Purdue University
West Lafayette, IN 47907

Abstract

This paper examines the feasibility of using statistical methods to train a part-of-speech predictor for unknown words. By using statistical methods, without incorporating hand-crafted linguistic information, the predictor could be used with any language for which there is a large tagged training corpus. Encouraging results have been obtained by testing the predictor on unknown words from the Brown corpus. The relative value of information sources such as affixes and context is discussed. This part-of-speech predictor will be used in a part-of-speech tagger to handle out-of-lexicon words.

1 Introduction

Part-of-speech tagging involves selecting the most likely sequence of syntactic categories for the words in a sentence. These syntactic categories, or *tags*, generally consist of parts of speech, often with feature information included. An example set of tags can be found in the Penn Treebank project (Marcus et al., 1993). Part-of-speech tagging is useful for speeding up parsing systems, and allowing the use of partial parsing.

Many current systems make use of a Hidden Markov Model (HMM) for part-of-speech tagging. Other methods include rule-based systems (Brill, 1995), maximum entropy models (Ratnaparkhi, 1996), and memory-based models (Daelemans et al., 1996). In an HMM tagger the Markov assumption is made so that the current word depends only on the current tag, and the current tag depends only on adjacent tags. Charniak (Charniak et al., 1993) gives a thorough explanation of the equations for an HMM model, and Kupiec (Kupiec, 1992) describes an HMM tagging system in detail.

One important area of research in part-of-speech tagging is how to handle unknown words. If a word is not in the lexicon, then the lexical probabilities must be provided from some other source. One common approach is to use affixation rules to "learn" the probabilities for words based on their suffixes or prefixes. Weischedel's group (Weischedel et al., 1993) examines unknown words in the context of part-of-speech tagging. Their method creates a probability distribution for an unknown word based on certain features: word endings, hyphenation, and capitalization. The features to be used are chosen by hand for the system. Mikheev (Mikheev, 1996; Mikheev, 1997) uses a general purpose lexicon to learn affix and word ending information to be used in tagging unknown words. His work returns a set of possible tags for unknown words, with no probabilities attached, relying on the tagger to disambiguate them.

This work investigates the possibility of automatically creating a probability distribution over all tags for an unknown word, instead of a simple set of tags. This can be done by creating a probabilistic lexicon from a large tagged corpus (in this case, the Brown corpus), and using that data to estimate distributions for words with a given "prefix" or "suffix". Prefix and suffix indicate substrings that come at the beginning and end of a word respectively, and are not necessarily morphologically meaningful.

This predictor will offer a probability distribution of possible tags for an unknown word, based solely on statistical data available in the training corpus. Mikheev's and Weischedel's systems, along with many others, uses language specific information by using a hand-generated set of English affixes. This paper investigates what information sources can be automatically constructed, and which are most useful in predicting tags for unknown words.

2 Creating the Predictor

To build the unknown word predictor, a lexicon was created from the Brown corpus. The entry for a word consists of a list of all tags assigned to that word, and the number of times that tag was assigned to that word in the entire training corpus. For example, the lexicon entry for the

word *advanced* is the following:

advanced ((VBN 31) (JJ 12) (VBD 8))

This means that the word *advanced* appeared a total of 51 times in the corpus: 31 as a past participle (VBN), 12 as an adjective (JJ), and 8 as a past tense verb (VBD). We can then use this lexicon to estimate $P(w_i|t_i)$.

This lexicon is used as a preliminary source to construct the unknown word predictor. This predictor is constructed based on the assumption that new words in a language are created using a well-defined morphological process. We wish to use suffixes and prefixes to predict possible tags for unknown words. For example, a word ending in *-ed* is likely to be a past tense verb or a past participle. This rough stemming is a preliminary technique, but it avoids the need for hand-crafted morphological information. To create a distribution for each given affix, the tags for all words with that affix are totaled. Affixes up to four characters long, or up to two characters less than the length of the word, whichever is smaller, are considered. Only open-class tags are considered when constructing the distributions. Processing all the words in the lexicon creates a probability distribution for all affixes that appear in the corpus.

One problem is that data is available for both prefixes and suffixes—how should both sets of data be used? First, the longest applicable suffix and prefix are chosen for the word. Then, as a baseline system, a simple heuristic method of selecting the distribution with the fewest possible tags was used. Thus, if the prefix has a distribution over three possible tags, and the suffix has a distribution over five possible tags, the distribution from the prefix is used.

3 Refining the Predictions

There are several techniques that can be used to refine the distributions of possible tags for unknown words. Some of these that are used in our system are listed here.

3.1 Entropy Calculations

A method was developed that uses the *entropy* of the prefix and suffix distributions to determine which is more useful. Entropy, used in some part-of-speech tagging systems (Ratnaparkhi, 1996), is a measure of how much information is necessary to separate data. The entropy of a tag distribution is determined by the following equation:

$$\text{Entropy of } i\text{-th affix} = -\sum_j \frac{n_{ij}}{N_i} \log_2\left(\frac{n_{ij}}{N_i}\right)$$

where

n_{ij} = j-th tag occurrences in i-th affix words
N_i = total occurrences of the i-th affix

The distribution with the smallest entropy is used, as this is the distribution that offers the most information.

3.2 Open-Class Smoothing

In the baseline method, the distributions produced by the predictor are smoothed with the overall distribution of tags. In other words, if $p(x)$ is the distribution for the affix, and $q(x)$ is the overall distribution, we form a new distribution $p'(x) = \lambda p(x) + (1 - \lambda)q(x)$. We use $\lambda = 0.9$ for these experiments. We hypothesize that smoothing using the open-class tag distribution, instead of the overall distribution, will offer better results.

3.3 Contextual Information

Contextual probabilities offer another source of information about the possible tags for an unknown word. The probabilities $P(t_i|t_{i-1})$ are trained from the 90% set of training data, and combined with the unknown word's distribution. This use of context will normally be done in the tagger proper, but is included here for illustrative purposes.

3.4 Using Suffixes Only

Prefixes seem to offer less information than suffixes. To determine if calculating distributions based on prefixes is helpful, a predictor that only uses suffix information is also tested.

4 The Experiment

The experiments were performed using the Brown corpus. A 10-fold cross-validation technique was used to generate the data. The sentences from the corpus were split into ten files, nine of which were used to train the predictor, and one which was the test set. The lexicon for the test run is created using the data from the training set. All unknown words in the test set (those that did not occur in the training set) were assigned a tag distribution by the predictor. Then the results are checked to see if the correct tag is in the n-best tags. The results from all ten test files were combined to rate the overall performance for the experiment.

5 Results

The results from the initial experiments are shown in Table 1. Some trends can be seen in this data. For example, choosing whether

Method	Open?	Con?	1-best	2-best	3-best
Baseline	no	no	57.6%	73.2%	79.5%
Baseline	no	yes	61.5%	75.0%	81.7%
Baseline	yes	no	57.6%	73.6%	83.2%
Baseline	yes	yes	61.3%	78.2%	87.0%
Entropy	no	no	62.2%	77.6%	83.4%
Entropy	no	yes	65.7%	78.9%	85.1%
Entropy	yes	no	62.2%	78.1%	86.9%
Entropy	yes	yes	65.4%	81.8%	89.6%
Endings	no	no	67.1%	83.5%	91.4%
Endings	no	yes	70.9%	86.5%	92.6%
Endings	yes	no	67.1%	83.6%	92.2%
Endings	yes	yes	70.9%	87.6%	93.8%

Open? - system uses open-class smoothing
Con? - system uses context information

Table 1: Results using Various Methods

to use the prefix distribution or suffix distribution using entropy calculations clearly improves the performance over using the baseline method (about 4-5% overall), and using only suffix distributions improves it another 4-5%. The use of context improves the likelihood that the correct tag is in the n-best predicted for small values of n (improves nearly 4% for 1-best), but it is less important for larger values of n. On the other hand, smoothing the distributions with open-class tag distributions offers no improvement for the 1-best results, but improves the n-best performance for larger values of n.

Overall, the best performing system was the system using both context and open-class smoothing, relying on only the suffix information. To offer a more valid comparison between this work and Mikheev's latest work (Mikheev, 1997), the accuracies were tested again, ignoring mistags between NN and NNP (common and proper nouns) as Mikheev did. This improved results to 77.5% for 1-best, 89.9% for 2-best, and 94.9% for 3-best. Mikheev obtains 87.5% accuracy when using a full HMM tagging system with his cascading tagger. It should be noted that our system is not using a full tagger, and presumably a full tagger would correctly disambiguate many of the words where the correct tag was not the 1-best choice. Also, Mikheev's work suffers from reduced coverage, while our predictor offers a prediction for every unknown word encountered.

6 Conclusions and Further Work

The experiments documented in this paper suggest that a tagger can be trained to handle unknown words effectively. By using the probabilistic lexicon, we can predict tags for unknown words based on probabilities estimated from training data, not hand-crafted rules. The modular approach to unknown word prediction allows us to determine what sorts of information are most important.

Further work will attempt to improve the accuracy of the predictor, using new knowledge sources. We will explore the use of the concept of a confidence measure, as well as using only infrequently occurring words from the lexicon to train the predictor, which would presumably offer a better approximation of the distribution of an unknown word. We also plan to integrate the predictor into a full HMM tagging system, where it can be tested in real-world applications, using the hidden Markov model to disambiguate problem words.

References

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.

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

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 14–27.

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

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

Andrei Mikheev. 1996. Unsupervised learning of word-category guessing rules. *Proceedings of the 34th Annual Meeting of the Association for Compuatational Linguistics*, pages 327–334.

Andrei Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing.*

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