Part-of-Speech Tagging from "Small" Data Sets

Eric Neufeld
Greg Adams, Henry Choy, Ron Orthner, Tim Philip, Ahmed Tawfik
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada, S7N 0W0
eric@spr.usask.ca

Abstract

Many probabilistic approaches to part-of-speech (POS) tagging compile statistics from massive corpora such as the LOB. Using the hidden Markov model method on a 900,000 token training corpus, it is not difficult achieve a success rate of 95 per cent on a 100,000 token test corpus.

However, even such large training corpora contain few relatively few words. For example, the LOB contains about 45,000 words, most of which occur only once or twice. As a result, 3–4 per cent of tokens in the test corpus are unseen and cause a significant proportion of errors. A corpus large enough to accurately represent all possible tag sequences seems implausible enough, let alone a corpus that also represents, even in small numbers, enough of English to make the problem of unseen words insignificant.

This work argues this may not be necessary, describing variations on HMM-based tagging that facilitate learning from relatively little data, including ending-based approaches, incremental learning strategies, and the use of approximate distributions.

1 Introduction

Although probabilistic approaches to linguistic problems were attempted earlier in the century[Zipf,1932], they were hampered by real difficulties of collecting and managing statistics, not to mention challenges to probabilistic methods in principle. New technology and the availability of tagged electronic corpora such as the million-word Lancaster-Oslo-Bergen (LOB) Corpus [Johansson,1980, Johansson *et al.*,1986] changed this situation dramatically, and a variety of probabilistic approaches to a variety of natural language processing problems have become popular for some years.

One success of the probabilistic approach has been using hidden Markov models (HMMs) to attach POS tags to unrestricted text. Given an actual stream of text, a sequence of tokens (instances of words) $w_1 \ldots w_n$, the HMM method computes the word-tag sequence (or simply tag sequence) $t_1 \ldots t_n$ that most probably generated the sequence, that is, that maximizes

$$P(t_1 \ldots t_n | w_1 \ldots w_n). \tag{1}$$

Such probabilities would be extremely difficult to collect in practice; furthermore their number is exponential in n. However, assuming that 1) the probability of a tag t_i directly depends only on the tag immediately preceding it, and that 2) the probability of any word w_i depends only upon the tag t_i that produced it, the preceding reduces to

$$\prod_{i=1}^{n+1} P(w_i|t_i)P(t_i|t_{i-1}), \tag{2}$$

where t_0, t_{n-1} are dummy word-tags marking the beginning and the end of a sequence. $P(w_i|t_i)$ is a *lexical probability*; $P(t_i|t_{i-1})$ is a *contextual probability*. Equation 2 defines a *bi-tagger*; in a $k+1$-tagger, the probability of a tag depends on the k preceding tags, but this only improves performance marginally[Foster,1991]. See [Charniak *et al.*,1993] for a good overview of this approach to part-of-speech tagging.

Typically, a POS tagger trains by collecting lexical and contextual probabilities from a large subset of a tagged electronic corpus such as the LOB corpus [Johansson,1980] and is tested on a smaller disjoint subset. In earlier work [Adams & Neufeld, 1993], a training corpus of 900,000 tokens and a test corpus of 100,000 tokens was used.

Clearly a difficulty arises when attempting to attach part-of-speech tags to *unseen* tag sequences and words [Adams & Neufeld, 1993, Church,1989, Foster,1991, Merialdo,1990, Meteer *et al.*,1991, Kupiec,1992], that is, tag sequences or tokens not occurring in the training corpus, since no lexical probabilities are known for them. About half of the words in the LOB only appear once, so many words (about 3–4 per cent [Adams & Neufeld, 1993] are encountered for the first time in the test corpus. Most of these (for example, *outjumped* and *galaxy*) are neither exotic nor highly specialized but simply reflect the vastness of human experience. A tagged training corpus of astronomical size would be required to solve this problem.

Managing this problem seems to create the greatest overhead when constructing such taggers. A variety of solutions to the problem of unseen tag sequences have been studied by other authors; our work has addressed the problem of unseen words. The ideal solution would be to collect probabilities from a tagged corpus sufficiently large to represent all tag sequences and all word/tag pairs, but the value of manually tagging such a corpus must be addressed. In previous work, we estimated probabilities for unseen word/tag combinations using word endings. In subsequent work, it occurred to us that the performance cost of using ending-based probabilities for relatively rare words might well be marginal. This would be valuable when it is difficult to store all the parameters.

The results below show this may not be necessary. A variety of techniques show that performance can be maintained in the presence of infrequent and unseen words, without necessarily increasing the size of the training corpus, and possibly even reducing it.

2 Ending-based strategies

In [Meteer *et al.*,1991], it is reported that the success rate for tagging unseen words is significantly improved by compiling word/tag statistics for about 35 preselected word-endings, such as *-ology* and *-tion*. Of course, selecting such a set of word-endings requires expert language knowledge. In [Adams & Neufeld, 1993], it is asked whether one could compile statistics on all word endings of some fixed length. This language-independent approach was attempted for all L-letter endings, $L = 1, \ldots, 4$. It is possible to look at the problem in another way. That is, initially collect statistics on all L-letter endings, but then collect statistics for the n most frequently occuring words in the corpus, and use the whole-word statistic whenever possible in the tagging process. The first approach gives a measure of value of the ending statistics; the second approach gives a sense of the value of whole-word information, and perhaps more clearly expresses the tradeoff between additional knowledge and improved performance as rare words are added to the model.

	Number of Most Frequent Words Put Back In LOB									
	0	5000	10000	15000	20000	25000	30000	35000	40000	45000
2-letter endings										
AC - Full ETL	83.4	95.0	95.6	95.9	96.1	96.2	96.3	96.3	96.3	96.4
GT - Full ETL	83.4	95.0	95.6	95.9	96.1	96.2	96.3	96.3	96.3	96.4
AC - Unit ETL	88.6	96.0	96.2	96.3	96.3	96.4	96.4	96.4	96.4	96.4
GT - Unit ETL	88.7	96.0	96.3	96.3	96.4	96.4	96.4	96.4	96.4	96.4
3-letter endings										
AC - Full ETL	90.8	95.5	95.9	96.1	96.2	96.3	96.3	96.4	96.4	96.4
GT - Full ETL	90.8	95.5	95.9	96.1	96.2	96.3	96.3	96.4	96.4	96.4
AC - Unit ETL	93.4	96.2	96.3	96.4	96.4	96.4	96.4	96.4	96.4	96.4
GT - Unit ETL	93.4	96.2	96.3	96.4	96.4	96.4	96.4	96.4	96.4	96.4
4-letter endings										
AC - Full ETL	94.3	95.7	95.9	96.0	96.1	96.1	96.1	96.2	96.2	96.2
GT - Full ETL	94.4	95.7	95.9	96.0	96.1	96.1	96.2	96.2	96.2	96.2
AC - Unit ETL	95.2	96.0	96.1	96.2	96.2	96.2	96.2	96.2	96.2	96.2
GT - Unit ETL	95.3	96.1	96.1	96.2	96.2	96.2	96.2	96.2	96.2	96.2

Figure 1: Tokens Correctly Tagged

In the above table, **AC** means *augmented corpus* and refers to the technique of adding one to all tag-sequence counts and all seen word counts before computing probabilities. **GT** means Good-Turing method and is a more sophisticated way of adjusting unseen tag sequences and unseen word/tag pairs. *Full ETL* means considering for each token the full set of tag possibilities for the ending; *Unit ETL* means attaching only whole-word possibilities when there only is a single possibility.

The table illustrates that the marginal value of an increased lexicon is low. In fact, about half of the vocabulary occurs only once, so that last $20,000$ words added back are based on small samples.

Another variation was to consider mixing sets of ending statistics. There is a tradeoff between *specificity* (ending length) and *accuracy* (sample size) generally. The tradeoff can be resolved [Kyburg,1983] by using the narrowest reference class for which there are adequate statistics. Roughly, this means using the narrower statistic unless the difference between it and a more general statistic can be explained by chance error. Statistics were collected for whole words and word-ending lengths from unity to four. (A tagger using just one letter endings can achieve a success rate of 70 per cent.) A $1 - \alpha$ confidence interval was computed for each statistic (although point values were used in the actual tagging process) and the narrowest reference term for which we had adequate statistics was used during testing. The intuition here is to find points at which specificity is more valuable than accuracy, as defined by sample size. Several values of α were tried; the best success rates were obtained by always using the narrowest statistic available.

3 Studying Tag Distance

The tagged LOB corpus has 151 different tags; other taggers have between 18 and 179 different tags, and choosing the number of tags involves a variety of linguistic and computational factors. Many of the tags in the LOB corpus are similar. For example, different tags are assigned to each form of the verb *to be*, each form of which occurs in similar places. The distance between two tags t_i, t_j may be defined in terms of the context c_k about them:

$$d(t_i, t_j) = \sum_k (p(t_i|c_k) - p(t_j|c_k))^2.$$

There are many applications of such a distance measure, including evaluating the performance of existing taggers or providing evidence to guide the formation of tag hierarchies. This measure was computed for all tags appearing in a 30,000 word corpus and seemed to be reasonable. The exclamation mark was closest to a question mark, the colon closest to a semi-colon, singular article closest to a plural article, plural noun closest to a singular noun and so on. However a few dozen tags were closest to the adverb tag, perhaps due to the relatively little context required by general adverbs. Increasing the training corpus to 900,000 words eliminated this to some extent.

4 Incremental Learning

The LOB corpus attaches one of 151 different possible tags to each of about one million tokens. Orthner [Orthner, 1994] hypothesizes the the fine grain size of the tag set impairs training. He therefore constructs a coarser set of 23 tags which is used to generate a simplifed corpus. Using backpropagation as well as neural nets, a tagger is first trained on the coarse tags, then on the large set. Orthner finds the tagger learns best when about equal numbers of training epochs with both tag sets are used, resulting in the so-called U-shaped learning curve. This is being ported to the HMM approach as follows. A battery of known techniques will be used to achieve the highest possible success rate on the coarse-grained tag set. (There are two reasons to believe this. First, in a related set of experiments, pattern recognition techniques were used to construct equivalence classes of "similar" tags based on occurences in similar contexts. Second, similar patterns have also been observed in the error output.) Then, tokens will be suffixed with their coarse-grained tag, and the process will recommence with the 151 tags, effectively giving the HMM a hierarchical strategy. Orthner used a representative training set of 34,721 tokens, a neural net with a hidden layer composed of 70 neurons. Each test involved 100 epochs, or presentations, of the corpus to the tagger for training, and a single epoch of tagging. The 100 epochs were then divided between the simplied and the unsimplified corpus. Preliminary results seem to suggest that the best strategy is to spend a considerable number of epochs training on the simplified corpus before training on the complex corpus. It appears to impair the tagger's efficiency to spend a short time on the simplified corpus. The next phase is to use the technique on the HMM method, because the neural net models have a narrow window of context.

5 Sensitivity analysis

Is the success rate sensitive to the actual numerical values or is it sensitive to their rank orderings? If the latter, it may be possible to construct qualitative tagging algorithms that avoid the calculations.

To test this, lexical probabilities were rounded off to the nearest value within small fixed subset of probabilities. The following quantization sets were used:

$R_1 = \{0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1\}$
$R_2 = \{0.0001, 0.01, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1\}$
$R_7 = \{0.00000001, 0.000001, 0.001, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$
$R_{15} = \{0.00001, 0.001, 0.5, 1\}$

The results appear below.

	Number of Most Frequent Words Put Back In LOB						
	0	3000	5000	10000	20000	30000	45000
R_1	84.0	87.4	88.8	90.1	91.1	91.5	91.8
R_2	90.8	92.9	93.6	94.4	94.9	95.1	95.3
R_7	89.7	92.1	93.0	93.9	94.6	94.8	95.0
R_{15}	88.3	91.1	92.1	93.1	93.8	94.1	94.3

Figure 2: Tokens Correctly Tagged

This lets us measure the value of exact probabilistic information. On one hand, in the best cases, it only gives up 1.2 per cent success rate. (On the other hand, it is very difficult to get this much accuracy using other techniques.) The quantization sets were contructed with several issues in mind. One is expressiveness, the number of values in the set. As expressiveness increases, so does accuracy. The other issue is the distribution of values near zero. An earlier set of experiments suggested that success rate improves with the number of values close to zero.

The table above gives results of a final carefully monitored set of experiments. It is surprising how well the tagger performs given such a small set of numbers. It is interesting that adding some very small values to R_1 (giving R_2) gives a dramatic improvement; in fact, R_2 consistently gives the best performance of the four quantization sets. It makes sense that performance will increase with expressiveness, but the role of the distribution is important. It is remarkable that R_{15}, with only four values results in as little as 1 per cent worse performance.

This suggests that the accuracy of the statistics is not as important as the rank ordering, and opens important possibilities. For example, in [Adams & Neufeld, 1993], unseen words were tagged by first consulting an external dictionary that gave legal tag types but contained no statistics. Some performance improvement could be obtained by attaching numbers from one of these fixed subsets to the list of legal tag types thus obtained.

6 Conclusions

The hidden Markov model works well up to a certain point (say, 95 per cent) but it seems difficult to push performance beyond that point. The marginal increase in performance given extra training corpus or vocabulary is very small. The techniques here suggest that it may be possible to achieve success rates of close to 95 per cent without having to train on massive corpora or carry about a massive lexicon. Not only do ending-based approaches work well, they learn quickly. To give an example, it was shown that using ending-based strategies on a very small (100,000 token) training corpus resulted in a decline in the success rate of only 3 per cent.

This work shows that there are many other techniques to exploit. For example, maximizing specificity helps if many kinds of ending statistics are available and furthermore, the range of numerical values is relatively unimportant. The work on incremental learning and tag similarities hints at a hierarchical approach that may improve overall performance.

Acknowledgements

This research was supported by the Natural Science and Engineering Research Council of Canada, the Institute for Robotics and Intelligent Systems, the University of Saskatchewan, and the City of Saskatoon. George Foster made software available.

References

[Adams & Neufeld, 1993] Adams, G. and Neufeld, E. (1993) Automated word-class tagging of unseen words in text. In *Proceedings of the Sixth International Symposium on Artificial Intelligence*, pages 390–397.

[Charniak *et al.*,1993] Charniak, E., Henrickson, C., Jacobson, N., and Perkowitz, M. (1993) Equations for part-of-speech tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 784–789.

[Church,1989] Church, K. W. (1989) A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Glasgow, U.K.

[Foster,1991] Foster, G. F. (1991) Statistical lexical disambiguation. Master's thesis, McGill University, Montreal.

[Johansson,1980] Johansson, S. (1980) The LOB Corpus of British English texts: Presentation and comments. *ALLC Journal*, 1(1):25–36.

[Johansson *et al.*,1986] Johansson, S., Atwell, E., Garside, R., and Leech, G. (1986) *The Tagged LOB Corpus: Users' Manual.* Norwegian Computing Centre for the Humanities, Bergen, Norway.

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

[Kyburg,1983] . Kyburg, Jr., Henry E. 1983. The reference class. Philosophy of Science, **50**:374–397.

[Merialdo,1990] Merialdo, B. (1990) Tagging text with a probabilistic model. In *Proceedings of the IBM Natural Language ITL*, pages 161–172, Paris.

[Meteer *et al.*,1991] Meteer, M., Schwartz, R., and Weischedel, R. (1991) POST: Using probabilities in language processing. In *IJCAI 91: Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 960–965, Sydney, Australia.

[Orthner, 1994] Orthner, Ron. (1994). Part of Speech Tagging: A Cognitive Simulation Approach. Cognitive Science Workshop, Baden-Baden, to appear.

[Zipf,1932] Zipf, G. K. (1932) *Selected Studies of the Principle of Relative Frequency in Language.* Harvard University Press, Cambridge, Massachusetts.