Tagset Reduction Without Information Loss

Thorsten Brants
Universität des Saarlandes
Computerlinguistik
D-66041 Saarbrücken, Germany
thorsten@coli.uni-sb.de

Abstract

A technique for reducing a tagset used for n-gram part-of-speech disambiguation is introduced and evaluated in an experiment. The technique ensures that all information that is provided by the original tagset can be restored from the reduced one. This is crucial, since we are interested in the linguistically motivated tags for part-of-speech disambiguation. The reduced tagset needs fewer parameters for its statistical model and allows more accurate parameter estimation. Additionally, there is a slight but not significant improvement of tagging accuracy.

1 Motivation

Statistical part-of-speech disambiguation can be efficiently done with n-gram models (Church, 1988; Cutting et al., 1992). These models are equivalent to Hidden Markov Models (*HMMs*) (Rabiner, 1989) of order $n-1$. The states represent parts of speech (*categories, tags*), there is exactly one state for each category, and each state outputs words of a particular category. The transition and output probabilities of the HMM are derived from smoothed frequency counts in a text corpus.

Generally, the categories for part-of-speech tagging are linguistically motivated and do not reflect the probability distributions or co-occurrence probabilities of words belonging to that category. It is an implicit assumption for statistical part-of-speech tagging that words belonging to the same category have similar probability distributions. But this assumption does not hold in many of the cases.

Take for example the word *cliff* which could be a proper (NP)[1] or a common noun (NN) (ignoring capitalization of proper nouns for the moment). The two previous words are a determiner (AT) and an adjective (JJ). The probability of *cliff* being a common noun is the product of the respective contextual and lexical probabilities $p(\mathrm{NN}|\mathrm{AT},\mathrm{JJ}) \cdot p(\mathit{cliff}|\mathrm{NN})$, regardless of other information provided by the actual words (*a sheer cliff* vs. *the wise Cliff*). Obviously, information useful for probability estimation is not encoded in the tagset.

On the other hand, in some cases information *not* needed for probability estimation is encoded in the tagset. The distributions for comparative and superlative forms of adjectives in the Susanne Corpus (Sampson, 1995) are very similar. The number of correct tag assignments is *not* affected when we combine the two categories. However, it does not suffice to assign the combined tag, if we are interested in the distinction between comparative and superlative form for further processing. We have to ensure that the original (interesting) tag can be restored.

There are two contradicting requirements. On the one hand, more tags mean that there is more information about a word at hand, on the other hand, the more tags, the severer the sparse-data problem is and the larger the corpora that are needed for training.

This paper presents a way to modify a given tagset, such that categories with similar distributions in a corpus are combined without losing information provided by the original tagset and without losing accuracy.

2 Clustering of Tags

The aim of the presented method is to reduce a tagset as much as possible by combining (*clustering*) two or more tags without losing information and without losing accuracy. The fewer tags we have, the less parameters have to be estimated and stored, and the less severe is the sparse data problem. Incoming text will be disambiguated with the new reduced tagset, but we ensure that the original tag is still uniquely identified by the new tag.

The basic idea is to exploit the fact that some of the categories have a very similar frequency distribution in a corpus. If we combine categories with

[1] All tag names used in this paper are inspired by those used for the LOB Corpus (Garside et al., 1987).

similar distribution characteristics, there should be only a small change in the tagging result. The main change is that single tags are replaced by a cluster of tags, from which the original has to be identified. First experiments with tag clustering showed that, even for fully automatic identification of the original tag, tagging accuracy slightly increased when the reduced tagset was used. This might be a result of having more occurrences per tag for a smaller tagset, and probability estimates are preciser.

2.1 Unique Identification of Original Tags

A crucial property of the reduced tagset is that the original tag information can be restored from the new tag, since this is the information we are interested in. The property can be ensured if we place a constraint on the clustering of tags.

Let $\mathcal{W}$ be the set of words, $\mathcal{C}$ the set of clusters (i.e. the reduced tagset), and $\mathcal{T}$ the original tagset. To restore the original tag from a combined tag (*cluster*), we need a unique function

$$f_{orig} : \mathcal{W} \times \mathcal{C} \mapsto \mathcal{T}, \qquad (1)$$

To ensure that there is such a unique function, we prohibit some of the possible combinations. A cluster is allowed if and only if there is no word in the lexicon which can have two or more of the original tags combined in one cluster. Formally, seeing tags as sets of words and clusters as sets of tags:

$$\forall c \in \mathcal{C}, t_1, t_2 \in c, t_1 \neq t_2, w \in \mathcal{W} : \quad w \in t_1 \Rightarrow w \notin t_2 \qquad (2)$$

If this condition holds, then for all words w tagged with a cluster c, exactly one tag t_{wc} fulfills

$$w \in t_{wc} \wedge t_{wc} \in c,$$

yielding

$$f_{orig}(w, c) = t_{wc}.$$

So, the original tag can be restored any time and no information from the original tagset is lost.

Example: Assume that no word in the lexicon can be both comparative (JJR) and superlative adjective (JJT). The categories are combined to {JJR,JJT}. When processing a text, the word *easier* is tagged as {JJR,JJT}. Since the lexicon states that *easier* can be of category JJR but not of category JJT, the original tag must be JJR.

2.2 Criteria For Combining Tags

The are several criteria that can determine the quality of a particular clustering.

1. Compare the trigram probabilities $p(B|X_i, A)$, $p(B|A, X_i)$, and $p(X_i|A, B)$, $i = 1, 2$. Combine two tags X_1 and X_2, if these probabilities coincide to a certain extent.
2. Maximize the probability that the training corpus is generated by the HMM which is described by the trigram probabilities.
3. Maximize the tagging accuracy for a training corpus.

Criterion (1) establishes the theoretical basis, while criteria (2) and (3) immediately show the benefit of a particular combination. A measure of similarity for (1) is currently under investigation. We chose (3) for our first experiments, since it was the easiest one to implement. The only additional effort is a separate, previously unused part of the training corpus for this purpose, the *clustering part*. We combine those tags into clusters which give the best results for tagging of the clustering part.

2.3 The Algorithm

The total number of potential clusterings grows exponential with the size of the tagset. Since we are interested in the reduction of large tagsets, a full search regarding all potential clusterings is not feasible. We compute the local maximum which can be found in polynomial time with a best-first search.

We use a slight modification of the algorithm used by (Stolcke and Omohundro, 1994) for merging HMMs. Our task is very similar to theirs. Stolcke and Omohundro start with a first order HMM where every state represents a single occurrence of a word in a corpus, and the goal is to maximize the a posteriori probability of the model. We start with a second order HMM (since we use trigrams) where each state represents a part of speech, and our goal is to maximize the tagging accuracy for a corpus.

The clustering algorithm works as follows:

1. Compute tagging accuracy for the clustering part with the original tagset.
2. Loop:
   (a) Compute a set of candidate clusters (obeying constraint (2) mentioned in section 2.1), each consisting of two tags from the previous step.
   (b) For each candidate cluster build the resulting tagset and compute tagging accuracy for that tagset.
   (c) If tagging accuracy decreases for all combinations of tags, break from the loop.
   (d) Add the cluster which maximized the tagging accuracy to the tagset and remove the two tags previously used.
3. Output the resulting tagset.

2.4 Application of Tag Clustering

Two standard trigram tagging procedures were performed as the baseline. Then clustering was performed on the same data and tagging was done with the reduced tagset. The reduced tagset was only internally used, the output of the tagger consisted of the original tagset for all experiments.

The Susanne Corpus has about 157,000 words and uses 424 tags (counting tags with indices denoting

Table 1: Tagging results for the test parts in the clustering experiments. Exp. 1 and 2 are used as the baseline.

	Training	Clustering	Testing	Result (known words)
1.	parts A and B	–	part C	93.7% correct
2.	parts A and C	–	part B	94.6% correct
3.	part A	part B	part C	93.9% correct
4.	part A	part C	part B	94.7% correct

multi-word lexemes as separate tags). The tags are based on the LOB tagset (Garside et al., 1987).

Three parts are taken from the corpus. Part A consists of about 127,000 words, part B of about 10,000 words, and part C of about 10,000 words. The rest of the corpus, about 10,000 words, is not used for this experiment. All parts are mutually disjunct.

First, part A and B were used for training, and part C for testing. Then, part A and C were used for training, and part B for testing. About 6% of the words in the test parts did not occur in the training parts, i.e. they are unknown. For the moment we only care about the known words and not about the unknown words (this is treated as a separate problem). Table 1 shows the tagging results for known words.

Clustering was applied in the next steps. In the third experiment, part A was used for trigram training, part B for clustering and part C for testing. In the fourth experiment, part A was used for trigram training, part C for clustering and part B for testing.

The baseline experiments used the clustering part for the normal training procedure to ensure that better performance in the clustering experiments is not due to information provided by the additional part.

Clustering reduced the tagset by 33 (third exp.), and 31 (fourth exp.) tags. The tagging results for the known words are shown in table 1.

The improvement in the tagging result is too small to be significant. However, the tagset is reduced, thus also reducing the number of parameters *without* losing accuracy. Experiments with larger texts and more permutations will be performed to get precise results for the improvement.

3 Conclusions

We have shown a method for reducing a tagset used for part-of-speech tagging without losing information given by the original tagset. In a first experiment, we were able to reduce a large tagset and needed fewer parameters for the n-gram model. Additionally, tagging accuracy slightly increased, but the improvement was not significant. Further investigation will focus on criteria for cluster selection. Can we use a similarity measure of probability distributions to identify optimal clusters? How far can we reduce the tagset without losing accuracy?

References

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA.

Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing (ACL)*, pages 133–140.

R. G. Garside, G. N. Leech, and G. R. Sampson (eds.). 1987. *The Computational Analysis of English*. Longman.

L. R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77(2), pages 257–285.

Geoffrey Sampson. 1995. *English for the Computer*. Oxford University Press, Oxford.

Andreas Stolcke and Stephen M. Omohundro. 1994. *Best-first model merging for hidden markov model induction*. Technical Report TR-94-003, International Computer Science Institute, Berkeley, California, USA.