Clustering Words by Syntactical Behavior

Wide R. Hogenhout and Yuji Matsumoto

29 March 1997

Abstract

We show how a treebank can be used to construct classes of words on the basis of their syntactical behavior. These classes represent distinct types of behavior with much more precision than parts of speech. We demonstrate an application of these classes to the problem of estimating the distribution of dependency relations between two words with sparse data. Experiments show a performance equivalent to other estimation methods when several thousands of classes are used, while the required storage space is reduced to about one third.

1 Introduction

The construction of classes of words, or calculation of distances between words, has frequently drawn the interest of researchers in natural language processing. Many of these studies aimed at establishing the semantical similarity between words (Brown et al., 1992; Dagan, Markus, and Markovitch, 1993; Dagan, Pereira, and Lee, 1994; Pereira and Tishby, 1992; Grefenstette, 1992).

We suggest a method for clustering words purely on the basis of syntactical behavior. We show how the necessary data for such clustering can easily be drawn from a publicly available treebank, and how distinct types of behavior can be discovered. Although a part of speech tag set can be thought of as a classification based on syntactical behavior, we can construct an arbitrary number of classes. For this goal we use a novel distance measure, which we experimentally compare to the more common measure known as divergence.

We proceed to demonstrate an example application of such classes. The distribution of the dependency relation between two words must necessarily be estimated with sparse data. By using a classification consisting of 3000 classes we can perform this estimation such that cross entropy is only slightly higher than for other estimation techniques, while the required storage space is reduced to about one third.

This, in turn, can be applied directly to broad coverage parsing systems.

2 Headwords and Dependencies

The distributions we use are based on the concept of *headwords*. Such headwords are chosen for every constituent in the parse tree by means of a simple set of rules. These have been used in various studies in this field, see (Collins, 1996; Magerman, 1995; Jelinek et al., 1994). Every headword is propagated up through the tree such that every parent receives a headword from the head-child.

Following the techniques suggested by (Collins, 1996), a parse tree can subsequently be described as a set of dependencies. Every word except the headword of the sentence depends on the lowest headword it is covered by. The syntactical relation is then given by the triple of nonterminals: the modifying nonterminal, the modified nonterminal, and the nonterminal that covers the joint phrase. Table 1 gives an example of such a description.

Our interest goes in particular to two distributions which reflect two fundamental properties of words, namely

$$p(R, t_h | w_d t_d) \tag{1}$$
$$p(R, t_d | w_h t_h) \tag{2}$$

where R indicates the triple representing the syntactical relation, w_d a dependent word that modifies headword w_h, and t_d and t_h their respective part of speech tags. These distributions indicate the syntactical behavior of words in the dependent role (1) and the syntactical behavior of words in the head role (2). Note that by ignoring the other word this will show syntactical behavior, such as that a particular verb is often used transitively, and not cooccurrences, such as *eat* with *cheese*.

Throughout this paper we will use the term "word" to refer to words for which the part of speech has already been disambiguated. In equations we emphasize this by writing $w_d t_d$ rather than just w_d.

Table 1: Dependencies for the sentence *John Smith works fast*

dependent word		head word		relation		
John	(proper noun)	Smith	(proper noun)	-	NP	-
Smith	(proper noun)	works	(verb)	NP	S	VP
fast	(adverb)	works	(verb)	-	VP	-

3 Similarity Between Words

Statistics of the distributions (1) and (2) described in the previous paragraph can easily be taken from a treebank. We took such data from the Wall Street Journal Treebank, calculating the probabilities with the Maximum Likelihood Estimator:

$$p(R, t_h | w_d t_d) = \frac{f(R, t_h, w_d t_d)}{f(w_d t_d)}$$

where f stands for frequency.

Compare the dependency behavior of the proper nouns *Nippon* and *Rep.* in table 2. As can be seen, the word *Rep.* occurs far more frequently, but their distributions are highly similar. Both always modified another proper noun, around 33% of the time forming an NP-SBJ and 67% of the time an NP. Both are a particular kind of proper noun that almost always modifies other proper nouns and almost never appears by itself.

The noun *company* is however very different from a noun such as *hostage*, since *company* often is the subject of a verb, while *hostage* is rarely in the subject position. Both are also very different from the noun *year*, which is frequently used as the object of a preposition.

The gerund *including* has an extremely strong tendency to produce prepositional phrases, as in *"Safety advocates, including some members of Congress,..."*, making it different from most other gerunds. A past tense such as *fell* has an unusual high frequency as the head of a sentence rather than a verb phrase, which is probably a peculiarity of the Wall Street Journal (*"Stock prices fell..."*).

Our observation is that among words which have the same part of speech, some word groups exhibit behavior that is extremely similar, while others display large differences. The method we suggest aims at making a set of classes based on such behavior. By using a clustering technique any number of classes can be obtained, far beyond what humans can be expected to recognize as distinct categories.

Table 2: Distributions for the proper nouns *Nippon* and *Rep.*

dependent word		headword tag	relation			freq.
Nippon	proper noun	proper noun	-	NP-SBJ	-	3
		proper noun	-	NP	-	6
Rep.	proper noun	proper noun	-	NP-SBJ	-	23
		proper noun	-	NP	-	45

4 Similarity Measure and Algorithm

Studies in distribution based clustering often use the KL-distance, see for example (Pereira, Tishby, and Lee, 1993; Dagan, Pereira, and Lee, 1994). This distance is not symmetrical, and since we are interested in a symmetrical measure, we could possibly use divergence (also known as Jeffery's Information), i.e. the sum of the KL-distances:

$$Div(p,q) = \sum_x p(x) \log \left(\frac{p(x)}{q(x)} \right) + q(x) \log \left(\frac{q(x)}{p(x)} \right). \tag{3}$$

We have tried this distance measure, and will present results for comparison, but in many cases we have found it to have undesirable effects, primarily because the goal is joining words (and their statistics) together to make one cluster, and a distorted image results from this measure when words have different total frequencies. Furthermore, the divergence is undefined if either distribution has a value of 0 and the other not. For this reason they would have to be smoothed with, for example, a part of speech based distribution, such as

$$\hat{p}(R, t_h | w_d t_d) = \lambda(R, t_h | w_d t_d) + (1 - \lambda)(R, t_h | t_d), \tag{4}$$

but we wanted to avoid using an unlexical distribution since we believe lexical information is more valuable.

Instead we suggest a different measure. Assume there are a number of patterns $i = 1...n$, and observed frequencies $a_1...a_n$ for word $w_a t_a$, and $b_1...b_n$ for word $w_b t_b$. Also, let $A = \sum_i a_i$ and $B = \sum_i b_i$. The Maximum Likelihood Estimates for w_a are thus calculated as $p_a(x) = a_x / A$ and likewise for w_b.

We now define the distance between words as

$$M(w_a t_a, w_b t_b) \stackrel{\text{def}}{=} \sum_i \frac{a_i}{A} \log \left(\frac{a_i}{A} \frac{(A+B)}{(a_i + b_i)} \right) + \frac{b_i}{B} \log \left(\frac{b_i}{B} \frac{(A+B)}{(a_i + b_i)} \right)$$

which can be interpreted as the sum of KL-distances between a hypothetical word that would be created if the observations of the words $w_a t_a$ and $w_b t_b$ would be joined together, and $w_a t_a$ and $w_b t_b$ respectively. Like divergence, this measure is symmetrical, although not a true distance since it does not obey the triangle inequality.

This measure seems more appropriate for two reasons. First, this distribution is better tailored toward making clusters where observations will be joined together. Second, we take this sum to be zero for values of i when $a_i = b_i = 0$ (no observations for either word), therefore pre-smoothing is not necessary.

The equation can easily be transformed into the form

$$\begin{aligned} M(w_a t_a, w_b t_b) &= \log\left(\frac{A+B}{A}\right) + \log\left(\frac{A+B}{B}\right) + \\ & \sum_i \frac{a_i}{A}\log\left(\frac{a_i}{a_i+b_i}\right) + \frac{b_i}{B}\log\left(\frac{b_i}{a_i+b_i}\right) \quad (5) \end{aligned}$$

which makes calculation significantly faster since patterns for which only one word has a non-zero frequency do not need to be calculated within the summation, as they always becomes zero.

The Algorithm The algorithm initially regards every word as a 1-element cluster. First, all clusters which have equal distributions are merged together. This phase has no influence on the distributions of clusters but enormously reduces the number of clusters (In our experiment by about 50%).

Next, the strategy of a *greedy algorithm* is followed, every time finding the two clusters that have the least distance between them and merging them until the desired number of clusters is reached. There are however two conditions. First, only words with the same part of speech may be clustered, so distances between words that have different parts of speech are never calculated. Second, if a cluster exists that contains only one word which occurred only once, it must be merged to whatever it is closest to before other clusters are merged.

The algorithm is actually applied *twice*. It is applied once to clustering for dependent-context (1), and once to clustering for head-context (2).

The computational expense of this method is modest. Since words with equal distributions are collapsed into one cluster, and only words with equal parts of speech need to be compared, we could easily cluster over 43,000 words to any number of clusters in one day on a Sparc Station. We notice however, that the new measure we suggest (5) made the process several times faster as compared to using divergence (3).

5 Comparison with Clustering for Bigrams

Clustering of words based on syntactical behavior has to our knowledge not been carried out before, but clustering has been applied to the estimation of bigram distributions in various studies. This context is rather different, since the conditioned event is a word instead of a relation. Also, many other clustering studies incorporate the objective of finding semantically similar words, whereas we ignore semantical similarity.

We present a *hard* clustering algorithm, in the tradition of (Brown et al., 1992). They apply clustering to n-gram models, estimating for example $P(w_i|w_{i-1})$ by finding a cluster for w_{i-1}. They report that required storage was reduced to one third, at the expense of a slight increase in perplexity.

Besides *hard* algorithms there have also been studies to *soft* clustering (Pereira, Tishby, and Lee, 1993; Dagan, Pereira, and Lee, 1994) where every word is smoothed with the nearest k words rather than placed in a class which supposedly has a uniform behavior. In (Dagan, Markus, and Markovitch, 1993) it was argued that reduction to a relatively small number of predetermined word classes or clusters may lead to substantial loss of information.

The fact that the number of classes we obtain is not so small, as we work with thousands of classes, may be one reason why our hard clustering is effective. Nevertheless we studied the possibility of applying soft clustering to syntactical relations, although by this method one would not obtain classes representing types of behavior. A significant obstacle we encountered when trying to apply this method is the fact that many words have k neighbors that are totally equal, and gain nothing if those are used for smoothing.

6 An Application: The Relation Between Words

The distribution of types of dependency relations between words is of importance to natural language processing, partly because of the direct practical application in parsing systems, but even more so because it can show fundamental properties of language. At the same time it is clear that good estimation techniques will be important due to the enormous number of parameters.

Consider the distribution of the dependency relation between two words, given that there is some dependency relation between them, and given their respective parts of speech:

$$p(R|w_d t_d, w_h t_h). \tag{6}$$

The used symbols carry the same meaning as in section 2. Our conjecture is that the classes we obtained will also indicate similarity within the distribution (6).

Almost the same distribution is used by (Collins, 1996), and we would like to refer to this publication for more details, but we note that for the goal of our research we have slightly simplified the relation. We ignore the distance between words, and also do not use the *reduced sentence*. When applied in a parsing system, these are undoubtedly valuable tools, and our work can easily be adjusted to accommodate such modifications.

7 Experimental Setup

The baseline model we use is the regular model smoothed with part of speech based probabilities using a method suggested by (Bahl, Jelinek, and Mercer, 1983). To be exact,

$$\begin{aligned}
\hat{p}_{lexical}(R|w_d t_d, w_h t_h) &= \lambda_1(w_d t_d, w_h t_h) p(R|w_d t_d, w_h t_h) + \\
&\quad \lambda_2(w_d t_d, w_h t_h) p(R|t_d, w_h t_h) + \\
&\quad \lambda_3(w_d t_d, w_h t_h) p(R|w_d t_d, t_h) + \\
&\quad \lambda_4(w_d t_d, w_h t_h) p(R|t_d, t_h)
\end{aligned}$$

which we refer to in the table als **lexical**. The basic probabilities are found with the Most-Likelihood estimator, and the interpolation parameters are estimated from data held out for this goal, but not separately for every word pair. As suggested by Bahl et al. they are divided into buckets based on their count, so that all word pairs with the same frequency have the same interpolation parameters. The frequencies were also joined into one bucket when there was too little data to estimate them. We used the same technique for all other models. In every bucket the four lamda values were optimized with pairwise optimizations until they all stabilized.

The class-based model we compare this with is

$$\begin{aligned}
\hat{p}_{classes}(R|w_d t_d, w_h t_h) &= \lambda_1(c_d, c_h) p(R|c_d, c_h) + \\
&\quad \lambda_2(c_d, c_h) p(R|t_d, c_h) + \\
&\quad \lambda_3(c_d, c_h) p(R|c_d, t_h) + \\
&\quad \lambda_4(c_d, c_h) p(R|t_d, t_h)
\end{aligned}$$

were c_d is the class of $w_d t_d$ (using the dependent-word clustering), and c_h the class of $w_h t_h$ (using the headword clustering). These classes were made with the new measure (5). In the table this model is identified as **classes**.

Since test data was not used for clustering, the test data contained a certain percentage of words that were never clustered by the algorithm. For this reason an imaginary extra cluster is assumed, containing all such "unseen" words. Probabilities conditioned on this class are always zero.

Exactly the same model has also been applied to the model identified as **divergence**. This model is completely identical to **classes**, but with the clusters produced using divergence (3) as the distance measure (equation (4) was used for smoothing word distributions), instead of our measure (5).

We also present results of a combined model using both lexical and class-based probabilities. This is calculated with the equation

$$\begin{aligned}\hat{p}_{comb}(R|w_d t_d, w_h t_h) \quad = \quad & \lambda_1(w_d t_d, w_h t_h) p(R|w_d t_d, w_h t_h) + \\ & \lambda_2(w_d t_d, w_h t_h) p(R|c_d, c_h) + \\ & \lambda_3(w_d t_d, w_h t_h) p(R|t_d, c_h) + \\ & \lambda_4(w_d t_d, w_h t_h) p(R|c_d, t_h) + \\ & \lambda_5(w_d t_d, w_h t_h) p(R|t_d, t_h)\end{aligned}$$

and identified as **combined**.

8 Experimental Results

The experiments were carried out with data from the Wall Street Journal Corpus. A set of about 740,000 patterns was used as training data, for clustering words, and to optimize smoothing parameters. A separate group of 60,000 patterns was used for testing. In order to assess statistical significance of results this experiment was carried out 5 times, each time using different test data.

In all clustering methods different sets of classes were made for the dependent position and the headword position. The number of classes we found to be optimal was about 3000 for both dependent position and headword position.

Table 3 shows the results for the given models. The cross entropy of predicting the relation of all patterns is given in the first column. The second column gives the entropy on patterns that occurred at least once in the training data ($f(R, w_d t_d, w_h t_h) > 0$). The third column indicates entropy on patterns that did not occur in the training data ($f(R, w_d t_d, w_h t_h) = 0$). All differences between models were found to be significant with a paired t-test ($p < 0.001$), except the difference between the pair marked with the symbol "$\star$."

The classes produced with the distance measure (5) display a cross entropy of only about 3% more than the lexical model. The size of the tables needed for the class-based probabilities is about one third of that of lexical tables. As expected, the performance is not as good as the lexical model for the more common patterns.

The classes that were produced with divergence (3) did not perform as well, illustrating the problems we mentioned earlier.

Since the lexical probabilities perform better on more frequent events and class-based probabilities better on infrequent events, a logical next step is combining them. The score of **combined** illustrates that this creates a gain in entropy of about 4.6%.

Table 3: Cross Entropy on Test Data

method	All Data	Occurred	Not Occurred
lexical	1.862	0.729	3.178
classes	1.899	0.988	2.958
divergence	2.198	1.423	3.098 ⋆
combined	1.811	0.707	3.093 ⋆

9 Conclusion

We have presented a method which constructs classes of words with similar syntactical behavior by clustering words using treebank data. This is different from previous research since we ignore semantical similarity and focus purely on syntactical behavior. Experiments show that these classes do represent meaningful types of words, and can be used to perform estimation of the distribution of dependency relations between words, while requiring only one third of the storage space. For this goal we have used a novel distance measure for distributional analysis, which clearly outperforms the measure known as divergence.

When applied in a parsing system in combination with other conditioning events, such as the distance within the sentence, problems with the data sparseness will become even more serious, so the need for this technique will be even more urgent.

References

Bahl, Lalit R., Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190.

Brown, P. F., S. A. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Collins, M. J. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 184–191.

Dagan, I., S. Markus, and S. Markovitch. 1993. Contextual similarity and estimation from sparse data. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 164–171.

Dagan, I., F. Pereira, and L. Lee. 1994. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 272–278.

Grefenstette, G. 1992. Finding semantic similarity in raw text: the Deese antonyms. In *Working Notes, Fall Symposium Series, AAAI*, pages 61–68.

Jelinek, F., J. Lafferty, D. Magerman, R. Mercer, A. Ratnaparkhi, and S. Roukos. 1994. Decision tree parsing using a hidden derivation model. In *ARPA: Proceedings of the Human Language Technology Workshop*, pages 272–277.

Magerman, D. M. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33d Annual Meeting of the Association for Computational Linguistics*, pages 276–283.

Pereira, F. and N. Tishby. 1992. Distributional similarity, phase transitions and hierarchical clustering. In *Working Notes, Fall Symposium Series, AAAI*, pages 108–112.

Pereira, F., N. Tishby, and L. Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190.