A Preliminary Study of Word Clustering Based on Syntactic Behavior

Wide R. Hogenhout and **Yuji Matsumoto**
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara 630-01, Japan
{marc-h,matsu}@is.aist-nara.ac.jp

Abstract

We show how a treebank can be used to cluster words on the basis of their syntactic behavior. The resulting clusters represent distinct types of behavior with much more precision than parts of speech. As an example we show how prepositions can be automatically subdivided by their syntactic behavior and discuss the appropriateness of such a subdivision. Applications of this work are also discussed.

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 finding classes based on co-occurrences, often combined with the aim of establishing semantic similarity between words (McMahon and Smith, 1996, 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 syntactic 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 syntactic behavior, we can construct an arbitrary number of clusters, or a binary tree of words that share their part of speech.

We discuss in detail a binary word tree for prepositions that was created by syntactic-behavior based clustering, to show what sort of properties are revealed by the clustering and what one can learn from this about language. We also discuss various ways in which this kind of clustering can be used in NLP applications.

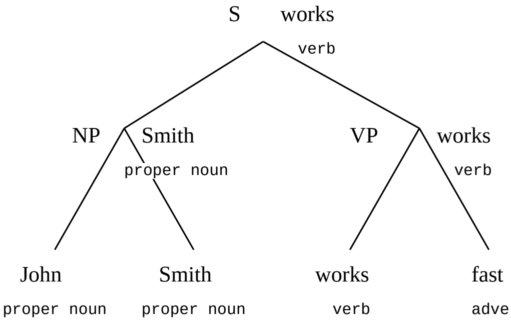

Figure 1: Sentence with Parse Tree and Headwords

2 Headwords and Dependencies

The data we extract 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. Figure 1 gives an example of a parse tree with headwords.

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 syntactic 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.

On one point our method is different from the method suggested by Collins. Collins uses a *reduced sentence* in which every basic noun phrase (i.e., a noun phrase that has no noun phrase as a child) is reduced to its headword. The reason for this is that it improves co-occurrence counts and adjacency statistics. We however do not reduce the sentence

Wide R. Hogenhout and Yuji Matsumoto (1997) **Word Clustering from Syntactic Behavior.** In T.M. Ellison (ed.) *CoNLL97: Computational Natural Language Learning,* ACL pp 16-24.
© 1997 Association for Computational Linguistics

since we do not need to consider adjacency statistics or unresolved ambiguities, and therefore never face the problem that a word in a basic noun phrase, that is not the headword, is adjacent to or modifies something outside of the basic noun phrase.

Table 1 gives the relations for one sentence, but instead of considering one sentence we collect such patterns for the whole corpus and study statistics for individual words. In this way it can be discovered that, for example, a particular verb is often used transitively, or that a particular preposition is mostly used to produce locative prepositional phrases. Words can be distinct or similar in this respect, but note that this is not related to semantic similarity. Words such as *eat* and *drink* have a semantic similarity, but may be completely different in their syntactic behavior, whereas *tend* and *appear* do not have an obvious semantic relation, but they do have a similarity since they can both be used as raising verbs, as will be exemplified later.

Throughout this paper we will use the term "word" to refer to words for which the part of speech has already been disambiguated. In tables and figures we emphasize this by indicating the part of speech together with the word.

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 Collecting Statistics for Individual Words

The next step we take, is eliminating one of the two words in this table of dependencies. Consider tables 2 and 3. These show we can take three "observations" from the sentence by eliminating either the headword or the dependent word. If headwords are eliminated we obtain three observations, for the words *John*, *Smith* and *fast*. If dependent words are eliminated we also obtain three observations, two for *works* and one for *Smith*.

By collecting the observations over the entire corpus we can see to/by what sort of words and with what kind of relations a word modifies or is modified. We consider the following distributions:

$$p(R, t_h | w_d t_d) \quad (1)$$

$$p(R, t_d | w_h t_h) \quad (2)$$

where R indicates the triple representing the syntactic relation, w_d a dependent word that modifies headword w_h, and t_d and t_h their respective part of speech tags. For example, in the second line of table 3, which corresponds to distribution 1, R is (NP,S,VP), t_h is "verb", w_d is "Smith" and t_d is "proper noun".

Statistics of the distributions (1) and (2) 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:

$$p(R, t_h | w_d t_d) = \frac{f(R, t_h, w_d t_d)}{\sum_{R', t'} f(R', t', w_d t_d)}$$

where f stands for frequency. Note that we only extract the dependency relations, and ignore the structure of the sentence beyond these relations. This shows the equation for distribution (1), distribution (2) is calculated likewise.

Compare the dependency behavior of the proper nouns *Nippon* and *Rep.* in table 4. The word *Nippon* is Japanese for *Japan*, and mainly occurs in names of companies. The word *Rep.* is the abbreviation of *Republic*, and obviously occurs mainly in names of countries. As can be seen, the word *Rep.* occurs far more frequently, but the distributions are highly similar. Both always modify another proper noun, about 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.

It also became clear that the noun *company* is 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 clustering based on such behavior. By using this technique any number of clusters can be obtained, sometimes far beyond what humans can be expected to recognize as distinct categories.

Table 2: Dependencies with dependent words eliminated.

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

Table 3: Dependencies with headwords eliminated.

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

Table 4: Distribution of dependencies of the words *Nippon* and *Rep.*, as proper nouns.

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

4 Comparison with Co-Occurrence Based Clustering

Clustering of words based on syntactic behavior has to our knowledge not been carried out before, but clustering has been applied with the goal of obtaining classes based on co-occurrences. Such clusters were used in particular for interpolated n-gram language models.

By looking at co-occurrences it is possible to find groups of words such as [*director, chief, professor, commissioner, commander, superintendent*]. The most prominent method for discovering their similarity is by finding words that tend to co-occur with these words. In this case they may for example co-occur with words such as *decide* and *lecture*.

The group of verbs [*tend, plan, continue, want, need, seem, appear*] also share a similarity, but one has to look at structures rather than meaning or co-occurrences to see why. All these verbs tend to occur in the same kind of structures, as can be seen in the following examples from the Wall Street Journal.

```
The funds' share prices tend to
swing more than the broader market.

Investors continue to pour cash
into money funds.

Cray Research did not want to fund
a project that did not include
Seymour.

No one has worked out the players'
average age, but most appear to be
in their late 30s.
```

What these verbs share is the property that they often modify an entire clause (marked as 'S' in the Wall Street Journal Treebank) rather than noun phrases or prepositional phrases, usually forming a subject raising construction. This is only a tendency, since all of them can be used in a different way as well, but the tendency is strong enough to make their usage quite similar. Co-occurrence based clustering ignores the structure in which the word occurs, and would therefore not be the right method to find related similarities.

As mentioned, co-occurrence based clustering methods often also aim at producing semantically meaningful clusters. Various methods are based on Mutual Information between classes, see (Brown et al., 1992, McMahon and Smith, 1996, Kneser and Ney, 1993, Jardino and Adda, 1993, Martin, Liermann, and Ney, 1995, Ueberla, 1995). This measure cannot be applied in our case since we look at structure and ignore other words, and consequently algorithms using that measure cannot be applied to the problem we deal with.

The mentioned studies use word-clusters for interpolated n-gram language models. Another application of hard clustering methods (in particular bottom-up variants) is that they can also produce a binary tree, which can be used for decision-tree based systems such as the SPATTER parser (Magerman, 1995) or the ATR Decision-Tree Part-Of-Speech Tagger (Black et al., 1992, Ushioda, 1996).

In this case a decision tree contains binary questions to decide the properties of a word.

We present a *hard* clustering algorithm, in the sense that every word belongs to exactly one cluster (or is one leaf in the binary word-tree of a particular part of speech). Besides *hard* algorithms there have also been studies to *soft* clustering (Pereira, Tishby, and Lee, 1993, Dagan, Pereira, and Lee, 1994) where the distribution of every word is smoothed with the nearest k words rather than placed in a class which supposedly has a uniform behavior. In fact, 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. On the other hand, when using soft clustering it is not possible to give a yes/no answer about class membership, and binary word trees cannot be constructed.

5 Similarity Measure and Algorithm

The choice of the clustering algorithm is to some extent independent from the way data is collected, but as mentioned clustering is carried out on the basis of distributional similarity, and methods using Mutual Information are not applicable. The algorithm we present here is meant to demonstrate how syntactic behavior can be used for clustering. However, we feel the optimal choice for the clustering method depends on the application it will be used for.

Studies in distribution based clustering often use the Kullback-Leibler (KL) distance, see for example (Pereira, Tishby, and Lee, 1993, Dagan, Pereira, and Lee, 1994). However, this distance is not symmetrical, and since we are (for the time being) interested in *hard clustering* it is desirable to have a symmetrical measure. We could possibly use Jeffery's Information, i.e. the sum of the KL-distances:

$$\begin{aligned} D(p,q) &= \sum_x p(x)\log\left(\frac{p(x)}{q(x)}\right) + \\ &\qquad q(x)\log\left(\frac{q(x)}{p(x)}\right). \end{aligned} \tag{3}$$

We have tried this distance measure, but in many cases we have found it to have undesirable effects, primarily because the goal of our algorithm 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, Jeffery's Information 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

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

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

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

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 Jeffery's Information, this measure is symmetrical, although not a true distance since it does not obey the triangle inequality.

This measure is 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

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

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, and works *bottom up* towards a set of clusters. 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. However, only words with the same part of speech may be merged, so distances between words that have different parts of speech are never calculated. Words can therefore receive a 'combined tag' consisting of their part of speech tag, and a syntactic behavior tag. This is similar to what McMahon (1996) refers to as a *structural tag.*

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

An obvious problem with this sort of clustering is low frequency words. For many words only a one or a few observations are available, which may give some information about what sort of word it is, but which does not give a reliable estimate of the distributions. We will mention a solution to this problem later. In the example we present only words for which at least 25 observations are available.

One problem with co-occurrence based clustering that has been pointed out in the past is that of almost-linear dendrograms, caused by the properties of Mutual Information. We have not encountered this problem with the described algorithm.

6 Result: the Case of Prepositions

We present a binary word tree that was produced by the algorithm described in the previous section. The main goal of this is to show what sort of properties are revealed by this clustering, and what kind of words are problematic. Even in situations where words are clustered by syntactic behavior without making a binary tree, it can be useful to study the type of properties that decide syntactic behavior.

Please refer to figure 2 for an example of the results obtained with clustering. This is a dendrogram that reflects the clustering process from loose words until the point were they are all merged into one cluster. The dendrogram shows the result for prepositions, although only those prepositions were considered for which at least 25 observations were available. In the division of words over the parts of speech we follow the tagging scheme of the Wall Street Journal Treebank, and for example subordinators such as *while, if* and *because* are included in the prepositions. Of course it is possible to use a more fine grained tag set, when available. On the other hand, as will be shown later, the algorithm does decide to classify most subordinators into one cluster.

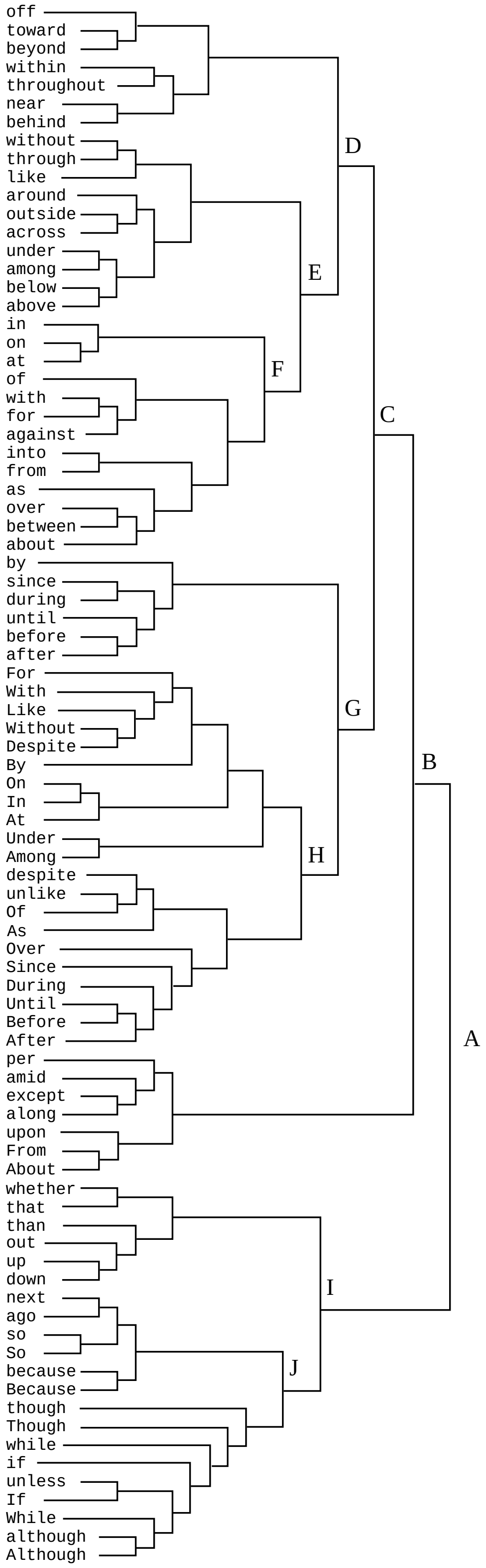

Figure 2: Clustering Result for Dependency Behavior of Prepositions

We will discuss the major distinctions made by the algorithm. At first it may not be clear why words should be divided in this way, but inspection of the data from the corpus shows that many of these choices are very natural. We also discuss in which cases the dendrogram does not form natural categories.

The first partition, marked A, is a quite natural division. The upper branch (from *off* through *About*) are prepositions that usually cover some phrase themselves, whereas the prepositions in the lower branch usually do not cover any phrase.

The preposition *whether* occurs, for example, in structures such as

```
`` We have no useful information
on (SBAR whether (S users are at
risk)),'' said James A. Talcott
of Boston's Dana-Farber Cancer
Institute.
```

where in our headword-selection scheme *whether* depends on the headword *are.* (Even if this is changed, they still become one cluster because of the typical patterns with S and SBAR.)

For comparison, the preposition *below* usually occurs in structures such as

```
Magna recently cut its quarterly
dividend in half and the company's
Class A shares are (VP wallowing
(PP-LOC far below their 52-week
high of 16.125 Canadian dollars
(US$ 13.73))).
```

where it is the headword of a prepositional phrase before it modifies the verb.

The partition marked with B is not a natural division; it rather separates a set of prepositions that do not fit in elsewhere. The prepositions from *per* through *About* are not similar to each other or to other prepositions in their behavior.

Partition C again resembles to groups that can be characterized easily. The prepositions *by* through *After*, the lower branch of C, depended almost exclusively on verbs. The prepositions from *off* through *about*, the upper branch of C, depend on more varied headwords. Most of these frequently depend on both nouns and verbs. The following example shows *around* depending on a noun, although *around* also tends to depend on cardinal numbers.

```
You now may drop by the Voice of
America offices in Washington and
read the text of what the Voice is
broadcasting to those 130 million
people (PP-LOC around the world)
who tune in to it each week.
```

An example for the lower branch of C is

```
A plan to bring the stock to market
before year end apparently (VP was
upset (PP by the recent weakness of
Frankfurt share prices) ).
```

The prepositions at the upper branch of partition D tend to form a higher amount of PP-TMP type phrases, as in

```
And in each case, he says, a sharp
drop in stock prices (VP began
(PP-TMP within a year)).
```

although, while this is strongly the case for the prepositions *within* and *throughout,* it is not the case for *behind.*

At partition E prepositions with a preference for verbs are at the upper branch. Prepositions that almost exclusively deal with verbs were separated at C, but here the distinction is less absolute. The prepositions at the upper branch of E have a chance of about two thirds to depend on a verb, whereas this is only one third at the lower branch.

Partition F is once again a very clear, natural division. The prepositions *in*, *on* and *at* have a strong tendency to form phrases of the type PP-LOC as in

```
Mr.  Nixon is traveling (PP-LOC
in China) as a private citizen,
but he has made clear that he is
an unofficial envoy for the Bush
administration.
```

while the prepositions at the lower branch, *of* through *about* have much lower frequencies for these locative phrases.

The division at G is also very clear when the data are inspected. The upper branch reflects prepositions for which the covering phrase (the middle part of the triple representing the grammatical relation) is mostly VP or NP. The prepositions *For* through *After* at the lower branch of G are mainly covered by phrases of type S. A preposition such as *during* is found in structures such as

```
Fujitsu said it (VP bid the
equivalent of less than a U.S.
penny on three separate municipal
contracts (PP-TMP during the past
two years)).
```

while a preposition such as *without* is usually found in the PP-S-VP pattern:

```
(S In fact, (PP without Wall
Street firms trading for their
own accounts), the stock-index
arbitrage trading opportunities for
the big funds (VP may be all the
more abundant).)
```

At H this is further divided in words that tend more to depend on loose words, PP type phrases (such as *without* in the last example) or S type phrases, at the lower branch, and those that usually depend on heads of a VP.

As for the division at point I, the prepositions *next* through *Although* share the property that their covering phrase (the middle part of the triple representing the grammatical relation) is often of the type SBAR-ADV or SBAR-PRP. The prepositions at the upper branch, *whether* through *down,* mainly share not having this property.

While the status of the upper branch of J is somewhat unclear, the lower branch of J is a perfectly clear and intuitive group. All of the words from *though* through *Although* appear almost exclusively in the patterns (–,SBAR-ADV,S), (–,S,S), (–,SBAR-PRP,S) and (–,SBAR-PRP,–). An example is

```
The group says standardized
achievement test scores are greatly
inflated because teachers often
''teach the test'' as Mrs.  Yeargin
did, (SBAR-ADV although (S most are
never caught)).
```

where in our headword scheme *are* becomes the headword of the SBAR-ADV type phrase.

Concluding, many of the divisions made by the algorithm are quite natural. There are some parts of speech (such as nouns and verbs) were a much larger number of words is included in the hierarchy, while some other parts of speech, for example personal pronouns, produce very small hierarchies. In general the hierarchy is more interesting for parts of speech that are used in a varied way, and less interesting for, for example, symbols such as the percentage sign, that are used in a monotone way.

It is interesting to see that capitalization turns out to be a meaningful predictor about the way a word will be used for some words, but not for others. The word pair *so* and *So*, and the pair *because* and *Because* are clustered next to each other, which indicates that they modify the same kind of structures, independent of whether they are at the beginning of the sentence. The word pair *under* and *Under*, and the pair *after* and *After* on the other hand are rather far apart, indicating that their usage changes substantially when they become the first word of the sentence.

7 Applications

A first application of this work, of which we carried out a first step in this article, is the lexicographical one of studying word behavior. Some properties of words, such as the peculiar behavior of the gerund *including* or the similarities between prepositions such as *though* and *while* only becomes clear once the corpus data is analyzed in the way we described. When inspecting manually, the binary word tree representation appears to be the most easy to understand.

A second application of the binary word tree can be found in decision-tree based systems such as the SPATTER parser (Magerman, 1995) or the ATR Decision-Tree Part-Of-Speech Tagger, as described by Ushioda (Ushioda, 1996). In this case it is necessary to use a hard-clustering method, such that a binary word tree can be constructed by the clustering process, as we did in the example in the previous sections.

A decision tree classifies data according to its properties by asking successive (often binary) questions. In the case of a part of speech tagger or a parsing system, it is particularly important for the system to ask lexicalizing questions. However, questions about individual words such as "Is this the word *display*?" are not efficient on a large scale since it would easily require thousands of questions. A binary tree allows one to separate the vocabulary into two parts at every question, which is efficient when these two parts are maximally different. In that case it is possible to obtain as much information as possible with a small number of questions. A condition for this application is that trees may not be very unbalanced, as the extreme case of a linear tree becomes equal to asking word-by-word. As mentioned, the method we suggest did not produce a very unbalanced tree for the parts of speech in the Wall Street Journal Treebank.

A third application can be found in Information Retrieval. This can be seen from the example of *including*: words with such behavior have little content because they have a rather functional role in the sentence. This can be seen in the sentence *"Safety advocates, including some members of Congress,..."* where terms such as *Safety advocates* or *members of Congress* indicate much more about the topic of the sentence than the relatively empty word *including.* It is possible to cluster words and decide which clusters are likely to indicate the topic, and which are not likely to do so. For this application a wider variety of algorithms can be applied; words can for example be *exchanged* or *shuffled* between classes to improve the entire model.

A fourth application is class-based smoothing of interpolated n-gram models. The co-occurrence based classes described in the literature are, of course, created with this as objective function, but on the other hand the classes we suggest clearly contain information that is inaccessible to co-occurrence based classes. It is possible that a combination of co-occurrence based classes and classes of syntactic behavior would give better results, but this would have to be demonstrated experimentally.

In some of these applications words with a low frequency cannot be ignored because of their quantity, but at the same time the algorithm cannot rely too heavily on their observations. A possible solution is to carry out clustering without these words, and distribute the low-frequency words over the leaves of the tree afterwards. A solution along this line was chosen for co-occurrence based clustering in (McMahon and Smith, 1996), where a first algorithm handles more frequent words, and a second algorithm adds the low-frequency words afterwards.

8 Conclusion

We have presented a method which constructs classes of words with similar syntactic behavior, or binary trees that reflect word similarity, by clustering words using treebank data. In this way it is possible to discover particular types of behavior, such as the peculiar behavior of the gerund *including*, verbs that modify an entire clause (raising verbs), nouns that prefer either subject position or object position, or prepositions that prefer locative phrases.

Most of the classes found in this way would not be found if clustering were performed on the basis of co-occurrences, as has been described in the literature. For example, the verbs [*tend, plan, continue, want, need, seem, appear*] share a particular sentence structure rather than, say, the sort of noun that becomes the object.

As became clear from the case study of prepositions, the clustering process reveals similarities in the syntactic structure in which words appear which in some cases can be clearly felt by intuition. For example, the words *in*, *on* and *at* often are the head of locative prepositional phrases, and a preposition such as *within* usually is the head of a temporal prepositional phrase. Using this method these intuitions can be quantified.

One of the applications we described is that of a decision-tree based system for syntactic analysis. We are currently applying the method in this application, which will be described in later publications.

9 Acknowledgements

We would like to express our appreciation to the anonymous reviewers who have provided many valuable comments and criticisms.

References

Black, E., F. Jelinek, R. Mercer, and S. Roukos. 1992. Decision tree models applied to the labeling of text with parts-of-speech. In *Proceedings DARPA Speech and Natural Language Workshop*, pages 117–121.

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.

Jardino, M. and G. Adda. 1993. Automatic word classification using simulated annealing. In *ICASSP 93*, pages II 41–44.

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.

Kneser, R. and H. Ney. 1993. Improved clustering techniques for class-based statistical language modelling. In *Proceedings of European Conference on Speech Communication and Technology*, pages 973–976.

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.

Martin, M., J. Liermann, and H. Ney. 1995. Algorithms for bigram and trigram word clustering. In *Proceedings of European Conference on Speech Communication and Technology*, pages 1253–1256.

McMahon, John G. and Francis J. Smith. 1996. Improving statistical language model performance with automatically generated word hierarchies. *Computational Linguistics*, 22(2):217–247.

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.

Ueberla, J. 1995. More efficient clustering of n-grams for statistical language modeling. In *Proceedings of European Conference on Speech Communication and Technology*, pages 1257–1260.

Ushioda, Akira. 1996. Hierarchical clustering of words and application to nlp tasks. In *Proceedings of the Fourth Workshop on Very Large Corpora*.