The order of prenominal adjectives in natural language generation

Robert Malouf
Alfa Informatica
Rijksuniversiteit Groningen
Postbus 716
9700 AS Groningen
The Netherlands
malouf@let.rug.nl

Abstract

The order of prenominal adjectival modifiers in English is governed by complex and difficult to describe constraints which straddle the boundary between competence and performance. This paper describes and compares a number of statistical and machine learning techniques for ordering sequences of adjectives in the context of a natural language generation system.

1 The problem

The question of robustness is a perennial problem for parsing systems. In order to be useful, a parser must be able to accept a wide range of input types, and must be able to gracefully deal with dysfluencies, false starts, and other ungrammatical input. In natural language generation, on the other hand, robustness is not an issue in the same way. While a tactical generator must be able to deal with a wide range of semantic inputs, it only needs to produce grammatical strings, and the grammar writer can select in advance which construction types will be considered grammatical. However, it is important that a generator not produce strings which are strictly speaking grammatical but for some reason unusual. This is a particular problem for dialog systems which use the same grammar for both parsing and generation. The looseness required for robust parsing is in direct opposition to the tightness needed for high quality generation.

One area where this tension shows itself clearly is in the order of prenominal modifiers in English. In principle, prenominal adjectives can, depending on context, occur in almost any order:

the large red American car

??the American red large car

*car American red the large

Some orders are more marked than others, but none are strictly speaking ungrammatical. So, the grammar should not put any strong constraints on adjective order. For a generation system, however, it is important that sequences of adjectives be produced in the 'correct' order. Any other order will at best sound odd and at worst convey an unintended meaning.

Unfortunately, while there are rules of thumb for ordering adjectives, none lend themselves to a computational implementation. For example, adjectives denoting size do tend to precede adjectives denoting color. However, these rules underspecify the relative order for many pairs of adjectives and are often difficult to apply in practice. In this paper, we will discuss a number of statistical and machine learning approaches to automatically extracting from large corpora the constraints on the order of prenominal adjectives in English.

2 Word bigram model

The problem of generating ordered sequences of adjectives is an instance of the more general problem of selecting among a number of possible outputs from a natural language generation system. One approach to this more general problem, taken by the 'Nitrogen' generator (Langkilde and Knight, 1998a; Langkilde and Knight, 1998b), takes advantage of standard statistical techniques by generating a lattice of all possible strings given a semantic representation as input and selecting the most likely output using a bigram language model.

Langkilde and Knight report that this strategy yields good results for problems like generating verb/object collocations and for selecting the correct morphological form of a word. It also should be straightforwardly applicable to the more specific problem we are addressing here. To determine the correct order for a sequence of prenominal adjectives, we can simply generate all possible orderings and choose the one with the highest probability. This has the advantage of reducing the problem of adjective ordering to the problem of estimating *n*-gram probabilities, something which is relatively well understood.

To test the effectiveness of this strategy, we took as a dataset the first one million sentences of the written portion of the British National Corpus (Burnard, 1995).[1] We held out a randomly selected 10% of this dataset and constructed a backoff bigram model from the remaining 90% using the CMU-Cambridge statistical language modeling toolkit (Clarkson and Rosenfeld, 1997). We then evaluated the model by extracting all sequences of two or more adjectives followed by a noun from the held-out test data and counted the number of such sequences for which the most likely order was the actually observed order. Note that while the model was constructed using the entire training set, it was evaluated based on only sequences of adjectives.

The results of this experiment were somewhat disappointing. Of 5,113 adjective sequences found in the test data, the order was correctly predicted for only 3,864 for an overall prediction accuracy of 75.57%. The apparent reason that this method performs as poorly as it does for this particular problem is that sequences of adjectives are relatively rare in written English. This is evidenced by the fact that in the test data only one sequence of adjectives was found for every twenty sentences. With adjective sequences so rare, the chances of finding information about any *particular* sequence of adjectives is extremely small. The data is simply too sparse for this to be a reliable method.

[1] The relevant files were identified by the absence of the `<settDesc>` (spoken text "setting description") SGML tag in the file header. Thanks to John Carroll for help in preparing the corpus.

3 The experiments

Since Langkilde and Knight's general approach does not seem to be very effective in this particular case, we instead chose to pursue more focused solutions to the problem of generating correctly ordered sequences of prenominal adjectives. In addition, at least one generation algorithm (Carroll et al., 1999) inserts adjectival modifiers in a post-processing step. This makes it easy to integrate a distinct adjective-ordering module with the rest of the generation system.

3.1 The data

To evaluate various methods for ordering prenominal adjectives, we first constructed a dataset by taking all sequences of two or more adjectives followed by a common noun in the 100 million tokens of written English in the British National Corpus. From 247,032 sequences, we produced 262,838 individual pairs of adjectives. Among these pairs, there were 127,016 different pair types, and 23,941 different adjective types. For test purposes, we then randomly held out 10% of the pairs, and used the remaining 90% as the training sample.

Before we look at the different methods for predicting the order of adjective pairs, there are two properties of this dataset which bear noting. First, it is quite sparse. More than 76% of the adjective pair types occur only once, and 49% of the adjective types only occur once. Second, we get no useful information about the syntagmatic context in which a pair appears. The left-hand context is almost always a determiner, and including information about the modified head noun would only make the data even sparser. This lack of context makes this problem different from other problems, such as part-of-speech tagging and grapheme-to-phoneme conversion, for which statistical and machine learning solutions have been proposed.

3.2 Direct evidence

The simplest strategy for ordering adjectives is what Shaw and Hatzivassiloglou (1999) call the *direct evidence* method. To order the pair $\{a,b\}$, count how many times the ordered sequences $\langle a,b\rangle$ and $\langle b,a\rangle$ appear in the training data and output the pair in the order which occurred more often.

This method has the advantage of being conceptually very simple, easy to implement, and highly accurate for pairs of adjectives which actually appear in the training data. Applying this method to the adjectives sequences taken from the BNC yields better than 98% accuracy for pairs that occurred in the training data. However, since as we have seen, the majority of pairs occur only once, the overall accuracy of this method is 59.72%, only slightly better than random guessing. Fortunately, another strength of this method is that it is easy to identify those pairs for which it is likely to give the right result. This means that one can fall back on another less accurate but more general method for pairs which did not occur in the training data. In particular, if we randomly assign an order to unseen pairs, we can cut the error rate in half and raise the overall accuracy to 78.28%.

It should be noted that the direct evidence method as employed here is slightly different from Shaw and Hatzivassiloglou's: we simply compare raw token counts and take the larger value, while they applied a significance test to estimate the probability that a difference between counts arose strictly by chance. Like one finds in a trade-off between precision and recall, the use of a significance test slightly improved the accuracy of the method for those pairs which it had an opinion about, but also increased the number of pairs which had to be randomly assigned an order. As a result, the net impact of using a significance test for the BNC data was a very slight decrease in the overall prediction accuracy.

The direct evidence method is straightforward to implement and gives impressive results for applications that involve a small number of frequent adjectives which occur in all relevant combinations in the training data. However, as a general approach to ordering adjectives, it leaves quite a bit to be desired. In order to overcome the sparseness inherent to this kind of data, we need a method which can generalize from the pairs which occur in the training data to unseen pairs.

3.3 Transitivity

One way to think of the direct evidence method is to see that it defines a relation $\prec$ on the set of English adjectives. Given two adjectives, if the ordered pair $\langle a,b\rangle$ appears in the training data more often then the pair $\langle b,a\rangle$, then $a \prec b$. If the reverse is true, and $\langle b,a\rangle$ is found more often than $\langle a,b\rangle$, then $b \prec a$. If neither order appears in the training data, then neither $a \prec b$ nor $b \prec a$ and an order must be randomly assigned.

Shaw and Hatzivassiloglou (1999) propose to generalize the direct evidence method so that it can apply to unseen pairs of adjectives by computing the *transitive closure* of the ordering relation $\prec$. That is, if $a \prec c$ and $c \prec b$, we can conclude that $a \prec b$. To take an example from the BNC, the adjectives *large* and *green* never occur together in the training data, and so would be assigned a random order by the direct evidence method. However, the pairs $\langle large, new\rangle$ and $\langle new, green\rangle$ occur fairly frequently. Therefore, in the face of this evidence we can assign this pair the order $\langle large, green\rangle$, which not coincidently is the correct English word order.

The difficulty with applying the transitive closure method to any large dataset is that there often will be evidence for both orders of any given pair. For instance, alongside the evidence supporting the order $\langle large, green\rangle$, we also find the pairs $\langle green, byzantine\rangle$, $\langle byzantine, decorative\rangle$, and $\langle decorative, new\rangle$, which suggest the order $\langle green, large\rangle$.

Intuitively, the evidence for the first order is quite a bit stronger than the evidence for the second. The first ordered pairs are more frequent, as are the individual adjectives involved. To quantify the relative strengths of these transitive inferences, Shaw and Hatzivassiloglou (1999) propose to assign a weight to each link. Say the order $\langle a,b\rangle$ occurs m times and the pair $\{a,b\}$ occurs n times in total. Then the weight of the pair $a \rightarrow b$ is:

$$-\log\left(1-\sum_{k=m}^{n}\binom{n}{k}\cdot\frac{1}{2}^{n}\right)$$

This weight decreases as the probability that the observed order did not occur strictly by chance increases. This way, the problem of finding the order best supported by the evidence can be stated as a general shortest path problem: to find the preferred order for $\{a,b\}$, find the sum of the weights of the pairs in the lowest-weighted path from a to b and from b to a and choose whichever is lower.

Using this method, Shaw and Hatzivassiloglou report predictions ranging from 81% to 95% accuracy on small, domain specific samples. However, they note that the results are very domain-

specific. Applying a graph trained on one domain to a text from another another generally gives very poor results, ranging from 54% to 58% accuracy. Applying this method to the BNC data gives 83.91% accuracy, in line with Shaw and Hatzivassiloglou's results and considerably better than the direct evidence method. However, applying the method is computationally a bit expensive. Like the direct evidence method, it requires storing every pair of adjectives found in the training data along with its frequency. In addition, it also requires solving the all-pairs shortest path problem, for which common algorithms run in $O(n^3)$ time.

3.4 Adjective bigrams

Another way to look at the direct evidence method is as a comparison between two probabilities. Given an adjective pair $\{a,b\}$, we compare the number of times we observed the order $\langle a,b\rangle$ to the number of times we observed the order $\langle b,a\rangle$. Dividing each of these counts by the total number of times $\{a,b\}$ occurred gives us the maximum likelihood estimate of the probabilities $P(\langle a,b\rangle|\{a,b\})$ and $P(\langle b,a\rangle|\{a,b\})$.

Looking at it this way, it should be clear why the direct evidence method does not work well, as maximum likelihood estimation of bigram probabilities is well known to fail in the face of sparse data. It should also be clear how we might improve the direct evidence method. Using the same strategy as described in section 2, we constructed a back-off bigram model of adjective pairs, again using the CMU-Cambridge toolkit. Since this model was constructed using only data specifically about adjective sequences, the relative infrequency of such sequences does not degrade its performance. Therefore, while the word bigram model gave an accuracy of only 75.57%, the adjective bigram model yields an overall prediction accuracy of 88.02% for the BNC data.

3.5 Memory-based learning

An important property of the direct evidence method for ordering adjectives is that it requires storing all of the adjective pairs observed in the training data. In this respect, the direct evidence method can be thought of as a kind of memory-based learning.

Memory-based (also known as lazy, nearest neighbor, instance-based, or case-based) approaches to classification work by storing all of the instances in the training data, along with their classes. To classify a new instance, the store of previously seen instances is searched to find those instances which most resemble the new instance with respect to some similarity metric. The new instance is then assigned a class based on the majority class of its nearest neighbors in the space of previously seen instances.

To make the comparison between the direct evidence method and memory-based learning clearer, we can frame the problem of adjective ordering as a classification problem. Given an unordered pair $\{a,b\}$, we can assign it some canonical order to get an instance ab. Then, if a precedes b more often than b precedes a in the training data, we assign the instance ab to the class $a \prec b$. Otherwise, we assign it to the class $b \prec a$.

Seen as a solution to a classification problem, the direct evidence method then is an application of memory-based learning where the chosen similarity metric is strict identity. As with the interpretation of the direct evidence method explored in the previous section, this view both reveals a reason why the method is not very effective and also indicates a direction which can be taken to improve it. By requiring the new instance to be identical to a previously seen instance in order to classify it, the direct evidence method is unable to generalize from seen pairs to unseen pairs. Therefore, to improve the method, we need a more appropriate similarity metric that allows the classifier to get information from previously seen pairs which are relevant to but not identical to new unseen pairs.

Following the conventional linguistic wisdom (Quirk et al., 1985, e.g.), this similarity metric should pick out adjectives which belong to the same semantic class. Unfortunately, for many adjectives this information is difficult or impossible to come by. Machine readable dictionaries and lexical databases such as WordNet (Fellbaum, 1998) do provide some information about semantic classes. However, the semantic classification in a lexical database may not make exactly the distinctions required for predicting adjective order. More seriously, available lexical databases are by necessity limited to a relatively small number of words, of which a relatively small fraction are adjectives. In practice, the available sources of semantic information only provide semantic classifications for fairly common adjectives, and

these are precisely the adjectives which are found frequently in the training data and so for which semantic information is least necessary.

While we do not reliably have access to the meaning of an adjective, we do always have access to its form. And, fortunately, for many of the cases in which the direct evidence method fails, finding a previously seen pair of adjectives with a similar form has the effect of finding a pair with a similar meaning. For example, suppose we want to order the adjective pair $\{$*21-year-old*, *Armenian*$\}$. If this pair appears in the training data, then the previous occurrences of this pair will be used to predict the order and the method reduces to direct evidence. If, on the other hand, that particular pair did not appear in the training data, we can base the classification on previously seen pairs with a similar form. In this way, we may find pairs like $\{$*73-year-old*, *Colombian*$\}$ and $\{$*44-year-old*, *Norwegian*$\}$, which have more or less the same distribution as the target pair.

To test the effectiveness of a form-based similarity metric, we encoded each adjective pair *ab* as a vector of 16 features (the last 8 characters of *a* and the last 8 characters of *b*) and a class $a \prec b$ or $b \prec a$. Constructing the instance base and testing the classification was performed using the TiMBL 3.0 (Daelemans et al., 2000) memory-based learning system. Instances to be classified were compared to previously seen instances by counting the number of feature values that the two instances had in common.

In computing the similarity score, features were weighted by their information gain, an information theoretic measure of the relevance of a feature for determining the correct classification (Quinlan, 1986; Daelemans and van den Bosch, 1992). This weighting reduces the sensitivity of memory based learning to the presence of irrelevant features.

Given the probability p_i of finding each class i in the instance base D, we can compute the entropy $H(D)$, a measure of the amount of uncertainty in D:

$$H(D) = -\sum_{p_i} p_i \log_2 p_i$$

In the case of the adjective ordering data, there are two classes $a \prec b$ and $b \prec a$, each of which occurs with a probability of roughly 0.5, so the entropy of the instance base is close to 1 bit. We can also compute the entropy of a feature f which takes values V as the weighted sum of the entropy of each of the values V:

$$H(D_f) = \sum_{v_i \in V} H(D_{f=v_i}) \frac{|D_{f=v_i}|}{|D|}$$

Here $H(D_{f=v_i})$ is the entropy of subset of the instance base which has value v_i for feature f. The information gain of a feature then is simply the difference between the total entropy of the instance base and the entropy of a single feature:

$$G(D, f) = H(D) - H(D_f)$$

The information gain $G(D, f)$ is the reduction in uncertainty in D we expect to achieve by learning the value of the feature f. In other words, knowing the value of a feature with a higher G gets us closer on average to knowing the class of an instance than knowing the value of a feature with a lower G does.

The similarity Δ between two instances then is the number of feature values they have in common, weighted by the information gain:

$$\Delta(X, Y) = \sum_{i=1}^{n} G(D, i)\, \delta(x_i, y_i)$$

where:

$$\delta(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases}$$

Classification was based on the five training instances most similar to the instance to be classified, and produced an overall prediction accuracy of 89.34% for the BNC data.

3.6 Positional probabilities

One difficulty faced by each of the methods described so far is that they all to one degree or another depend on finding particular pairs of adjectives. For example, in order for the direct evidence method to assign an order to a pair of adjectives like $\{$*blue*, *large*$\}$, this specific pair must have appeared in the training data. If not, an order will have to be assigned randomly, even if the individual adjectives *blue* and *large* appear quite frequently in combination with a wide variety of other adjectives. Both the adjective bigram method and the memory-based learning method

reduce this dependency on pairs to a certain extent, but these methods still suffer from the fact that even for common adjectives one is much less likely to find a specific pair in the training data than to find some pair of which a specific adjective is a member.

Recall that the adjective bigram method depended on estimating the probabilities $P(\langle a,b\rangle|\{a,b\})$ and $P(\langle b,a\rangle|\{a,b\})$. Suppose we now assume that the probability of a particular adjective appearing first in a sequence depends only on that adjective, and not the the other adjectives in the sequence. We can easily estimate the probability that if an adjective pair includes some given adjective a, then that adjective occurs first (let us call that $P(\langle a,x\rangle|\{a,x\})$) by looking at each pair in the training data that includes that adjective a. Then, given the assumption of independence, the probability $P(\langle a,b\rangle|\{a,b\})$ is simply the product of $P(\langle a,x\rangle|\{a,x\})$ and $P(\langle x,b\rangle|\{b,x\})$. Taking the most likely order for a pair of adjectives using this alternative method for estimating $P(\langle a,b\rangle|\{a,b\})$ and $P(\langle a,b\rangle|\{a,b\})$ gives quite good results: a prediction accuracy of 89.73% for the BNC data.

At first glance, the effectiveness of this method may be surprising since it is based on an independence assumption which common sense indicates must not be true. However, to order a pair of adjectives, this method brings to bear information from all the previously seen pairs which include either of adjectives in the pair in question. Since it makes much more effective use of the training data, it can nevertheless achieve high accuracy. This method also has the advantage of being computationally quite simple. Applying this method requires only one easy-to-calculate value be stored for each possible adjective. Compared to the other methods, which require at a minimum that all of the training data be available during classification, this represents a considerable resource savings.

3.7 Combined method

The two highest scoring methods, using memory-based learning and positional probability, perform similarly, and from the point of view of accuracy there is little to recommend one method over the other. However, it is interesting to note that the errors made by the two methods do not completely overlap: while either of the methods gives the right answer for about 89% of the test data, one of the two is right 95.00% of the time. This indicates that a method which combined the information used by the memory-based learning and positional probability methods ought to be able to perform better than either one individually.

To test this possibility, we added two new features to the representation described in section 3.5. Besides information about the morphological form of the adjectives in the pair, we also included the positional probabilities $P(\langle a,x\rangle|\{a,x\})$ and $P(\langle b,x\rangle|\{b,x\})$ as real-valued features. For numeric features, the similarity metric Δ is computed using the scaled difference between the values:

$$\delta(x_i,y_i) = \frac{x_i - y_i}{max_i - min_i}$$

Repeating the MBL experiment with these two additional features yields 91.85% accuracy for the BNC data, a 24% reduction in error rate over purely morphological MBL with only a modest increase in resource requirements.

4 Future directions

To get an idea of what the upper bound on accuracy is for this task, we tried applying the direct evidence method trained on both the training data and the held-out test data. This gave an accuracy of approximately 99%, which means that 1% of the pairs in the corpus are in the 'wrong' order. For an even larger percentage of pairs either order is acceptable, so an evaluation procedure which assumes that the observed order is the only correct order will underestimate the classification accuracy. Native speaker intuitions about infrequently-occurring adjectives are not very strong, so it is difficult to estimate what fraction of adjective pairs in the corpus are actually unordered. However, it should be clear that even a perfect method for ordering adjectives would score well below 100% given the experimental set-up described here.

While the combined MBL method achieves reasonably good results even given the limitations of the evaluation method, there is still clearly room for improvement. Future work will pursue at least two directions for improving the results. First, while semantic information is not available for all adjectives, it is clearly available for some. Furthermore, any realistic dialog system would make use of some limited vocabulary

Direct evidence	78.28%
Adjective bigrams	88.02%
MBL (morphological)	89.34% (*)
Positional probabilities	89.73% (*)
MBL (combined)	91.85%

Table 1: Summary of results. With the exception of the starred values, all differences are statistically significant ($p < 0.005$)

for which semantic information would be available. More generally, distributional clustering techniques (Schütze, 1992; Pereira et al., 1993) could be applied to extract semantic classes from the corpus itself. Since the constraints on adjective ordering in English depend largely on semantic classes, the addition of semantic information to the model ought to improve the results.

The second area where the methods described here could be improved is in the way that multiple information sources are integrated. The technique method described in section 3.7 is a fairly crude method for combining frequency information with symbolic data. It would be worthwhile to investigate applying some of the more sophisticated ensemble learning techniques which have been proposed in the literature (Dietterich, 1997). In particular, boosting (Schapire, 1999; Abney et al., 1999) offers the possibility of achieving high accuracy from a collection of classifiers which individually perform quite poorly.

5 Conclusion

In this paper, we have presented the results of applying a number of statistical and machine learning techniques to the problem of predicting the order of prenominal adjectives in English. The scores for each of the methods are summarized in table 1. The best methods yield around 90% accuracy, better than the best previously published methods when applied to the broad domain data of the British National Corpus. Note that McNemar's test (Dietterich, 1998) confirms the significance of all of the differences reflected here (with $p < 0.005$) with the exception of the difference between purely morphological MBL and the method based on positional probabilities.

From this investigation, we can draw some additional conclusions. First, a solution specific to adjective ordering works better than a general probabilistic filter. Second, machine learning techniques can be applied to a different kind of linguistic problem with some success, even in the absence of syntagmatic context, and can be used to augment a hand-built competence grammar. Third, in some cases statistical and memory based learning techniques can be combined in a way that performs better than either individually.

6 Acknowledgments

I am indebted to Carol Bleyle, John Carroll, Ann Copestake, Guido Minnen, Miles Osborne, audiences at the University of Groningen and the University of Sussex, and three anonymous reviewers for their comments and suggestions. The work described here was supported by the School of Behavioral and Cognitive Neurosciences at the University of Groningen.

References

Steven Abney, Robert E. Schapire, and Yoram Singer. 1999. Boosting applied to tagging and PP attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Lou Burnard. 1995. Users reference guide for the British National Corpus, version 1.0. Technical report, Oxford University Computing Services.

John Carroll, Ann Copestake, Dan Flickinger, and Victor Poznanski. 1999. An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*, pages 86–95, Toulouse.

Philip R. Clarkson and Ronald Rosenfeld. 1997. Statistical language modeling using the CMU-Cambridge Toolkit. In G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors, *Eurospeech '97 Proceedings*, pages 2707–2710.

Walter Daelemans and Antal van den Bosch. 1992. Generalization performance of backpropagation learning on a syllabification task. In M.F.J. Drossaers and A. Nijholt, editors, *Proceedings of TWLT3: Connectionism and Natural Language Processing*, Enschede. University of Twente.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2000. TiMBL: Tilburg memory based learner, version 3.0, reference guide. ILK Technical Report 00-01, Tilburg University. Available from `http://ilk.kub.nl/~ilk/papers/ilk0001.ps.gz`.

Thomas G. Dietterich. 1997. Machine learning research: four current directions. *AI Magazine*, 18:97–136.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Irene Langkilde and Kevin Knight. 1998a. Generation that exploits corpus-based statistical knowledge. In *Proceedings of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 704–710, Montreal.

Irene Langkilde and Kevin Knight. 1998b. The practical value of *n*-grams in generation. In *Proceedings of the International Natural Language Generation Workshop*, Niagara-on-the-Lake, Ontario.

Fernando Pereira, Naftali Tishby, and Lilian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 30th annual meeting of the Association for Computational Linguistics*, pages 183–190.

J. Ross Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1:81–106.

Randolf Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, London.

Robert E. Schapire. 1999. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings of Supercomputing*, pages 787–796, Minneapolis.

James Shaw and Vasileios Hatzivassiloglou. 1999. Ordering among premodifiers. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 135–143, College Park, Maryland.