Language, Information and Computation(PACLIC 11), 1996, 383-392

Fast Statistical Grammar Induction

Wide R. Hogenhout Yuji Matsumoto
Nara Institute of Science and Technology
{marc-h,matsu}@is.aist-nara.ac.jp

Abstract

The statistical induction of context free grammars from bracketed corpora with the Inside Outside Algorithm has often inspired researchers, but the computational complexity has made it impossible to generate a large scale grammar. The method we suggest achieves the same results as earlier research, but at a much smaller expense in computer time. We explain the modifications needed to the algorithm, give results of experiments and compare these to results reported in other literature.

1 Introduction

The availability of large treebanks creates the opportunity to model the structures humans recognize in sentences that appear in everyday natural language. One well known method for modeling such structures is the Inside Outside Algorithm, which was first described by Baker (1979).

The statistical induction of context free grammars has the attractiveness that it does not require any presumptions about the grammar that is being created, other than those that limit the size. However, the major problem with this kind of modeling has always been the computational complexity; the algorithm requires $O(n^3|w|^3)$ of training time per sentence w for a grammar with n nonterminals, per iteration.

The algorithm has been used in two different ways, both of which reduce the computational complexity. First, there is a line of research (Black, Garside, and Leech, 1993; Hogenhout and Matsumoto, 1996) that concentrates on the reestimation of hand written grammars. This has none or much less problems with time complexity since the structure of the grammar is already decided and usually generates a limited number of parses for a given sentence. However, it completely loses the original attractiveness of modeling without presumptions.

A second line of research, which concentrates on inducting a new grammar from scratch, is described in, amongst others, (Pereira and Schabes, 1992; Schabes, Roth, and Osborne, 1993).

In these experiments the algorithm was applied to a treebank. The brackets of the treebank were used to reduce the number of possible structures by disallowing any structure that crosses some treebank bracket. This greatly

reduces training time, and they were able to parse short sentences of the Wall Street Journal corpus with 90.2% accuracy using a 15-nonterminal grammar.

The extension to the algorithm and the experiments we present aims at improving the efficiency of induction of grammars from scratch. We have been able to strongly reduce the time complexity of the training, and at the same time achieved equivalent results when parsing short sentences of the Wall Street Journal Corpus.

Instead of starting with a grammar consisting of all possible rules for a given number of nonterminals, we start with a small number of nonterminals and gradually increase this to the desired number. At the same time we remove rules that have become obsolete so we can work with a much smaller grammar.

In this paper we describe the method in detail and report on the results obtained in preliminary experiments.

2 Inducting Statistical Grammars

The Inside Outside Algorithm makes it possible to start with an unstructered grammar. That means a number of nonterminals (n) and a number of parts of speech (m) are chosen and all possible rules for these symbols are created. Usually Chomsky Normal Form rules are used:

$$X_i \rightarrow X_j X_k \qquad (1 \leq i, j, k \leq n)$$

$$X_i \rightarrow t_j \qquad (1 \leq i \leq n; 1 \leq j \leq m)$$

where t stands for a part of speech. After training it is possible (but not necessary) to discard those rules that obtained a very low probability.

We refer to (Baker, 1979; Lari and Young, 1990) for the details of the Inside Outside Algorithm.

2.1 Induction from Bracketed Corpora

The experiments reported in (Pereira and Schabes, 1992) were limited in the size of the training corpus (770 sentences) and in the number of nonterminals (15). Apart from the limited size of the grammar and the training set, the linguistical simplicity of the corpus that was used also gave reason for doubt.

Schabes, Roth, and Osborne (1993) report on an experiment using the linguistically more complex Wall Street Journal Corpus. This proved that more complex structures can be learned in the same way. Various sizes were tried for the training set, but this had no significant affect on the performance.

In our experiments we used the same number of nonterminals, the same corpus and about the same amount of training data, but we created the grammar in much less time.

2.2 Enforcing Structure in the Grammar

We mention some experiments that aimed at reducing the computational complexity of grammar induction. They are aimed at either giving structure to

the grammar before parsing, reducing the number of rules in the grammar, or reducing the number of rules involved in training.

Fujisaki et al. (1989) describe an experiment where training was only used to find the parameters of a predefined subset of the rules. They trained with an ambiguous corpus of slightly more than 4200 sentences, on average about 11 words long. The resulting grammar was tested on 84 sentences, but comparison with other experiments is rather difficult.

Sharman, Jelinek, and Mercer (1990) describe an experiment where the grammar was in ID/LP format (Immediate Dominance and Linear Precedence), and received initial probabilities from the counts in a treebank. In this way, the grammar already had a strong shape before training started. In contrast to (Fujisaki et al., 1989) they used a treebank to do the training and they also used longer sentences. This grammar was tested on 42 sentences, but here also it is very difficult to compare the results because of the different test sets.

Another experiment with the Inside Outside Algorithm with restrictions on the grammar is described by Briscoe and Waegner (1992). The restrictions they place are similar to those in $\bar{\text{X}}$ theory. Every nonterminal has a number of bars (zero, one or two), and is specified for noun and verb (e.g., a noun is classified +noun and -verb, an adverb is classified +noun and +verb). Every rule must be consistent with some constraints in order to be permitted. Most importantly, the left side nonterminal must be matched by a right side nonterminal that has the same specifications for noun and verb, and has the same number of bars or one less. Briscoe and Waegner also give higher probabilities to what they call explicit rules to give the grammar more structure. Unfortunately, it is hard to evaluate this in terms of performance since they only give results in terms of entropy.

3 The Details of Step-by-Step Induction

The Gradual Induction we describe is based on the intuition that a small grammar can gradually be corrected and improved in order to make a bigger one. Imagine a grammar with n nonterminals, which has been trained for a certain number of iterations. We can take one nonterminal from this grammar, removing all rules that contain the nonterminal, and replace it with two new ones. In a sense we split up a nonterminal into two, thus creating a grammar with $n+1$ nonterminals.

More formally, we take the following steps.

1. Select a nonterminal X_q that is being used in the grammar
2. Remove all rules of the form $X_q \rightarrow X_j X_k$, $X_q \rightarrow t_j$, $X_j \rightarrow X_q X_k$, $X_k \rightarrow X_j X_q$, for every j, k.
3. Create all rules possible with the two nonterminals X_q and $X_{q'}$. In other words, all rules of the form $X_i \rightarrow X_j X_k$, where either i, j or k is X_q

or $X_{q'}$. This includes multiple occurrences, so for example $X_q \rightarrow X_q X_q$ and $X_q \rightarrow X_{q'} X_q$ are also created.

4. Also create the rules $X_q \rightarrow t_j$ and $X_{q'} \rightarrow t_j$ for every $j \leq m$.

5. Give these new rules randomized probabilities, so that their total probability mass equals that of the nonterminal X_q before it was removed

If no rules are deleted this can be used to create a grammar that is completely equal to the grammar that would be inducted directly from $n + 1$ nonterminals. But we are more interested in the possibility of removing those rules that received a very low probability, thereby keeping a small grammar while the number of nonterminals increases.

The intuition behind this process is that when the number of nonterminals is low (as it unvariably is) one nonterminal will take on various roles (represent various grammatical entities) and this will have a negative effect on the grammar. Separating one nonterminal into two and randomizing the related probabilities allows the algorithm to separate these roles while the rest of the grammar does not undergo major changes.

Selection One question we left open is the selection of a nonterminal: one needs some criterion to decide what nonterminal should be separated. In the experiments we are reporting on, we chose the nonterminal with the highest count as it is given by the Inside Outside Algorithm, in other words we used the nonterminal with the highest frequency in the training corpus.

The motivation is that this nonterminal has the most data available for training. Choosing another nonterminal may result in nonterminals that have too little training data to give meaningful estimates to their rules. However, there are other clues for choosing a nonterminal, and we almost never know if this is the best choice.

What we do know is that in our experiments this always was the best choice for the first separation. We also noticed that selecting a nonterminal with a low count is not very productive. Nevertheless, our criterion for selecting a nonterminal is not optimal.

Amount of Rules If no rules would be removed before separating a nonterminal, the size of the grammar would simply be n^3. Our approach aims at reducing this number. To keep things simple, we allow kn^2 rules where k is what we call the *tolerance factor*. By setting k to some value we can decide how many rules the grammar will contain.

The procedure is as follows:

1. Train the grammar for a certain number of iterations

2. Discard the rules of the form $X_i \rightarrow X_j X_k$ with the lowest probabilities until kn^2 such rules remain

3. Separate one nonterminal, and repeat the process

We been experimenting with various values of k. Most of these experiments showed that the value of k does not have much influence on the results. When n is between 7 and 9 there is some negative affect if k is smaller than 4, so it was set at 4. When n became higher than 9 we discovered no significant decrease in performance when k was as low as 1. Only $k < 1$ gave an inferior performance.

4 The Experiment

For the experiment we used rules in Chomsky Normal Form. Given a number of nonterminals n, an initial grammar is created consisting of the rules as mentioned in section 2.

4.1 Pilot Experiment

We first conducted a pilot experiment to answer the following question: can the entropy of the grammar be kept at the same level when a nonterminal is separated in two, can it improve, or will it deteriorate? This experiment is only meant to study the viability of the approach.

We compared three grammars:

- (a) a grammar with 7 nonterminals (regular procedure, no separations)
- (b) a grammar with 8 nonterminals (regular procedure, no separations)
- (c) a grammar with 7 nonterminals where one nonterminal was separated, turning it into an 8 nonterminal grammar

Figure 1 shows the results in cross entropy of the three grammars. Grammar b has more nonterminals than a, so it takes more time (this is not visible in the figure) to train and achieves a lower cross entropy. During the first 30 iterations grammar a and c are equal, then c suddenly increases enormously at the point where a nonterminal is separated.

This is caused by the los of information of the distribution of one nonterminal. Since the two new nonterminals receive random distributions the cross entropy of the grammar suddenly deteriorates. However, this soon improves and the final entropy is actually lower than grammar b, which had 8 nonterminals from the beginning.

This shows that separating nonterminals can be a good idea. It does however not show that the entropy will always be lower than a grammar that was trained with the same number of nonterminals from the start. In our experience it varies at separations, although it improves at most separations. But that is a conjecture rather than a conclusion, since (due to time limitations) we cannot train grammars with much more nonterminals for comparison.

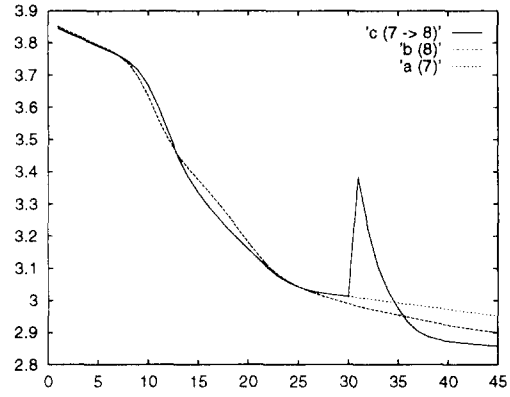

Figure 1: Entropy values for Pilot Experiment.

4.2 Experiment with 15 Nonterminals

We proceeded to perform an experiment with the Wall Street Journal treebank. The value for m (number of parts of speech) was 31 (this is lower than the 47 in the experiment from Schabes, Roth, and Osborne (1993) since some similar low frequency parts of speech were merged.) The training set consisted of 1000 sentences and tests were performed on 100 sentences that were not used in training.

The experiment started with a 7-nonterminal grammar. This was trained for 30 iterations. After this the nonterminal with the highest count was separated into two and the resulting grammar was trained for 15 iterations. This was repeated until the grammar had 15 nonterminals, every time training the grammar for 15 iterations after separating a nonterminal into two.

At first, after training the grammar for 30 or 15 iterations and before separating a nonterminal, the number of rules of the form $X_i \rightarrow X_j X_k$ was reduced to $4n^2$. (When the number of rules is not reduced there would be n^3 such rules.) When the number of nonterminals became 10, the number of rules was further reduced to only n^2 (before every separation). Since we do one more deletion at the end, the final grammar had n^2 rules of the form $X_i \rightarrow X_j X_k$.

The rules that were removed, were simply those with the smallest probability (irrespective of the nonterminals they contained). Theoretically this could result in nonterminals becoming obsolete when their rules are removed, but we never encountered this problem.

The final 15 nonterminal grammar we inducted thus had 225 rules of the type $X_i \rightarrow X_j X_k$. We also fixed the number of rules of the type $X_i \rightarrow t_j$, so that it remained at 7 for every part of speech. The final number of rules was therefore $225 + 7 * 31 = 442$.

5 Results

The result in entropy on test data and training data is indicated in figure 2. The peaks in this figure represent the points where some nonterminal is separated into two. This includes randomizing the probabilities of the rules involved and therefore leads to an upward jump in entropy. But after this severe loss the entropy values become lower than they would have become with less nonterminals almost every time. For example, the 7-nonterminal grammar did not come under an entropy of 3, also not after a running much more iterations than have been indicated here, whereas the final 15 nonterminal grammar came at less then 2.5. This shows that our method gradually improves the grammar, even though a large part of the grammar is discarded before every separation.

Table 1 compares our results with those of the grammar inducted in (Schabes, Roth, and Osborne, 1993), with a grammar that gives a right-branching structure to everything except the final punctuation and a grammar directly abstracted from the corpus. (These figures have been taken over from (Schabes, Roth, and Osborne, 1993)).

Figure 3 shows the accuracies for the test set differentiated by length. Please note that this is cumulative; for example 20 on the x-axis means "shorter than 20 words." This shows that our inducted grammar scores equivalent to the grammar inducted in (Schabes, Roth, and Osborne, 1993).

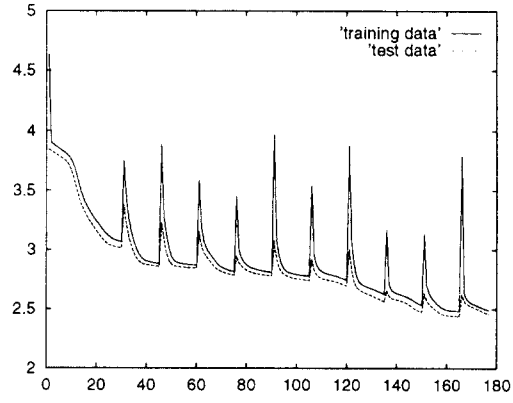

Figure 2: Entropy values for training data and test data against number of iterations.

6 Future Perspectives

After this experiment there are a number of questions left to be answered in the future. First of all, we randomize the probabilities of the new nonterminals after a separation. This temporarily causes an enormous increase in entropy

Table 1: Comparison of Bracketing Accuracies

Length	0-10	0-15	10-19	20-30
Inducted gram.	92.0%	91.7%	83.8%	72.0%
Schabes et. al.	94.4%	90.2%	82.5%	71.5%
Right linear	76%	70%	63%	50%
Treebank gram.	46%	31%	25%	

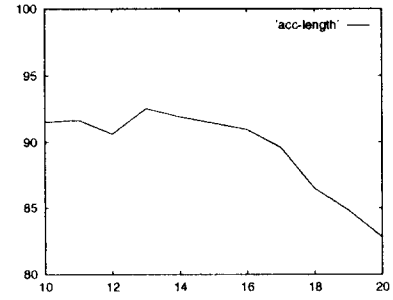

Figure 3: Bracket Accuracy against maximum sentence length.

which is only recovered after training for a few iterations. This is essentially inefficient, since valuable information is being discarded.

An alternative would be to retain most of the original distribution, only introducing a small amount of noise. For example, a new rule receives a probability of 0.8 times the old value, and 0.2 times a random number between 0 and 1. This will reduce the amount of information that is lost during a separation.

One question that remains is the future of statistical grammar induction. While this technique strongly speeds up the process, inducting a large scale grammar is still far from possible. Also, broad coverage parsing is being done with more success elsewhere, see for example (Collins, 1996; Magerman, 1995).

We therefore see this experiment as an experiment in automatically discovering grammatical structures. We also feel grammar induction can play a role in discovering groups of brackets that have similar behavior. Another application can be evaluating a fine grained tag set, since the success of the grammar strongly depends on the word tags.

Although this proposal is limited to simple parts of speech, we feel that for context free grammars to be more mature, they should use headwords, either of categories as in (Hogenhout and Matsumoto, 1996), or of words on the lexical level as in (Collins, 1996; Magerman, 1995).

7 Conclusion

We have briefly discussed some of the existing literature on grammar induction from bracketed corpora and presented an improvement to the Inside Outside Algorithm that makes grammar induction possible in less time. We have presented results of experiments that show this can be used without loss of performance.

It is expected that these results can be improved on in the future, by retaining part of the distribution of the nonterminal that is being separated.

We consider our results a success since we can have shown we construct a grammar with a performance that is equivalent to earlier attempts, but at a much lower cost in terms of computer time. It is not possible to give the time gain exactly, but from the fact that the *size* our grammar grows with a speed of $O(n^2)$ will make this clear.

On the other hand, although this speeds up the process, the possibility of learning a full fledged grammar in this way is still not within reach.

References

Baker, J. K. 1979. Trainable grammars for speech recognition. *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550.

Black, E., R. Garside, and G. Leech. 1993. *Statistically-Driven Computer Grammars of English: The IBM/Lancaster Approach.* Rodopi.

Briscoe, T. and N. Waegner. 1992. Robust stochastic parsing using the inside-outside algorithm. In *Workshop Notes, Statistically-Based NLP Techniques, AAAI*, pages 33–41.

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.

Fujisaki, T., F. Jelinek, J. Cocke, E. Black, and T. Nishino. 1989. A probabilistic method for sentence disambiguation. In *Proceedings of the 1st International Workshop on Parsing Technologies*, pages 105–114.

Hogenhout, W. R. and Y. Matsumoto. 1996. Training stochastical grammars on semantical categories. In Ellen Riloff Stefan Wermter and Gabriele Scheler, editors, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing.* Springer, pages 160–172.

Lari, K. and S. J. Young. 1990. The estimation of stochastic context-free grammars using the Inside-Outside Algorithm. *Computer Speech and Language*, 4:35–56.

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 Y. Schabes. 1992. Inside Outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135.

Schabes, Y., M. Roth, and R. Osborne. 1993. Parsing the wall street journal with the inside-outside algorithm. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 341–347.

Sharman, R., F. Jelinek, and R. Mercer. 1990. Generating a grammar for statistical training. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 267–274.