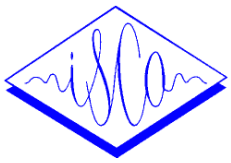

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

COMPUTER ASSISTED GRAMMAR CONSTRUCTION

H-H. Shih *S.J. Young*

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, England

ABSTRACT

This paper presents a system for computer assisted grammar construction (CAGC) and its application in speech processing. The CAGC system is designed to infer linguistically-motivated broad-coverage stochastic context-free grammars (SCFGs) for large corpora, without requiring significant manual contributions. Our approach utilizes an extended inside-outside learning algorithm [1] to train a hybrid SCFG [2] from a bracketed training set. The bracketing information is derived by an automatic surface bracketing system (AUTO) specifically designed for this purpose[3]. Experimental results, evaluated by using Parseval metrics [4], demonstrate that the CAGC system is capable of inferring a grammar from a subset of the Wall Street Journal (WSJ) tagged text corpus and that the inferred grammar achieves high coverage and good precision. As an application, the inferred grammar acts as a language model for rescoring N-best outputs from a speech recognizer [5].

1. INTRODUCTION

Grammatical Inference (GI) from large corpora has recently attracted significant interest in Natural Language Processing research [6] [7] [8] [9]. The aim of GI is to infer or re-estimate a grammar from a set of observations of a language. A successful inferred grammar not only covers these observations but also predicts the unseen members of the language.

Two elements are needed for efficient GI: an initial starting grammar and learning algorithm. The initial grammar can be an empty grammar as in [8] and [10], a hand-crafted grammar as in [11], a grammar containing all possible rules generated over a set of non-terminals and terminals as in [1] or a hybrid grammar utilizing an explicit-implicit technique as in [7].

Several learning algorithms [12] [13] [14] have been proposed in GI. In particular, Baker's inside-outside algorithm has been widely used to infer stochastic grammars of natural language [14]. However, its two inherent problems, high computational complexity and no guarantee of inferring a linguistically-motivated grammar, inhibit its practical application.

In an attempt to overcome these obstacles, Pereira [1] and Black [11] extended the inside-outside algorithm to utilize constituent information from a treebank, a hand-parsed tree corpus, to supervise its training procedure. Their work showed that the extended algorithm significantly reduced the computational requirement as well as improved performance of the inferred grammar. However, they required a manually-parsed treebank to provide constituent information.

In this paper, we present a novel system for computer assisted grammar construction that utilizes a hybrid explicit-implicit rule technique to achieve a high coverage rate, and the extended inside-outside algorithm making use of AUTO generated constituent information to obtain a linguistically-motivated inferred grammar.

2. SYSTEM OVERVIEW

Figure 1 shows an overview of the CAGC system. The first part of the system falls into two stages: construction of an initial hybrid SCFG and phrase-bracketing of the raw text data by the AUTO system. In the second part of the system, the SCFG is inferred from the bracketed text data using the extended inside-outside algorithm.

An explicit-implicit technique was used to generate our hybrid SCFG. The explicit part of the SCFG (core grammar) was manually developed using a grammar development environment tool (GDE) [15] to form a skeleton of the SCFG. The implicit part, generated from four templates [2], consists of all possible rules which do not appear in the core grammar but which are nevertheless linguistically plausible. This is done by filtering all possible rule forms using constraints which enforce structural features such as headedness [7]. In order to bias the learning process towards a linguistically-motivated optimum, the explicit rules are given higher initial probabilities than the implicit rules [3].

The AUTO system was designed to generate phrase bracketing information by converting the raw text data into a constituent-rich training set, The AUTO system utilizes heuristic knowledge to bracket the raw text data using an integrated bottom-up and top-down approach. The bottom-up process creates basic constituents of sentence structures on top of an input tag (or parts-of-speech) sequence and gives a clearer view of the structure of the whole sentence than that is provided by a bare input tag sequence. A top-down procedure starts with the whole sentence S and then finds the key constituents to form the clauses from which a subject and a predicate are bracketed. Full details of the AUTO algorithm are given in [3].

In the second part of the CAGC system, the inside-outside inference procedure, incorporating a bottom-up chart parser [16], iteratively re-estimates the probabilities of the production rules. The updated probabilities are calculated according to the weighted frequency counts of the rules used in parses licensed by the grammar which was generated at the previous iteration. At the end of each iteration, the rules with probabilities falling below a pre-defined (empirically determined) threshold are discarded. This reduces the size of the inferred grammar and the computational expense of further re-estimation.

The re-estimation process continues until either the change in the total log probability (sum of log probabilities of all possible parses generated for all the training data) between iterations is less than a predefined minimum or the number of iterations reaches a maximum. The final inferred grammar is generated when either criteria is satisfied.

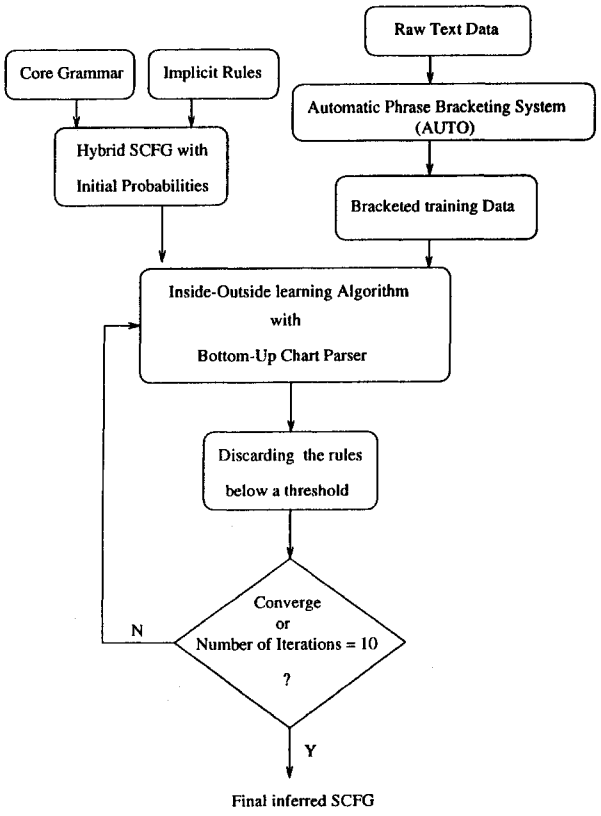

Figure 1. A Block Diagram of the CAGC System

3. EXPERIMENTS AND RESULTS

The training and test data are taken from the WSJ tagged text corpus, however, the number of parts-of-speech has been increased from the original 48 to 62 [3]. In the experiment 3.1. and 3.2. there were 1521 training and 505 testing sentences and none of them contains punctuation. In the experiment 3.3. where punctuation is involved, the training set was expanded to 4112 and the test set to 1505 sentences.

3.1. Explicit/NULL_BC/AUTO_BC

This experiment was designed to investigate the extent to which bracketing information is utilized during the inside-outside re-estimation process and its effects on trained grammars. Three training procedures were carried out: training the explicit grammar from the raw text, training the hybrid grammar from raw text (NULL_BracketingConstraint) and training the hybrid grammar from bracketed text (AUTO_BC). Results for the three trained grammars are shown in Table 1.

Grammar Types	Explicit Grammar	Hybrid Grammars	
		NULL_BC	AUTO_BC
Rules Remain	33.99% 2022/5949	40.91% 6029/14736	18.54% 2733/14736
Coverage	89.80%	98.60%	97.20%
Recall	76.11%	71.35%	85.58%
Precision	58.59%	55.23%	63.43%
Crossings	2.87	3.35	2.00

Table 1. Performance of the Explicit/Hybrid Grammars Trained From Raw/Bracketed Corpus

From Table 1, it can be seen that both the hybrid grammars achieved significantly greater coverage than the explicit grammar. After training, the AUTO_BC grammar has less than half the number of rules of the NULL_BC grammar and its only disadvantage is a slightly reduced coverage. Although the NULL_BC grammar has the best coverage among the three, its performance is the worst with respect to the other metrics. The AUTO_BC grammar maintains considerable improvement in coverage and it has significantly improved performance over the other metrics.

3.2. AUTO_BC/PENN_BC

In this experiment, the hybrid grammar was trained using the corresponding Penn treebank training corpus (PENN_BC). The purpose of this experiment is to compare the effectiveness of the AUTO generated brackets with the Penn treebank brackets. Table 2 shows the performance comparison between these two approaches.

	AUTO_BC	PENN_BC
Rules Remain	18.54%	21.29%
Coverage	97.20%	97.80%
Recall	84.76%	84.76%
Precision	62.67%	63.98%
Crossings	2.13	1.93

Table 2. A Performance Comparison between AUTO_BC and PENN_BC Grammars

As can be seen, the performances of these two grammars are very similar. These results illustrate that although the AUTO system makes errors, they do not seem to degrade the learning process significantly.

3.3. Punctuation

Punctuation often signals non-syntactic relations between constituents. For example, in sentences (a1) and (a2) there is no syntactic relation between the two clauses but a discourse relation of narration or elaboration:

(a1) John pushed Max; he fell.
(a2) Max fell – John has pushed him.

However, punctuation occurs frequently in sentences. In the subset of the WSJ corpus collected in our experiments, 67.8% of the sentences contain at least one punctuation, of which 84% contain only *comma(s)*. The punctuation mark *comma*, unlike others such as a semicolon or dash, usually plays a syntactic role in the sentences. Sentences (b1) and (b2) demonstrate this point:

(b1) Sam has been to Europe, the states and Asia.
(b2) Sam, a manager of the company, went abroad.

The punctuation mark *comma* acts as a conjunct in (b1) and also introduces an apposition of a proper noun in (b2). It is therefore reasonable to seek to include sentences containing *comma* into our grammar construction task. In this experiment, the core grammar was modified to handle this punctuation and the modified hybrid AUTO_BC grammar was then retrained from the expanded training data. Table 3 shows results for the test set of 1505 sentences.

No. of ,	Recall	Precision	Crossings	Parsed/ Total
0	86.05%	63.89%	2.00	500/505
1	82.30%	59.41%	2.86	460/465
2	77.57%	54.57%	4.39	308/309
3	75.06%	51.34%	5.72	152/154
4	67.63%	46.12%	8.50	44/44
5+	67.55%	46.23%	8.82	28/28
0 ~ 5+	81.12%	58.45%	3.46	1492/1505

Table 3. System performance on sentences containing comma(s)

The performance differs according to the number of commas in the test sentences. The figures of the three metrics drop sharply when the number of commas in the sentences increases, especially from 3 commas to 4 commas. We believe that as the number of commas increases, it complicates the structure of the sentences, increases the number of ambiguities and consequently makes the selection of the right parse more difficult.

Although the individual performance degrades as the number of commas increases, the overall performance is still good and acceptable. This is due to the fact that the sentences containing a high number of commas form less than 5% of the whole test set. It is also worth noting that the performance of the AUTO_BC grammar for sentences without punctuation is slightly different to that shown in Table 2. This is because more training data was involved in this experiment and therefore slight improvement was obtained.

4. APPLICATIONS

The potential of the inferred grammar as a language model in speech recognition was investigated by rescoring N-best sentence outputs from the speech recognizer [5]. In this experiment, the grammar merely acted as a language model, as opposed to a trigram model, to reorder the N-best sentences according to their Viterbi parse scores. 136 sentences from the WSJ 20k open vocabulary task were tested and their N-best outputs were ensured to contain the correct sentences.

Because our grammar model works on parts-of-speech, the Acquilex tagger [17] was employed in this work to tag the N-best sentence outputs before applying the grammar. The results are shown in Table 4 in terms of word error rates and ranking of correct sentences.

Models	Word Error	Ranking		
		Top1	Top5	Top10
Acoustic	15.7%	1.5%	14.7%	32.4%
A.+Grammar	13.7%	11.8%	33.8%	50.0%
A.+Trigram	4.9%	60.3%	79.4%	86.8%

Table 4. Performances of the Grammar Model

As can be seen, the grammar model reduced the word error rate by 2%. In the ranking of the correct sentences, the grammar model made a 10% improvement in Top1 and almost 20% in Top5. This figures demonstrate the grammar model is capable of increasing the performance to a certain extent. The trigram model performed much better than the grammar model. However, this was expected because, unlike the trigram model, parts-of-speech are the only source of lexical information used in our grammar model. The following example gives a clearer view of what happened in the grammar rescoring task.

A correct sentence and its parts-of-speech sequence is :

(c1) In/IN a/AT minute/NN the/DT deal/NN is/BEZ closed/VBN.

and three incorrect sentences in the N-best output are:

(c2) In/IN a/AT manner/NN the/DT deal/NN is/BEZ closed/VBN.
(c3) In/IN a/AT return/NN the/DT deal/NN was/BEDZ closed/VBN.
(c4) In/IN a/AT turner/NN a/AT deal/NN is/BEZ closed/VBN down/RP.

In the incorrect sentences, the word *minute* was replaced by *manner*, *return* or *turner* in (c2)(c3)(c4) which was given the same part-of-speech as *minute*, the correct word *is* was substituted by *was* in (c3), the word *the* was replaced by *a* in (c4), and an insertion error *down* occurred in (c4) after the word *closed*. However, all these substitution/insertion errors are perfectly acceptable in terms of their syntactic roles. What is more, the tagger treats each sentence in the N-best output as a correct sentence and therefore an unknown word or a word with more than one possible parts-of-speech will be given a most likely part-of-speech according to its context. As a result, the grammar model is unable to discriminate the correct sentence in the N-best output.

5. CONCLUSION

A system for computer assisted grammar construction has been presented in this paper. The experimental results demonstrate that the CAGC system can successfully infer a grammar which has high coverage and good precision for a subset of the WSJ corpus. Two techniques employed in the system contributed to this success.

Firstly, the method of generating an initial SCFG ensures broad-coverage of the inferred grammar and provides good bootstrapping for the learning process. Secondly, the extended inside-outside algorithm successfully utilizes constituent information derived by the AUTO bracketing system to constrain the training process and establish similar constituent structures in the inferred grammar.

Although not competitive to a trigram model when compared only on the basis of recognition rate, the inferred SCFG did attribute useful parses to the majority of the WSJ utterances and this is viewed as an important step in the longer term goal of speech understanding rather than speech recognition.

Another area where syntax is important is in speech synthesis. We are therefore investigating in collaboration with the Boston University the extent to which the naturalness of synthetic speech can be improved by utilizing syntactic structures in a prosody/syntax model for speech synthesis [18].

ACKNOWLEDGEMENT

We would like to thank Gareth Jones for helpful discussions on this work and also Ted Briscoe for useful contributions and for allowing us access to the Acquilex tagger.

REFERENCES

[1] F. Pereira and Y. Schabes. Inside-outside reestimation for partially bracketed corpora. In *30th Annual Meeting of the ACL*, pages 128–135, June 1992.

[2] N. Waegner. *Stochastic Models for Language Acquisition.* PhD thesis, Cambridge University, England, 1993.

[3] H-H. Shih and S.J. Young. A system for computer assisted grammar construction. Technical Report TR.170, Engineering Department, Cambridge University, England, June 1994.

[4] E. Black, S. Abney, D. Flicknger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammar. In *DARPA Speech and Natural Language Workshop*, pages 306–311, 1991.

[5] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *ARPA Workshop on Human Language Technology*, pages 286–291, 1994.

[6] Y. Schabes, M. Roth, and R. Ostorne. Paring the wall street journal with the inside-outside algorithm. In *European Chapter Meeting of the Association for Computational Linguistics*, 1993.

[7] E. Briscoe and N. Waegner. Undergeneration and robust paring. In J. Arts, P. de Haan, and N. Oostdijk, editors, *English language corpora: design, analysis and exploitation*, pages 14–19. Rodopi, Amsterdam, 1993.

[8] G. Carroll and E. Charniak. Learning probabilistic dependency grammars from labelled text. In *AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language*, pages 25–32, 1992.

[9] T. Fujisaki, F. Jelinek, and E. Black. A probabilistic method for sentence disambiguation. In *1st international workshop on Parsing Teschnologies*, pages 85–94, 1989.

[10] S. Miller and H. Fox. Automatic grammar acquisition. In *ARPA Workshop on Human Language Technology*, pages 256–259, 1994.

[11] E. Black, J. Lafferty, and S. Roukos. Development and evaluation of a broad-coverage probabilistic grammar of English-language computer manuals. In *30th Annual Meeting of the ACL*, pages 185–192, June 1992.

[12] Peter Wyard. Context free grammar induction using genetic algorithm. In *Colloquium on Grammatical Inference: Theories, Applications and Alternatives*, 1993.

[13] H. Rulot, N. Prieto, and E. Vidal. Learning accurate finite-state structural models of words through the ECGI algorithm. In *Proceedings of the ICASSP 89*, volume 2, pages 643–646, 1989.

[14] J.K. Baker. Trainable grammar for speech recognition. In *Speech Communication Papers for the 97th Meeting of the acoustical Society of America (D. Klatt and J. Wolf, eds)*, pages 547–550, 1979.

[15] E. Briscoe, C. Grover, B. Bogurraev, and J. Carroll. A formalism and environment for the development of a large grammar of english. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 703–708, Milan, Italy, 1987.

[16] G. Gazdar and C. Mellish. *Natural Language Processing in PROLOG.* Addison-Wesley, 1988.

[17] D. Elworthly. Part-of-speech tagging and phrasal tagging. Technical Report Acquilex-II Working Paper 10, Computer Laboratory, Cambridge University, England, 1993.

[18] M. Ostendorf and N. Veilleux. A hierarchical stochastic model for automatic prediction of prosodic boundary location. *Computational Linguistics*, 20(1):27–54, 1994.