AN AUTOMATICALLY ACQUIRED CFG FOR SPEECH UNDERSTANDING AND HYPOTHESES REORDERING

Michael Barlow*, Stephanie Dal†, Tatsuo Matsuoka¥ and Sadaoki Furui¥

*School of Computer Science
University of NSW, ADFA

†Ecole Nationale Superieure des Telecommunication

¥Human Interface Laboratories
Nippon Telegraph & Telephone

ABSTRACT - The paper describes the generation and use of a context free grammar as a component in both a speech recognition and speech understanding system. An N-best speech recogniser was run on sentences from the ATIS-2 distribution and the top 25 hypotheses for each were produced. Post-processing grammar models-- bigrams, trigrams, co-occurrence, finite state, and semantic MM were employed to reorder the hypotheses. All showed considerable reduction in sentence error rate. The incorporation of the CFG lead to a further, significant reduction with the best showing more than a halving in original error rate. The speech understanding experiments comprised a finite-state grammar based system for translating class-A sentences into database queries. Incorporation of the CFG dramatically improved translation rate as well as reducing the finite-state grammar's perplexity & complexity.

INTRODUCTION

In the last ten to fifteen years considerable emphasis has been placed upon large vocabulary, speaker independent, continuous speech recognition systems. Even more recently considerable research effort has been focussed upon speech understanding systems. In such cases an integral component of any automatic system is a language or grammar model to both improve the recognition accuracy and form the basis of the understanding component.

Spontaneous speech presents a unique challenge due to its very spontaneous and oft times ungrammatical nature. Conventional rule based grammars predicated on written language have been found to be inadequate and statistically robust techniques such as n-grams have found ascension.

While such techniques as n-grams are adequate for speech recognition, if the additional step of speech understanding is to be undertaken a grammar model that may be employed to extract the meaning of a sentence is required. Clearly there appears scope, and indeed a requirement, for the combination of the "localised" properties of an n-gram grammar with the more "spanning" properties of understanding targeted grammars such as a finite-state grammar.

One compelling approach to this problem is to employ a context-free grammar (CFG) in conjunction with another grammar model. The CFG may be generated automatically from training data and serve to cluster and exploit the "locality" of the spoken data which may then be incorporated into the second grammar. If the process of CFG and subsequent grammar generation is automated the approach may be quickly ported to other tasks without the usual, human-intensive 're-wiring' of the grammar(s).

This paper will describe two sets of experiments conducted using DARPA's ATIS (Air Travel Information Service) (MADCOW, 1992) task, both employing an automatically acquired CFG (AACFG). The first set comprises reordering experiments on a speech recogniser's hypotheses. Five different grammar models (bigram, trigram, co-occurrence, finite state, and semantic markov model) were employed to rescore the hypotheses. Incorporation of the AACFG for all five grammar models witnessed a significant reduction in sentence error rate. The second set of experiments comprise translation of ATIS class-A sentences into an unambiguous query language for the database via the mechanism of a finite-state grammar. Once again incorporation of the AACFG led to a dramatic improvement in the performance of the system.

THE DATA

The ATIS task represents many of the most difficult problems in speech recognition today: continuous, spontaneous speech from a large number of speakers. Many of the sentences contain

such 'artifacts' as restarts, pauses, and corrections (*"I'd like to book a, are there, is there a first-class fare for the flight that departs at 6:40p.m."*), are contextually dependent (*"No, on Thursday."*), or are simply unanswerable (*"What is the payload of an African Swallow?"*).

The task itself comprises answering user's spoken queries from a database of flight and airport details for 35 North American cities. Query of the database is achieved through generation of an SQL query or equivalent, unambiguous query language.

Individual sentences are split into one of three classes dependent upon a system's requirements to answer the question posed. Class-A sentences are self contained queries that require no further information. Class-D sentences require information that the user has specified in a previous sentence(s). Class-X sentences are simply unanswerable.

Table 1 details the breakdown of data from the ATIS-0, 2 and 3 distributions, that was employed for the various components of the experimental systems. In all circumstances there was no overlap between training data and testing data for the same experiment.

TASK	DATA EMPLOYED
CFG generation	9338 sentences from ATIS-0, ATIS-2 and ATIS-3 training sets
bigram, trigram, co-occurence & finite-state training	5307 sentences from ATIS-0, ATIS-2 training sets
semantic MM training	1200 sentences from ATIS-0, ATIS-2 training sets
hypoth. reordering testing	455 sentences from ATIS-2 test set
translator training	1915 class-A sentences from ATIS-2
translator testing	213 class-A sentences from ATIS-2

Table 1. Breakdown of data employed for experimental training and testing.

GENERATION OF THE CFG

Automatic acquisition of the CFG was achieved through a process of grammar inference (McCandless & Glass 1994).

Each iteration of the algorithm sees the merging of two units u_i and u_j on the basis of the divergence between the left and right bigram contextual probabilities for the two units. Units may be words, non-terminals or "phrases"--groups of two or more words and/or non-terminals often found together.

The divergence value for two units *ui* and *uj* is found via the following formulation:

$$\| u_i, u_j \| = d(P_i,P_j) + d(P_j,P_i)$$

$$d(P_i,P_j) = \sum_{c \in \text{Context}} P_i(c) \times \log \frac{P_i(c)}{P_j(c)}$$

$$P_i(c) = P(c \mid u_i)$$

The two units showing the least divergence are merged to form a new non-terminal.

A phrase queue of size n (100 for these experiments) is kept at each iteration by finding those units that occur together most often by using the following formulation:

$$D(u_i,u_j) = -N(u_i,u_j) \times \log \frac{N(u_i,u_j)}{N(u_i) \times N(u_j)}$$

Table 2 shows the state of the grammar after 1, 5 and 8 iterations. In all, one hundred iterations of the algorithm were performed.

Due to the slowness of grammar inference the algorithm was seeded with a small hand-written CFG that included some numbers and dates. It can be seen that the 4'th non-terminal created merged two of these classes (the dates "first"..."ninth" and those "tenth" to "thirty first").

Hypotheses reordering experiments were run with the grammar inferred at each step. While performance generally increased as the grammar grew in size there appeared no definite cut-off based

either on reordering performance or automated measures of the algorithm's parameters. Indeed reordering performance occasionally fluctuated and at times dropped slightly with the introduction of new non-terminals before flattening out from approximately 50-iterations onwards. As such a final stage was added to the grammar inference:- after the algorithm was terminated a human reviewed the non-terminals generated and eliminated any that appeared non-sensical (only a very small number of non-terminals were altered in this way).

ITERATION 1	ITERATION 5	ITERATION 10
NT_0: what is | what's	NT_0: what is | what's | what're	NT_0: what is | what's | what're
	NT_1: petersburg | paul	NT_1: petersburg | paul
	NT_2: kinds | type	NT_2: kinds | type
	NT_3: <dates1> | <dates2>	NT_3: <dates1> | <dates2>
		NT_4: most | least
		NT_5: hi | hello
		NT_6: long | much

Table 2. CFG at iterations 1, 5, and 8 of the grammar inference algorithm.

After the 100 iterations of the inference algorithm and the human post-processing a CFG comprising 46 non-terminals and several hundred units (words and phrases) was derived.

GRAMMAR MODELS

A total of five different stochastic grammar models were employed in the hypotheses reordering experiments and evaluated one against the other, both without and with the incorporation of the AACFG. These grammars were: bigram, trigram, co-occurrence, finite state, and semantic markov model.

Bigrams & Trigrams

Both bigram and trigram grammar models were trained using the available data with back-off smoothing (Jelinek & Mercer, 1980) into lower order n-grams (bigrams, unigrams). Flooring was also implemented to handle data unseen during training.

Co-occurrence

A left-to-right word co-occurrence grammar of the form: $P(w_i \mid w_j,\ j>i)$ was trained on the available data. Right-to-left and full co-occurrence grammar models were also trained and tested, and though their results are not reported here they are very similar to that of the left-to-right variant.

Finite-State Grammar

A stochastic finite state grammar model was built using the ECGI (Error Correcting Grammar Inference) algorithm (Rulot et. al 1989, Prieto & Vidal 1992). A penalty score (probability) of 0.1 was imposed for substitution, deletion, and insertion errors (ie. in those cases where a sentence could not be parsed without an error).

Figure 1 shows a simple finite state grammar built using the ECGI algorithm and 4 training sentences.

Semantic Markov Model

A semantic markov model (Pieraccini et. al. 1991, Barlow et. al., 1995) was trained with a set of 1200 hand-labelled sentences from the ATIS-0 and ATIS-2 distributions. Each word within a sentence was assigned to one of 35 semantic classes. The classes were then equated to markov model states and transition and emission probabilities obtained for a fully ergodic model.

HYPOTHESES REORDERING EXPERIMENTS

A number of hypotheses reordering experiments were conducted and will be summarised here. In brief a speech recogniser was run on a set of sentences and its top twenty-five hypotheses for each sentence were generated. The previously described five grammar models were then used to rescore the hypotheses and a new sentence recognition rate was obtained allowing the contrast of the different grammar models as well as evaluating their performance both with and without the CFG .

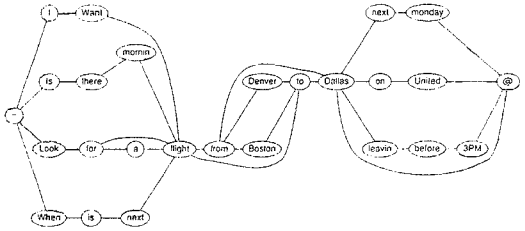

Figure 1. A simple finite-state grammar built with the ECGI algorithm.

The Speech Recogniser

An N-best, tree-trellis based speech recogniser (Chou et. al. 1994) was employed to process the input utterances into a set of hypotheses. The recogniser employs context-dependent inter-word triphone models in a continuous HMM framework, with a bigram word grammar.

Conduct

The N-best speech recogniser was run on a set of 455 sentences taken from the ATIS-2 test set and generated 25 hypotheses for each sentence. The hypotheses were then examined and only those sentences for which at least one acceptable hypothesis existed were retained, leaving a total of 355 sentences. Acceptable sentences were those of equivalent meaning to the original utterances (e.g., the sentences *"Find flights from Pittsburgh to San Francisco"* and *"Find flights Pittsburgh to San Francisco"* would be considered equivalent) while still being grammatically and syntactically correct. This criteria was selected with the end-goal of an understanding system in mind.

The remaining 8875 (355 sentences by 25 hypotheses each) hypotheses were scored by each of the five grammar models in turn and the 25 hypotheses for each sentence reordered on the basis of the combined likelihood scores of both the speech recogniser and the grammar model in question. The sentence was deemed correct if the top-hypothesis of the reordered list was an acceptable one for that sentence. The top hypothesis was found using the following formulation:

$$\arg\max_{i} \quad P_R(h_i) P_G^N(h_i)^w$$

where h_i is the i'th hypothesis, $P_R()$ is the recogniser's likelihood score, $P_G^N()$ is the grammar model's normalised (on basis of number of words) likelihood score, and w is a weighting factor which varies for the different grammars.

In the case where the AACFG was combined with the grammar models both the training and testing (output of the recogniser) data were first parsed using the AACFG. Where matches were found the corresponding non-terminal symbol was substituted for the matching word(s) (e.g., a city name such as "San Francisco" would be replaced with the non-terminal symbol that clustered/matched cities). This modified training and testing data was then used to (respectively) train the grammar models and as input for the testing.

Results

Figure 2 shows the results of the hypotheses reordering experiments for all five grammars in question and both with and without the AACFG incorporated.

Several results are immediately clear from the figure. Firstly, the addition of any post-processing grammar significantly improves the recogniser's performance (a reduction in error rate from 25.6% to a mean of 19.3% without the CFG).

Secondly, and in the case of all 5 grammars, the addition of the CFG led to further, significant reductions in error rate (average case of 19.0% to 16.3%).

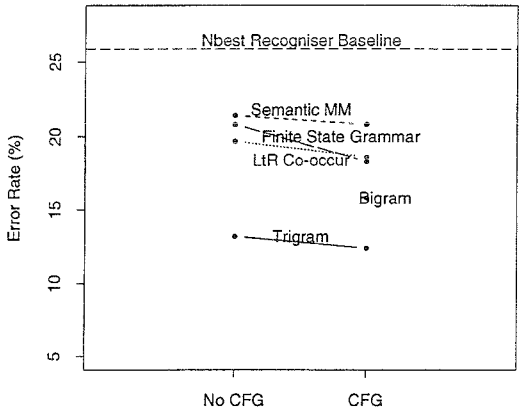

Figure 2. Hypotheses reordering results for the 5 grammar models both with and without the incorporation of the CFG.

Finally it is possible to compare the performances of the five individual grammar models. Clearly trigrams perform significantly better than the other four models. It is worth noting that of the five models, four had exactly the same training data and only the semantic markov model differed in having significantly less due to the requirement that its training data be hand labelled.

SPEECH UNDERSTANDING EXPERIMENTS

A second set of experiments involving a speech understanding task were carried out using the previously described and generated CFG. These experiments comprised the translation of ATIS class-A sentences into WIn (Wizard Input-an unambiguous English language query of the database that is directly and simply translatable to SQL) sentences.

The translation was achieved by building finite-state grammars using the ECGI algorithm for both input sentences and output WIn sentences. Using the paired (input sentence and corresponding WIn query) training data mapping rules between the two grammars were derived so as to achieve the translation of an input sentence. As for the hypotheses reordering trials, experiments were conducted in which the input (training and testing) sentences were unprocessed and where they were pre-processed by the CFG and the results contrasted.

Translation System

As described more fully in (Matsuoka et. al.,1996) a translation system was built using ECGI derived stochastic finite-state grammars of the input spoken language and output WIn sentences with a training set of 1915 sentences.

Translation was achieved via a set of mapping probabilities between arcs of the input grammar and arcs of the output grammar: $P(R_o \mid R_i)$ -- probability that an arc of the output grammar R_o will be taken given an arc of the input grammar R_i was taken.

An input sentence for translation would be parsed by the input grammar and a path through the grammar obtained. Using the mapping probabilities associated with these arcs the output grammar's arc probabilities would be modified (multiplied by the mapping probabilities associated with the input grammar arcs that were traversed). A viterbi based algorithm was then used to find the best path through the modified output grammar. This path defines the output sentence.

Results

Table 3 shows the results of grammar training and translation experiments for 213 test sentences from ATIS-2 when the CFG is excluded versus included (sentences pre-processed by CFG) in the process.

	No CFG	CFG
Number of States	1925	1053
Number of Arcs	5187	2723
Perplexity	14.92	8.47
Translation Rate	33.3%	62.4%

Table 3. Translator performance with and without the CFG incorporated.

It is clear from the table that not only does the CFG greatly reduce the complexity and perplexity of the input grammar but it also significantly improves the translation rate, almost doubling it. Indeed this improvement in translation rate is no doubt due to the reduction in parameters that need estimating in the system and hence leading to a more reliable and statistically robust estimation of the parameters.

CONCLUSIONS

A series of speech recognition and speech understanding experiments were conducted in which the utility of an automatically acquired context-free grammar was evaluated.

The speech recognition experiments comprised a hypotheses reordering task using five stochastic grammars:- bigrams, trigrams, co-occurrence, finite-state and semantic markov model. In all cases the CFG's incorporation led to further, significant reductions in the error rate, with the best system (trigrams employing the CFG) more than halving the recogniser's error rate.

The speech understanding experiments comprised a finite-state grammar based translation system. Incorporation of the CFG in the processing of the input sentences greatly reduced the complexity of the input finite-state grammar and lead to nearly a doubling in the translation rate obtained.

Considerable scope for further work exists. Eliminating the need for human post-processing of the CFG is clearly desirable. The translation system shows much room for improvement and though not reported here experiments have shown that parameters of the ECGI training algorithm play a significant role in this respect.

ACKNOWLEDGMENTS

This work was carried out within Furui Laboratory, Human Interface Laboratories, Nippon Telegraph and Telephone, Japan. We are grateful to Professor Enrique Vidal of Universidad Politecnia de Valencia for providing the ECGI software.

REFERENCES

Barlow M., Matsuoka T. & Furui F. (1995) *Markov Model Reordering of Sentence Hypotheses*, Proc. Acoust. Soc. Japan, Spring Meeting, vol 1, 175-176.

Chou W., Matsuoka T., Juang B.H. & Lee C.H. (1994) *An Algorithm for High Resolution and Efficient Multiple String Hypothesization for Continuous Speech Recognition Using Inter-Word Models"*, Proc. ICASSP-94, vol 2, 153-156.

Jelinek F., & Mercer R.L. (1980) *Interpolated Estimation of Markov Source parameters from Sparse Data*, in Pattern Recognition in Practice, Gelsema & Kanal Eds., North Holland Publishing, 381-397.

MADCOW (1992) *Multisite Data Collection for a Spoken Language Corpus*, Proc. Fifth DARPA Workshop on Speech and Natural Language, Harriman, NY.

Matsuoka T., Hasson R., Barlow M. & Furui S. (1996) *Language Model Acquisition from a Text Corpus for Speech Understanding*, Proc. ICASSP-96.

McCandless, M.K. & Glass J.R. (1994) *Empirical Acquisition of Language Models for Speech Recognition*, Proc. ICSLP-94, 835-838, Yokohama.

Prieto N. & Vidal E. (1992) *Learning Language Models through the ECGI Method*, Speech Communication, vol 11, 299-309.

Rulot H., Prieto N. & Vidal E. (1989) *Learning Accurate Finite-State Structural Models of Words through the ECGI Algorithms*, Proc. ICASSP-89, vol 1, 643-646.