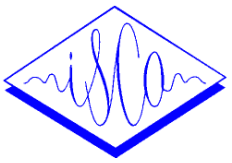

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

Continuous Speech Recognition with Parse Filtering

Ken Hanazawa and Shinsuke Sakai

Computer & Communication Media Research

NEC Corporation

ABSTRACT

We propose “parse-filtering”, a new approach to continuous speech recognition. With it, word sequence hypotheses generated on the basis of N-gram language models are verified by grammar-based parsing during the search for the best-scoring hypothesis, and unparsable hypotheses are filtered out immediately as the search proceeds. Experimental results show that this method yields a higher sentence accuracy than can be achieved with a trigram language model alone. Error reductions are, respectively, 10.0% for word error rate and 12.3% for sentence error rate.

1. INTRODUCTION

To build a high performance speech recognition system, we not only need an accurate acoustic model but also an accurate language model. Two approaches have commonly been employed: 1) that using such statistical language models as bigram and trigram [1], and 2) that using such linguistic knowledge as CFG (context free grammar) in place of statistical language information [2]. In fact, however, both of these conventional approaches have serious drawbacks.

Statistical language models, for example, are limited in their constraining capability. With bigram and trigram, for instance, it is difficult to apply constraints to long word sequences. Further, since a shortage in learning data necessitates smoothing, which sometimes causes improper sequences of words, an accurate solution cannot be guaranteed. The problem with linguistic-knowledge based models is slightly different. When a word sequence hypothesis is generated on the basis of a linguistic knowledge/grammar such as CFG, for example, its grammatical correctness can be guaranteed as long as the grammar used has been correctly described, but when two hypotheses have both been deemed to be grammatically correct, it is difficult to determine which of them is more likely.

A number of approaches have been proposed that combine statistical language models and linguistic knowledge. Harper et al. applies CDG (constraint dependency grammar) to a word graph used in DARPA Resource Management with respect to sentences generated from templates [3], but it is unclear if their approach takes advantage of acoustic and N-gram scores in making decisions on the recognition output. Meteer applies the finite state network representation of a grammar (which appears to be a regular grammar) to air traffic control tasks [4]. Their approach requires a finite-state representation of the grammar to be used for recognition, however, and it seems best suited to small-domain speech understanding. Tsukada uses FSA (finite-state automaton) approximation of CFG to parse the recognized partial segments of an utterance for the purpose of robust speech recognition [5]. Since the FSA parsing is applied to best hypotheses only, however, this approach cannot take advantage of information hidden in lower-rank hypotheses.

Our goal is to apply sentence-level linguistic constraints to the evaluation of hypotheses while taking full advantage of the local measure of reliability that is provided by conventional N-gram language models in a large vocabulary continuous speech recognition system.

2. PARSE-FILTERED SEARCH

2.1. An Overview of Parse Filtering

When a speech recognition system using a traditional N-gram language model produces a grammatically incorrect result, this result may still contain sequences of correct word chains, even though the sentence as a whole is incorrect. In this case, if we are able to look at the lower rank hypotheses, we may often find a correct hypothesis among them. That is to say, it may often be possible to find a correct result if a system can determine, on the basis of such grammatical knowledge as CFG, that the best hypothesis is incorrect. This is the principle of parse-filtering.

For example, in the case illustrated in Figure 1, an incorrect word sequence hypothesis “Is my room have a view?” is selected as the best recognition result of an acoustically similar input utterance “Does my room have a view?”. If the system then determines this best hypothesis to be incorrect on the basis of grammatical knowledge, it can then choose the second-likeliest hypothesis “Does my room have a view?”, which is grammatically correct. In cases in which, rather than word substitution errors, it is the creation of insertion errors by lip smacks or inhales that results in an incorrecthypothesis, a correct hypothesis may also often be found among the lower ranks.

2.2. Speech Recognition System

As an alternative way of implementing parse-filtering for the purpose of verifying word sequence hypotheses that have been generated on the basis of acoustic and N-gram language

models, we use a multi-pass search method that first generates word sequence hypotheses in the form of a word graph and then conducts a parse-filtered best-first search of that graph (see Fig.2).

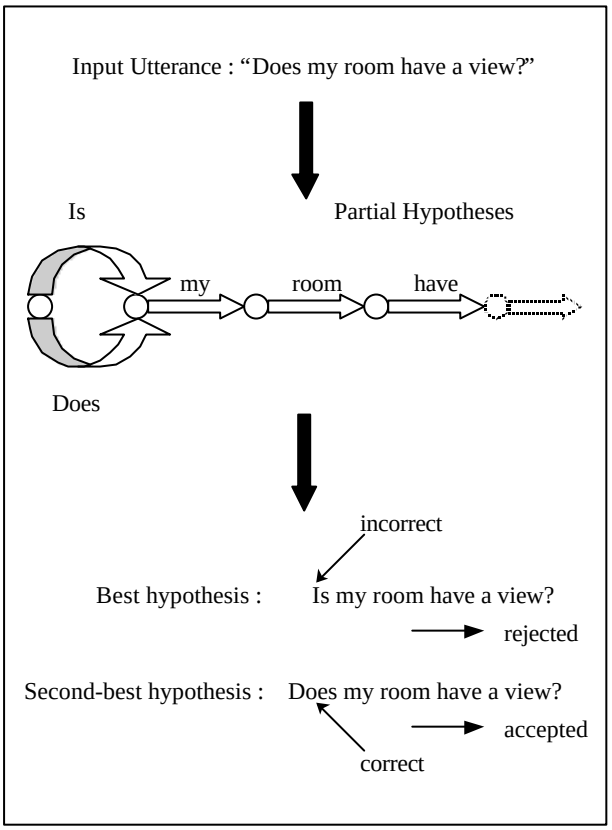

Figure 1: Example of parse-filtering.

Acoustic features are first extracted from input speech and then decoded by means of a Viterbi beam search, employing both an acoustic model and a rough language model (e.g., a bigram language model). Decoding results in the generation of a word graph that contains word sequence hypotheses (first-pass). Next, a rescoring process uses a more elaborate language model, such as trigram language model, to find the likeliest hypothesis from within this word graph, and also uses grammatical knowledge, such as CFG, to try to verify this hypothesis on the basis of parse-filtering (second-pass). In our proposed method, we use a feature-attached CFG as our grammatical knowledge base.

2.3. Parse-Filtered Word Graph Search

In our proposed method, parse-filtering is applied at the second pass rescoring module, which we refer to as the "parse-filtered rescoring module". As shown in Figure 3, a word graph, generated as the result of a first pass is input to the module, which then outputs word sequence hypothesis as the recognition result.

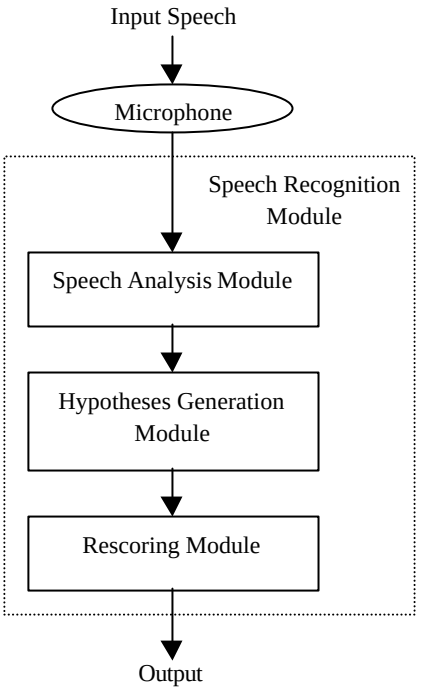

Figure 2: Outline of our speech recognition system.

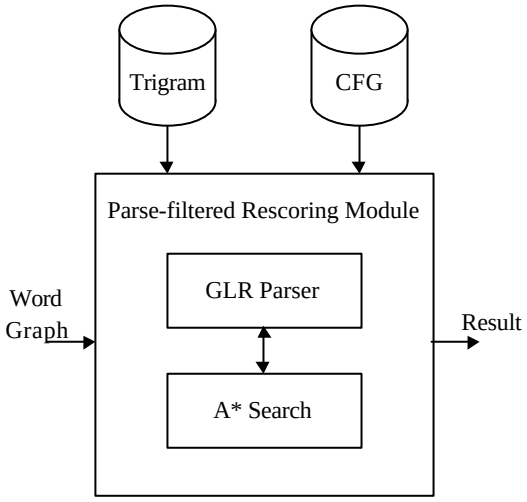

Figure 3: Parse-Filtered rescoring module.

2.3.1. Search Strategy

To search for the best hypothesis, our system employs the A* search algorithm [6]. Specifically, the parse-filtered rescoring process first ranks partial hypotheses within a word graph on the basis of a score obtained for each as the sum of 1) the acoustic score yielded in the first pass and 2) a trigram language score. To each of these "partial hypothesis scores" it adds a "heuristic score", which consists of the same sum as obtained for a section that begins with the end of a partial hypothesis and extends to the end of the word graph. Since

there may be more than one route to the end of the word graph, more than one of these sums may be produced; as the heuristic score to be added, the system chooses the highest among them.

2.3.2. Parse-Filtering

In the parse-filtered rescoring process, a partial sentence hypothesis is partially parsed on the basis of a feature-attached CFG using the LR parsing algorithm [7]. Specifically, since a grammar of natural language has ambiguities, a generalized LR parsing algorithm that uses a graph-structured stack [8] is employed. If the parser determines a partial hypothesis to be grammatically improper, the partial hypothesis is rejected, a second-likeliest partial hypothesis is located, and parsing is again conducted. When it has been extended to the end of the word graph, the result is output by the system as a whole-sentence hypothesis.

When all the hypotheses in the word graph are rejected by the parse-filtering, the system outputs a hypothesis that has the highest sum of an acoustic model score and a trigram language model score.

2.4. Grammar

2.4.1. Framework of Grammar

Our proposed method uses a feature-attached CFG as a grammar. The non-terminals can have any multiple features. Figure 4 shows a few examples of feature-attached CFG rules.

In this framework, a partial hypothesis is parsed based on the following rules.

- During the parsing process of a hypothesis, if a feature-name appears in two or more positions of a grammar rule, the corresponding parts of the partial hypothesis need to have the same feature-value. For example in Figure 4, on the right side of grammar rule (a), if feature N in non-terminal NP corresponding to a part of a partial hypothesis has a feature-value, feature N in non-terminal VP in the same partial hypothesis needs to have the same value. And the feature-value of feature N on the left side of the same rule becomes the same value of feature N on the right side after the reducing action. Using this framework, we can describe constraints such as number agreement in English grammar.
- If a feature-value is specified in a grammar rule, the corresponding part needs to match it. For example in Figure 4, on the right side of grammar rule (a), the feature-value of feature C in non-terminal NP corresponding to a part of a partial hypothesis has to be SBJ.
- Even if a part of a partial hypothesis has a feature-value, it is ignored unless the right side of the corresponding grammar rule has the same feature.
- A part of a partial hypothesis resulting from a reducing action only has features that appear on the left side of the corresponding grammar rule. For example in Figure 4, on the grammar rule (a), feature C will be ignored after the reducing action.

(a) ST(X,T,N,P) -> NP(N,P,C=SBJ,G=NO)
VP(F=Y,T,N,P,X)
(b) NP(N,P,C,W,G=NO) -> NP(N,P,C,W,G=NO)
PP(X=NO)
(c) NC(N=SG,G=NO) -> accessory
(d) V(T=PRES,N=SG,P=3RD,V=I) -> arrives

(Features used in the grammar; F: Finitude. T: Tense and form of a verb. N: Number. P: Person. X: Gap. C: Case. G: Genitive. W: Question form of a noun phrase. V: Verb. A: Form of adjectives and adverbs.)

Figure 4: Example rules of feature-attached CFG. (a) Basic rule of a sentence. (b) Modification by preposition. (c) Vocabulary rule of noun. (d) Vocabulary rule of verb.

2.4.2. Grammar Building

All feature-attached CFG rules are written by hand using a text of 3,000 English sentences. We wrote basic grammar rules, features, and vocabulary rules to cover the sentences. Ten features describe constraints on tense, number, person, etc. Further, we added vocabulary rules that cover evaluation test data for a preliminary experiment. Thus, this CFG is vocabulary-closed to the evaluation test data.

3. EXPERIMENT

3.1. Experimental Conditions

Using the feature-attached CFG described above, we did a preliminary experiment of English continuous speech recognition. Table 1 shows an outline of the experimental conditions, and Table 2 shows the evaluation test data. The CFG covers 99.0% of the evaluation test data (i.e., the CFG can parse 99.0% of the evaluation test sentences successfully).

In our system (Figure 2), the first pass employs both acoustic model and bigram language model scores, and the second pass employs acoustic model and trigram language model scores, and feature-attached CFG for parse-filtering. Bigram and trigram language models are trained by a text corpus of 90,000 sentences that contains 3,000 sentences used in making CFG rules.

Language model	First-pass:	word bigram
	Second-pass:	word trigram
Vocabulary	9000 words	
Grammar	Rules:	8,040
	Number of terminals:	8,204

Table 1: Outline of experimental conditions.

Speakers	10 males and 10 females
Total utterances	3,600
Domain	Travel conversation

Table 2: Evaluation test data. Each speaker uttered 180 sentences in a read speech style.

	Word Acc.	Sentence Acc.
Trigram rescoring	89.0%	69.9%
Parse-filtered rescoring	90.1%	73.6%

Table 3: Experimental result of continuous speech recognition. "Word Acc." means word accuracy, and "Sentence Acc." means sentence accuracy.

3.2. Experimental Results

Table 3 shows the experimental result obtained. For comparison purposes, we also show the performance obtained in using traditional rescoring with a trigram language model alone. The graph error rate of the first pass is 2.1%.

From Table 3, it is clear that the performance of parse-filtered rescoring is better than that of traditional trigram language model rescoring. The improvement in word accuracy is not particularly large, but the improvement in sentence accuracy is rather large. This difference appears to be due to the effect of applying sentence-level linguistic constraints. Error reduction rates are, respectively, 10.0% for word error rate and 12.3% for sentence error rate. In addition, the additional computational cost, i.e., computational time and memory resource, to apply parse-filtering at the rescoring process is small.

Thus, even though the evaluation was vocabulary-closed, our proposed method is basically effective.

4. CONCLUSION

In this paper, we proposed a parse-filtering approach to obtain accurate speech recognition results. The following points distinguish our approach from other linguistic-knowledge-based approaches: 1) An overgenerating grammar is used not to create hypotheses but to verify them. 2) Therefore, the grammar can be a general linguistic grammar with a large vocabulary, and the additional computational cost to apply it is very small. 3) In addition, it takes full advantage of both acoustic model and N-gram language model scores in evaluating hypotheses.

Experimental results show that our proposed method can perform higher sentence accuracy than can be achieved with a trigram language model alone. Since greater improvement was obtained in sentence accuracy than in word accuracy, we can see the effect of applying sentence-level linguistic constraints to a traditional trigram language model based rescoring method.

In future work, we intend to extend the CFG rules to cover a broader domain, and to examine the effectiveness of our proposed method by employing a vocabulary-open evaluation.

REFERENCES

[1] F. Jelinek, "Self-Organized Language Modeling for Speech Recognition", Readings in Speech Recognition, Morgan Kaufmann, pp.450-506, 1990.

[2] H. Ney, "Dynamic programming parsing for context free grammars in continuous speech recognition", IEEE Trans., SP-39, 2, pp.336-340, 1990.

[3] M. P. Harper, et. al, "Interfacing a CDG parser with an HMM word recognizer using word graphs", ICASSP99, 1999.

[4] M. Meteer, and J. R. Rohlicek, "Statistical language modeling combining N-gram and context-free grammars", ICASSP93, II-37, 1993.

[5] H. Tsukada, et. al, "Integration of grammar and statistical language constraints for partial word-sequence recognition", EUROSPEECH97, pp.2759-2762, 1997.

[6] S. J. Russell and P. Norvig, "Artificial Intelligence – A Modern Approach", Prentice-Hall, inc., 1995.

[7] A. Aho, R. Sethi, and J. Ullman, "Compilers: Principles, Techniques, and Tools", Addison-Wesley, 1986.

[8] M. Tomita, "An Efficient Augmented-Context-Free Parsing Algorithm", Computational Linguistics, Vol.13, No.1-2, 1987.