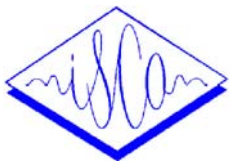

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

A TWO-STAGE SPEECH RECOGNITION METHOD WITH AN ERROR CORRECTION MODEL

Yoshiharu Abe, Hiroyasu Itsui, Yuzo Maruta and Kunio Nakajima

Human Media Technology Dept., Information Technology R&D Center,
Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura, 247-8501 Japan

ABSTRACT

A novel multi-pass speech recognition method is presented. The method is organized as two stages. The first stage decodes the input speech based on an acoustic model and outputs the most probable sequence of basic units. The second stage searches for the most probable word sequence in the decoding output of the first stage. The novel point is use of an error correction model (ECM) in the second stage. With the ECM the second stage can recover decoding errors in the first stage.

The ECM is realized as a statistical model, whose parameters are estimated from training data. The first stage is realized by a one-pass DP algorithm with triphone models. The second stage is realized by a best-first search algorithm with the ECM and a N-gram language model.

The presented method was evaluated with large vocabulary continuous speech recognition. When we used N-best decoding outputs of the first stage and a 64K word trigram language model we achieved the word accuracy of 89.1% for open data with test-set perplexity of 129.

1 INTRODUCTION

Multi-pass search methods are used for large vocabulary continuous speech recognition (for example, [1],[2]). In usual multi-pass search, if candidate for the correct solution is pruned on the way before the final pass, the correct solution cannot be ever recovered. On the contrary, to avoid the pruning of the correct solution we have to use very large search space. To keep the search space relatively small and obtain correct result if the candidate for the correct solution is pruned, we have to use a mechanism to recover the error in the search.

Wakita, et al.[3] proposed a method of recovering recognition error, which uses characteristics of the recognition error for training data. They entered phoneme candidates in the recognizer output. Kurimo[4] proposed a method of obtaining correct phoneme strings, in which they applied a production rule to the recognizer output and used an N-best HMM system for re-scoring.

Our goal is to recognize large vocabulary continuous speech. To find the most probable word sequence with small search space the search is divided into two stages. The first stage searches for the most probable sequence of basic units in the input speech. The second stage searches for the most probable word sequences in the decoding output of the first stage. With an error correction model (ECM) the second stage can recover the decoding errors in the first stage and is possible to yield the correct word sequence, even if the correct candidate is pruned in the first stage. The ECM is realized as a statistical model, whose parameters are estimated from training data.

The decoding in the first stage is based on the one-stage DP algorithm[5]. The decoding outputs of the first stage can be optimal and we can use the context-dependent phone models for intra- and inter-basic units. The word sequence search in the second stage can be realized by a best-first search algorithm with an N-gram language model.

In evaluation, we show experimental results for large-vocabulary continuous speech recognition. As for acoustic features we compare segment statistics[6] with frame statistics with delta components. We also describe result when using N-best decoding output of the first stage.

2 TWO STAGE SEARCH

2.1 Overview

Figure 1 shows a block diagram of one implementation of the two stage search. The first stage searches the most probable sequence of basic units using the acoustic model. The second stage searches the most probable word sequence using the language model. To recover error introduced in the first stage, the second stage refers to the ECM. The role of the ECM is to recover correct word sequence, even if basic units constituting the correct word sequence is dropped in the first stage.

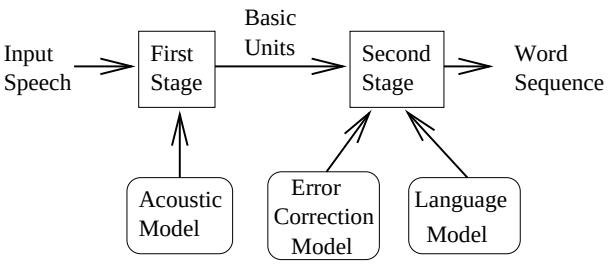

Figure 1: Block diagram

2.2 Basic Unit Recognition

The most probable sequence of basic units is decoded from the input speech using one-stage DP algorithm[5]. We can

P1: $P(bc|z)$
P2: $P(c|z)$
P3: $P(c|yz)$
P4: $P(\phi|z)$
P5: $P(c|\phi)$
P6: $P(abc|z) \quad (z = \text{pau})$

$abc\ldots$: Recognized basic units
$xyz\ldots$: Lexical basic units, ϕ: null

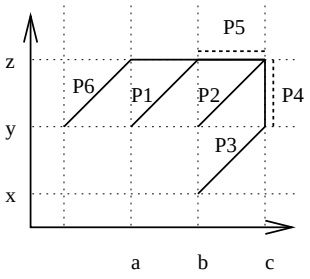

Figure 2: Symmetric ECM

P1: $P(bc|z)$
P2: $P(c|z)$
P4: $P(\phi|z)$
P6: $P(abc|z) \quad (z = \text{pau})$

$abc\ldots$: Recognized basic units
$xyz\ldots$: Lexical basic units, ϕ: null

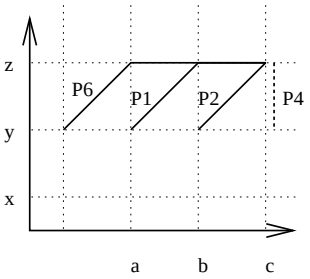

Figure 3: Asymmetric ECM

use syllables or phonemes as basic units. Since the number of the basic units is much smaller than the size of the word lexicon, the search space is very small. Therefore we can use context-dependent phone models for both intra- and inter-basic units. In addition a finite state automaton of basic units can be used to constrain possible sequences of the basic units. And a beam search technique may be used to prune unnecessary candidates.

2.3 Error Correction Model

The ECM is a statistical model. The ECM gives the probability that the recognized sequence of basic units is decoded given the correct sequence of basic units. Let a,b,c and x,y,z be the recognized sequence of basic units and correct sequence of basic units, respectively. The ECM gives a probability $P(abc\ldots|xyz\ldots)$. We use two kinds of ECMs. A symmetric ECM has paths shown in figure 2. The path, P6, is added for absorbing insertions at the beginning and the end of a speech section. An asymmetric ECM has paths shown in figure 3. The probability $P(abc\ldots|xyz\ldots)$ is computed by connecting those paths. If there are multiple paths we take the optimal path.

2.4 Word Sequence Search

The second stage searches for the most probable word sequence that matches to the sequence of basic units decoded by the first stage. Not only a language model but also the ECM is used to recover the correct solution which may drop in the first stage. Let $W_1^k = \{w_1, w_2, \ldots, w_k\}$ be a hypothesized word sequence and $X_1^i = \{x_1, x_2, \ldots, x_i\}$ be the recognized sequence of basic units. The most probable word sequence is found as a word sequence which maximizes a probability

$$P(W_1^k|X_1^i) = P(X_1^i|W_1^k)P(W_1^k)/P(X_1^i) \qquad (1)$$

Since the denominator of the right-hand side is constant, the problem is equivalently rewritten to

$$\text{find } W_1^k \text{ such that } P(X_1^i|W_1^k)P(W_1^k) \rightarrow \max \qquad (2)$$

The term $P(W_1^k)$ represents *a language probability*. It can be computed based on an N-gram language model. The term $P(X_1^i|W_1^k)$ represents how often the word sequence corresponds to the sequence of basic units. In order to compute this term, we introduce a hypothesized sequence of basic units. Let Y_1^m be the hypothesized sequence of basic units. The first term can be rewritten as

$$P(X_1^i|W_1^k) = \sum_{Y_1^m} P(X_1^i|Y_1^m)P(Y_1^m|W_1^k) \qquad (3)$$

where the term $P(X_1^i|Y_1^m)$ represents *a confusion probability*. Since we make the hypothesized sequence of basic units Y_1^m by concatenating the sequences of basic units in the word lexicon according to the hypothesized word sequence, the term $P(Y_1^m|W_1^k)$ is unity and the equation (3) is rewritten as

$$P(X_1^i|W_1^k) = P(X_1^i|Y_1^m) \qquad (4)$$

Therefore the problem (2) is rewritten as

$$\text{find } W_1^k \text{ such that } P(X_1^i|Y_1^m)P(W_1^k) \rightarrow \max \qquad (5)$$

To solve the maximization problem of (5), a best-first search algorithm can be used. The score of hypothesis, a partial sequence of words, is computed as the product of the confusion probability $P(X_1^i|Y_1^m)$ and the language probability $P(W_1^k)$. The most probable word sequence is obtained to repeat a pop-and-push-cycle for the best hypothesis until the word sequence of the hypothesis reaches the end of the input sequence of basic units.

The score of each hypothesis decreases monotonically according as the hypothesis is expanded. Thus the scores of hypotheses in the stack are compared without any normalization of the scores. To reduce computation amount and search space, suboptimal A* heuristic[7] is useful. Because the heuristic is suboptimal we have to find multiple solutions and select the most probable solution. Pruning techniques, such as beam search and end-time spotting, are also useful.

3 EVALUATION

The presented method was evaluated with large-vocabulary continuous speech recognition experiments.

3.1 Acoustic Model

The training data is comprised of total of 30,386 newspaper sentences and ATR phonetic balanced sentences spoken by 96 male and 96 female speakers. Test data is comprised of total of 200 newspaper sentences spoken by 10 male and 10 female speakers. Those sentences and speakers are different from those of training data. All speech data were taken from the ASJ Continuous Speech Corpus (JNAS corpus). Original 16kHz speech samples were down-sampled to 11 kHz and converted to 13 mel-frequency cepstrum coefficients (MFCC13) at every 10ms. Then 9-frame segment statistics were extracted. Dimension of 9-frame segment was reduced to 40 using KL-expansion. We used triphone models as the context-dependent models. There were a total of 25 phones. Each triphone has 3 states. Each state shared 2,000 single state HMMs. The output density was comprised of 16 diagonal Gaussians.

3.2 Syllable Recognition

We used 125 Japanese syllables as the basic units in the first stage. Each syllable is comprised of 1 to 3 phones. There were two parameters in the syllable recognition, namely, *a syllable transition penalty* and *a beam search threshold*. The syllable transition penalty was added to the logarithmic score when the path moved from a syllable to the next one. We investigated appropriate condition for those parameters. Syllable error rates against syllable transition weights are shown in Figure4. The accuracy of syllable recognition was 80.63% in the best condition. The best value of syllable transition penalty was 40. Syllable error rates against the beam search thresholds are shown in Figure5 when the syllable transition penalty was set to 40. The syllable accuracy abruptly became zero with threshold around 40, because the beam search threshold became lower than the syllable transition penalty. In the succeeding experiments we used syllable transition penalty of 40 but we did not use the beam search.

We compared the performance of the 9-frame segment statistics with MFCC13 with delta components. The result is shown in table1.

Table 1: Comparison of Acoustic Features

Acoustic Features	Number of Features	% Syllable Accuracy
MFCC13 + Delta	26	79.24
MFCC13 × 9 frames	40 (KL)	80.63

3.3 Error Correction Model

The ECM were trained using the decoding output of the first stage for 27,204 sentences, a part of the training data, which included the total of 1,138,287 syllables. To estimate confusion probabilities we aligned each decoded output of syllables with the correct sequence of syllables and

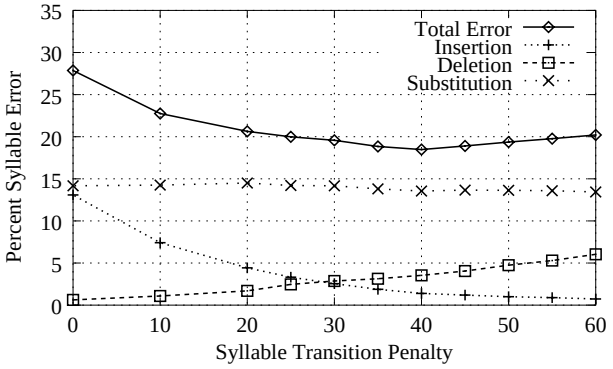

Figure 4: Syllable error rates against syllable transition penalties

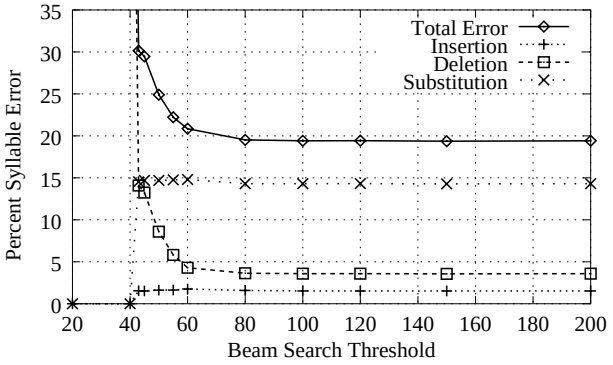

Figure 5: Syllable error rates against beam search thresholds

counted occurrences for each path of ECMs shown in the figures 2 and 3. For example, the confusion probability $P(abc|xyz)$ was computed by

$$P(abc|xyz) = C(abc, xyz)/\sum_{*} C(*, xyz) \qquad (6)$$

where C(.) denotes the count and '$*$' denotes possible syllable combinations.

3.4 Language Model

The lexicon was comprised of 63.8K words. Sentences of the test data included 1.06%(22/2068) of unknown words. Bigram and trigram were estimated from corpus of 56.7M words of newspapers taken from a part of the Mainichi Newspaper CD-ROM 1991–1993. There were 7.98M kinds of bigrams and 19.3M kinds of trigrams. The lower bound of test set perplexities were 172.3 for bigram and 129.4 for trigram.

3.5 Using N-best Decoding

We also investigated usage of N-best decoding outputs of the first stage. The word sequence search was applied to the respective decoding output. The score of word sequence search was re-scored by adding the score of the decoding output. We found that the re-scoring was not effective, that is, zero weight for the decoding output was best in all cases. In the next experiment we used only the score of the second stage.

3.6 Experimental Result

In order to fairly evaluate the performance without being affected by the difference of the words in the lexicon the word accuracy was computed by counting insertion-deletion-substitution of morphemes which were parsed by the Japanese morphological analysis system 'ChaSen' [8]. Test data included total of 1870 morphemes. Table 2 shows comparative results for two types of ECMs when 1-best decoding output was used.

The results for using N-best decoding with the symmetric ECM are shown in table 3. In the table, 'syllables' means the number of candidates to be remained at each syllable node in the syllable automaton and 'sentences' means the number of N-best decoding outputs being fed to the word sequence search.

Table 2: Comparison of two types of ECMs

Type of ECM	Language Model	Test Set Perplexity	% Word Accuracy
Symmetric	Bigram	172.3	85.78
Symmetric	Trigram	129.4	85.88
Asymmetric	Bigram	172.3	84.55
Asymmetric	Trigram	129.4	85.72

Table 3: Use of N-best decoding outputs

Number of N-best		% Word Accuracy	
Syllables	Sentences	LM=3gram	LM=2gram
1	1	84.01	84.12
2	10	86.45	NA
2	20	87.65	NA
2	50	88.56	NA
5	10	87.17	86.31
5	20	87.59	87.33
5	50	89.09	88.24
10	10	87.06	NA
10	20	88.70	NA
10	50	89.04	NA

3.7 Discussion

As shown in table 2 the symmetric ECM is slightly superior to the asymmetric one for both bigram and trigram language models. Comparison of bigram and trigram results there are slight improvement when using trigram language model.

As shown in table 3 use of the N-best decoding outputs was effective. The word accuracy of 1-best case was lower than the result shown in table 2. The reason is as follows. Each spoken sentence was divided into one or more sections, from which N-best decoding outputs were obtained, and the word sequence search was applied to each speech section. As the result, some N-gram, which covers multiple sections, was not used.

We achieved the word accuracy of 89.09% in the best condition. Test data included 1.06% of unknown words for 63.8K word lexicon. The accuracy might be improved if we use a larger lexicon.

4 CONCLUSION

The two stage speech recognition method has been described. The first stage decodes the most probable sequence of basic units. Then the second stage searches the most probable word sequence. The novel point was the ECM introduced in the second stage. In evaluation we have shown that the segmental statistics was better than the frame statistics and the use of N-best decoding outputs of the first stage was effective to improve the word accuracy. Using the 63.8K word trigram language model and N-best decoding outputs we achieved the word accuracy of 89.1%.

ACKNOWLEDGMENTS

In evaluation of the presented method we used the ASJ Continuous Speech Corpus — the Japanese Newspaper Article Sentences(JNAS) —, a part of the Mainichi Newspaper CD-ROM 1991–1993 and the Japanese morphological analysis system *ChaSen*. We would like to acknowledge all institutions and people for allowing use of those corpora and software for research purpose.

REFERENCES

[1] Frank K.Soong, Eng-Fong Huang: "A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition," ICASSP91, pp.705–708(1991).

[2] H.Ney, X.Aubert: "A word graph algorithm for large vocabulary, continuous speech recognition," ICSLP94, pp.1355–1358(1994).

[3] Yumi Wakita, Harald Singer, Yoshinori Sagisaka: "Phoneme candidate re-entry modeling using recognition error characteristics over multiple HMM states (*in Japanese*)," IEICE Transactions D-II, J79-D-II, pp.2086–2095(1996).

[4] Mikko Kurimo: "Improving vocabulary independent HMM decoding results by using dynamically expanding context," ICASSP98, pp.833–836(1998).

[5] H.Ney: "The use of a one-stage dynamic programming algorithm for connected word recognition," IEEE Trans. ASSP-32, 2, pp.263–271(1984).

[6] Seiichi Nakagawa, Kazumasa Yamamoto: "Speech recognition by hidden Markov model using segmental statistics (*in Japanese*)," IEICE Trans. D-II, J79-D-II, pp.2032–2038(1996).

[7] D.B.Paul: "An efficient A* algorithm for continuous speech recognition with a stochastic language model," ICASSP92, pp.I-25–I-28(1992).

[8] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Osamu Imaichi, Tomoaki Imamura: "Japanese morphological analysis system ChaSen manual (*in Japanese*)," Nara Institute of Science and Technology Technical Report, NAIST-IS-TR97007(1997).