CONTINUOUS SPEECH RECOGNITION BY CONTEXT-DEPENDENT PHONETIC HMM AND AN EFFICIENT ALGORITHM FOR FINDING N-BEST SENTENCE HYPOTHESES

ITOU Katunobu, HAYAMIZU Satoru†, and TANAKA Hozumi.

Tokyo Institute of Technology, Meguro-ku, Tokyo 152, JAPAN
†Electrotechnical Laboratory, Tsukuba-shi, Ibaraki 305, JAPAN

Abstract

In this paper, a continuous speech recognition system, "niNja" (Natural language INterface in JApanese), is presented. Efficient search algorithms are proposed to get high accuracy and to reduce the required computations. First, an LR parsing algorithm with context-dependent phone models is proposed. Second, scores of the same phone models in different hypotheses at the phone-level are represented by the single score of the best hypothesis. The system is tested for the task with 113 word vocabulary, word perplexity 4.1. It produces sentence accuracy of 97.3% for the 10 open speakers's 110 sentences and the error reduction is as much as 77% comparing with the case using context independent phone models.

1 INTRODUCTION

In this paper, we describe niNja, the continuous speech recognition system for large vocabulary applications[1, 2]. This system is phone-based and time-synchronous. So, although the system preserves only the hypotheses within a threshold, many hypotheses are generated during processing. We propose two solutions to improve the performance of the system.

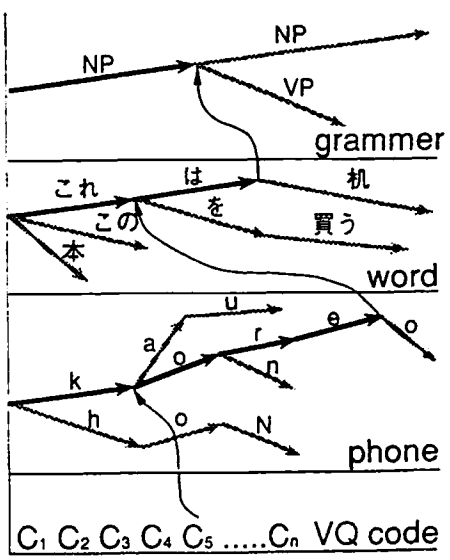
To get high accuracy, the context-dependent phone model is incorporated in many systems [3, 4, 5, 6]. However, for continuous speech applications, what model is connected at word boundaries and how such model is dealt with has not been studied enough. Within a word, we can know the phonetic context in advance by dictionary. However, especially for large vocabulary applications, at word boundaries, it is difficult to know the phonetic context in advance. Therefore, we must consider how to deal with the models according to where they occur. In this paper, we propose a new lexical access algorithm with a dictionary with an LR-table and context-dependent phone models including between-word processing.

To get high efficiency, scores of the same phone models in different hypotheses at the phone-level are represented by the one score of the best hypothesis and the differences. In other words, different hypotheses for one phone model share one best phone model score to save on the computation.

In this paper, we report the results of the two experiments to show the effect of the two proposed methods. In the first experiment, we evaluated the context dependent phone model. In the second experiment, we evaluated how much computation is reduced by the efficient algorithm.

2 SYSTEM OVERVIEW

The system integrates multi-level (acoustic, phonetic, lexical, and linguistic) knowledge sources.



The relations between knowledge sources and how hypotheses are kept in each level are shown as follows. Context-dependent phone hidden Markov Models (HMM) are used to construct phone level hypotheses from vector quantization (VQ) codes. In this level, the only HMMs that are predicted by dictionaries are used. To construct

0-7803-0532-9/92 $3.00

word level hypotheses from phone level hypotheses with dictionaries, *LR-dictionary*, that is the LR-parsing method modified for lexical access, is used. To construct syntactic level hypotheses from word level hypotheses with context free grammar (CFG), the LR-parsing method is used. Each level keeps hypotheses as a tree structure. Tree structures of different levels are linked and a hypothesis in an upper level has corresponding hypotheses in a lower level.

This system constructs and prunes partial sentence hypotheses using all available knowledge time-synchronously. These partial sentence hypotheses are kept as *hypothesis cells*. One hypothesis cell contains a score of the hypothesis and pointers to each level hypothesis. Each pointer represents the location of the hypothesis in the tree structure. The same constraint at each level is evaluated only once, and so the system avoids extra computation.

The architecture of the system is quite similar to HMM-LR[7]. But, we modify the LR-parsing method for lexical access.

3 CONTEXT DEPENDENT PHONE HMM

It has been found that context dependent phone modeling and clustering of context dependent phones produce very good results in speech recognition.

In this system, we use the context dependent phone models clustered by tree-based phone modeling[4]. This modeling is able to predict stochastic characteristics of phone models in unknown phonetic contexts.

To use context dependent phone model for continuous speech recognition, the problem how to determine the phonetic context arises[6]. Within a word, we can know the phonetic context by using a dictionary in advance. However, especially for large vocabulary applications, at word boundaries, it is difficult to know the phonetic context in advance. We propose the new lexical access algorithm to consider these characteristics.

3.1 Determination of the phone contexts within a word

We describe the dictionary entry as CFG with phone symbols as terminals.

```
noun → h o N
noun → z a q sh i
```

First, we translate them according to a phone-context map. A phone-context map contains all triphone and corresponding model as follows in part.

context	phone model
h(i,o)	h3
o(h,N)	o10
a(z,q)	a4
q(a,sh)	q0
sh(q,i)	sh5

So, the dictionary rules are translated as follows.

```
noun → h(*,o) o10 N(o,*)
noun → z(*,a) a4 q0 sh5 i(sh,*)
```

At a word boundary, for example 'h' or 'N' of "h o N", only the previous (left-hand) context or following (right-hand) context is determined. Such phones translate to a *wild-card model*. `h(*,o)` means phone 'h' which has 'o' as the following context and any phones as the previous one. `N(o,*)` means phone 'N' which has 'o' as the previous context.

Then, the LR-dictionary is made from these rules. (It is shown in part.)

state	h(*,o)	z(*,a)	o10	N(o,*)
0	sh1	sh3		
1			sh2	
2				re0

3.2 Handling with the wild-card model

The lexical access method is modified to deal with the wild-card model. We explain the case of the utterance of "koko ni hoN ga ..." by way of example.

3.2.1 For reduce action (at the end of the word)

When phone 'i' of "ni" is processed, the parser makes reduce action with the wild-card model `i(n,*)`. Phone models which are translated from the wild-card model are determined with a phone-context map in advance: `i5`, `i7`, `i13`, `i15` and `i17` as follows. (Each model determines some phones that can follow.) As these models are predicted, the parser goes on processing.

model	following phone
i5	a, a-, e, e-, i, i-, o, o-, u, u-
i7	ch, f, k, ky, p, py, ry, s, sh, t, ts
i13	b, by, d, dy, j, r, w, y, z
i15	N, g, gy, m, my, n, ny
i17	#, h, hy, q, >

3.2.2 At the beginning of the word

When phone 'h' of "hoN" is processed following "koko ni", as before, five hypotheses are constructed. Next, a new lexical access process starts. If the system uses the LR-dictionary that is described above, the initial state has two wild-card models as a lookahead symbol. Here, the model `i15` has phone 'h' as the succeeding phone, and so the hypothesis which has `i15` can make shift action. The hypothesis determines the context of the wild-card model to be `h(i,o)`. The context `h(i,o)` is translated to the model `h2`. Then the model following the model `i15` is `h2`. Also the model `i17` can makes shift action with the wild-card model `z(*,a)` in the same way. The other three model are not able to get the succeeding phone, and so they are rejected at this step.

4 EFFICIENT ALGORITHM

The time-synchronous exact algorithm for finding the N-Best sentence hypotheses requires computation in proportion to a large number of partial hypotheses which are kept during processing[8]. Many partial hypotheses have the same phone. For example, both words "koko" and "asoko" have phone 'k' and 'o'. In case such a phone has the same beginning time, the best scoring hypothesis is computed and the other hypotheses are approximated to have a path that is the same as the best scoring hypothesis. Therefore, this algorithm requires computation as same as 1-best algorithm for finding paths. However, the computation for updating hypotheses is almost linear to the number of the preserved hypotheses.

4.1 Algorithm

Within a phone we use the time-synchronous viterbi search algorithm, with only one theory at each state (of HMM). The initial state of a phone model at each frame has a hypotheses list, which has all hypothesis cells that are going to search the phone model. In the hypotheses list, hypothesis cells record the difference in scores from the best hypothesis except the best scoring hypotheses cell. The best scoring hypotheses cell records score as it is.

The score of the initial state of a phone model is set by the best scoring cell in the hypotheses list. At each final state of the phone (at each frame), the system updates the hypotheses list keeping at the initial state of the path. The score of the path is for the best scoring hypotheses cell. The score of the other cells in the list is calculated to subtract the recorded difference from the score of the path.

Some phone model that is next connected to each hypothesis is determined with the LR-dictionary and (if necessary) the LR-table. The cells are set at the initial state of the next phone model at the frame. So, all phone model have been processed, then new hypotheses lists are constructed at the initial state of each phone model.

4.2 Comparison

There have been other efficient algorithms proposed for finding multiple sentence hypotheses. Lattice N-best algorithm[9] is quite similar to our algorithm, but there are two differences. The first difference is that this algorithm is at the phone-level, while Lattice N-best algorithm is at the word-level. In multiple sentence hypotheses, the number of appearing phones are constant for every task. However, the number of appearing words varies with the difficulty of the task. Therefore, our algorithm have a greater reduction of computation than Lattice N-best algorithm. The second difference is how to preserve partial sentence hypotheses. In Lattice N-best algorithm, partial sentence hypotheses are preserved as a lattice. In our algorithm, partial sentence hypotheses are preserved as hypothesis cells. Our algorithm is suitable to use multi-level constraints as early as possible.

5 EXPERIMENT

The data used here includes 1542 words each by five speakers and 150 sentences each by two speakers. The testset consists of 11 sentences each from ten speakers. The texts and the speakers in the testset are not included in the training data. Therefore the experiments are speaker and word independent recognition.

The frame shift is 5msec and the sampling frequency is 15kHz. A melcepstrum analysis of order 14 is done. A single codebook of melcepstrum, delta-melcepstrum, delta-power is used. The codebook size is 1024.

Phone HMMs are discrete models and have 4 states, 3 loops, and left-to-right structure. They are trained using a forward-backward algorithm. We use 43 (context independent model), 128, 256, 512, and 1024 models.

Allophone clustering algorithm is decision-tree-based and acoustic features of neighboring phones are used in the clustering. The criterion for finding the best split at each node is to maximize the mutual information.

In the first experiment, we evaluated the context dependent phone HMM. For this experiment, we used a dictionary which had 113 words. The perplexity of the grammar is 4.1. The recognition results are shown as follows. As can be seen, the more the number of models increases, the more the recognition rate is improved. At 1024 models, context dependent modeling reduces the error rate by 77% as compared with context independent modeling.

Number of models	43	128	256	512	1024
sentence correct (%)	88.2	90.0	92.7	94.5	97.3
bunsetu accuracy (%)	94.2	96.4	97.6	97.9	99.1

In the second experiment, we evaluated how much computation was reduced by the efficient algorithm. For this experiment, we used a dictionary which had about 500 words. The grammar was that the word can connect to any word which is included in the dictionary, so the grammar has almost no limitation. The results are shown as follows.

	total number of the hypothesis cells	total times of matching phone models
high threshold	881,363	239,285
low threshold	1,764,036	249,376

The total number of the hypothesis cells of the low threshold is twice as many as the high threshold. However, both total times for matching the phone model are about the same. Using the proposed method, we can make the threshold low with almost no extra computation.

6 CONCLUSION

In this paper, we have presented the continuous speech recognition system, niNja. We showed how we integrate multi-level knowledge sources and two methods for improvement of this system performance.

First, we showed the lexical access algorithm with a dictionary by LR table and context-dependent phone models including between-word processing to get high accuracy.

Second, we showed that scores of the same phone model in different hypotheses at the phone-level are represented by one score of the best hypothesis cell and the differences to get high efficiency.

In recognition experiments, we demonstrated that context-dependent modeling reduces the error rate by as much as 77%.

ACKNOWLEDGMENT

The authors sincerely wish to express their thanks to the members of Tanaka Laboratory of TIT and to those of Speech Processing Section of ETL. The continuous speech corpus used here is a part of "Continuous Speech Corpus for Research of Acoustical Society of Japan". Special thanks to those who supported to create the corpus.

References

[1] K. Itou, S. Hayamizu, and H. Tanaka. Continuous speech recognition by the generalized LR parsing. *IEICE Tech. Rep.*, 90(374):49–56, 1990. in Japanese.

[2] K. Itou, S. Hayamizu, and H. Tanaka. Continuous speech recognition by context-dependent phonetic HMM. In *Proc. Fall Meeting ASJ*, pages 41–42. ASJ, 1991. in Japanese.

[3] S. Hayamizu, K. Tanaka, and K Ohta. A large vocabulary word recognition system using rule-based network representation of acoustic characteristic variations. In *Proc. ICASSP-88*, pages 211–214. IEEE, 1988.

[4] S. Hayamizu, K. F. Lee, and H. W. Hon. Description of acoustic variations by tree-based phone modeling. In *Proc. ICSLP-90*, pages 705–708. IEEE, 1990.

[5] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul. Context dependent modeling for acoustic phonetic recognition of continuous speech. In *Proc. ICASSP-85*, pages 1205–1208. IEEE, 1985.

[6] K. F. Lee, H. W. Hon, Mei. Yuh. Hwang, and Sanjoy Mahajan. Recent progress and future outlook of the SPHINX speech recognition system. *Computer Speech and Language*, 4(1):57–69, 1990.

[7] K. Kita, T. Kawabata, and H Saito. HMM continuous speech recognition using predictive LR parsing. In *Proc. ICASSP-89*, pages 703–706. IEEE, 1989.

[8] R. Schwartz and Y. L. Chow. The N-best Algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proc. ICASSP-90*, pages 81–84. IEEE, 1990.

[9] R. Schwartz and S. Austin. A comparison of several approximate algorithms for finding multiple (N-BEST) sentence hypotheses. In *Proc. ICASSP-91*, pages 701–704. IEEE, 1991.