INTERSPEECH 2009 BRIGHTON

Fast Speech Recognition for Voice Destination Entry in a Car Navigation System

Hoon Chung, JeonGue Park, HyeonBae Jeon and YunKeun Lee

Electronics and Telecommunications Research Institute, Daejeon, Korea

{hchung, jgpark, hbjeon, yklee}@etri.re.kr

Abstract

In this paper, we introduce a multi-stage decoding algorithm optimized to recognize very large number of entry names on a resource-limited embedded device. The multi-stage decoding algorithm is composed of a two-stage HMM-based coarse search and a detailed search. The two-stage HMM-based coarse search generates a small set of candidates that are assumed to contain a correct hypothesis with high probability, and the detailed search re-ranks the candidates by rescoring them with sophisticate acoustic models. In this paper, we take experiments with 1-millions of point-of-interest (POI) names on an in-car navigation device with a fixed-point processor running at 620MHz. The experimental result shows that the multi-stage decoding algorithm runs about 2.23 times real-time on the device without serious degradation of recognition performance.

Index Terms: speech recognition, multi-stage decoding

10.21437/Interspeech.2009-290

1. Introduction

One of the rapidly increasing markets of speech recognition is a car. As the most natural human interface, speech can provide a convenient and safe way to control in-car equipments such as a car audio device, a navigation system and so on. However, in order for speech recognition to survive as a practical user interface, it has to overcome several obstacles. Most of all, it has to work robust irrespective of various noise conditions and speaker variations, and it also responses fast on most low performance in-car devices. Among these issues, we discuss fast speech recognition to recognize 1-millions of POI names for voice destination entry (VDE) on a car navigation system. VDE is a feature that enables users to select destination by saying where to drive instead of keying in the address. In Korea, it prefers to select destination by saying POI or landmark names than saying its address. It is a challenging task to develop a speech recognition system that can cover whole POIs because there are about 3.3 millions of POI names in Korea[1]. As a first step to solve this problem, we discuss about fast decoding algorithm in this paper.

In general, fast decoding algorithms are broadly classified into one of two categories. The first category is a method to reduce the complexity of the state output probability computation, and the second is a method to reduce search space. The first category is divided into two subcategories, acoustic space selection and fast Gaussian evaluation. Gaussian selection [2] and GMM selection [3] are included in the acoustic space selection, and sub-vector quantization [4] and sub-space distribution clustering HMM (SDCHMM) [5] are in the fast Gaussian evaluation category. The second category reduces search space through a multi-pass search strategy. The most representative of this category is the fast match [6]. In this approach, computationally inexpensive acoustic models are initially used to produce a reduced search space represented as the N-best hypotheses or a word lattice and then a detailed match using more sophisticated acoustic models re-ranks the hypotheses.

In this paper, we use a multi-stage decoding algorithm to maximize the recognition speed. The algorithm is composed of a two-stage HMM-based coarse search and a detailed search. The algorithm has common architecture with human speech recognition (HSR) in that speech recognition has completed through a three stage decoding procedure: acoustic feature to phone conversion, phone to word conversion and rescoring [7]. The contribution of this work is to present another statistical framework to handle HSR especially from the aspect of fast decoding.

The remainder of this paper is organized as follows: In Section 2, we review the computational complexity of Viterbi decoding algorithm in CDHMM-based speech recognition. In Section 3, we give a brief overview of general multi-stage decoding algorithm. In Section 4, we describe the proposed multi-stage decoding algorithm in detail. We give experimental results on a 1-millions of Korea POI task domain in Section 5.

2. Complexity of Viterbi decoding

In CDHMM-based speech recognition, where the state output probability is represented as a mixture of Gaussian probability density functions, recognition is a process to find an optimal word sequence $\bar{W} = w_1, w_2, \dots, w_N$ which produces maximum a posterior probability for an observation $X = x_1, x_2, \dots, x_T$ as follows:

$$\begin{aligned}\bar{W} &= argmax_W\{P(W|X)\} \\ &\approx argmax_W\{P(X|W)P(W)\} \\ &= argmax_W\left\{\sum_{m=1}^{M} P(X|M)P(M|W)P(W)\right\} \\ &= argmax_W max_M P(X|M)P(M|W)P(W)\end{aligned} \quad (1)$$

where M is a sequence of subwords comprising W, $P(X|M)$ is acoustic model which is modelled with CDHMM, $P(M|W)$ is pronunciation model and $P(W)$ is language model. The $argmax_W max_M\{\}$ operation is usually implemented with Viterbi decoding algorithm, and the complexity is defined as follows[13]:

$$O((B+K)NT) \quad (2)$$

where N is the total number of sates comprising all word models, K is the number of previous states, B is the number of operations to compute the state output probability and T is the total number of frames. As described previously, there are two directions to accelerate the recognition speed: one is to reduce the complexity B and the other is to reduce the search space NT. In this paper, we adopt multi-stage decoding algorithm to exploit both search space and complexity reduction. In multi-stage decoding, the first stage decoding drops feature rate from analysis-frame rate to phonetic segment rate, and also reduces multi-dimensional input vector to a single-dimensional scalar

by performing phone recognition. The second stage decoding is a kind of lexical access that generates N-best candidates having the minimum edit distances for an input phone sequence or lattice.

3. Multi-stage decoding algorithm

Fig. 1 depicts the block diagram of multi-stage decoding [7].

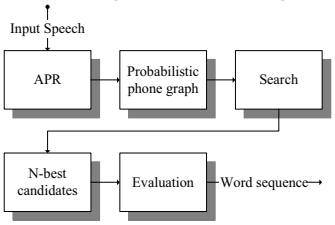

Fig. 1: A block diagram of general multi-stage decoding

For an input feature sequence, automatic phone recognition (APR) decodes the feature sequence into a phone lattice. The resultant phone lattice may contain errors such as substitutions, insertions and deletions. There can be even no canonical phone sequence in the lattice due to the limitations of the APR performance. The second stage recovers N-best candidates from the error-prone phone lattice, where probabilistic edit distance is used for distance measure between two phonetic symbol strings. The evaluation stage rescores the N-best hypotheses and then generates re-ordered results.

The multi-stage decoding algorithm is virtually based on the assumption that probabilistic knowledge sources such as acoustic, pronunciation and language model, are independent each other, and then final results can be decoded by applying individual knowledge sources at each stage as follows:

$$\begin{aligned} &acoustic\ model\ decoding: && \bar{M} = argmax_M P(X|M) \\ &pronunciation\ model\ decoding: && \bar{W} = argmax_W P(\bar{M}|W) \\ &language\ model\ decoding: && \bar{S} = argmax_W P(\bar{W}) \end{aligned} \quad (3)$$

It seems not easy that multi-stage decoding outperforms the typical 1-pass decoding if same knowledge sources are used due to the knowledge independence assumption, but this assumption can provide a systematic framework to exploit various knowledge sources in decoding stage without demanding work to revise a decoding module [11].

In this paper, we deal with the multi-stage decoding algorithm from the aspect of fast decoding. Multi-stage decoding can be treated as a combination of coarse search and detailed search, where APR and lexical access as coarse search and rescoring as detailed search.

4. Two-stage HMM-based Coarse Search

Fig. 2 shows the block diagram of the multi-stage algorithm customized for fast decoding.

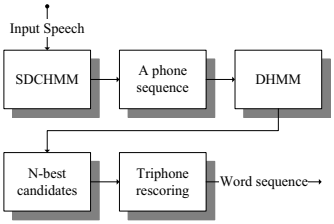

Fig. 2: A block diagram of the proposed multi-stage decoding

The proposed algorithm is composed of a two-stage HMM-based coarse search and a detailed search. The coarse search is composed of SDCHMM-based automatic phone recognition (APR) and discrete HMM (DHMM)-based lexical access, and the detailed search is CDHMM-based triphone rescoring.

The difference of the proposed decoding algorithm is that we use a single phone sequence as an input to lexical access and implement lexical access with DHMM framework instead of a phone lattice and dynamic programming.

We showed that there is little degradation of lexical access even if using a single phone sequence instead of a phone lattice from the aspect of N-best performance, but it gives much faster results [12]. Assuming that error-prone phone results as an observation sequence for reference words, it is a typical noisy channel problem and we use DHMM in estimating the channel characteristics.

4.1. SDCHMM-based APR

The role of APR is to reduce both feature frame rate and feature dimension while minimizing the loss of information for fast and accurate next stage decoding. In order to achieve the goal with memory efficiency, we use SDCHMM-based automatic phone recognition system. 47 context-independent phones including silence are modeled with SDCHMMs. Each phone is first trained with a 3-state left-to-right CDHMM, and then all of the CDHMMs are transformed into SDCHMMs where a one-dimensional feature is put into one stream.

A single phone sequence as a result of APR implies reduction of search space and feature dimension. It reduces search space by dropping feature frame rate from every speech analysis frame to phonetic segment rate, and reduces feature dimension by representing phonetic segment with a 1-dimensional phone symbol. Fig. 3 shows an example of phone segmentation as a result of APR for an utterance spoken as */school/* in Korean. It shows that frame length reduces from 100 to 7, and it means that complexity of next stage can be reduced from $O\big((B+K)N \cdot 100\big)$ to $O\big((1+K)N \cdot 5\big)$, where silence parts are excluded in lexical access.

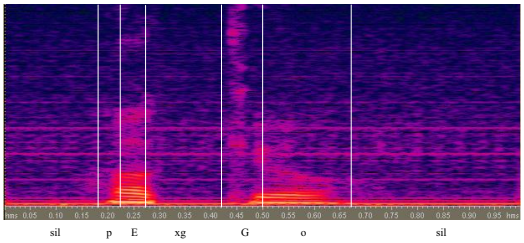

Fig. 3: An example of phonetic segmentation

4.2. DHMM-based Lexical Access

In DHMM-based lexical access, a recognized phone sequence is treated as a sequence of discrete observations. Recognized phone sequences probably contain errors such as substitutions, insertions and deletions. Most common way to estimate such errors is to learn phone confusion probability by collecting occurrence frequencies. However, in this paper, we use DHMM for that purpose, and we call DHMMs modeling phone errors as lexical model from the analogy of the acoustic model.

Fig. 4 shows a DHMM topology of lexical model and the relationship between confusion probabilities, where a is a state transition probability and b is a state output probability.

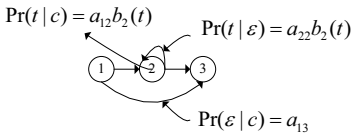

Fig. 4: DHMM-based confusion probability

where t is a recognized phone, c is a reference phone, and ε represents a null-sounding phone for insertion and deletion. The output probability of the 2nd state represents substitution probability, the state transition probability from the 1st state to the 3rd state denotes deletion probability and the insertion probability is represented by multiplication of output probability and self-state transition probability. To train lexical models, utterances are first converted into a phone sequence using the same APR and resultant phone sequences are aligned against a reference transcription, and DHMM parameters are re-estimated by the EM-algorithm.

In decoding stage, lexical word models are constructed by concatenating DHMMs corresponding to their pronunciations, and Viterbi algorithm is used to find N-best candidates.

4.3. Pruning Scheme

In lexical access, we apply global path constraint and beam pruning to remove unlikely hypotheses. Global path constraint pre-excludes a part of search space by making state transitions being restricted to prescribed time slots. Fig 5 depicts an example of global path constraint where states to be explored are filled with grey for the recognized phone sequence of /*school*/ in Korean and its reference string.

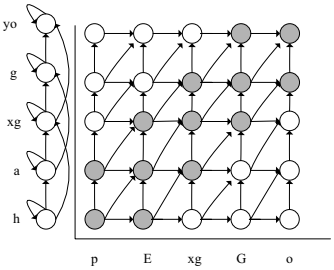

Fig. 5: An example of search space reduction in lexical access

5. Experimental Results

5.1. Korean POI Task Domain

There are about 3.3 millions of POIs used in commercial navigation system. For evaluation in this experiment, we select 1 millions of POI names randomly.

In APR and DHMM-based lexical access, context-independent Korean monophones are used for sub-word units, and context-dependent triphones are trained for the detailed search. In APR, each context independent phone is modeled with 3-state left-to-right SDCHMM whose state output probability is composed of 32 Gaussian probability density functions (PDFs). In lexical access, error patterns for each context independent phone are modeled with 1-state DHMM. In detailed search, triphones are modeled with 3-state left-to-right CDHMM with 16 Gaussian components per a state.

We prepare three sets of speech corpus. A training corpus for acoustic models for APR and rescoring has 90400 utterances (400 males, 400 females), a training set for DHMM-based lexical models for the lexical access has 101,870 utterances (450 males, 450 females) and the test set has 1,920 utterances. The training corpus is collected on various driving conditions from idling to high speed. In testing, a speech corpus collected on idling and low driving speed is used.

The speech signal is sampled at 16KHz, and the frame length is 20ms with 10ms shift. Each speech frame is parameterized as a 39-dimensional feature vector containing 12 MFCCs, C0 energy, their delta and delta-delta feature.

All of the decoding related algorithms are implemented with single precision arithmetic for fixed-point processor.

Recognition performance is measured by word error rate(WER) and recognition speed by a real-time factor (xRT). It is defined as the division of the total recognition time by the total time of the speech utterances on a workstation with a single processor operating at 3GHz. We will also show the recognition speed measured on an in-car navigation device.

5.2. The Baseline performance

As a baseline system, we use a typical CDHMM-based speech recognition system, where the same triphone models trained for detailed search are used as acoustic models and word models are constructed by concatenating HMMs according to their pronunciation. In decoding, lexical tree is constructed, beam threshold is set to 200.0, and a score cache algorithm is used to evaluate state output probability once at each frame. Any other fast decoding methods are not applied. Table 1 shows the performance of the baseline system.

Table 1. The performance of baseline system

WER(%)		Recognition
1-best	10-best	time(xRT)
15.80	6.94	4.71

5.3. The Performance of SDCHMM-based APR

We measure the performance of APR by varying phone language models. Phone language models are trained from the 3.3M POI text corpus, and syllable FSN is a finite state network that allows only the phone sequence which is valid to construct Korean syllable sequences, where C1 is a set of 19 initial consonants, V is a set of 19 vowels and C2 is a set of 7 final consonants.

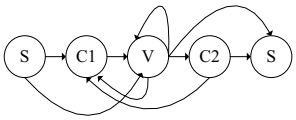

Fig. 5: syllable FSN

Table 2 shows the performance of APR. Even though phone 3-gram is used, the accuracy is not high.

Table 2. Phone accuracies on various phone language models

Phone language model	Accuracy (%)
phone 1-gram	40.14
syllable FSN	49.02
phone 2-gram	55.46
phone 3-gram	58.15

We use results of APR using phone 3-gram as an input to the DHMM-based lexical access.

5.4. The Performance of Multi-Stage Decoding

DHMM-based lexical access generates 500-best candidates and detailed search generates final 10-best results. There are several parameters which affect the performance of the multi-stage decoding.
In detailed search, we apply the same thresholds that are used in measuring the baseline performance.
In DHMM-based lexical access, there are two parameters: one is application of global path constraint and the other is beam threshold. Table 4 shows the affect of global path constraint.

Table 3. Performance on global path constraint

Path constraint	WER(%)		Response time (xRT)
	1-best	10-best	
Disable	16.82	7.03	0.72
Enable	20.80	9.79	0.35

Table 4 shows the performance on various lexical beam thresholds in global path constraint being enabled.

Table 4. Performance on lexical beam thresholds

Beam threshold	WER(%)		Response time(xRT)
	1-best	10-best	
40.0	20.80	9.79	0.35
30.0	20.80	9.79	0.31
20.0	22.63	10.7	0.26
19.0	22.94	11.62	0.25
18.0	22.63	11.93	0.24
17.0	22.02	11.31	0.23
16.0	22.02	11.62	0.21
15.0	23.55	14.07	0.20

Table 5 shows the response time on an in-car device with a fixed-point processor running at 620MHz with the same setup in Table 4.

Table 5. Response time in a navigation device

Beam	20.0	19.0	18.0	17.0	16.0	15.0
xRT	2.60	2.46	2.39	2.33	2.23	2.12

The proposed algorithm can achieve much faster recognition than the baseline system but there is a little degradation of recognition performance. There must be many factors which affect the degradation of recognition performance. However, confining to multi-stage decoding itself, the main cause of the deterioration is low accuracy of APR. Low accuracy lose too much information to recover correct hypothesis in lexical access.

The proposed system is commercialized to after-market GPS navigation system through 2nd market-share company in Korea. In the system, VDE completes through two steps. Users first select a metropolitan city or province and then select POI in the selected area by voice. 8-best candidates are displayed on a screen and users set a destination by touching one of candidates. In this hierarchy interface, the maximum number of names to be recognized at once does not exceed 0.5 millions and real time factor is about 1.04.

6. Conclusions

In this paper, we describe a multi-stage decoding algorithm to achieve fast recognition for very large number of entry names on a resource-limited device. The algorithm is configured with a coarse search and a detailed search. The coarse search is sub-divided into APR and lexical access. For a fixed-frame rate stream of multi-dimensional feature vectors, APR reduces feature dimension and search space by carrying out phonetic segmentations, and DHMM-based lexical access generates N-best candidates from the phonetic sequence.
The proposed algorithm improves recognition speed minimum 10-fold compared to the conventional systems at a similar level of recognition.

7. Acknowledgements

This work was supported by the IT R&D program of MKE/IITA [2009-S-001-01, Development of large vocabulary interactive distributed/embedded VUI for new growth engine industries].

8. References

[1] Hoon Chung, Jeon Park, Yun Lee, Ikjoo Chung, "Fast speech recognition to access a very large list of items on embedded devices", IEEE Transactions on Consumer Electronics, vol 54, pp. 803-807

[2] M.J.F. Gales, K.M. Knill and S.J. Young, "State-based Gaussian selection in large vocabulary continuous speech recognition using HMM's," IEEE Trans. Speech and Audio Processing, pp. 152-161, 1999

[3] A. Lee, T. Kawahara, K. Shikano. "Gaussian mixture selection using context-independent HMM," in Proc. IEEE Int. Conf. Acoust. Speech, signal Processing, vol. 1, pp. 69-72, 2001

[4] M. Ravishankar, R. Bisiani, E. Thayer. "Sub-Vector Clustering To Improve Memory And Speed Performance Of Acoustic Likelihood Computation," in Proc. Of European Conf. on Speech Communication and Technology, 1997

[5] Bocchieri, E. Mak, B.: 'Subspace Distribution clustering Hidden Markov Model', IEEE Trans. Speech and Audio Signal Processing, 2001, Vol. 3, pp. 264-275

[6] L.R. Bahl, S.V. Gennaro, P.S. Gopalakrishnan, R.L. Mercer. "A fast approximate acoustic match for large vocabulary speech recognition," in Proc. Of European Conf. on Speech Communication and Technology, pp. 1156-1158, 1989

[7] O. Scharenborg "Parallels between HSR and ASR: How ASR can contribute to HSR. Proceedings of Interspeech, Lisbon, (2005) Portugal, pp. 1237-1240

[8] J.B. Allen. "How do humans process and recognize speech?" IEEE Trans. Speech and Audio Processing, vol. 2, pp. 567-577, 1994.

[9] Ten Bosch, L., O. Scharenborg. "ASR Decoding in a Computational Model of Human Word Recognition." Proc. INTERSPEEH 2005, Lisbon, Portugal , 1241-1244. 2005

[10] Brill, E. and Moor, R. "An improved error model for noisy channel spelling correction," Proc. ACL 2000

[11] D Moore, J Dines, MM Doss, J Vepa, O Cheng, T Hain, "Juicer: A weighted finite state transducer speech coder," Proc. MLMI 2006 Washington DC.

[12] Hyungbae Jeon, Kyuwoong Hwang, Hoon Chung,. Seunghi Kim, Jun Park and Yunkeun Lee , "Using Confidence Vector in Multi-Stage Speech Recognition", Proc IJCNLP 2008

[13] M.T Johnson: "Capacity and complexity of HMM duration modeling techniques," IEEE Signal Processing Letters, vol. 12, pp. 407- 410, 2005