IMPROVED SEARCH STRATEGY FOR LARGE VOCABULARY CONTINUOUS MANDARIN SPEECH RECOGNITION

Tai-Hsuan Ho[1], Kae-Cherng Yang[2], Kuo-Hsun Huang[1], Lin-Shan Lee[123]

[1]Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, R.O.C.
[2]Department of Electrical Engineering, National Taiwan University, Taiwan, R.O.C.
[3]Institute of Information Science, Academia Sinica, Taiwan, R.O.C.

tai@speech.ee.ntu.edu.tw

ABSTRACT

This paper presents a new search strategy for large vocabulary continuous Mandarin speech recognition considering the special structure of Chinese language. This strategy is composed of a forward and a backward passes, between which a high-quality syllable lattice is generated to bridge the syllable-level and word-level decoding processes. In the forward pass, considering the small number of syllables in Chinese language, a frame-synchronous stack decoder is used to integrate the high-order syllable N-Gram language model, so as to generate a very accurate and compact syllable lattice. In the backward pass, considering the special monosyllabic wording structure in Chinese language, the search space for the word-level decoding is expanded dynamically from the syllable lattice, and the best word sequence is extracted based on the knowledge provided by the word pronunciation lexicon and the word N-Gram language model. In the preliminary experiments, it was found that, with this strategy, the character error rate can be reduced by more than 20% as compared with a previous system using syllable-aligned lattice approach on a speaker-adaptive continuous speech recognition task.

1. INTRODUCTION

During the past decade, several efficient search strategies have been proposed for large-vocabulary continuous speech recognition, such as A* search, stack decoding [1,2], and word graph or word lattice approaches [3,4,5]. Among these strategies, it is believed that the word lattice approach is the most efficient for dictation of Mandarin speech. The word graph or word lattice approach has been studied intensively for western languages [3,4,5]. There are many merits for this strategy. First of all, a word graph is much more compact in encompassing all promising sentence hypotheses than an n-best list, for which a huge *n* is usually required for large vocabulary continuous speech recognition. Furthermore, a word graph can be generated efficiently using left-to-right beam search algorithms, in connection with a tree-organized pronunciation lexicon.

Instead of generating a word lattice directly, a syllable lattice can be generated for large vocabulary continuous Mandarin speech recognition, considering the special monosyllabic structure of Chinese language. In Chinese language, each word is composed of one to several characters, and all the characters are monosyllabic. There exist more than 10,000 commonly used Chinese characters, but only 1,345 phonologically allowed syllables. Most of the syllables are shared by tens of homonym characters. As a result, combination of these 1,345 syllables gives almost unlimited number of Chinese words, and the existence of the syllable level in Mandarin Chinese becomes an essential key to large vocabulary Mandarin speech recognition [6,7]. For example, in a previously developed very successful recognition system [6], the acoustic matching for the 1,345 syllables is first performed to select enough syllable candidates to construct a syllable-aligned lattice, and the linguistic decoding is then performed in the second stage to obtain the best output sentence. The selection of syllable candidates is based on the syllable boundaries in the recognized syllable sequence with the highest score. In this way, the candidates of a particular speech segment are forced to have the same boundaries, and the lattice generated can be very compact. However, due to the aligned syllable boundaries, potential information loss herein is apparently not negligible.

Here we propose an improved search strategy for Mandarin speech dictation based on forward and backward passes, between which a syllable lattice is generated to bridge the syllable-level and word-level decoding processes. The small number of syllables in Chinese language makes it possible to construct a compact yet accurate syllable lattice using a linear syllable pronunciation lexicon in the forward pass. On the other hand, the monosyllabic wording structure in Chinese language also makes it possible to decode the syllable lattice into a word sequence efficiently and accurately using a tree-organized word pronunciation lexicon in the backward pass. As opposed to the previous system [6], the syllables in the syllable lattice generated in the forward pass are not aligned at all, so as to include as much information as possible. High-order language model, such as syllable trigram, can also be integrate early in the forward pass. This is why compact yet accurate syllable lattices can be obtained, and in this way, tighter pruning thresholds became feasible with the strong constraints provided by trigram grammar. In the backward pass, a multiple-stack backward decoder is employed to extract the best word sequence, in which the word-level search trees are expanded dynamically from the syllable lattice. Quite several useful pruning techniques are also applied in the backward pass, such that a significant portion of the word-level search space is eliminated without degrading the recognition accuracy.

In the following, the new search strategy proposed here

will be briefly summarized first in section 2. The forward and backward passes are then explained in details in sections 3 and 4, respectively. The experimental results are finally presented in section 5.

2. SEARCH STRATEGY

The proposed search strategy contains 2 passes, namely, a forward and a backward passes. Between these 2 passes, a syllable lattice is generated to bridge the syllable-level and word-level decoding processes. In the forward pass, a frame-synchronous stack decoder is exploited to decode the speech signal at the syllable level. Linear pronunciation lexicon for the 1345 syllables is used. Possible syllables are hypothesized time-synchronously based on the acoustic matching probabilities. With the stack decoding algorithm and the linear lexicon, high-order language model can be easily integrated in this stage, and tighter pruning thresholds can be used. The resultant lattice is therefore very accurate yet compact as well.

The backward pass aims to extract the best word sequence from the input syllable lattice, based on the knowledge provided by the word pronunciation lexicon and the N-Gram language model. Since the input lattice is composed of syllable hypotheses instead of words, we need to create the corresponding word lattice for the purpose of word sequence extraction. However, because in Mandarin Chinese most of the syllables have tens of homonym characters, the word lattice created directly from the syllable lattice will be very huge. Here we derive a tree search algorithm using multiple stacks, in which the search space of word-level decoding is expanded dynamically based on the local scores of partial paths. By applying proper pruning techniques, many unpromising paths are pruned halfway, and their subsequent word-level search tree expansion can be eliminated.

3. FORWARD : SYLLABLE LATTICE GENERATION

The forward pass generates a syllable lattice for the subsequent decoding pass, in which the syllable hypotheses are not necessarily aligned at all. In this pass, a linear syllable pronunciation lexicon is used. With some minor modifications, such a syllable lattice can be generated very easily as a by-product of Viterbi beam search with n-gram constraints for $n \leq 2$. Whenever a syllable ending state is active after the beam pruning, its corresponding syllable hypothesis is created. However, the quality of such a lattice generated in this way could be questionable. In Viterbi search, each node keeps track of only the best preceding node, which may result in syllable hypotheses with many different end times but sharing the same start time. Many potential hypotheses are thus very likely to be truncated halfway. One remedy for this problem is to use very loose beam width to generate a gigantic lattice for the next pass, but the overall efficiency of the processes is sacrificed.

To avoid the problems mentioned above, we need to duplicate the states in the search space whenever there are more than one promising preceding paths coinciding at the same state. Here frame-synchronous stack decoder is used to achieve the above goal of state duplication. Conceptually, the stack contains all promising partial paths of HMM state sequences, which are primarily sorted by the end time and secondarily sorted by the accumulated path scores. Therefore, when the system processes a hypothesis ending at time *t*, all of the other hypotheses ending before time *t* have been processed. If the stack size is unlimited and no path merger is performed, the frame-synchronous stack decoder becomes a breadth-first tree search algorithm. If all partial paths with the same ending state are merged, it becomes a frame-synchronous Viterbi search algorithm.

An important merit of using stack decoding algorithm is the convenience in integrating high-order language models, but the search space will be increased exponentially. To effectively constrain the search space, two pruning techniques are applied. First, a certain beam width is used to prune partial paths with scores not close enough to the locally optimal path. Secondly, the histogram pruning previously proposed [9] is also adopted to limit the number of active paths in the stack . Because the high-order language model has been integrated, tighter pruning thresholds can be used. As a result, the lattice generated in this pass can be very compact while accurate as well.

The complete algorithm for generating the syllable lattice as described above, given input speech $x_1,x_2,\ldots x_T$, is listed in the following. Since the decoder is frame-synchronous, only 2 stacks are active while decoding.

1. Set $t = 0$ and push initial path into *stack*(*0*).
2. Pop the best path *h* from *stack*(*t*).
3. Extend *h* to all of its possible succeeding nodes at $t+1$, integrate acoustic matching scores for x_{t+1}.
4. If any above path extension is a syllable model transition, integrate the corresponding syllable *n*-gram language model probability.
5. Push all the newly extended path into the *stack*($t+1$).
6. If *stack*(*t*) is not empty, go to step 2.
7. For *stack*($t+1$), perform beam width and histogram pruning
8. If any of the active path in *stack*($t+1$) is at a syllable ending, create a syllable hypothesis.
9. Increase time *t* by 1, and go to step 2 until the whole utterance is decoded.

In addition to the above algorithm, the word conditioned search technique previously proposed [8] is also adopted to further shrink the lattice. When it is applied previously [8], because of the use of tree-organized pronunciation lexicon, the word identity is not known until the tree leaf is reached. When *n*-gram language model is used, a separate copy of the lexical tree have to be kept for each of the (*n*-1) predecessor words. In approach proposed here, because linear pronunciation lexicon is used instead, the syllable identity is known immediately at the starting state of the syllable model. Therefore two paths can be merged when they have the same (*n*-2) predecessor syllables. In the case of syllable trigram here, where $n=3$, only the very preceding syllable need to be checked before merging. Also, in order not to prune the path too strictly, the criterion for path merger is relaxed slightly by some small threshold. So two paths are merged only when the difference between their scores exceeds the small threshold. In this way, most of the potential syllable hypotheses will be preserved in the lattice.

4. BACKWARD : SYLLABLE LATTICE DECODING

The backward pass aims to efficiently extract the best word sequence from the syllable lattice, based on the knowledge provided by the word lexicon and the word N-Gram language model. A backward tree-organized word pronunciation lexicon is utilized here. The pronunciation of each word is composed of the corresponding syllables of its component characters. Figure 1 shows a simplified example of such a backward tree-organized pronunciation lexicon. Each arc in the tree is associated with a syllable, each lexical state except for the leaves stands for a pronunciation suffix, and on each tree leaf is a list of homonym words.

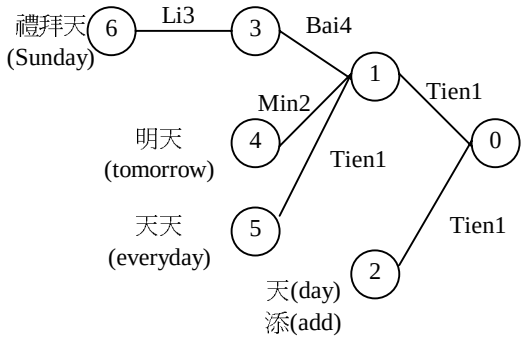

Figure 1. A simplified backward tree-organized word lexicon

In the syllable lattice, each syllable hypothesis *s* contains information including start time $t_{start}(s)$, end time $t_{end}(s)$, the accumulated forward partial path score at syllable level, and the syllable hypothesis score. The backward pass uses stack decoding algorithm very similar to that used in the forward pass. It is frame-synchronous in the backward direction, that is, the shortest path is always decoded first. For each time *t*, a stack, *stack*(*t*), is initiated to preserve the promising paths starting at *t*. Each path in *stack*(*t*) contains its current lexical state in the lexicon tree, and the backward word sequence that has been decoded so far. At each iteration, the decoder pops the shortest path from the stack, extends it by one syllable in the backward direction, and pushes it into the corresponding stack. The path extension is described in detail as follows. For each path in *stack*(*t*), given a syllable hypothesis *s* whose $t_{end}(s)=t-1$, a new path *h'* is created, with its lexical state updated according to the tree lexicon. When the lexical state of *h'* reaches the tree leaf, one more word is decoded, and word N-Gram language model probability is integrated into the path score. In this case, a new tree copy of the lexicon is initiated for this path, and its lexical state is reset to the tree root. The newly created path *h'* is now starting at time $t_{start}(s)$, and is pushed into the corresponding stack, *stack*($t_{start}(s)$). Note that the path proceeds by one syllable at each extension. Therefore, many stacks are active while decoding, and the number of active stacks is roughly equal to the average duration of the syllable hypothesis in the lattice.

Many pruning schemes are applicable to limit the search space. First, path merger is performed at each stack operation, based on the concept of word conditioned search using tree-organized lexicon [8]. Two paths are merged at *stack*(*t*) when both have the same lexical state and identical (*n*-1) successor words, if word *n*-gram language model is used. Secondly, beam width pruning can effectively truncate less promising paths based on local information. Different beam widths are selected for acoustic model probability, language model probability, and the combined score of these 2 knowledge sources. Thirdly, histogram pruning [9] can again be utilized to limit the number of active paths per time frame. The detailed algorithm is listed in the following :

1. Set $t = T$ and push initial sentence hypothesis, of lexical state 0, into *stack*(*T*).
2. Pop the best sentence hypothesis *h* from *stack*(*t*).
3. For each syllable *s* in lattice with $t_{end}(s)=t-1$
   - 3.1 perform path extension as described above, and push newly created path *h'* into *stack*($t_{start}(s)$).
   - 3.2 perform path merger, beam width pruning and histogram truncation.
4. If *stack*(*t*) is not empty, go to step 2.
5. Decrease time *t* by 1, and go to step 2 until the whole lattice is decoded.

5. EXPERIMENTAL RESULTS

The experiments were carried out on a speaker-adaptive continuous Chinese speech recognition task. Speech data of isolated syllables and phonetically balanced sentences from 40 male speakers is used to train the speaker-independent acoustic models. The speech adaptation data is composed of 260 short sentences, corresponding to about 10-minute speech in normal speaking rate. The test material contains 15 Chinese articles, about 4,000 characters, excerpted from local newspapers. Three speakers were enrolled in the test. Each speaker produced a set of the adaptation material, and 5 articles of the test materials.

The testing system is described in the following. The input speech signal is sampled at 11.025kHz, and 12 mel-frequency cepstral coefficients (MFCC) are extracted every 10 milliseconds, to which the pitch for tone recognition is concatenated to form a 13-dimensional feature vector. The first and second order differences are also computed and concatenated to the original vector to form a 39-dimensional feature vector. The acoustic models are HMM with continuous Gaussian Mixture density functions for INITIAL/FINAL units [6], where INITIAL is the initial consonant part of a syllable and FINAL is the final vowel part plus optional medial and nasal endings. A total of 113 context-dependent INITIAL models and 167 tone-dependent FINAL models were used. Each HMM is composed of 3 states and 4 Gaussian functions per state.

The lexicon used contains 48210 words, including 13012 distinct Chinese characters. In the tree-organized lexicon, 7121 of the lexical states are internal and 36523 are tree leaves. Language models were trained by text corpus of about 20 million words from newspapers. Word bigram, syllable bigram, and syllable trigram parameters were estimated from these text data. Table 1 lists the word and syllable perplexity values for the test materials mentioned above. It can be observed that the syllable

perplexity is much smaller than that for word, because there are only 1,345 syllables, much smaller than the number of words in the lexicon. Also, syllable trigram yields much smaller perplexity than syllable bigram. It is therefore believed that the integration of syllable trigram can moderately reduce the error rates.

		Word bigram	syllable bigram	syllable trigram
perplexity of 15 test articles	Word perplexity	307	--	--
	Syllable perplexity	--	67	28

Table 1. The word and syllable perplexities of the test data of 15 articles

In the experiments, the syllable-aligned lattice generated by Viterbi beam search without syllable grammar as used in the previous system [6] mentioned in section 1 is taken as the baseline system, in which the lattice includes only syllable candidates aligned to the same boundaries for each speech segment. Here we investigated the effect of integrating syllable *n*-gram into both the baseline system and the new search strategy proposed here, as well as the achievable performance of the new search strategy. The same word bigram language model is used for both approaches in word-level decoding. Table 2 lists the character error rates. We see that, for the baseline syllable-aligned lattice approach, integration of the syllable bigram only slightly improves the accuracy. This is because the boundary alignment assumption yields substantial search errors, thus inevitably offsets the benefits brought by the syllable bigram. On the other hand, the syllable lattice with the new search strategy outperforms the baseline significantly. In particular, when we upgrade the syllable grammar from bigram to trigram in the new syllable lattice approach, the error rate is further reduced by 10.5%, which implies a 20.6% error rate reduction as compared with the baseline system even including the syllable bigram.

Approaches	Syllable-aligned lattice		Syllable lattice with new search strategy	
Syllable Grammar	No Grammar	Bigram	Bigram	Trigram
Character error rate	11.0	10.7	9.5	8.5

Table 2. Word error rates of syllable-aligned lattice approach and the new search strategy using syllable n-gram

6. CONCLUSION

In this paper, a new search strategy based on syllable lattice and forward and backward passes is proposed for continuous Mandarin speech dictation. A frame-synchronous stack decoding algorithm is used in the forward pass to integrate high-order syllable *n*-gram language model to generate an accurate and compact syllable lattice. A multiple-stack backward decoder is used in the backward pass to dynamically expand the word-level search space to extract the best word sequence from the syllable lattice. Very encouraging character error rate reduction has been obtained with preliminary experiments on a speaker-adaptive continuous Mandarin speech dictation task.

ACKNOWLEDGMENTS

Tai-Hsuan Ho is grateful to Mr. Yen-Lu Chow, Dr. Jerome. R. Belegarda, Mr. John. W. Butzberger, and Dr. Devang Naik for many discussions on the idea of this paper while he visited Speech Group of Apple Computer Inc. in 1996. We also would like to thank Mr. Alex J. Liou and Shiu-Fon Huang of Acer Inc. for their technical assistance.

7. REFERENCES

[1] D.B.Paul, "Algorithms for an Optimal A* Search and Linearizing the Search in the Stack Decoder", *Proc. ICASSP'91*, pp.693-696, 1991

[2] P. Kenny, et al, "A* - Admissible Heuristics for Rapid Lexical Access.", *Proc. ICASSP'91*, pp. 689-692, 1991

[3] X. Aubert and H. Ney, "Large Vocabulary Continuous Speech Recognition Using Word Graph.", *Proc. ICASSP'95*, Detroit, Michigan, USA, Vol.1, pp. 49-52, 1995

[4] H. Murveit, J. Butzberger, V. Digalakis, M. Weintraub, "Large-Vocabulary Dictation using SRI's Decipher Speech Recognition System : Progressive Search Techniques." *Proc. ICASSP'93*, Minneapolis, MN, USA, Vol.2, pp. 319-322, 1993

[5] J.L. Gauvian, L.F. Lamel, G. Adda, "The LIMSI Continuous Speech Dictation System : Evaluation on the ARPA Wall Street Journal Task.", *Proc. ICASSP'94*, Adelaod, Australia, Vol.1, pp. 557-560, 1994.

[6] H.M.Wang, et al, "Complete Recognition of Continuous Mandarin Speech for Chinese Language with Very Large Vocabulary using Limited Training Data", *IEEE trans. On Speech and Audio Processing*, Vol.5, NO. 2, March 1997, pp.195-200

[7] T.H. Ho, et al, "Fast and Accurate Continuous Speech Recognition for Chinese Language with Very Large Vocabulary", *Proc. EUROSPEECH '95*, Madrid, Spain, Vol.1, pp.211-214, 1995

[8] S. Ortmanns, H. Ney, F. Seide, I. Lindam, "A Comparison of Time Conditioned and Word Conditioned Search Techniques for Large Vocabulary Speech Recognition", *Proc. ICSLP'96*, Vol.4, pp.2091-2094, 1996

[9] V. Steinbiss, G.-H. Tran and H. Ney, "Improvements in Beam Search", *Proc. ICSLP'94*, Yokohama, Japan, Vol.4, pp. 2143-2146, 1994