Searching for the Missing Piece

W.N. Choi, Y.W. Wong, Tan Lee and P.C. Ching

Department of Electronic Engineering
The Chinese University of Hong Kong
{wnchoi, ywwong, tanlee, pcching}@ee.cuhk.edu.hk

ABSTRACT

Tree-trellis forward-backward algorithm has been widely used for N-best search in continuous speech recognition. In conventional approaches, the heuristic score used for the A* backward search is derived from the partial-path scores recorded during the forward pass. The inherently delayed use of language model in the lexical tree structure leads to inefficient pruning and the partial-path scores recorded is an under-estimated heuristic score. This paper presents a novel method of computing the heuristic score that is more accurate than the partial-path score. The goal is to recover high-score sentence hypotheses that may have been pruned halfway during the forward search due to the delayed use of LM. For the application of Hong Kong stock information inquiry, the proposed technique shows a noticeable performance improvement. In particular, a relative error-rate reduction of 12% has been achieved for top-1 sentences.

1. Introduction

In the tree-trellis forward-backward search algorithm [1][2], the forward pass involves a time-synchronous Viterbi search. At each time frame whenever a word-ending state is encountered, the following quantities are retained for word lattice generation: (i) word identity; (ii) start time and end time of the word; and (iii) the partial-path score of the word hypotheses. The partial-path score results from the joint contribution of the acoustic model and the language model, up to that particular state.

The backward search starts at the last frame. It is an A* heuristic search based on the partial-path score at each lattice node generated during the forward search. For linear lexicon, this partial-path score would be a perfect estimation of the heuristic score. For tree lexicon, since the language model can only applied at the word end. The LM information is not fully contributed to the partial-path score, it is not accurate and will result in inefficient pruning.

The pruned path in the forward pass could be re-activated in the backward pass as word hypotheses ending at the same time instant are connected to those starting at the next frame. This makes it possible for the backward search to recover the sentence hypotheses that might have been pruned halfway during the forward search. Due to the delay use of LM, the partial-path scores recorded at the word lattice nodes would result in an under-estimated heuristic score. Thus we propose to compute the heuristic score at each word-end node as the maximum score that can be attained for that node at that time.

In the next section, we will first give an overview of our two-pass search algorithm. In Section 3, we will review the generation of word lattice based on the modified word-conditioned search [3]. The implementation details for constructing the word lattice based on a class-based language model will be discussed. Then we present the method of computing the heuristic score during the forward search. Backward A* search is then employed based on this heuristic score to produce N-best list or the best sentence hypothesis. In Section 4, the proposed method is evaluated in the application of Hong Kong stock information inquiry. The attained error-rate reduction and the quality of N-best list after using the proposed heuristic score are analyzed. Finally we conclude in Section 5.

2. Overview of the Search Framework

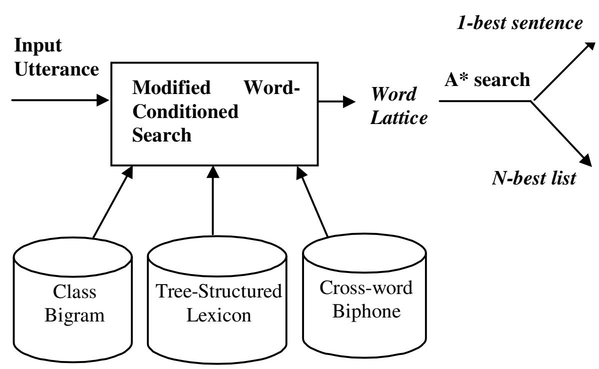


Figure 1: Two-pass search framework

Figure 1 illustrates an overview of the two-pass search framework [3]. In the forward pass, a word lattice is produced using a modified word-conditioned search with a tree-structured lexicon, cross-word biphone acoustic models and class bigram language model. The way of incorporating class bigram with the lexical tree is described as in [3]. The heuristic score computed in the forward search is stored at each word-end node. In the backward search, A* heuristic search is employed to produce N-best hypotheses by re-scoring with either class bigram or trigram. Since the backward search is performed at the word level, the best sentence hypothesis can

be extracted with a tight beam in a fraction of computational effort compared to the forward search.
Our work attempts to search for the missing piece pruned halfway in the forward search by computing an exact heuristic score.

3. Modified Two Pass Search

3.1. Word Lattice Generation by Modified Word-Conditioned Search

The forward search employs time-synchronous search with tree-structured lexicon as described in [3]. With only minor modification, the word-conditioned lexical tree search can be used to produce the word lattice. For the class bigram language model, each active state hypothesis is identified by the following DP quantities: state index, model (HMM) index, lexical-tree node index and class history. There are two types of recombination that may occur at every time frame.

- State recombination — Within the lexical tree, the state hypotheses are recombined if they have the same DP quantities; and
- Class bigram LM recombination — At the word level the best predecessor class for each class hypothesis is chosen.

To generate the word lattice, the following information is collected whenever a word-ending state is processed:

i. word and class identity;
ii. predecessor word and class identity;
iii. start time and end time of the word;
iv. word acoustic score and
v. partial-path score at that word node.

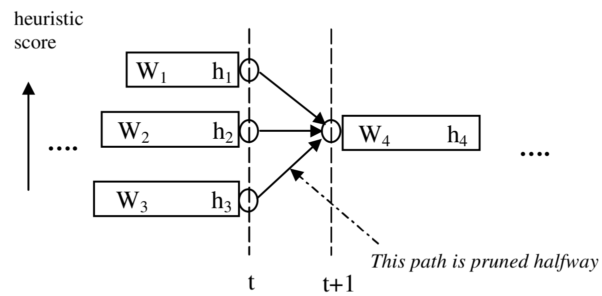
In order to reduce the number of word copies with different start time, word pair approximation is applied [4]. The word boundary between the word pair is assumed to be independent of further predecessor words. Since a word may be associated with more than one class, multiple class identities must be retained at the word-end node.

It should be noted that the collection of word-ending information has been done before the class bigram recombination is performed. The state hypotheses with different word identities but belonging to the same class are not merged until the recording of word ending information is completed. By recording the word hypotheses before the LM recombination, different word hypotheses can be preserved as far as possible for subsequent search process.

In the conventional backward A* heuristic search, the partial-path scores kept at word-end nodes would be used as the heuristic estimates. If the same acoustic and language models are applied to both the forward and backward search, this heuristic score is a very accurate estimate of the best score that the extended path would be able to attain. However, this is not the case for tree lexicon if pruning is applied. As we will show in the next section, due to the delay use of LM and the inefficient pruning, the partial-path score obtained in the forward time-synchronous search might not be the best score that the word-ending state can achieve.

3.2. Computation of the Heuristic Score

The partial-path score is no longer perfect for the backward A* heuristic search due to the inherent problem of delayed use of language model in lexical tree. The pruning based on the less reliable score causes the partial-path score obtained at the word-ending state may not be the best score that it can achieve. In this section, we present the method for computing the heuristic score in the forward search that is more accurate than the partial-path score.



h1, h2, h3 and h4 are the heuristic score computed at the corresponding word end nodes

Figure 2: Estimation of heuristic score in the forward pass

In Figure 2, the path originated from word W_3 is pruned halfway. The partial-path score computed at the word-end node W_4 does not take this path into account. However, since the path from word W_3 to word W_4 is valid for the backward search, the partial-path score computed at the word-end node W_4 may not correspond to the highest attainable score. For bigram language model, we propose to compute the heuristic score as shown in the following steps:

i) For a word-end node **W**, record its word acoustic score.
ii) Back-trace the start time of the word **W** and collect all the predecessor words W_i information.
iii) The heuristic score of the word end node **W** is computed using the following equation.

$$h_{\mathbf{W}} = \max_i \{ h_i + P(\mathbf{W}|W_i) \} + \text{acoustic word score } \mathbf{W}$$

where $h_{\mathbf{W}}$ and h_i are the heuristic scores stored in the word-end nodes **W** and W_i respectively. The heuristic scores computed in this way guarantee that it is the best score that the word-end node can achieve given the available knowledge source.

To reduce the computational effort for computing the heuristic score, we record the maximum heuristic score for all word-end nodes with the same class identity at every time frame. Since class bigram are used, the number of class identities is much smaller than that of word identities. The computation of heuristic score will only cause a small computational overhead for the forward search.

3.3. A* Heuristic Search

The backward A* heuristic search is performed at the word level. At each time instant, a word is extended for the best partial-path according to the A* heuristic. In this case, no acoustic model evaluation at the state level is required. The computational load of the backward search is only a very small fraction of the forward search. The backward search may yield multiple hypotheses or generate a single best sentence at a fast speed using a very tight beam [5]. The A* heuristic is defined as f* = h* + g + LM score. h* is the heuristic score stored in the word-end nodes during the forward time-synchronous search and g is the partial-path score evaluated from the end of utterance to the word start node considered. Since there is a transition from the word not yet extended to the word that has already been extended. LM score is added to the A* heuristic to bridge the connection between h* and g as shown in Figure 3.

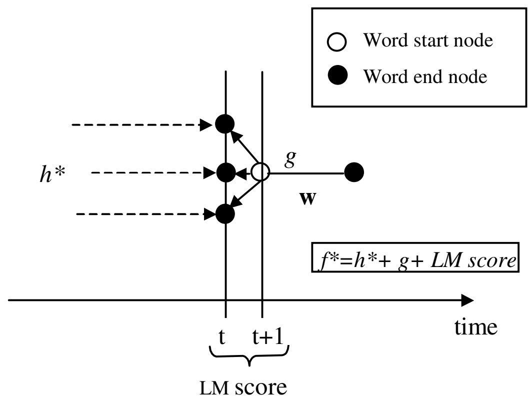


Figure 3: Illustration of the A* heuristic

If the same language model is used in both forward and backward search, the proposed heuristic estimate is exact. The best sentence hypothesis can be found efficiently and accurately. The N-best list extracted in this way is also in the order of increasing cost. For trigram re-scoring, the proposed heuristic score is still a better estimate of heuristic than the partial-path score computed in the forward search as shown in the following experiments.

4. EXPERIMENTAL RESULTS

The proposed heuristic estimate was evaluated in a domain-specific application of Cantonese continuous speech recognition. The application deals with naturally spoken queries on stock information. A typical query looks like:

我想入五千手匯豐
I want to buy five thousands lots of Hong Kong Bank

The lexicon contains 1,125 words. A few of them have pronunciation variants. The 1,125 words are grouped into 267 classes. Each word may be associated with more than one class. The major classes are shown as in Table 1.

Class Name	Examples of words
Stock Name	匯豐銀行, 新鴻基, 長實…
Buy Action	入, 買入, 收入 …
Sell Action	抛, 放, 沽出 …
Digit	一, 二, 三 …

Table 1: Major Categories of the Stock Query task

The acoustic models are 785 decision-tree based cross-word biphones and the number of mixtures per state is 8 [6]. The training set for the language model contains 2095 queries with 17357 words.
The test data contains 1,400 queries recorded from 14 speakers. The perplexities of test data are 43.8 and 34.8 in the cases of class bigram and class trigram respectively.
A number of experiments were carried out to demonstrate the effectiveness of the proposed method in comparison with the conventional partial-path score. All the experiments are performed on an 800Mhz Pentium III PC.

4.1. Simple back-tracking vs A* heuristic search

In the first set of experiments, we compare the character recognition rate of the best sentence hypothesis found in the forward search and the backward search. As discussed in the Section 3.3, the backward search is performed at the word level and the heuristic is exact. The backward search can be made extremely fast. Thus, the best sentence hypothesis found by using A* heuristic search will not increase the computational load compared to simply back-tracking the sentence hypothesis. This is shown by the figure RTF. In Table 2, the character recognition accuracy of the best sentence hypothesis found by using the simple back-tracking and A* heuristic search with proposed heuristic is showed.

Max. no of state hypotheses	A* search		Simple back-tracking	
	Character Accuracy	RTF	Character Accuracy	RTF
200	74.32	1.1	71.58	1.1
400	80.18	1.5	77.47	1.4
600	81.58	2.2	81.18	2.0

Table 2: The character recognition rate for the best sentence hypothesis found by using simple back-tracking and A* heuristic search

The results showed in Table 2 suggest that it is possible to improve the recognition accuracy by using the A* heuristic search with the proposed heuristic score instead of just simply back-tracking the best sentence hypothesis. The gain in recognition accuracy is more significant for tighter beam. In this condition more potentially high-score sentence hypotheses are pruned halfway. Therefore, the missing highest score sentence hypothesis in the forward search can be found in the backward search using the proposed heuristic estimate.

4.2. N-best list evaluation using class bigram re-scoring

In the second experiment, the N-best hypotheses produced with the proposed heuristic estimate and the conventional partial-path score were evaluated. The maximum number of state hypotheses used for all N-best list evaluation is 400. Though the same language model was applied to both the forward and backward search, the N-best list extracted in the backward search using the partial-path score is not in the order of increasing cost. On the other hand, since the proposed heuristic estimate is exact, the N-best list extracted in this way is ordered. To compare the quality of the N-best list produced by two different heuristic estimates, the first N sentences are extracted and the oracle recognition accuracy is compared. The oracle character accuracy is the character accuracy of the best sentence hypothesis extracted in the N-best lists. Table 3 shows the oracle character accuracy using class bigram re-scoring.

	Oracle Character Accuracy (proposed heuristic)	**Oracle Character Accuracy (partial-path score)**
3-best	82.68	82.49
5-best	83.39	83.27
10-best	84.06	84.01

Table 3: Backward search using class bigram re-scoring

4.3. N-best list evaluation using class trigram re-scoring

Then the class trigram re-scoring is performed in the backward search, we showed that the proposed heuristic score is still a more effective heuristic than the partial-path score if more complex language model is used in the backward search. Since more complex language model is applied, both the partial-path score and the heuristic computed are not exact. The N-best list extracted is not in the order of increasing cost. We compared the first N sentence extracted in the backward search using the partial-path score and the proposed heuristic score. The result is shown in Table 4.

	Oracle Character Accuracy (proposed heuristic)	**Oracle Character Accuracy (partial-path score)**
3-best	83.27	82.53
5-best	84.15	83.31
10-best	84.70	84.06

Table 4: Backward search using class trigram re-scoring

From table 4, we observed that the proposed heuristic score is still a better choice than the partial-path score even though a more complex language model is applied in the backward pass.

5. CONCLUSIONS

In this paper, we present a novel method of computing the heuristic score in the tree-trellis search framework for Chinese continuous speech recognition. By using the proposed heuristic, we could extract the best hypothesis with score that is better than the best hypothesis found in the forward search. The error rate reduction for the best sentence hypothesis found in the backward pass is 12% as compared with that obtained from the forward pass without increasing the computational load significantly.

6. ACKNOWLEDGEMENT

This research is partly supported by a research grant from the Hong Kong Research Grants Council.

7. REFERENCES

[1] E.-F. Huang, F.K. Soong, and H.-C. Wang, "The use of tree-trellis search for large-vocabulary Mandarin polysyllabic word speech recognition", *Computer Speech and Language*, Vol.8, No.1, pp.39 - 50, 1994.

[2] F.K. Soong and E.-F. Huang, "A tree-trellis based search for finding the N-best sentence hypotheses in continuous speech recognition", *Proceeding of ICASSP-1991*, Toronto, pp.705 - 708, May 1991.

[3] W.N. Choi, Y.W. Wong, Tan Lee and P.C. Ching, "Lexical tree decoding with a class-based language model for Chinese speech recognition", *Proceeding of ICSLP-2000*, Vol.1, pp.174 - 177, October 2000.

[4] R. Schwartz, S. Austin: "A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses", *Proceeding of ICASSP-1991*, Toronto, pp.701-704, May 1991.

[5] N. Ström, "Continuous speech recognition in the WAXHOLM dialogue system", *Quarterly Progress and Status Report*, Department of Speech, Music and Hearing, KTH, No.4/1996, pp.67 - 110.

[6] Y.W. Wong, "Large Vocabulary Continuous Speech Recognition for Cantonese," *MPHIL Thesis*, The Chinese University of Hong Kong 2000.