# NEW DEVELOPMENTS IN THE INRS CONTINUOUS SPEECH RECOGNITION SYSTEM

*Z. Li, M. Héon and D. O'Shaughnessy*

INRS-Télécommunications, Université du Québec
16 Place du Commerce, Verdun, Québec, Canada H3E 1H6

## ABSTRACT

New techniques are developed for the second pass search in our large vocabulary continuous speech recognition system. The merging of recognition hypotheses is proposed in order to linearize the exponential growth of the tree structure in the depth first search. Branching ordering of the first pass word graph and pruning at both word and phone levels are used to further speed up the search. The algorithm has been shown to be effective on the speaker-independent WSJ task.

## 1. INTRODUCTION

For large vocabulary continuous speech recognition, a two-pass search strategy has advantages in that we may use inexpensive models in the first pass to prune the search space, and then scrutinize the data using a powerful language model and detailed acoustic-phonetic models in the second pass. We use the terms coarse and fine to distinguish between the models used in the first and the second passes. The natural interface between the first and the second passes is a word graph [1,2]. To build the word graph, we have used the monotone graph search algorithm [1,3]. To speed up the search of the lexicon, we proposed a transcription graph [4,5] which may take advantage of the sharing at the phone level rather than only at the word level in the word graph.

The objective of the second pass is to find the highest scoring recognition hypothesis by searching the word graph produced by the first pass. An important difference between the first and the second pass searches is that, in the first pass, each partial transcription that is hypothesized has a unique segmentation associated with it whereas in the second pass we allow for multiple segmentation hypotheses in order to score partial transcriptions exactly.

In selecting the search technique in the second pass, we note that an exhaustive traversal is inefficient unless the word graph is very small. A stack search is efficient but it has an adverse effect since we use a backward heuristic which is calculated by using the first pass models. To avoid an inadmissible heuristic in the stack search, we have implemented a depth first search in the second pass. Since the depth first search opens the search space into a tree, it fails to take advantage of the implicit graph structure of the search space. In order to make the search effort linear rather than exponential in the length of the file, we propose to merge the recognition hypotheses. To further speed up the search, branching ordering of the first pass word graph and pruning at both word and phone levels are used in the course of the search.

## 2. THE SECOND PASS SEARCH

The search space for the second pass is a word graph whose nodes are quadruple $(\bar{t}, \mathbf{D}, n, \bar{\sigma})$ where $\bar{t}$ is a first pass end time, $\mathbf{D}$ is a look-ahead phone string, $n$ is a node in the lexical graph and $\bar{\sigma}$ is a coarse language model state [5]. A branch is labeled by a pair

$(w, \mathbf{F})$ where $w$ is a word and $\mathbf{F}$ a transcription of $w$. A second pass partial recognition hypothesis is essentially a partial path in the word graph together with the information needed to support scoring with the fine acoustic phonetic models and the fine language models. The partial recognition hypothesis $\eta$ is a quintuple $(\pi, b, \mathbf{E}, \{\alpha_t\}, \{\Lambda_\sigma\})$ where

(i) $\pi$ is a pointer to the parent partial recognition hypothesis (if any).

(ii) $b$ is a branch in the word graph.

(iii) $\mathbf{E}$ is a look-behind phone string.

(iv) $\{\alpha_t\}$ is an array of forward scores centered on $\bar{t}$ whose width is controlled by the uncertainty $\Delta$.

(v) $\{\Lambda_\sigma\}$ is an array of language model scores indexed by fine language model states $\sigma$.

Suppose $b$ is a branch in the word graph originating in $(\bar{t}, \mathbf{D}, n, \bar{\sigma})$ and terminating in a node $(\bar{t}', \mathbf{D}', n', \bar{\sigma}')$. Let $w$ be the word label and $\mathbf{F}$ the phonetic transcription of $w$ in the complete lexical theory corresponding to $b$. We use the branch $b$ to generate a new partial recognition hypothesis $\eta' = (\pi', b', \mathbf{E}', \{\alpha'_{t'}\}, \{\Lambda'_{\sigma'}\})$ as follows:

(i) $\pi = \eta$.

(ii) $\mathbf{E}'$ is the look-behind phone string defined by the condition that the tail end of $\mathbf{DF}$ is $\mathbf{E}'\mathbf{D}'$.

(iii) $\{\alpha'_{t'}\}$ is the array of forward scores centered on $\bar{t}'$ obtained by propagating $\{\alpha_t\}$ through the phone string $\mathbf{DF}\backslash\mathbf{D}'$ (that is, $\mathbf{DF}$ with $\mathbf{D}'$ removed).

(iv) For each language model state $\sigma'$ such that $P(w, \sigma'|\sigma) > 0$ for some state $\sigma$ in $\eta$,

$$\Lambda'_{\sigma'} = \max_{\sigma} \Lambda_\sigma P(w, \sigma'|\sigma).$$

In the case where the coarse and fine language models are congruent, it is only necessary to consider states $\sigma'$ which map onto $\bar{\sigma}'$ in the correspondence between coarse and fine language model states.

## 2.1 Branch Ordering

The efficiency in the depth first search depends on how quickly the optimal path can be reached since the pruning is more effective if the highest scoring path is found at an early stage. We may use the first pass information by ordering the branches in the word graph to get the optimal path earlier.

Each branch in the graph has an a priori score obtained by combining the first pass acoustic phonetic and language model scores. In the forward-backward pruning of the word graph, we also assign a backward score to each node. It follows that a backward score can be assigned to each branch by combining the a priori score of the branch with the backward score at the target node of the branch. For each node in the graph, the branches originating in the node are ordered using this scoring function. This is the ordering used for the depth first traversal.

## 2.2 Merging

The depth first search opens the search space into a tree, and it ignores the fact that the possible extensions of all partial paths in the implicit graph that end in the same node are identical. So to make the search efficient, we need another data structure to keep track of visits to the nodes in the implicit graph. We define a *merge bucket* to be a quadruple $(\bar{t}, \mathbf{D}, \mathbf{E}, n, \sigma)$ where $\bar{t}$, $\mathbf{D}$ and $n$ are taken from a node in the word graph, $\mathbf{E}$ is a look-behind phone string and $\sigma$ is a fine language model state. The merge bucket has associated with it an array of forward scores $\{A_t\}$ where $t$ is within the uncertainty of $\bar{t}$. At any stage in the course of the search the forward score $A_t$ is given by $A_t = \max_\eta \Lambda_\sigma \alpha_t$ where $\eta$ ranges over all partial recognition hypotheses that have the property that they end in a word graph node for which $\bar{t}, n, \mathbf{D}$ are the time, the lexical graph node and the look-ahead phone string, that $\mathbf{E}$ is the look-behind phone string and that $\sigma$ is contained in the list of language model states.

Whenever a new partial recognition hypothesis is generated, the scores in the corresponding merge buckets are updated. If for a given partial recognition hypothesis $\eta$ and a language model state contained in the list of language model states for $\eta$, the forward scores of the partial recognition hypothesis are all dominated by the forward scores in the corresponding merge bucket,

then the language model state can be pruned away without running the risk of failing to find the highest scoring global recognition hypothesis. If this happens for all language model states, then the entire partial recognition hypothesis can be pruned away.

### 2.3 Pruning

There are two pruning envelopes, one for word boundaries and another for phone boundaries, which keep track the best forward scores for each frame. Separate thresholds are used against the envelopes for pruning phones and words in the course of the search since the forward score used to calculate the word envelope includes the language model score. Due to the block processing, the envelopes could extend over several blocks. Pruning is also carried out after finishing searching a block, in order to prevent unnecessary recognition hypotheses from passing to the next block.

## 3. EXPERIMENTAL RESULTS

To test the system, the acoustic models were trained by using the ARPA *Wall Street Journal*-based speaker-independent CSR corpus (WSJ0+1), and had separate male and female models. Right-context VQ models [6] (with a single full-covariance matrix and a set of 256 means for all distributions) and a bigram language model are used in the first pass. In the second pass, we use right-context continuous models (with 1 covariance matrix and 16 means per distribution) and a trigram language model. In order to minimize the overall memory requirement in our system, we have investigated the effect of eliminating the low-count statistics of the language model. We have also tested the system on an evaluation test set with the language models which exclude the low-count statistics, which shows that the search algorithm is effective.

### 3.1 Low-Count Statistics Effect

For the language modeling, we have use the statistics from the three-year 1987-89 WSJ text data. In a 5,000-word task, the memory requirement can be reduced by half for bigrams if the count-1 statistics are excluded, whereas the memory requirement for trigrams can be reduced by 92% if the count-4 and below are excluded. To show the low-count statistics effect in the language modeling, we have compared the full statistics with reduced statistics in terms of perplexities as well as word error rates. In Table 1 we have compared the full bigrams with bigrams excluding count-1 statistics for four female speakers on the development set (dev-92). We say that a word search error occurs whenever a recognition error can be attributed to the fact that a word to be recognized fails to appear in the word graph. We see that the word search errors are virtually the same in both cases.

| | Perplexity | Word Search Error |
|---|---|---|
| full bigrams | 119.3 | 0.96% |
| reduced bigrams | 120.2 | 0.96% |

**Table 1:** Comparison of full bigrams versus reduced bigrams in perplexities and word search error rates on the development set.

The comparison between full models and reduced models is given in Table 2, where using full models means using full bigrams in the first pass and full trigrams in the second pass and using reduced models means using bigrams with counts greater than 1 in the first pass and trigrams with counts greater than 4 in the second pass. The word error rate is 10.76% with the full language models and 10.83% with the reduced language models for four female speakers on the development set (dev-92) where, in this particular experiment, the right-context VQ models are used in the second pass. It shows that the low-count statistics have little impact on the word error rate.

| | Perplexity | Word Error |
|---|---|---|
| full models | 93.3 | 10.76% |
| reduced models | 95.2 | 10.83% |

**Table 2:** Comparison of full language models versus reduced language models in perplexities and word error rates on the development set.

### 3.2 Results on the Evaluation Set

We have also tested the system on the evaluation set (Nov92-5k-si-nvp) with a 5,000-word vocabulary where the reduced language models are used. In Table 3 the third column gives the word search error rates whereas the fourth column gives the word error rates. The average word search error rate 0.75% and the average word error rate 7.57% have been achieved.

| Speaker | Sex | Word Search Error | Word Error |
|---|---|---|---|
| 440 | M | 0.46% | 6.61% |
| 441 | F | 1.41% | 11.77% |
| 442 | F | 0.97% | 8.17% |
| 443 | M | 0.61% | 6.53% |
| 444 | F | 0.68% | 8.24% |
| 445 | F | 0.67% | 7.99% |
| 446 | M | 0.73% | 4.22% |
| 447 | M | 0.46% | 7.15% |
| average | | 0.75% | 7.57% |

**Table 3:** The word search error rates and the word error rates on the evaluation set.

## 4. CONCLUSION

We have presented new search techniques for the second pass in our large vocabulary continuous speech recognition system. In order to avoid the exponential growth of the tree structure in the depth first search, it is necessary to merge the recognition hypotheses in the course of the search. To further speed up the search, pruning is imposed at both word and phone levels where two thresholds are used in order to take into account the language model effect. We have also carried out branching ordering of the first pass word graph before doing the depth first search, which allows the envelope pruning to be more effective.

The experiments on the reduction of the counts of language model statistics led to a 92% reduction of the memory requirement without degrading the performance. That also shows that the system is robust to the low-count language model statistics.

## ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

## REFERENCES

[1] P. Kenny, P. Labute, Z. Li and D. O'Shaughnessy, "New graph search techniques for speech recognition," *Proceedings ICASSP 94*, vol. 1, pp. 553–556, April 1994.

[2] M. Oerder and H. Ney, "Word graphs: an efficient interface between continuous speech recognition and language understanding," *Proceedings ICASSP 93*, vol. 2, pp. 119–122, April 1993.

[3] N. Nilsson, *Principles of Artificial Intelligence*, Tioga Publishing Company, 1980.

[4] Z. Li, P. Kenny and D. O'Shaughnessy, "Searching with a transcription graph," *Proceedings ICASSP 95*, vol. 1, pp. 564–567, May 1995.

[5] Z. Li and D. O'Shaughnessy, "Using a transcription graph for large vocabulary continuous speech recognition," to appear in *Proceedings ICASSP 96*, May 1996.

[6] Z. Li, P. Kenny and D. O'Shaughnessy, "Hybrid hidden Markov models in speech recognition," *Proceedings Eurospeech 95*, vol. 1, pp. 795–798, September 1995.