Submission Cover Sheet for 1999 IEEE SoutheastCon

Submission Type (Please Check One):

☐ *Full Length Paper - (Due 11/1/98)* ☑ *Concise Paper - (Due 12/1/98)*

Title of manuscript:
FAST SEARCH ALGORITHMS FOR CONTINUOUS SPEECH RECOGNITION

Authors and affiliations (List authors in the order to appear on the manuscript. List affiliations (companies, universities, etc.) and addresses exactly as they should appear. If there are more than 4 authors, please attach a second sheet.

1. Name: **Jie Zhao**
Affiliation: **Mississippi State University**
Address: **PO Box 9571**
Mississippi State, MS 39762

2. Name: **Jonathan Hamaker**
Affiliation: **Mississippi State University**
Address: **PO Box 9571**
Mississippi State, MS 39762

3. Name: **Neeraj Deshmukh**
Affiliation: **Mississippi State University**
Address: **PO Box 9571**
Mississippi State, MS 39762

4. Name: **Aravind Ganapathiraju**
Affiliation: **Mississippi State University**
Address: **PO Box 9571**
Mississippi State, MS 39762

Corresponding Author:
Name: **Jie Zhao**
Address: **PO Box 9571**
Mississippi State, MS 39762

Telephone: **601-325-8335**
FAX: **601-325-3149**
E-mail: **zhao@isip.msstate.edu**

Key Words: **Speech Recognition, Search, Fast-match**

For Technical Committee Use Only -- Do Not Write Below

Date Received: _____ Manuscript # Assigned _____

Author Notification Date: _____ Date Camera Ready Received: _____

Decision: _____ Session: _____

Notes: _____

Submission Cover Sheet for 1999 IEEE SoutheastCon

Submission Type (Please Check One):

☐ *Full Length Paper - (Due 11/1/98)* ☑ *Concise Paper - (Due 12/1/98)*

Title of manuscript:
FAST SEARCH ALGORITHMS FOR CONTINUOUS SPEECH RECOGNITION

Authors and affiliations (List authors in the order to appear on the manuscript. List affiliations (companies, universities, etc.) and addresses exactly as they should appear. If there are more than 4 authors, please attach a second sheet.

5. Name **Dr. Joseph Picone**
Affiliation **Mississippi State University**
Address **PO Box 9571**
Mississippi State, MS 39762

6. Name
Affiliation
Address

Corresponding Author:
Name: **Jie Zhao**
Address: **PO Box 9571**
Mississippi State, MS 39762

Telephone: **601-325-8335**
FAX: **601-325-3149**
E-mail: **zhao@isip.msstate.edu**

Key Words: **Speech Recognition, Search, Fast-match**

For Technical Committee Use Only -- Do Not Write Below

Date Received: Manuscript # Assigned

Author Notification Date: Date Camera Ready Received:

Decision: Session:

Notes:

FAST SEARCH ALGORITHMS FOR CONTINUOUS SPEECH RECOGNITION

J. Zhao, J. Hamaker, N. Deshmukh, A. Ganapathiraju, J. Picone
Institute for Signal and Information Processing
Mississippi State University
Mississippi State, Mississippi 39762, USA
e-mail: {zhao, hamaker, deshmukh, ganapath, picone}@isip.msstate.edu
Ph (601) 325-3149 - Fax (601) 325-3149

Abstract **- The most important component of a state-of-the-art speech recognition system is the decoder, or search engine. Given this importance, it is no surprise that many algorithms have been devised which attempt to increase the efficiency of the search process while maintaining the quality of the recognition hypotheses. In this paper, we present a Viterbi decoder which uses a two-pass fast-match search to efficiently prune away unlikely parts of the search space. This system is compared to a state-of-the-art Viterbi decoder with beam pruning in evaluations on the OGI Alphadigits Corpus. Experimentation reveals that the Viterbi decoder after a first pass fast-match produces a more efficient search when compared to Viterbi with beam pruning. However, there is significant overhead associated with the first pass of the fast-match search.**

INTRODUCTION

In recent years we have seen great advances [1] in speech recognition technology. Typically these systems are restricted to a particular domain such as automatic dictation or command and control applications. With these restrictions, developers are able to create highly efficient systems which run in real-time with very low error rates. However, the primary goal of speech research is to produce systems that allow users to interact naturally without restrictions to either content or style of speech. Unfortunately, the resources required for such conversational speech recognition systems are far beyond the hardware currently available in the consumer marketplace.

The majority of this resource consumption is owed to the search process inherent in finding the string of words spoken. The decoder searches through every possible word path to find the most likely string of words according to statistical models of speech. The Viterbi search algorithm [2] is used at the core of most state-of-the-art decoders, but the search space for this algorithm is impractical even for speech recognition tasks of moderate complexity. Thus, there is a need for algorithms that can intelligently limit the search space while not affecting the word error rate (WER).

In this work we present a Viterbi decoder incorporating a two-pass fast-match decoding strategy. The first pass, called the fast-match search, quickly finds an approximate solution by applying a simple heuristic at each time step of the search. The second pass uses the knowledge gained from the first pass to perform a more detailed search. The art to this type of algorithm is determining the heuristic which can find a high quality partial solution using very limited resources.

THE VITERBI ALGORITHM AND BEAM PRUNING

Most state-of-the-art speech recognition systems use the Viterbi search algorithm. This algorithm is a dynamic programming algorithm [3, 4] which builds a breadth-first search through a network of Hidden Markov Models (HMMs) and maintains the most likely path score at each state in this network for each frame or time step. This search process is time-synchronous: it processes all states at the current frame completely before moving on to the next frame (this is in contrast to stack-based searches like A*). At each frame, the path scores for all current paths are computed based on a comparison with the governing acoustic and language models. When all the speech data has been processed, the path with the highest score is the best hypothesis. In the worst case (each state can transition to every state at every frame), the complexity of the search quickly grows out of bounds even for moderate-sized tasks.

In most state-of-the-art decoders, pruning techniques are used to reduce the Viterbi search space and to improve the search speed. Most of these involve setting a threshold at each frame in the search where only paths whose score is higher than that threshold are extended to the next frame. All others are pruned away. Pruning is very effective since the

normal search produces many paths which are quickly eliminated from contention and can be pruned with no penalty. However, the thresholds must be set prudently as search errors can occur due to overpruning.

The most commonly used pruning technique is beam pruning which advances only those paths whose score falls within a specified range. Consider the path which is in state S at frame t and whose score is given by $q(s, t)$. At each frame the state with the highest path score, $q_{max}(s, t)$, is found. In beam pruning the pruning threshold is set to be

$$q_{Th} = q_{max}(s, t) + b(t) \quad , \qquad (1)$$

where $b(t)$ is the beam width which is chosen to be appropriate for the application in question. All states whose path score falls inside this threshold are considered active and extended further; any others are pruned away.

While beam pruning is generally effective at reducing the search space, it is also highly data dependent. The appropriate value for the beam varies from utterance to utterance. Yet most recognition systems allow one only to set a beam width for all utterances. Thus, we would like to examine methods for pruning which alleviate this data dependence.

FAST-MATCH SEARCH

In the field of continuous speech recognition, fast-match search is typically seen as the process that quickly provides a short list of most likely candidate words from the vocabulary of several thousand possibilities. Subsequently, a detailed model for only the words in the short list is used to match those words to the acoustic signal. The fast-match process reduces the number of hypotheses to a manageable level for decoding with detailed acoustic models [5, 6].

In this work we explored a word level fast-match technique which uses a two-pass Viterbi decoding strategy. The first pass is the fast-match search, and the second pass is a detailed search. The difference between the two is in the way that paths are pruned. In the first pass, the fast-match search extends only the *M* word-level paths with highest scores to that point in the search. All other path objects are regarded as inactive and do not propagate further. The basic idea behind the fast-match search is shown in Figure 1.

In this figure, the dotted line indicates that only the words ending in the same frame can be compared and that only a fixed number of words are active at any one time. At the end of the fast-match search, a best hypothesis is found. It should be noted that this best hypothesis is only assumed to be a rough estimate of the score of the true best hypothesis. It is expected that the true best hypothesis will have a better score than this estimate.

In the second pass, the path score of the fast-match hypothesis is used as the pruning threshold. At each frame the fast-match score at that frame is used as a rough estimate of what the score should be. Any hypothesis falling below that level is pruned. The aim of this is to only keep those paths that have a reasonable chance of being the overall best hypothesis. The overall best hypothesis may have a score which is worse than the fast-match threshold for some frame in the search but would get better if allowed to continue. Since the fast-match search prunes based on information in only the current frame, this best path will be pruned away resulting in a search error.

SOFTWARE

The software developed during this work was built on top of the public-domain decoder available from the Institute for Signal and Information Processing [7, 8]. The core search algorithm used in the ISIP decoder is based on a hierarchical variation of the standard Viterbi-style time-synchronous search paradigm. At each frame of the utterance being decoded, the system maintains complete

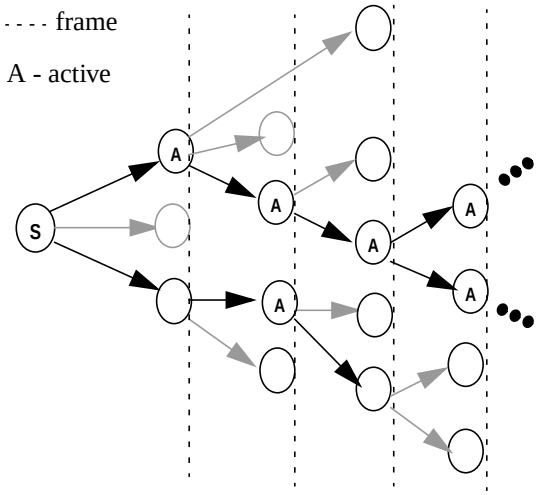

Figure 1. Fast-match search, *M*=2

history for each active path at each level in the search hierarchy via special scoring data structures (markers). Each path marker keeps its bearings in the search space hierarchy by indexing the current history (or word graph) node, lexical tree node and the triphone model. It also maintains the path score and a backpointer to its predecessor.

At each instantiation of a model, a state-level path marker is projected from the previous model-level marker and added to a state-level list of path markers. For each frame, the active states are evaluated only once. The state-level markers are compared and the best marker for each different instance of the state is projected to the next states as governed by the state transition probabilities (Viterbi decoding). The score for each state is stored locally and added to the projected path marker score. A marker exiting the model is added to the model-level marker list and used to project the next context-dependent markers. Similarly, model-level path markers at word ends are promoted to the word level and used to project paths into the subsequent words.

The decoder is equipped with a number of advanced pruning features. It allows the user to set a separate beam at each level in the search hierarchy. The beam width at each level is determined empirically, and the beam threshold is computed with respect to the best scoring path marker at that level. Since the identity of a word is known with a much higher likelihood at the end of the word compared to its beginning; and to curb the fan-out caused by the language model list of possible next words, typically the word-level beam is set much tighter compared to the other two.

By setting an upper limit on the number of active triphone instances per frame, we can effectively regulate the memory usage (and hence computation time) of the decoder. If the number of active hypotheses exceeds this limit then only the best hypotheses are allowed to continue while the rest are pruned off.

EXPERIMENTS

To get a measure of the effectiveness of fast-match pruning in comparison to beam pruning, a series of experiments were run using the OGI Alphadigits Corpus [9]. The Alphadigits Corpus is a telephone database collected over digital phone lines. The approximately 3000 subjects of the Alphadigits Corpus were volunteers responding to a posting on the USEnet. The subjects were given a list of either 19 or 29 alphanumeric strings to speak. The strings in the lists were each six words long, with 1102 total unique strings giving a balanced coverage of vocabulary and contexts (e.g. “R Z 8 3 6 B”). We chose 26 utterances from the official test set with lengths ranging from 1.3 seconds to 6.1 seconds to get a measure of the dependence of each method on the data.

The first experiment explored the way in which the fast-match search results were dependent on the length of the

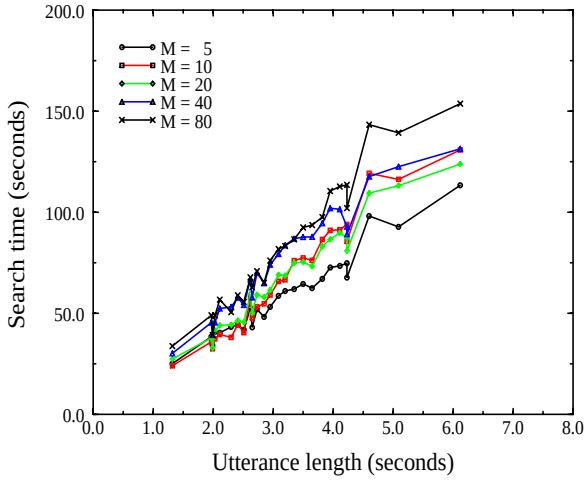

Figure 2. Plot of fast-match search versus the utterance duration for varying *M* values

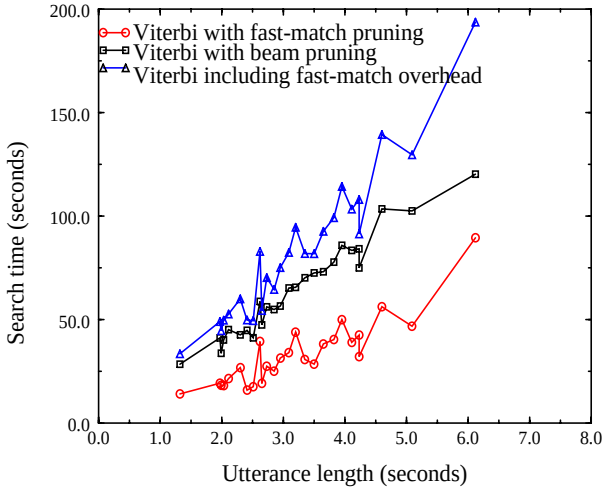

Figure 3. Comparison of Viterbi beam pruning and fast-match pruning ($M = 5$)

utterance and on the number of words extended at each frame (*M*). As seen in Figure 2 the search time increases linearly as the utterance length increases. Also, the search time is directly related to the number of word paths extended at each frame. An interesting point to note is that the WER did not decrease as the *M* parameter was varied. Given that, the best choice for the fast-match parameter is the one which requires the least resources. For the remainder of the experiments we set the number of word paths to extend to five.

Next we did a comparison of the effects of the three pruning methods: no pruning, beam pruning and fast-match pruning. Compared to Viterbi with no pruning it is not surprising that both beam pruning and fast-match pruning techniques greatly improved the efficiency of the decoding process in terms of both speed and memory. Figure 3 demonstrates that Viterbi with fast-match pruning is also much more efficient than Viterbi with beam pruning for all utterances examined. There was a slight degradation in error performance but most of that is attributable to a single, extremely noisy utterance which suffered from poor acoustic match with the models. However, this is without taking into account the overhead incurred during the first pass. In this case, the overhead makes the overall two-pass process less efficient than choosing a beam empirically.

CONCLUSIONS

In this work we have presented an efficient search algorithm that leverages the knowledge gained in a simple first pass search to do an efficient detailed search in the second pass. The experiments have shown that using the fast-match pruning technique in the second pass can greatly reduce the search complexity over conventional pruning techniques. However, the cost of the first pass in our implementation causes the overall search process to be less efficient than using only beam pruning.

There are many techniques that could be used in the first pass search that may increase the overall efficiency of the two-pass search. One example is to use a simple model such as monophone acoustic models for the first pass of the search. The second pass would then use the more complex context-dependent triphone models for decoding. We also know that, by the end of the search process, the path scores for the best candidates are typically close together. Thus, it seems likely that a time-variant beam (wide at the beginning of the utterance and narrow at the end) would produce better results than a static beam width.

REFERENCES

[1] D. Pallett, J. Fiscus, A. Martin and M. Przybocki, "1997 Broadcast News Benchmark Test Results: English and Non-English," *Proceedings of the Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, USA, February 1998.

[2] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260-269, April 1967.

[3] J. Deller, J. Proakis and J. Hansen, *Discrete-Time Processing of Speech Signals*, pp. 623-670, Macmillan Publishing Company, New York, 1993.

[4] M. Ravishankar, "Efficient Algorithms for Speech Recognition," Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 1996.

[5] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Progressive-Search Algorithms for Large Vocabulary Speech Recognition," *Proceedings of the DARPA Human Language Technology Workshop*, Cambridge, Massachusetts, USA, March 1993.

[6] E. Eide and L. Bahl, "A Time-Synchronous Tree Based Search Strategy in the Acoustic Fast Match of an Asynchronous Speech Recognition System," *Proceedings of the International Conference on Spoken Language Processing*, pp. 2915-2918, Sydney, Australia, November 1998.

[7] N. Deshmukh, A. Ganapathiraju, J. Hamaker and J. Picone, "An Efficient Public Domain LVCSR Decoder," *Proceedings of the Hub-5 Conversational Speech Recognition (LVCSR) Workshop*, Linthicum Heights, Maryland, USA, September 1998.

[8] N. Deshmukh, A. Ganapathiraju, J. Zhao, X. Zhang, Y. Wu, J. Hamaker and J. Picone, "Large Vocabulary Conversational Speech Recognition," *http://www.isip.msstate.edu/projects/speech_recognition/*, Institute for Signal and Information Processing, Mississippi State University, 1999.

[9] "Alphadigit v1.0," *http://cslu.cse.ogi.edu/corpora/alphadigit/*, Center for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology, USA, 1997.