ISIP PUBLIC DOMAIN LVCSR SYSTEM

A. Ganapathiraju, N. Deshmukh, J. Hamaker, V. Mantha, Y. Wu, X. Zhang, J. Zhao and J. Picone

Institute for Signal and Information Processing
Department for Electrical and Computer Engineering
Mississippi State University, Mississippi State, MS 39762
{ganapath, deshmukh, hamaker, mantha, wu, zhang, zhao, picone}@isip.msstate.edu

ABSTRACT

In this paper, we describe the first version of our baseline public domain speech recognition system. This system contains most functionality normally expected in an LVCSR system, including word-graph generation and rescoring, cross-word acoustic modeling, state-tying and Baum-Welch training. The decoder has changed substantially since last year's workshop, and includes a much more powerful search engine. As a demonstration of its capabilities, we were able to generate and rescore SWB word-graphs (with a 15.6% word graph error rate) to obtain a 45.6% WER using crossword acoustic models and a trigram language model.

1. INTRODUCTION

Over the past two years, ISIP has been developing a state-of-the-art public domain speech-to-text system. An early prototype of the decoder, which forms the core of this system, was demonstrated at last year's workshop [1] on a SWB [2] word-graph rescoring task. Since then we have added several components to the system, including an acoustic front-end and HMM training capabilities, making it a full-fledged large vocabulary conversational speech recognition (LVCSR) system. In this paper we describe key features that have been added to this system in the past year, and quantify system performance in terms of required computing resources and word error rate (WER).

The current system, which has been in release for several months, is still preliminary in the sense that it does not use the full repertoire of ISIP foundation classes (IFCs). The system is currently being rewritten to be based on these IFCs so that it is extensible, and portable across languages and platforms. We expect this task to be completed by Fall'99, at which time we will make our first production release. Users can participate in the development of this version of the system using ISIP's anonymous CVS server.

2. ENHANCEMENTS TO THE BASELINE SYSTEM

Our primary goal for the first release of the system was to provide the most important functionality for performing SWB word-graph rescoring experiments. This allowed us to focus on the complexity of the search problem. For SWB, this is particularly important because many existing public domain systems seem to break on the SWB task. Reasons for this include the sheer size of the application and the acoustic ambiguity (too many hypotheses with similar likelihoods). A search engine must be extremely efficient so that it can maintain a deep stack of significantly different word sequences, and avoid pruning the eventual winning hypothesis. Having accomplished this task, we have focused in the past year on augmenting the decoder with functions required by a comprehensive LVCSR system. Below, we highlight the various modules that have been added in the past year.

Acoustic Front-End: In an attempt to provide an easy migration path for existing recognition acoustic models, we provide an industry standard front-end [3, 4] that generates mel-spaced cepstral coefficients along with their temporal derivatives (delta) and the acceleration (double-delta) coefficients. The user can choose from a wide range of windowing functions, including the standard Hamming and Hanning windows. Cepstral mean normalization is also an integral part of the front-end.

Baum-Welch Training: The most effective training technique for three-state context-dependent continuous density HMMs is the Baum-Welch algorithm [5]. Another commonly used approach is the Viterbi algorithm [5], which uses a one-best approach to estimate parameters. It is a simple extension of the Viterbi decoder used in the ISIP system, and therefore was an obvious choice for the first implementation. We have subsequently introduced Baum-Welch training into our system. Both implementations include most of the standard features such as the capability to estimate multiple mixtures and the use of model and word level transcriptions. Training can be easily run in a batch mode to allow training across multiple processors — a feature crucial to the large experimental setups required for state-of-the-art performance.

State-Tying Using Decision Trees: One of the main concerns with training context-dependent models is the lack of sufficient training data. Several models typically end up with very few instances and suffer from bad estimates. To avoid this, states of models with similar phonetic contexts are allowed to share training data, which yields better parameter estimates. Clustering of states that can share data using phonetic context information is called phonetic state tying, and is typically performed using statistical decision trees [6]. This also allows the system to generate models for unseen contexts.

Search: The core search algorithm used in the system is based on a hierarchical variation of the standard Viterbi-style time-synchronous search paradigm. The current version of the decoder supports N-gram decoding and generation of word graphs, network decoding and word graph rescoring, evaluation of the word graph WER, and forced alignment. The core search algorithm is described in detail in [7]. The decoder can handle both word-internal and cross-word context-dependent models and uses a tree-based organization [8] of the pronunciation lexicon to efficiently process large vocabulary tasks.

3. IMPLEMENTATION ISSUES IN LVCSR DECODING

The decoder is the most complex component of an ASR system and it dominates the resources required to run an application. A decoder needs to efficiently manage the large search space generated using cross-word context-dependent acoustic models and an N-gram language model (LM) for large vocabulary tasks such as SWB. The following sections provide a brief synopsis of some key implementation issues for the decoder.

Lexical Trees: The decoder represents the phonetic lexicon in terms of a pronunciation prefix tree [8], which provides the framework for the propagation of paths. A copy of this lexical tree is needed per word ending to represent the next set of hypothesized words. For large vocabulary applications, even a few such copies of the lexical tree overshoot the available memory. Our decoder avoids this memory explosion by dissociating the LM scores from the lexical tree and using only a single tree that is independent of the predecessor words. The LM score for a word is calculated on an as-needed basis and stored in the path instance associated with the corresponding history word and lexical tree node. Since the tree is made up of monophones, the system dynamically generates triphones by traversing the lexical tree nodes at each step.

Language Model Lookahead: Due to the fact that common phones are shared in the lexical trees, the identity of a word (and hence the associated LM score) is uniquely known only at the terminal node for the branch containing that word. Therefore the LM score can be applied only at the end of the word, and not at its beginning. This delay allows for undesirable growth in the complexity of the search by pruning fewer paths. Our decoder uses language model lookahead to overcome this problem [9]. Here, while the true LM score is added only at a leaf node of the tree, at each non-terminal node the maximum possible LM score for that branch of the tree is used for pruning purposes.

Pruning: In order to conserve computational and memory resources, it is imperative to identify low-scoring partial paths that are unlikely to get any better, and stop propagating them. A number of heuristic criteria are applied to halt the traversal through such paths and return their memory to the system for reuse. Three important pruning criteria used in the decoder are: (1) a traditional multi-level beam; (2) a limit on the number of active phone model instances (MAPMI); (3) a limit on the number of active word endings. Good decoder performance and resource utilization involves careful adjustment of these three thresholds — no single pruning dominates when the system is properly tuned (though traditional beam pruning is the least effective of the three). The impact of performance as a function of pruning is shown below in Figure 1.

Finally, the unique location of a path in the search space is described by a data structure known as its *instance*. This is defined in terms of the lexical node, the appropriate word history (e.g. N-gram, word network node) and the identity of the HMM in use. The instance of a path governs its associated evaluations, merging and propagation through the search network. Therefore pruning based on the maximum number of allowed instances is very crucial in allowing the decoder to handle resource intensive tasks such as lattice generation for SWB.

4. EVALUATIONS

We have used two tasks to benchmark our decoder during development: the OGI AlphaDigit Corpus and SWB. In a related project, we have been retranscribing and resegmenting the SWB Corpus in an effort to finally provide a clean version of the database suitable for technology development. The original release of SWB was based on acoustic segmentations and its transcriptions had an inherent word error rate of approximately 8%. We have completed resegmentation of the entire 2438 conversations, and have completed new transcriptions for 569 conversations. A preliminary release of 795 conversations was made for this workshop so that researchers could begin assessing the impact of the new data.

To benchmark the performance of our system we ran several comparison experiments using existing models trained during WS97 [10]. We first evaluated the system in word-graph rescoring mode using cross-word triphone models and WS97 word graphs. This was followed by generation of word graphs using our system with a bigram LM and word-internal models. The WER for these new word graphs was measured to be 15.6% and required about 200 xRT on a 333 MHz Pentium processor. These new lattices

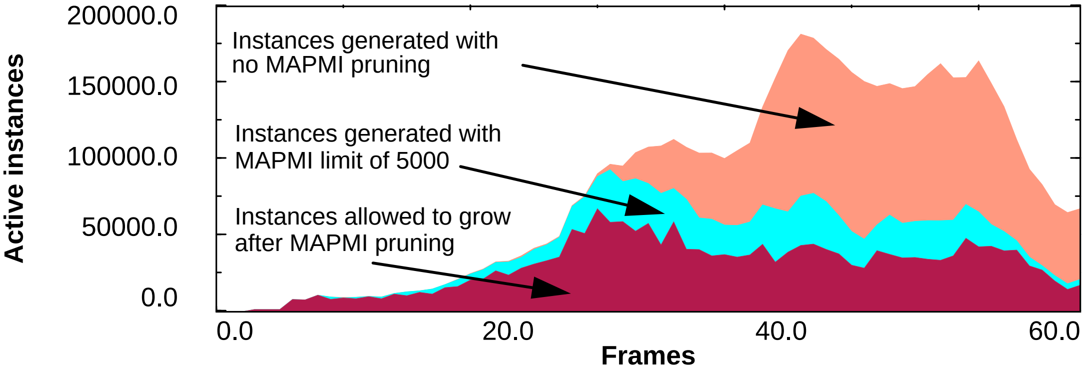

Figure 1: Effect of pruning on the complexity (and therefore efficiency) of the search process.

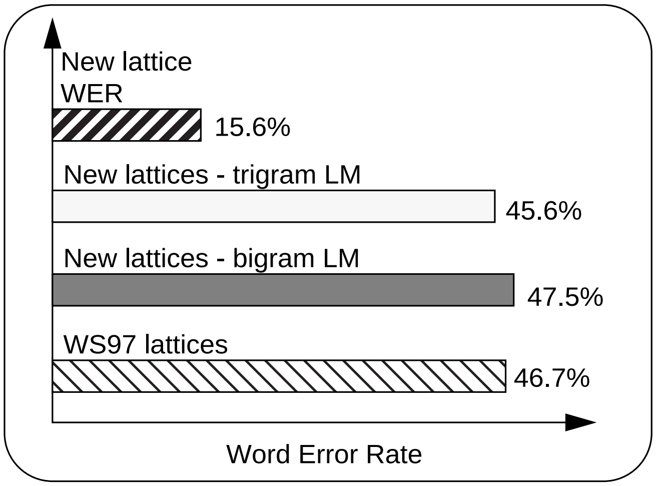

Figure 2: A comparison of performance for several rescoring experiments using crossword models.

were rescored using cross-word models and a trigram language model. Figure 2 compares the performance of these systems.

5. CONCLUSIONS

We have released into the public domain a complete speech-to-text system capable of efficiently handling large vocabulary tasks such as SWB. The system now includes a cepstral front-end, Viterbi and Baum-Welch training modules and a time-synchronous one-pass tree-based Viterbi decoder. Other salient features of the system include decision tree-based state tying, word graph generation and rescoring, and word graph compaction.

The system has been evaluated using several configurations of acoustic and language models. Performance is comparable to other systems on the same applications, both in terms of accuracy, complexity and memory usage. We are in the process of expanding the system to handle broadcast news tasks (our first step will be Chinese) to validate its extensibility to foreign languages and different domains. The larger LMs used in these applications appear to be a major challenge. In the following year, we will hold our first training workshop aimed at introducing various sites to this toolkit, as well as an industrial design review forum.

REFERENCES

[1] N. Deshmukh, et al, “An Efficient Public Domain LVCSR Decoder,” *Proc. Hub-5 LVCSR Workshop*, Linthicum Heights, Maryland, USA, September 1998.

[2] J. Godfrey, et al, “SWITCHBOARD: Telephone Speech Corpus for Research and Development,” *Proc. ICASSP’92*, San Fran., CA, USA, pp. 517-520, March 1992.

[3] P. Woodland, et al, *HTK Version 1.5: User, Reference and Programmer Manuals*, Cambridge University Engineering Dept. and Entropic Research Laboratories Inc., 1995.

[4] J. Picone, “Signal Modeling Techniques in Speech Recognition,” *IEEE Proceedings*, vol. 81, no. 9, pp. 1215-1247, Sept. 1993.

[5] J. Deller, J. Proakis, and J.H.L. Hansen, *Discrete Time Processing of Speech Signals*, Macmillan Publishing Company, New York, New York, USA, 1993.

[6] S.J. Young, et al, “Tree-Based Tying For High Accuracy Acoustic Modeling,” *Proc. ARPA Workshop on Human Lang. Tech.*, pp. 286-291, Plainsboro, NJ, Sept. 1994

[7] N. Deshmukh, A. Ganapathiraju, and J. Picone, “Hierarchical Search for Large Vocabulary Conversational Speech Recognition,” to appear in *IEEE Signal Processing Magazine*, September 1999.

[8] J. Odell, “The Use of Context in Large Vocabulary Speech Recognition,” *Ph.D. Thesis*, Univ. of Cambridge, March 1995.

[9] H. Ney and S. Ortmanss, “Dynamic Programming Search for Continuous Speech Recognition,” to appear in *IEEE Signal Processing Magazine*, Sept. 1999.

[10] A. Ganapathiraju, J. Hamaker, and J. Picone, “Syllable-Based Large Vocabulary Continuous Speech Recognition,” submitted to the *IEEE Trans. on Speech and Audio Processing*, Dec. 1997.