# A PUBLIC DOMAIN SPEECH-TO-TEXT SYSTEM

*M. Ordowski+, N. Deshmukh*, A. Ganapathiraju*, J. Hamaker*, and J. Picone**

*Institute for Signal and Information Processing
Department for Electrical and Computer Engineering
Mississippi State University, Mississippi State, MS 39762 USA
{deshmukh, ganapath, hamaker,picone}@isip.msstate.edu
+Department of Defense, Ft. Meade, MD 20755, USA
mordow@afterlife.ncsc.mil

## ABSTRACT

The lack of freely available state-of-the-art Speech-to-Text (STT) software has been a major hindrance to the development of new audio information processing technology. The high cost of the infrastructure required to conduct state-of-the-art speech recognition research prevents many small research groups from evaluating new ideas on large-scale tasks. In this paper, we present the core components of an available state-of-the-art STT system: an acoustic processor which converts the speech signal into a sequence of feature vectors; a training module which estimates the parameters for a Hidden Markov Model; a linguistic processor which predicts the next word given a sequence of previously recognized words; and a search engine which finds the most probable word sequence given a set of feature vectors.

## 1. INTRODUCTION

Over the years, the complexity of the Speech-to-Text (STT) tasks have increased by many factors. In particular, the Switchboard [1] conversational speech recognition task exemplifies this situation. Unfortunately, the infrastructure required to deal with this increased complexity has also been magnified. The Institute for Signal and Information Processing (ISIP) has been committed to providing the research community with free software tools for digital information processing via the Internet to facilitate worldwide synergistic development of speech recognition technology. Our primary goal is to leverage state-of-the-art STT technology and harness the Internet as a means to share resources and provide unrestricted global access to the relevant software and data. To that end, ISIP hopes to lessen the burden of infrastructure requirements.

The ISIP Speech-to-Text system readily provides to the speech research community a toolkit that is capable of handling complex tasks such as Switchboard and has the flexibility to handle multilingual conversational speech recognition tasks such as Call Home [2].

As we approach our ultimate vision, our goals for the ISIP Speech-to-Text system include the following:

- unrestricted access via the Internet
- detailed documentation and operating instructions
- a state-of-the-art system with periodic upgrades
- object-oriented software design
- on-line technical support

The focus of this paper is on the core of any Speech-to-Text system, the decoder. A decoder is perhaps the sole determinant of whether an STT system is state-of-the-art in both performance and software design. Experiments will be presented to show that the decoder is state-of-the-art and we leave it will be up to user to determine if the software design is flexible to alteration. Next, we will cover the basics elements in acoustic processing and training which complete the Speech-to-Text system. Finally, some experiments with the complete system will be presented.

## 2. ISIP DECODER

A state-of-the-art public domain decoder needs to efficiently and transparently handle tasks of varied complexity, from connected digits to spontaneous conversations. The ISIP decoder can be executed in several different modes of operation.

- word graph acoustic/language model rescoring
- network grammar decoding
- n-gram decoding
- word graph generation
- forced alignments

### 2.1 DECODER FUNCTIONALITY

The core search algorithm used in the ISIP decoder is based on a hierarchical variation of the Viterbi-style time-synchronous search paradigm [3]. At each frame of the utterance, the system maintains a complete history for each active path at each level in the search hierarchy via special 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. The path score and a backpointer to its predecessor are also maintained.

### 2.2 DECODER PRUNING

The ISIP decoder employs two different heuristic pruning techniques to prevent growth of low-scoring hypotheses. Each of these has significant impact on the memory requirements and execution time of the decoder.

*Beam pruning*: The decoder allows the user to set a separate beam at each level in the search hierarchy (word, model, state). 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. At time $t$, the best scoring hypothesis in state $s$, model $m$, or word $w$ has a path score given by

$$Q_{max}(x, t) = \forall x, max\{Q(x, t)\}$$

where $x$ is either $s$, $m$, or $w$. For each state, model, or word beam width of $B_x$, we prune all hypotheses $(x, t)$ such that

$$Q(x, t) < Q_{max}(x, t) + B_x$$

where $x$ is either $s$, $m$, or $w$. Since the identity of a word is known with a much higher likelihood at the end of the word, a word-level beam is usually set to a much tighter value compared to the other two beams.

*Maximum active phone model instance (MAPMI) pruning*: By setting an upper limit on the number of active model (triphone) instances per frame, we can effectively regulate the memory usage of the decoder [4]. If the number of active hypotheses exceeds a limit, then only the best hypotheses are allowed to continue while the rest are pruned off.

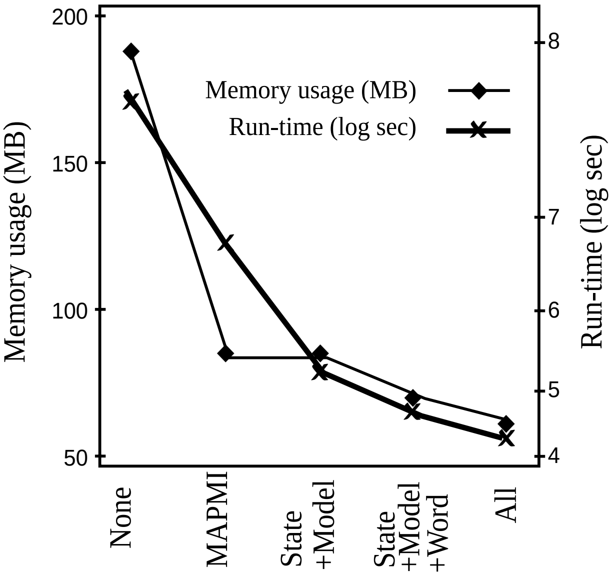

Figure 1: Effect of various pruning criteria using the ISIP decoder for a typical Switchboard utterance.

Effective pruning is critical in recognition tasks similar to Switchboard. The reason for this is often not well understood within the community. Given a situation where the acoustic models are poor matches to the acoustic data, as exemplified in telephone based applications like Switchboard that involve spontaneous speech, the number of hypotheses which are close in probability to the correct hypothesis is large. As a result, an explosive fan out of the unpruned hypotheses will occur unless strict attention is paid to pruning methods. Figure 1, shows the effects for a typical Switchboard lattice rescoring application using the ISIP decoder. As expected, the MAPMI pruning applies strict limits on memory usage. The state, model and word beams provide more direct control in preventing the fan-out caused by surviving end traces. This is one aspect that makes the ISIP decoder different from a standard Viterbi time-synchronous search paradigm.

### 2.3 Lexical Processing

The same lexical processor is used in all decoding modes. In n-gram decoding, the linguistic search space is constrained by a probabilistic model such as a bigram or trigram (with back-off) to predict the next possible word(s) [5]. The ISIP decoder can generate word graphs by constraining the linguistic search space either by a grammar or a n-gram language model. The Speech-to-Text system includes the software to take a grammar constructed in a Bakis-Naur Form [6] to a word graph that can be used as input to the decoder.

The pronunciation of a word is stored in a lexical tree. The ISIP decoder uses one lexical tree, and acquires language model weights on an as-needed basis. This implementation has two benefits. First, this implementation makes for a more efficient n-gram decoding scheme. In this case, a virtual copy of the tree is created for each n-gram word history (a dynamic tree approach). Second, the lexical tree is an efficient scheme to incorporate the language model probabilities early in the search stage [7].

### 2.4 Memory Management

All major data structures are handled by a centralized memory manager which works to minimize the memory load of the system by reusing as much memory as possible, and allocating memory in large blocks (thereby minimizing the overhead in creating memory).

### 2.5 Decoder Resource Usage

The current version of the ISIP decoder supports several modes of lattice rescoring and lattice generation. Figures 2, 3 and 4 show the resource usage of the system in some of these modes on utterances of different durations on the Switchboard task. All benchmarks described below are derived from experiments run on a 333MHz Pentium II processor with 512MB memory.

## 3. ACOUSTIC PROCESSOR

The ISIP Speech-to-Text system uses the most common front-end used in speech recognition systems today. The signal processing module produces industry-standard cepstral coefficients, coefficients that describe their derivatives (deltas) and acceleration (double-deltas). Key features, such as cepstral mean subtraction have been implemented.

Several other features that make a signal processing module robust to noise and compensate for the artifacts of frame-based computations have also been implemented. The user is able to choose from a wide range of windowing functions to include the standard Hamming and Hanning windows. The standard signal processing features like, pre-emphasis and energy normalization are also supported.

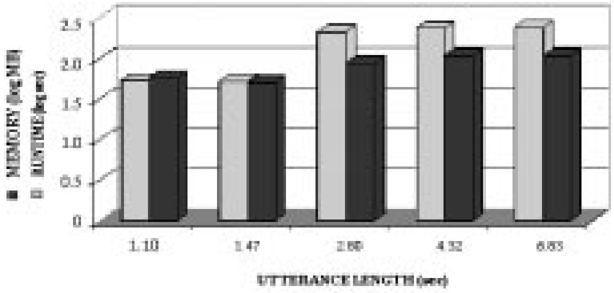

Figure 2: Cross-word triphone lattice rescoring using the ISIP decoder on a typical Switchboard utterance.

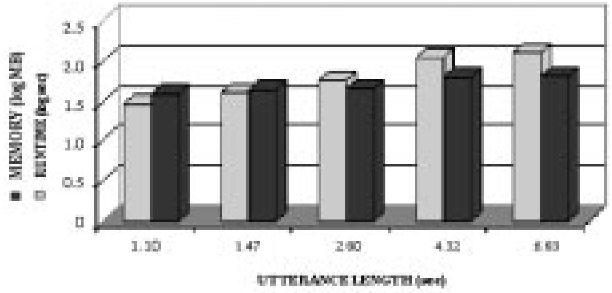

Figure 3: Word-Internal triphone lattice rescoring using the ISIP decoder for typical Switchboard utterance.

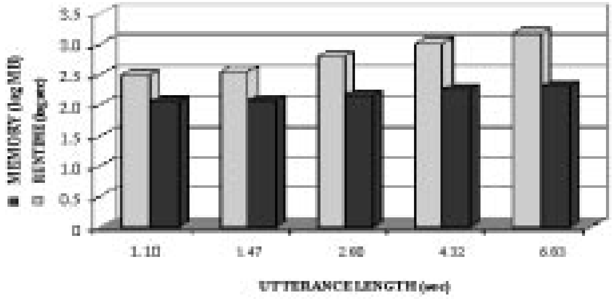

Figure 4: Lattice Generation using the ISIP decoder on a typical Switchboard utterance.

## 4. TRAINING

Training the Speech-to-Text system is currently organized as a set of utilities in which the Viterbi training module forms the core. The choice of Viterbi training was motivated by the simplicity, ease of implementation, and minimal modifications to the decoder to form a complete Speech-to-Text system. Baum-Welch training will be available in the very near future.

The organization of the training paradigm has been implemented in a distributed fashion. This has a major advantage for large task such as Switchboard [1]. Training in a distributed manner allows for training to be restarted from various stages. Allowing better management of the vast amounts of disk space needed and hundreds of hours to compute recognition models on large tasks.

The following utilities are available in training:

- Initialization of monophone models using global mean and diagonal covariance
- The core Viterbi training module
- Initialization of context-dependent models from a clustering process
- Generation of an arbitrary number of Gaussian mixture components

## 5. SOFTWARE AND INTERFACE

The ISIP decoder is designed in an object-oriented fashion and written completely in C++. For efficient access and sorting purposes the principal data structures are handled via linked lists and hash tables. Efficient modules for memory management ensure that used memory is periodically freed and reused. The software structure allows for a hierarchical representation of search space extensible to higher levels such as phrases and sentences. Hooks are provided to apply various kinds of acoustic distributions.

The decoder includes a Tcl-Tk based graphical interface (GUI) that allows the user to specify various decoding parameters through a simple point-and-click mechanism. It also provides a frame-by-frame display of the top word hypotheses, triphones and memory statistics; thus serving as a debugging and educational tool.

## 6. EXPERIMENTS AND RESULTS

ISIP has conducted several verification experiments to show that the Speech-to-Text system performs at a level which is competitive with state-of-the-art systems. ISIP has found it to be a difficult challenge to run a controlled experiment which puts a known state-of-the-art decoder against the ISIP decoder in a head-to-head competition. In the absence of the ultimate experiment, we present recognition results from a complete system for the Alpha-Digits [8] task and an evaluation on Switchboard from the 1997 Summer Workshop [9] at Johns Hopkins University.

At ISIP, the OGI Alpha-Digits task has been used to test several systems in the past. This task consists of approximately 3,000 subjects, each of whom spoke some subset of 1,102 phonetically-balanced prompting strings of letters and digits. This experiment used the ISIP training toolkit to train cross-word triphone models. The training set consisted 52,545 utterances from 2,237 speakers. Disjoint from the training set, the evaluation set was made up of 3,329 utterances from 739 speakers [10]. Table 1 shows the breakdown of errors.

| | WER | Subs. | Dels. | Ins. |
|---|---|---|---|---|
| ISIP | 15.6% | 13.8% | 1.0% | 0.8% |

Table 1: ISIP Speech-to-Text system on Alpha-Digits

The final system presented in this paper is the Switchboard recognizer. The system presented here is missing some prominent features such as vocal tract length normalization and speaker adaptation, and focuses on a core acoustic modeling system using context-dependent phones. The acoustic models were trained from 60 hours of data [11]. Using the ISIP system, lattices were generated on a evaluation set of 2,437 utterances [11] using word-internal triphone models and a bigram language model. The lattices were then rescored using cross-word triphone models. Table 2, shows the breakdown of errors of this system.

| | WER | Subs. | Dels. | Ins. |
|---|---|---|---|---|
| ISIP | 47.3% | 32.2% | 11.4% | 3.7% |

Table 2: ISIP Speech-to-Text system on Switchboard

## 7. CONCLUSIONS

We have presented a public domain Speech-to-Text system where the core module, the decoder, is state-of-the-art in terms of recognition performance as well as consumption of CPU and memory resources. Although, a direct comparison of a state-of-the-art recognition system has not been possible, we strongly feel that this decoder is representative of state-of-the-art when compared to similar systems reported in 1997 Johns Hopkins Summer Workshop [9].

The addition of Baum-Welch training, and decision tree clustering will bring the ISIP Speech-to-Text toolkit closer to similar proprietary systems. At the time of this publication, cross-word lattice generation had just been brought on-line. It is expected that the two additions to the training module and cross-word triphone lattice generation will further strengthen ISIP position that the Speech-to-Text system will provide the speech research community a toolkit which is state-of-the-art in both performance and software design.

In May of 1999, ISIP released a new web site, *http://www.isip.msstate.edu/asr*, making the latest version of the ISIP Speech-to-Text system available to all and introduced a new capability that allows speech recognition job submission over the Internet.

## 8. REFERENCES

[1] J. Godfrey, E. Holliman, and J. McDaniel, “SWITCHBOARD: Telephone Speech Corpus for Research and Development,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, California, USA, pp. 517-520, March 1992.

[2] *http://www.ldc.upenn.edu*, Linguistic Data Consortium, University of Pennsylvania, USA, 1996.

[3] S.J. Young, N.H. Russell and J.H.S. Thornton, “Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems,” *Cambridge University Engineering Department Technical Report* CUED/F-INFENG/TR.38, Cambridge University, UK, 1989.

[4] J.J. Odell, V. Valtchev, P.C. Woodland and S.J. Young, “A One-Pass Decoder Design for Large Vocabulary Recognition,” in *Proceedings of the DARPA Human Language Technology Workshop*, pp. 405-410, March 1995.

[5] R. Rosenfeld, “A Maximum Entropy Approach to Adaptive Statistical Language Modeling,” Computer, Speech and Language, vol. 10, pp.187-228, 1996.

[6] J.H. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1979.

[7] S. Ortmanns, H. Ney and A.Eiden, “Language Model Look-ahead for Large Vocabulary Speech Recognition,” in *Proceedings of the Fourth International Conference on Spoken Language Processing*, pp. 2095-2098, October 1996.

[8] R. Cole, et al., “Alphadigit Corpus,” http://www.cse.ogi.edu/CSLU/corpora/alphadigit, Center for Spoken Language Understanding, Oregon Graduate Institute, 1997.

[9] F. Jelinek, “1997 Large Vocabulary Continuous Speech Recognition Summer Research Workshop Technical Reports,” Center for Language and Speech Processing, Johns Hopkins University, Research Notes no. 30, January 1998.

[10] J. Hamaker, A. Ganapathiraju, and J. Picone, “A Proposal for a Standard Partitioning of the OGI AlphaDigit Corpus,” *http://isip.msstate.edu/projects/speech_recognition/research/syllable/alphadigits/data/ogi_alphadigits/eval_trans.text*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, U.S.A., August 1997.

[11] J. Picone, “Workshop 1997—Partitioning of the Switchboard Database”, *http://isip.msstate.edu/projects/speech_recognition/research/syllable/switchboard/data*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, U.S.A., August 1997.

# A PUBLIC DOMAIN SPEECH-TO-TEXT SYSTEM

*M. Ordowski+, N. Deshmukh*, A. Ganapathiraju*, J. Hamaker*, and J. Picone**

*Institute for Signal and Information Processing
Department for Electrical and Computer Engineering
Mississippi State University, Mississippi State, MS 39762 USA
{deshmukh, ganapath, hamaker,picone}@isip.msstate.edu
+Department of Defense, Ft. Meade, MD 20755, USA
mordow@afterlife.ncsc.mil

## ABSTRACT

The lack of freely available state-of-the-art Speech-to-Text (STT) software has been a major hindrance to the development of new audio information processing technology. The high cost of the infrastructure required to conduct state-of-the-art speech recognition research prevents many small research groups from evaluating new ideas on large-scale tasks. In this paper, we present the core components of an available state-of-the-art STT system: an acoustic processor which converts the speech signal into a sequence of feature vectors; a training module which estimates the parameters for a Hidden Markov Model; a linguistic processor which predicts the next word given a sequence of previously recognized words; and a search engine which finds the most probable word sequence given a set of feature vectors.

## 1. INTRODUCTION

Over the years, the complexity of the Speech-to-Text (STT) tasks have increased by many factors. In particular, the Switchboard [1] conversational speech recognition task exemplifies this situation. Unfortunately, the infrastructure required to deal with this increased complexity has also been magnified. The Institute for Signal and Information Processing (ISIP) has been committed to providing the research community with free software tools for digital information processing via the Internet to facilitate worldwide synergistic development of speech recognition technology. Our primary goal is to leverage state-of-the-art STT technology and harness the Internet as a means to share resources and provide unrestricted global access to the relevant software and data. To that end, ISIP hopes to lessen the burden of infrastructure requirements.

The ISIP Speech-to-Text system readily provides to the speech research community a toolkit that is capable of handling complex tasks such as Switchboard and has the flexibility to handle multilingual conversational speech recognition tasks such as Call Home [2].

As we approach our ultimate vision, our goals for the ISIP Speech-to-Text system include the following:

- unrestricted access via the Internet
- detailed documentation and operating instructions
- a state-of-the-art system with periodic upgrades
- object-oriented software design
- on-line technical support

The focus of this paper is on the core of any Speech-to-Text system, the decoder. A decoder is perhaps the sole determinant of whether an STT system is state-of-the-art in both performance and software design. Experiments will be presented to show that the decoder is state-of-the-art and we leave it will be up to user to determine if the software design is flexible to alteration. Next, we will cover the basics elements in acoustic processing and training which complete the Speech-to-Text system. Finally, some experiments with the complete system will be presented.

## 2. ISIP DECODER

A state-of-the-art public domain decoder needs to efficiently and transparently handle tasks of varied complexity, from connected digits to spontaneous conversations. The ISIP decoder can be executed in several different modes of operation.

- word graph acoustic/language model rescoring
- network grammar decoding
- n-gram decoding
- word graph generation
- forced alignments

### 2.1 DECODER FUNCTIONALITY

The core search algorithm used in the ISIP decoder is based on a hierarchical variation of the Viterbi-style time-synchronous search paradigm [3]. At each frame of the utterance, the system maintains a complete history for each active path at each level in the search hierarchy via special 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. The path score and a backpointer to its predecessor are also maintained.

## 2.2 DECODER PRUNING

The ISIP decoder employs two different heuristic pruning techniques to prevent growth of low-scoring hypotheses. Each of these has significant impact on the memory requirements and execution time of the decoder.

*Beam pruning*: The decoder allows the user to set a separate beam at each level in the search hierarchy (word, model, state). 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. At time $t$, the best scoring hypothesis in state $s$, model $m$, or word $w$ has a path score given by

$$Q_{max}(x, t) = \forall x, max\{Q(x, t)\}$$

where $x$ is either $s$, $m$, or $w$. For each state, model, or word beam width of $B_x$, we prune all hypotheses $(x, t)$ such that

$$Q(x, t) < Q_{max}(x, t) + B_x$$

where $x$ is either $s$, $m$, or $w$. Since the identity of a word is known with a much higher likelihood at the end of the word, a word-level beam is usually set to a much tighter value compared to the other two beams.

*Maximum active phone model instance (MAPMI) pruning*: By setting an upper limit on the number of active model (triphone) instances per frame, we can effectively regulate the memory usage of the decoder [4]. If the number of active hypotheses exceeds a limit, then only the best hypotheses are allowed to continue while the rest are pruned off.

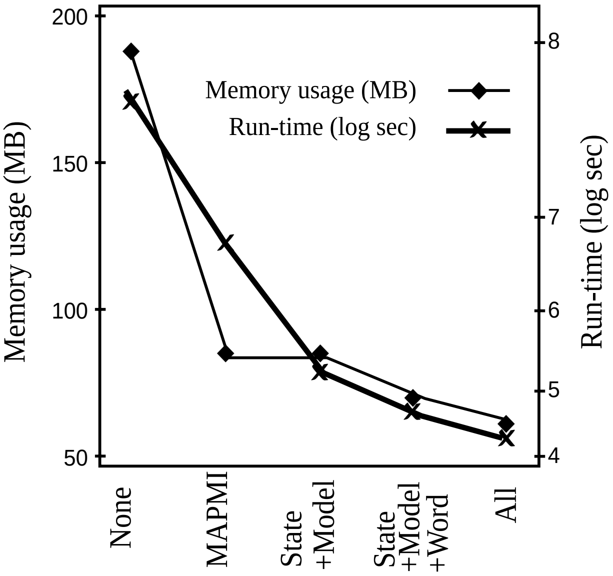

Figure 1: Effect of various pruning criteria using the ISIP decoder for a typical Switchboard utterance.

Effective pruning is critical in recognition tasks similar to Switchboard. The reason for this is often not well understood within the community. Given a situation where the acoustic models are poor matches to the acoustic data, as exemplified in telephone based applications like Switchboard that involve spontaneous speech, the number of hypotheses which are close in probability to the correct hypothesis is large. As a result, an explosive fan out of the unpruned hypotheses will occur unless strict attention is paid to pruning methods. Figure 1, shows the effects for a typical Switchboard lattice rescoring application using the ISIP decoder. As expected, the MAPMI pruning applies strict limits on memory usage. The state, model and word beams provide more direct control in preventing the fan-out caused by surviving end traces. This is one aspect that makes the ISIP decoder different from a standard Viterbi time-synchronous search paradigm.

## 2.3 Lexical Processing

The same lexical processor is used in all decoding modes. In n-gram decoding, the linguistic search space is constrained by a probabilistic model such as a bigram or trigram (with back-off) to predict the next possible word(s) [5]. The ISIP decoder can generate word graphs by constraining the linguistic search space either by a grammar or a n-gram language model. The Speech-to-Text system includes the software to take a grammar constructed in a Bakis-Naur Form [6] to a word graph that can be used as input to the decoder.

The pronunciation of a word is stored in a lexical tree. The ISIP decoder uses one lexical tree, and acquires language model weights on an as-needed basis. This implementation has two benefits. First, this implementation makes for a more efficient n-gram decoding scheme. In this case, a virtual copy of the tree is created for each n-gram word history (a dynamic tree approach). Second, the lexical tree is an efficient scheme to incorporate the language model probabilities early in the search stage [7].

## 2.4 Memory Management

All major data structures are handled by a centralized memory manager which works to minimize the memory load of the system by reusing as much memory as possible, and allocating memory in large blocks (thereby minimizing the overhead in creating memory).

## 2.5 Decoder Resource Usage

The current version of the ISIP decoder supports several modes of lattice rescoring and lattice generation. Figures 2, 3 and 4 show the resource usage of the system in some of these modes on utterances of different durations on the Switchboard task. All benchmarks described below are derived from experiments run on a 333MHz Pentium II processor with 512MB memory.

# 3. ACOUSTIC PROCESSOR

The ISIP Speech-to-Text system uses the most common front-end used in speech recognition systems today. The signal processing module produces industry-standard cepstral coefficients, coefficients that describe their derivatives (deltas) and acceleration (double-deltas). Key features, such as cepstral mean subtraction have been implemented.

Several other features that make a signal processing module robust to noise and compensate for the artifacts of frame-based computations have also been implemented. The user is able to choose from a wide range of windowing functions to include the standard Hamming and Hanning windows. The standard signal processing features like, pre-emphasis and energy normalization are also supported.

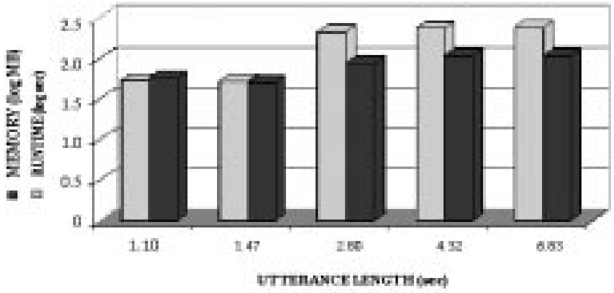

Figure 2: Cross-word triphone lattice rescoring using the ISIP decoder on a typical Switchboard utterance.

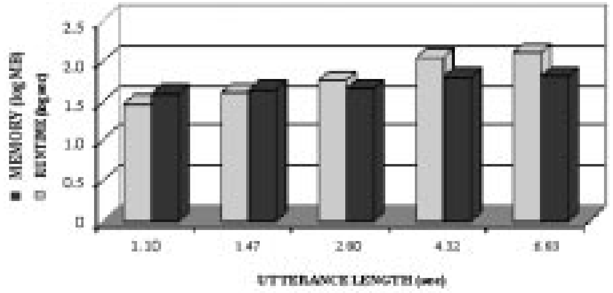

Figure 3: Word-Internal triphone lattice rescoring using the ISIP decoder for typical Switchboard utterance.

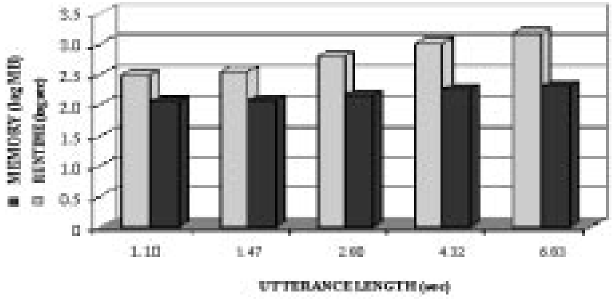

Figure 4: Lattice Generation using the ISIP decoder on a typical Switchboard utterance.

## 4. TRAINING

Training the Speech-to-Text system is currently organized as a set of utilities in which the Viterbi training module forms the core. The choice of Viterbi training was motivated by the simplicity, ease of implementation, and minimal modifications to the decoder to form a complete Speech-to-Text system. Baum-Welch training will be available in the very near future.

The organization of the training paradigm has been implemented in a distributed fashion. This has a major advantage for large task such as Switchboard [1]. Training in a distributed manner allows for training to be restarted from various stages. Allowing better management of the vast amounts of disk space needed and hundreds of hours to compute recognition models on large tasks.

The following utilities are available in training:

- Initialization of monophone models using global mean and diagonal covariance
- The core Viterbi training module
- Initialization of context-dependent models from a clustering process
- Generation of an arbitrary number of Gaussian mixture components

## 5. SOFTWARE AND INTERFACE

The ISIP decoder is designed in an object-oriented fashion and written completely in C++. For efficient access and sorting purposes the principal data structures are handled via linked lists and hash tables. Efficient modules for memory management ensure that used memory is periodically freed and reused. The software structure allows for a hierarchical representation of search space extensible to higher levels such as phrases and sentences. Hooks are provided to apply various kinds of acoustic distributions.

The decoder includes a Tcl-Tk based graphical interface (GUI) that allows the user to specify various decoding parameters through a simple point-and-click mechanism. It also provides a frame-by-frame display of the top word hypotheses, triphones and memory statistics; thus serving as a debugging and educational tool.

## 6. EXPERIMENTS AND RESULTS

ISIP has conducted several verification experiments to show that the Speech-to-Text system performs at a level which is competitive with state-of-the-art systems. ISIP has found it to be a difficult challenge to run a controlled experiment which puts a known state-of-the-art decoder against the ISIP decoder in a head-to-head competition. In the absence of the ultimate experiment, we present recognition results from a complete system for the Alpha-Digits [8] task and an evaluation on Switchboard from the 1997 Summer Workshop [9] at Johns Hopkins University.

At ISIP, the OGI Alpha-Digits task has been used to test several systems in the past. This task consists of approximately 3,000 subjects, each of whom spoke some subset of 1,102 phonetically-balanced prompting strings of letters and digits. This experiment used the ISIP training toolkit to train cross-word triphone models. The training set consisted 52,545 utterances from 2,237 speakers. Disjoint from the training set, the evaluation set was made up of 3,329 utterances from 739 speakers [10]. Table 1 shows the breakdown of errors.

| | WER | Subs. | Dels. | Ins. |
|---|---|---|---|---|
| ISIP | 15.6% | 13.8% | 1.0% | 0.8% |

Table 1: ISIP Speech-to-Text system on Alpha-Digits

The final system presented in this paper is the Switchboard recognizer. The system presented here is missing some prominent features such as vocal tract length normalization and speaker adaptation, and focuses on a core acoustic modeling system using context-dependent phones. The acoustic models were trained from 60 hours of data [11]. Using the ISIP system, lattices were generated on a evaluation set of 2,437 utterances [11] using word-internal triphone models and a bigram language model. The lattices were then rescored using cross-word triphone models. Table 2, shows the breakdown of errors of this system.

| | WER | Subs. | Dels. | Ins. |
|---|---|---|---|---|
| ISIP | 47.3% | 32.2% | 11.4% | 3.7% |

Table 2: ISIP Speech-to-Text system on Switchboard

## 7. CONCLUSIONS

We have presented a public domain Speech-to-Text system where the core module, the decoder, is state-of-the-art in terms of recognition performance as well as consumption of CPU and memory resources. Although, a direct comparison of a state-of-the-art recognition system has not been possible, we strongly feel that this decoder is representative of state-of-the-art when compared to similar systems reported in 1997 Johns Hopkins Summer Workshop [9].

The addition of Baum-Welch training, and decision tree clustering will bring the ISIP Speech-to-Text toolkit closer to similar proprietary systems. At the time of this publication, cross-word lattice generation had just been brought on-line. It is expected that the two additions to the training module and cross-word triphone lattice generation will further strengthen ISIP position that the Speech-to-Text system will provide the speech research community a toolkit which is state-of-the-art in both performance and software design.

In May of 1999, ISIP released a new web site, *http://www.isip.msstate.edu/asr*, making the latest version of the ISIP Speech-to-Text system available to all and introduced a new capability that allows speech recognition job submission over the Internet.

## 8. REFERENCES

[1] J. Godfrey, E. Holliman, and J. McDaniel, “SWITCHBOARD: Telephone Speech Corpus for Research and Development,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, California, USA, pp. 517-520, March 1992.

[2] *http://www.ldc.upenn.edu*, Linguistic Data Consortium, University of Pennsylvania, USA, 1996.

[3] S.J. Young, N.H. Russell and J.H.S. Thornton, “Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems,” *Cambridge University Engineering Department Technical Report* CUED/F-INFENG/TR.38, Cambridge University, UK, 1989.

[4] J.J. Odell, V. Valtchev, P.C. Woodland and S.J. Young, “A One-Pass Decoder Design for Large Vocabulary Recognition,” in *Proceedings of the DARPA Human Language Technology Workshop*, pp. 405-410, March 1995.

[5] R. Rosenfeld, “A Maximum Entropy Approach to Adaptive Statistical Language Modeling,” Computer, Speech and Language, vol. 10, pp.187-228, 1996.

[6] J.H. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1979.

[7] S. Ortmanns, H. Ney and A.Eiden, “Language Model Look-ahead for Large Vocabulary Speech Recognition,” in *Proceedings of the Fourth International Conference on Spoken Language Processing*, pp. 2095-2098, October 1996.

[8] R. Cole, et al., “Alphadigit Corpus,” http://www.cse.ogi.edu/CSLU/corpora/alphadigit, Center for Spoken Language Understanding, Oregon Graduate Institute, 1997.

[9] F. Jelinek, “1997 Large Vocabulary Continuous Speech Recognition Summer Research Workshop Technical Reports,” Center for Language and Speech Processing, Johns Hopkins University, Research Notes no. 30, January 1998.

[10] J. Hamaker, A. Ganapathiraju, and J. Picone, “A Proposal for a Standard Partitioning of the OGI AlphaDigit Corpus,” *http://isip.msstate.edu/projects/speech_recognition/research/syllable/alphadigits/data/ogi_alphadigits/eval_trans.text*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, U.S.A., August 1997.

[11] J. Picone, “Workshop 1997—Partitioning of the Switchboard Database”, *http://isip.msstate.edu/projects/speech_recognition/research/syllable/switchboard/data*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, U.S.A., August 1997.