Applying SPHINX-II to the DARPA Wall Street Journal CSR Task

F. Alleva, H. Hon, X. Huang, M. Hwang, R. Rosenfeld, R. Weide

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

This paper reports recent efforts to apply the speaker-independent SPHINX-II system to the DARPA Wall Street Journal continuous speech recognition task. In SPHINX-II, we incorporated additional dynamic and speaker-normalized features, replaced discrete models with sex-dependent semi-continuous hidden Markov models, augmented within-word triphones with between-word triphones, and extended generalized triphone models to shared-distribution models. The configuration of SPHINX-II being used for this task includes sex-dependent, semi-continuous, shared-distribution hidden Markov models and left context dependent between-word triphones. In applying our technology to this task we addressed issues that were not previously of concern owing to the (relatively) small size of the Resource Management task.[1]

1. Introduction

Extending a continuous speech recognition system to a larger vocabulary and more general task domain requires more than a new dictionary and language model. The primary problem in the application of the SPHINX-II [1] system to the Wall Street Journal (WSJ) CSR task was to extend the Viterbi beam-search used in the SPHINX [2] system to be able to run experiments given the constraints of available processing and memory resources.

First, we developed a practical form of between-word co-articulation modeling that was both time and memory efficient. The use of left context dependent between-word triphones is a departure from the left and right between-word context modeling but it allows the system to retain partial between-word co-articulation modeling despite the size and complexity of the task. Second, we significantly reduced the size of the memory required. To reduce the memory requirements of our search component it was necessary to change the Viterbi evaluation to use an in-place algorithm instead of a non-in-place one. Additionally we replaced the stack data structure used to recover the word sequence from the search, with a dictionary data structure. We decoupled the proto-type HMM state transition probabilities from the word specific HMM instances to avoid duplicating memory. We also found that our pointerless implementation of the HMM topology saved us both memory and time. Finally, we improved decoding efficiency substantially. One way to improve decoder efficiency is to reduce the search space. SPHINX-II reduces the search space with three pruning thresholds that are applied at the state, model, and word levels. In addition, evaluating a state requires an acoustic score computation and a graph update operation. Both of these operations run in constant time over one state. For discrete models, the cost of computing the acoustic score was on a par with the graph update operation since the acoustic score was computed by table lookup. With the introduction of semi-continuous models the cost of computing the acoustic score in the straight forward implementation is as much as an order of magnitude greater than the discrete model. This increase directly effects the overall time required by the search. To address this problem we decomposed the search into four phases. Shared distribution probability computation, HMM arc evaluation, active HMM instance evaluation and language model application. The shared distribution probability computation and HMM arc evaluation allow us to share computations that potentially would be repeated many times. Lastly, the introduction of full backoff language models made the previous approach of precomputing the entire table of non-zero arc probabilities impractical. For the SPHINX-II CSR decoder we use a cache table of active states in the language model to reduce the cost of accessing the language model.

2. Review of the SPHINX-II System

In comparison with the SPHINX system [2], the SPHINX-II system [1] has reduced the word error rate by more than 50% on most tasks by incorporating between-word coarticulation modeling [3], high-order dynamics [4], sex-dependent semi-continuous hidden Markov models [4], and shared-distribution models [5]. This section will

[1] This research is sponsored by the Defense Advanced Research Projects Agency, DoD, through ARPA Order 7239, and monitored by the Space and Naval Warfare Systems Command under contract N00039-91-C-0158. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or of the United States Government.

review SPHINX-II that will be used as the baseline acoustic modeling system for this study.

2.1 Signal Processing

The input speech signal is sampled at 16 kHz with a preemphasized filter, 1 - 0.9 Z^{-1}. A Hamming window with a width of 20 msec. is applied to the speech signal every 10 msec. A 32nd-order LPC analysis is used to compute the 12th-order cepstral coefficients. A bilinear transformation of cepstral coefficients is employed to approximate the mel-scale representation. In addition, relative power is also computed together with cepstral coefficients. The speech features used in SPHINX-II include LPC cepstral coefficients; 40-msec. and 80-msec differenced LPC cepstral coefficients; second-order differenced cepstral coefficients; and power, 40-msec differenced power, second-order differenced power. These features are vector quantized into four independent codebooks by the Linde-Buzo-Gray algorithm [6], each of which has 256 entries.

2.2 Training

Training procedures are based on the forward-backward algorithm. Word models are formed by concatenating phonetic models; sentence models by concatenating word models. There are two stages of training. The first stage is to generate the shared-distribution mapping table. Forty-eight context-independent *discrete* phonetic models are initially estimated from the uniform distribution. Deleted interpolation [7] is used to smooth the estimated parameters with the uniform distribution. Then context-dependent models are estimated based on the context-independent ones. There are 16,713 triphones in the DARPA WSJ-CSR training corpus when both within-word and left-context-dependent between-word triphones are considered. To simplify training, one codebook discrete models were used, where the acoustic features consist of the cepstral coefficients, 40-msec differenced cepstrum, and power and 40-msec differenced power. After the 16,713 discrete models are obtained, the shared-distribution clustering procedure [5] is applied to create the senones, 6255 in the case of the WSJ-CSR task. The second stage is to train 4-codebook models. We first estimate 51 context independent, four-codebook discrete models with the uniform distribution. With these context independent models and the senone table, we then estimate the shared-distribution SCHMMs. Because of substantial difference between male and female speakers, two sets of sex-dependent SCHMMs are are separately trained to enhance performance.

To summarize, the configuration of the SPHINX-II for WSJ-CSR system is:

- four codebooks of acoustic features,
- semi-continuous, shared-distribution triphones models, over
- left-context-dependent between-word and within-word triphone models,
- sex-dependent SCHMMs.

2.3 Recognition

For each input utterance, the *artificial* sex is first determined automatically [8, 9]. After the sex is determined, only the models of the determined sex are activated during recognition. This saves both time and memory. For each input utterance, a Viterbi beam search is used to determine the optimal state sequence in the language network.

3. New Techniques for CSR Decoding

3.1 Left Context Dependent Cross-Word Models

Using context dependent acoustic models across word boundaries presents two problems. The first of which is training the models and the second of which is using them in a decoder. The training problem is a relatively simple one. Since we are using a supervised training procedure it is simply a matter of transcribing the acoustic sequence to account for the cross-word phonetic context. An additional complication is introduced when optional silences can appear between words but this is also relatively easy to deal with by adding the appropriate optional phonetic sequences. One question that does arise is whether context dependent models for word beginning, word ending and word middle should be considered separately. In SPHINX-II they are kept separate [10].

The decoding problem is difficult since instead of a single word sequence to consider there are many alternative word sequences to consider. Consider the extension of a single word sequence $W_{1..n}$. Each possible one word extension of W gives rise to a particular phonetic right context at the end of w_n. There may be as many as N of these, where N is the number of basic phonetic units in the system. A similar problem appears when considering the best word sequence prior to a word w_{n+1}, each possible prior word, w_n, gives rise to a particular phonetic left context for the start of w_{n+1}. The final case to consider is a word that is exactly one phonetic unit in length. Here the number of possibilities to consider is order N^2. None the less, for small tasks (< 1000 words) with artificial grammars, it is possible to precompile only the relevant phonetic transitions since not all possible transitions will be allowed by the artificial grammar. When a larger and more natural task is considered, one such as WSJ CSR, these techniques are

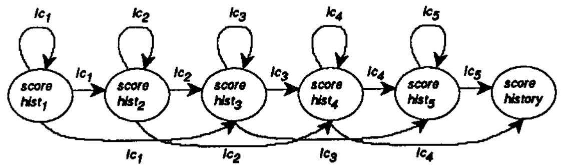

Figure 1: When decoding with the Bakis model the output distributions, lc_i, depend only on the name of the model. In the multiplexed Bakis model each lc_i is a function of the model name *and* the word sequence history, $hist_i$.

not applicable because of memory and run time constraints.

We made two important modifications in the application of cross-word context dependent phonetic models. The first was to model only the left context at word beginnings and ignore the right context at word endings. The second was to use the word-sequence-history information in each state to select the appropriate left context model for that state. See figure 1. An advantage afforded by left-context-only-modeling is that on each inter-word transition only one context is considered since the left context is uniquely determined by the word history $W_{1..n}$. If the right context is modeled all possible right contexts must be considered at word endings since the future is not yet known. The advantages afforded by using the best-word-sequence to select the appropriate left context model come in both space and time savings. Space is saved since only one model is needed at word beginnings rather than *N*. Time is saved since only one model is evaluated at word beginnings.

3.2 Memory Organization

The WSJ-CSR task is significantly different from the previous CSR tasks in the size of the lexicon and in the style of the language model. The lexicon is nearly an order of magnitude larger than previous lexicons and the language model contains more than two orders of magnitude more transitions than the Resource Management task. Several changes were required in the decoder design so that it could be run with out paging to secondary storage because of limited memory. Our redesign entailed changing the Viterbi evaluation to use an in-place algorithm, changing the management of history pointers to use a hash table rather than a stack, decoupling the proto-type HMM state transition probabilities from the word specific HMM instances, and changing from a statically compiled language model to dynamically interpreted language model. Finally, the pointerless implementation of the HMM topology continued to save both memory and time.

In Place Viterbi Evaluation. In our previous decoder the Viterbi evaluation used a separate set of source and destination states. The advantage to this approach is that states may be updated without regard to order. The disadvantage to this approach is that two sets fields must be kept for each state. By changing to an in-place evaluation only one set of fields is needed. Another feature of the previous decoder was that a word HMM was instantiated by making a copy of the appropriate HMMs and concatenating them together. As result duplicate copies of the arc transition probabilities would be made for each occurrence of HMM_i in a word. To save this space a pointer to the proto-type HMM is kept in the instance HMM and the arc transition probabilities are omitted.

The pointerless topology is a feature of the previous decoder [11] that implicitly encodes the topology of the model in the evaluation procedure. Not only does this save the memory and time associated with pointer following but it also allows, at no additional cost, the order dependent evaluation required by the in place Viterbi evaluation. Taken together these changes reduced the per state memory cost from 28 bytes/state to 8 bytes/states.

History Pointers and Language Model. By using a *dictionary* data structure instead of a *stack* data structure we reduced the amount of memory devoted to the word history sequences by an order of magnitude. The reduction comes because the *dictionary* does not differentiate identical word histories with differing segmentations. Besides the memory savings an advantage to this approach is that word histories can be rapidly compared for equality. A disadvantage is that the true segmentation cannot be recovered using this data structure. Finally, a consequence of using a fully backed-off language model is that it was no longer practical to precompile a graph that encoded all the language model transitions. Instead the language model is dynamically interpreted at run time.

3.2 Search Reduction

Viterbi beam search depends on the underlying dynamic programming algorithm that restricts the number of states to be |S|, where is S is the set of Markov states. For the bigram language model |S| is a linear function of *W*, the size of the lexicon. Therefore the time to decode an utterance is O(|S| * *l*) where *l* is the length of the input. The problem, at least when bigram language models are used, is not to develop a more efficient algorithm but to develop strategies for reducing the size of S. Beam search does this by considering only those states that fall with in some beam. The beam is defined to be all those states *s*, where *score(s)* is with in ε of the *best_score*(S). In the WSJ-CSR task the size of S has increased by almost an order of magnitude. With this motivation a refinement of the beam search strategy was developed that reduces the number of the states kept in the beam by a factor of two.

In the previous implementation of the decoder the beam was defined as $beam = \{s \mid score(s) > \sigma + best\_score(\mathrm{S})\}$. To further reduce the size of the beam two additional pruning thresholds have been added. The first threshold, π, is nominally for phone level pruning and the second, ω, is nominally for word level pruning. The set of states, **P** that π is applied to corresponds to the final (dummy) states of each instance of a phonetic model. The set of states **W**, that ω is applied to corresponds to the final (dummy) states of the final phonetic models of each word. The inequality relationship among the three beam thresholds is given by eqn. 1. The set containment relationship among the three sets is given by eqn. 2.

1. $\sigma \geq \pi \geq \omega$ 2. $\mathbf{S} \supset \mathbf{P} \supset \mathbf{W}$.

The motivation for partitioning the state space into subsets of states that are subject to different pruning thresholds comes from the observation that leads to the use of a pruning threshold in the first place. A state s is most likely to participate in the final decoding of the input when $score(s)$ is closest to $best\_score(\mathrm{S})$. Similarly a phonetic sub-word unit is most likely to participate in the final decoding when $score(p)$ is closest to $best\_score(\mathrm{S})$. Likewise for the word units. The difference between the state sets **P** and **W** and the state set **S** is that there is more than a single state of contextual information available. Put another way, when there is more information a tight pruning threshold can be applied with out an increase in search errors. Currently all the pruning thresholds are determined empirically. Informally we have found that the best threshold settings for π and ω are two and four orders of magnitude tighter than σ.

3.3 Search Decomposition

The search is divided into four phases.

1. shared distribution probability computation
2. HMM arc probability evaluation
3. active HMM instance evaluation
4. language model application

For each time frame the shared distribution probability computation first computes the probabilities of the top $N=4$ codewords in the codebook. Then the top N codewords and their probabilities are combined with each of $D=6255$ discrete output probability distribution functions. Although not all distributions will be used at every frame of the search a sufficiently large number are used so that computation on demand is less efficient.

The D output probabilities are then combined with the $M=16,713$ models in the HMM arc probability evaluation. Here we only compute the arc probabilities of those HMMs that have active instances as part of a word. Two advantages accrue from separating the arc probability computation from the state probability computation. First the arc transition probability and acoustic probability need only be combined once. Second this naturally leads to storing HMM arc transition probabilities separately from the HMM instances which results in a space savings.

Decoder Development Summary			
Condition	Error %	Size (Mb)	× Real Time
baseline	24.7%	172	167
+ left context	19.5%		
+ Intr. Lang. Model		77	217
+ Word Hist. Dict.		57	
+ Inplace Viterbi		53	
+ Multiple Pruning			63
+ Acoustic Score			53
+ HMM Arc			46
+ LM. Cache	19.5%	57	40

Table 1: The effect of each change to the decoder is summarized in terms of error rate, memory size and run time. The baseline result refers to the results obtained with original decoder that implemented no cross word modeling.

The active HMMs, ie. those HMM instances corresponding to phones in an active word, are updated with arc probabilities from the corresponding HMM protc-type. In this case updating an HMM means combining all the HMM instance state probabilities with the appropriate arc probabilities of the proto-type HMM and performing the Viterbi update procedure.

For each word history h ending at time t the language model is consulted for the vector of probabilities corresponding to the probability of each of one word extension of h. Between the language model and the word transition module sits a cache. For the WSJ-CSR 5000 word system, a 200 entry LRU[2] cache provides a hit rate of 92%. The cache reduces the cost of using this language model by an order of magnitude. For the a 5000 word lexicon, a 200 entry cache requires four megabytes.

4. WSJ-CSR Experimental Setup

The WSJ corpus consists of approximately 45-million words of text published by the Wall Street Journal between the years 1987 and 1989. This corpus was made available through the Association for Computation Linguistics/Data Collection Initiative (ACL/DCI) [12].

[2] LRU - least recently used

4.1. Language Models

For the purposes of the February dry run eight standard bigram language models were provided by D. Paul at Lincoln Labs [13]. The language models were trained only on the WSJ data that was not held out for acoustic training and testing. The language models are characterized along three dimensions, lexicon size (5k or 20k), closed or open vocabulary, and verbalized (vp) or non-verbalized pronunciation (nvp). The distinction between open closed vocabulary models is in the method used to chose the lexicon. For the open vocabulary the lexicon approximately consists of the *N* most common words in the corpus. For the closed vocabulary, a set of *N* words were selected in a manner that would allow the creation of a sub-corpus that would have 100% lexical coverage by this closed vocabulary. For further details see [14]. The development test set perplexities for the eight language models are given in **table 2.**

4.2. Training and Evaluation Acoustic Data Sets

The base line speaker independent training data set provided by the National Institute of Standards and Technology (NIST) [15] consisted of 7240 utterances of read WSJ text equally divided among VP and NVP texts. The texts chosen to train the system were quality filtered to remove very long and very short short sentences as well as removing sentences containing words not among the 64k most frequently occurring words in the WSJ corpus [13]. The data was collected from 84[3] speakers, equally divided among male and female persons. Data recording was performed at three different locations, MIT, SRI and TI. At all three locations the same close speaking, noise canceling microphone was used however environmental conditions vary from a sound both to a laboratory environment. At CMU we used a subset of the 7240 utterances, excluding 89 of the 7240 utterances because they contained cross talk or over-laying noise events as indicated by the detailed orthographic transcription (DOT) of the utterance.

	Lexicon Size			
	5k		20k	
	closed	open	closed	open
vp	80	72	158	135
nvp	118	105	236	198

Table 2: Perplexity of the eight standard language models on the development test set. VP - verbalized pronunciation. NVP - non-verbalized pronunciation.

The speaker independent evaluation data set consisted of eight data sets containing a total of 1200 utterances from 10 speakers. Again each data set was equally divided among male and female speakers. For further details on the evaluation test sets see [14].

4.3 Acoustic Configuration

The configuration of SPHINX-II for WSJ-CSR consists of 16,713 phonetic models that share 6255 semi-continuous distributions. For between word modeling only the left context is considered. There is no speaker normalization component or vocabulary adaptation component. The dictionary provided by Dragon Systems was programatically converted into the CMU style phonetic baseforms with some additional manual post processing to fix problems with the transcription of flaps /dx/.

4.4 Results

The official NIST results are given in the following table. Each line of the table gives results for a particular test from the si_evl test suite. The test sets are 5 (5000 word closed), 20 (20000 word closed), sp (spontaneous) and rs (read spontaneous). These four test sets are further subdivided to vp and nvp conditions. The final condition for each test is the language model used. For these tests only two models, 5c (5000 word closed) and 5o (5000 word open) were used. For further details on the testing datasets see [14]. The table is largely self explanatory other than the column labeled 2σ. This column is simply two times the standard deviation of the average word error rate computed from word error rates on a sentence by sentence basis. As expected the vp tests out perform the nvp tests and the the open language model out performs the closed language model when the test data set contains words from outside the language models lexicon. It should be noted however that the vp portion of the test is probably the more difficult set since when we remove the highly reliable punctuation words words from the scoring, the error rate for the remaining words is actually higher than the one obtained in the nvp case. We attribute this to the increased number of disfluencies caused by verbalized pronunciation and to the detrimental effect on the bigram language model.

5. Summary

The successful application of SPHINX-II to the WSJ-CSR task demonstrates the utility of distribution sharing for training a large number of triphones with a relatively small amount of data. We also have demonstrated the utility of the Viterbi-beam search for decoding in the context of a much larger task. Beyond the algorithmic improvements made to the decoder a major factor in reducing decoding time to just under 50 times real-time, is the availability of crisp acoustic models.

[3] One of the speakers in the training data set was recorded twice but at different sites and so this person is counted as two different speakers.

Sphinx II WSJ CSR Performance			
Test Condition	Insertion	Error	2σ
si_evl5.nvp-5c	2.1%	19.5%	± 2.38
si_evl5.vp-5c	3.0%	18.4%	± 2.47
si_evl5.5c	2.7%	18.9%	± 1.72
si_evl20.nvp-5o	7.5%	37.9%	± 3.30
si_evl20.vp-5o	6.6%	32.7%	± 3.34
si_evl20.5o	7.1%	35.2%	± 2.37
si_evl20.nvp-5c	7.6%	43.6%	± 3.56
si_evl20.vp-5c	6.9%	36.1%	± 3.43
si_evl20.5c	7.2%	39.6%	± 2.46
si_evlrs.nvp-5o	10.3%	50.4%	± 5.86
si_evlrs.vp-5o	7.7%	41.4%	± 4.29
si_evlrs.5o	8.9%	45.4%	± 3.46
si_evlsp.nvp-5o	11.7%	56.0%	± 5.64
si_evlsp.vp-5o	9.2%	45.5%	± 4.42
si_evlsp.vp	10.3%	50.2%	± 3.47

Future plans include introducing our speaker normalization and vocabulary adaptation technology as well as experimenting with longer range language models.

REFERENCES

1. Huang, X. and Alleva, F. and Hon, H. and Hwang, M. and Rosenfeld, R., "The SPHINX-II Speech Recognition System: An Overview", Technical Report CMU-CS-92-112, School of Computer Science, Carnegie Mellon University, February 1992.

2. Lee, K.F. and Hon, H.W. and Reddy, R., "An Overview of the SPHINX Speech Recognition System", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, January 1990, pp. 35-45.

3. Hwang, M.Y. and Hon, H.W. and Lee, K.F., "Modeling Between-Word Coarticulation in Continuous Speech Recognition", *Proceedings of Eurospeech*, Paris, FRANCE, September 1989, pp. 5-8.

4. Huang, X.D. and Alleva, F.A. and Hayamizu, S. and Hon, H.W. and Hwang, M.Y. and Lee, K.F., "Improved Hidden Markov Modeling for Speaker-Independent Continuous Speech Recognition", *DARPA Speech and Language Workshop*, Morgan Kaufmann Publishers, Hidden Valley, PA, June 1990, pp. 327-331.

5. Hwang, M.Y. and Huang, X.D., "Subphonetic Modeling with Markov States - Senone", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1992.

6. Linde, Y. and Buzo, A. and Gray, R.M., "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communication*, Vol. COM-28, No. 1, January 1980, pp. 84-95.

7. Jelinek, F. and Mercer, R.L., "Interpolated Estimation of Markov Source Parameters from Sparse Data", in *Pattern Recognition in Practice*, E.S. Gelsema and L.N. Kanal, ed., North-Holland Publishing Company, Amsterdam, the Netherlands, 1980, pp. 381-397.

8. Huang, X.D, "A Study on Speaker-Adaptive Speech Recognition", *DARPA Speech and Language Workshop*, Morgan Kaufmann Publishers, San Mateo, CA, Feb 1991.

9. Soong, F. and Rosenberg, A. and Rabiner, L. and Juang, B., "A Vector Quantization Approach to Speaker Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1985, pp. 387-390.

10. Hwang, M.Y. and Hon, H.W. and Lee, K.F., "Modeling Inter-Word Coarticulation Using Generalized Triphones", *The 117th Meeting of the Acoustical Society of America*, Syracuse, NY, May 1989.

11. Alleva, F., "Search Organization for Large Vocabulary Continuous Speech Recognition", *NATO ASI Speech Recognition and Understanding: Recent Advances, Trends and Applications*, 1990.

12. Liberman, M., "Text on Tap: the ACL/DCI", *DARPA Speech and Natural Language Workshop*, October 1989, pp. 173-188.

13. Paul, D.B., "New Results with the Lincoln Tied-Mixture HMM CSR System", *DARPA Speech and Language Workshop*, Morgan Kaufmann Publishers, San Mateo, CA, Feb 1991.

14. Paul, D.B., Baker, J.M., "The Design for the Wall Street Journal-based CSR Corpus", *DARPA Speech and Language Workshop*, Morgan Kaufmann Publishers, San Mateo, CA, Feb 1992.

15. National Institute of Standards and Technology, Pallet, D., "WSJ Pilot Corpus", Limited distribution CD-ROM, 1991.