THE SPHINX SPEECH RECOGNITION SYSTEM

**Kai-Fu Lee, Hsiao-Wuen Hon, Mei-Yuh Hwang,
Sanjoy Mahajan, Raj Reddy**

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper describes SPHINX, an accurate large-vocabulary speaker-independent continuous speech recognition system. An earlier version of SPHINX was described in ICASSP '88. We have since made several enhancements including generalized triphone models, word duration modeling, function-phrase modeling, between-word coarticulation modeling, and corrective training. On the 997-word resource management task, SPHINX attained a word accuracy of 96% with a grammar (perplexity 60), and 82% without grammar (perplexity 997).

1. Introduction

This paper describes SPHINX, a large-vocabulary speaker-independent continuous speech recognition system. SPHINX is based on discrete hidden Markov models (HMMs) with LPC-derived parameters. In order to deal with the problem of speaker independence, we added knowledge to these HMMs in several ways. We represented additional knowledge through the use of multiple codebooks. We also enhanced the recognizer with word duration modeling. In order to model co-articulation in continuous speech, we introduced the use of function-word-dependent phone models, function-phrase-dependent phone models, generalized triphone models, and between-word coarticulation modeling. More recently, we also modified the corrective training algorithm [1] for continuous speech recognition.

In this paper, we will describe the above components of the SPHINX System, with emphasis on the recent improvements. Interested reader may refer to [2] or [3] for more details.

On the 997-word DARPA resource management task, SPHINX achieved speaker-independent word recognition accuracies of 82% and 96%, with grammars of perplexity 997 and 60, respectively. Results on new test speakers are also presented.

2. Speech Representation

The speech is sampled at 16 KHz, and pre-emphasized with a filter of $1-0.97z^{-1}$. Then, a Hamming window with a width of 20 msec is applied every 10 msec. Autocorrelation analysis with order 14 is followed by LPC analysis with order 14. Finally, 12 LPC-derived cepstral coefficients are computed from the LPC coefficients, and these LPC cepstral coefficients are transformed to a mel-scale using a bilinear transform.

These 12 coefficients are vector quantized into a codebook of 256 prototype vectors. In order to incorporate additional speech parameters, we created two additional codebooks. One codebook is vector quantized from *differential coefficients*. The differential coefficient of frame n is the difference between the coefficient of frame n+2 and frame n–2. This 40 msec. difference captures the slope of the spectral envelope. The other codebook is vector quantized from *energy* and *differential energy* values.

3. Context-Independent HMM Training

SPHINX is based on phonetic hidden Markov models. We identified a set of 48 phones, and a hidden Markov model is trained for each phone. Each phonetic HMM contains three discrete output distributions of VQ symbols. Each distribution is the joint density of the three codebook pdf's, which are assumed to be independent. The use of multiple codebooks was introduced by Gupta, *et al.* [4].

We initialize our training procedure with the TIMIT phonetically labeled database. With this initialization, we use the forward-backward algorithm to train the parameters of the 48 phonetic HMMs. The training corpus consists of 4200 task-domain sentences spoken by 105 speakers. For each sentence, word HMMs are constructed by concatenating phone HMMs. These word HMMs are then concatenated into a large sentence HMM, and trained on the corresponding speech. Because the initial estimates are quite good, only two iterations of the forward-backward algorithm are run. This training phase produces 48 *context-independent* phone models. In the next two sections, we will discuss the second training phase for *context-dependent* phone models.

4. Function Word/Phrase Dependent Models

One problem with continuous speech is the unclear articulation of function words, such as *a, the, in, of*, etc. Since the set of function words in English is limited and function words occur frequently, it is possible to model each phone in each function word separately. By explicitly modeling the most difficult sub-vocabulary, recognition rate can be increased substantially. We selected a set of 42 function words, which contained 105 phones. We modeled each of these phones separately.

We have found that function words are hardest to recognize when they occur in clusters, such as *that are in the*. The

CH2673-2/89/0000-0445 $1.00 © 1989 IEEE

words are even less clearly articulated, and have strong inter-word coarticulatory effects. In view of this, we created a set of phone models specific to *function phrases*, which are phrases that consist of only function words. We identified 12 such phrases, modified the pronunciations of these phrases according to phonological rules, and modeled the phones in them separately. A few examples of these phrases are: *is the*, *that are*, and *of the*.

5. Generalized Triphone Models

The function-word and function-phrase dependent phone models provide better representations of the function words. However, simple phone models for the non-function words are inadequate, because the realization of a phone crucially depends on context. In order to model the most prominent contextual effect, Schwartz, *et al.* [5] proposed the use of *triphone models*. A different triphone model is used for each left and right context. While triphone models are sensitive to neighboring phonetic contexts, and have led to good results, there are a very large number of them, which can only be sparsely trained. Moreover, they do not take into account the similarity of certain phones in their affect on other phones (such as /b/ and /p/ on vowels).

In view of this, we introduce the *generalized triphone model*. Generalized triphones are created from triphone models using a clustering procedure:

1. An HMM is generated for every triphone context.
2. Clusters of triphones are created; initially, each clusters consists of one triphone.
3. Find the *most similar* pair of clusters which represent the same phone, and merge them.
4. For each pair of same-phone clusters, consider moving every element from one to the other.
    1. Move the element if the resulting configuration is an improvement.
    2. Repeat until no such moves are left.
5. Until some convergence criterion is met, go to step 2.

To determine the distance between two models, we use the following distance metric:

$$D(a,b) = \frac{(\prod_i (P_a(i))^{N_a(i)}) \cdot (\prod_i (P_b(i))^{N_b(i)})}{\prod_i (P_m(i))^{N_m(i)}} \quad (1)$$

where $D(a,b)$ is the distance between two models of the same phone in context a and b. $P_a(i)$ is the output probability of codeword i in model a, and $N_a(i)$ is the count of codeword i in model a. m is the merged model by adding N_a and N_b. In measuring the distance between the two models, we only consider the output probabilities, and ignore the transition probabilities, which are of secondary importance.

Equation 1 measures the ratio between the probability that the individual distributions generated the training data and the probability that the combined distribution generated the training data. Thus, it is consistent with the maximum-likelihood criterion used in the forward-backward algorithm. This distance metric is equivalent to, and was motivated by, entropy clustering used in [6] and [7].

This context generalization algorithm provides the ideal means for finding the equilibrium between trainability and sensitivity. Given a fixed amount of training data, it is possible to find the largest number of trainable detailed models. Armed with this technique, we could attack any problem and find the "right" number of models that are as sensitive and trainable as possible. This is illustrated in Figure 1, which shows that the optimal number of models increases as the training data is increased.

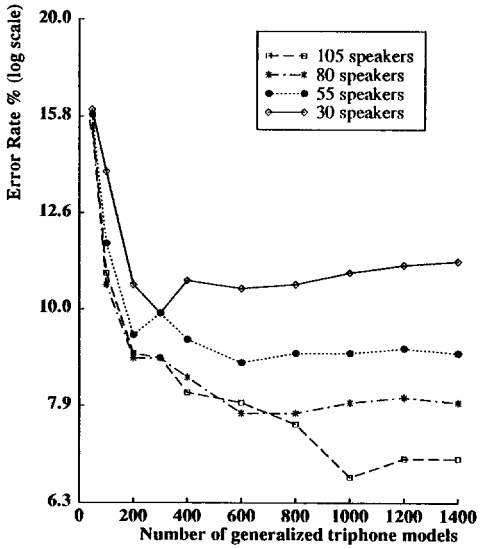

Figure 1: Error rate as a function of the amount of training and the number of models.

6. Between-Word Coarticulation Modeling

Triphone and generalized triphone models are powerful subword modeling techniques because they account for the left and right phonetic contexts, which are the principal causes of phonetic variability. However, these phone-sized models consider only intra-word context. A simple extension of triphones to model between-word coarticulation is problematic because the number of triphone models grows sharply when between-word triphones are considered. For example, there are 2381 within-word triphones in our 997-word task. But there are 7057 triphones when between-word triphones are also considered.

Therefore, generalized triphones are particularly suitable

for modeling between-word coarticulation. We first generated 7057 triphone models that accounted for both intra-word and inter-word triphones. These 7057 models were then clustered into 1000 generalized triphone models. Few program modifications were needed for training, since the between-word context is always known. However, during recognition, most words now have multiple initial and final states. Care must be taken to to ensure that each legal sentence has one and only one path in the search. Details of our implementation can be found in [8].

7. Summary of Training Procedure

The SPHINX training procedure operates in two stages. In the first stage, 48 context-independent phonetic models are trained. In the second stage, the models from the first stage are used to initialize the training of context-dependent phone models, which could be generalized triphone models and/or the function word/phrase dependent models.

Although we have 4200 sentences for training, this is still not sufficient to estimate the 2.5 million parameters in our models without smoothing. In order to estimate the probabilities of the unobserved and rare symbols, we interpolate the context-dependent model parameters with the corresponding context-independent ones. We use *deleted interpolation* [9] to derive appropriate weights in the interpolation.

The SPHINX training procedure is shown in Figure 2:

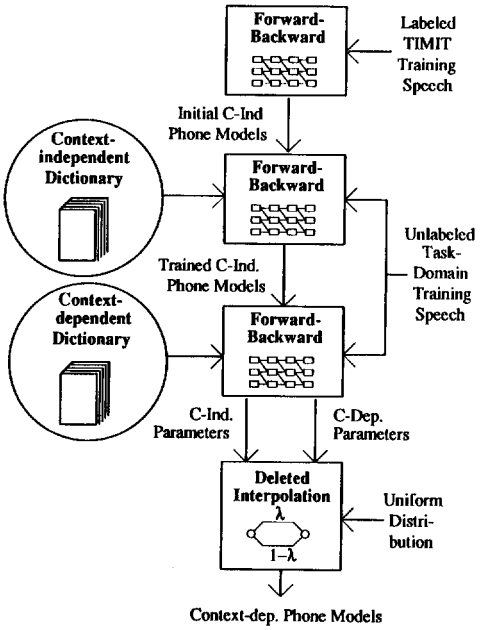

Figure 2: The SPHINX Training Procedure.

8. HMM Recognition with Word Duration

For recognition, we use a Viterbi search that finds the optimal state sequence in a large HMM network. At the highest level, this HMM is a network of word HMMs, arranged according to the grammar. Each word is instantiated with its phonetic pronunciation network, and each phone is instantiated with the corresponding phone model. Beam search is used to reduce the amount of computation.

One problem with HMMs is that they do not provide very good duration models. We incorporated word duration into SPHINX as a part of the Viterbi search. The duration of a word is modeled by a univariate Gaussian distribution, with the mean and variance estimated from a supervised Viterbi segmentation of the training set. By precomputing the duration score for various durations, this duration model has essentially no overhead.

9. Corrective Training

We have just completed several preliminary experiments with the corrective training algorithm proposed by Bahl, *et al.* [1]. The corrective training algorithm attempts to maximize the recognition rate on the training set. The algorithm proposed by Bahl, *et al.* was successfully applied to isolated-word speaker-dependent recognition. We modified the algorithm in several ways to deal with speaker-independent continuous speech recognition. This will be described in a forthcoming paper [10].

10. Results

The SPHINX System was tested on 150 sentences from 15 speakers. These sentences were the official DARPA test data for evaluations in March and October 1987. The word accuracies for various versions of SPHINX with the word-pair grammar (perplexity 60) and the null grammar (perplexity 997) are shown in Table 1. Word accuracy is defined as the percent of words correct minus the percent of insertions.

Version	No Grammar	Word Pair
1 Codebook	25.8%	58.1%
3 Codebooks	45.3%	84.4%
+Duration	49.6%	83.8%
+Fn-word	57.0%	87.9%
+Fn-phrase	59.2%	88.4%
+Gen-triphone	72.8%	94.2%
+Between-word	77.9%	95.5%
+Corrective	81.9%	96.2%

Table 1: Results of various versions of SPHINX.

We found duration modeling to be helpful when no grammar was used. Modeling function words and generalized triphones both led to substantial improvements. We also found that generalized triphones outperformed triphones, while saving 60% memory. More detailed descriptions and results on contextual modeling can be found in [2] or [3].

The improvements from function-phrase dependent modeling encouraged us to implement between-word triphone models. This led to substantial improvements with no increase in the number of models. Finally, we have also shown the effectiveness of our extension of the corrective training algorithm to speaker-independent continuous speech.

SPHINX was evaluated on the June 1988 test set, which contains 12 speakers, with 25 sentences per speaker. Thus far, we have only run the version of SPHINX that corresponded to the "Between-word 78.0% 95.7%" line in Table 1. We obtained accuracies of 70.2% and 93.0% on the two grammars. The reason for this degradation was that the first test set contained one extremely good speaker, while the June 1988 test set contained two extremely poor speakers. If we discard these three speakers, the performance of SPHINX on the remaining 24 speakers is consistent. This suggests that speaker-independent systems can work extremely well on a great majority of speakers, but speaker adaptation may be needed on atypical speakers.

11. Conclusion

This work addressed the problem of large-vocabulary speaker-independent continuous speech recognition. At the outset, we chose to use hidden Markov modeling, a powerful mathematical learning paradigm. We also decided to use vector quantization and discrete HMMs for expedience and practicality. Then we attacked the problems of large vocabulary, speaker independence, and continuous speech within our discrete HMM framework.

It is well known that HMMs will perform better with detailed models. It is also well known that HMMs need considerable training. This need is accentuated in large-vocabulary, speaker-independence, and discrete HMMs. However, given a fixed amount of training, model specificity and model trainability are two incompatible goals. More specificity usually reduces trainability, and increased trainability usually results in over-generality.

Thus, our work can be viewed as finding an equilibrium between specificity and trainability. To improve trainability, we used one of the largest speaker-independent speech databases. To facilitate sharing between models, we used deleted interpolation to combine robust models with detailed ones. By combining poorly trained (within and between word generalized triphone, function word and phrase dependent phone) models with well-trained (context-independent, uniform) models, we improved trainability through sharing.

To improve specificity, we used multiple codebooks of various LPC-derived features, and integrated external knowledge sources into the system. We also introduced the use of function-word-dependent phone modeling, function-phrase-dependent phone modeling, and generalized triphone modeling, both within and between words.

Through these techniques we demonstrated that accurate large-vocabulary speaker-independent continuous speech recognition is feasible. We report recognition accuracies of 82% and 96% with grammars of perplexity 997 and 60. These results were made possible by ample training data, powerful learning paradigms, and detailed modeling techniques.

Acknowledgments

The authors wish to thank the CMU Speech Group for their support and contributions. We would also like to thank DARPA and NSF for their support.

References

1. Bahl. L.R., Brown, P.F., De Souza, P.V., Mercer, R.L., "A New Algorithm for the Estimation of Hidden Markov Model Parameters", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1988.

2. Lee, K.F., *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System*, PhD dissertation, Computer Science Department, Carnegie Mellon University, April 1988.

3. Lee, K.F., *Automatic Speech Recognition: The Development of the SPHINX System*, Kluwer Academic Publishers, Boston, 1989.

4. Gupta, V.N., Lennig, M., Mermelstein, P., "Integration of Acoustic Information in a Large Vocabulary Word Recognizer", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1987, pp. 697-700.

5. Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., Makhoul, J., "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1985.

6. Lucassen, J.M., "Discovering Phonemic Baseforms: an Information Theoretic Approach", Research Report RC 9833, IBM, February 1983.

7. Brown, P., *The Acoustic-Modeling Problem in Automatic Speech Recognition*, PhD dissertation, Computer Science Department, Carnegie Mellon University, May 1987.

8. Hwang, M.Y., Hon, H.W., Lee, K.F., "Between-Word Coarticulation Modeling for Continuous Speech Recognition", Technical Report, Carnegie Mellon University, March 1989.

9. Jelinek, F., 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.

10. Lee, K.F., Mahajan, S., "Corrective and Reinforcement Learning for Speaker-Independent Continuous Speech Recognition", Technical Report CMU-CS-89-100, Carnegie Mellon University, January 1989.