The SPHINX-II Speech Recognition System: An Overview

Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon,
Mei-Yuh Hwang, Ronald Rosenfeld

January 15, 1992

CMU-CS-92-112

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In order for speech recognizers to deal with increased task perplexity, speaker variation, and environment variation, improved speech recognition is critical. Steady progress has been made along these three dimensions at Carnegie Mellon. In this paper, we review the SPHINX-II speech recognition system and summarize our recent efforts on improved speech recognition.

This research was sponsored by the Defense Advanced Research Projects Agency and monitored by the Space and Naval Warfare Systems Command under Contract N00039-91-C-0158, ARPA Order No. 7239.

Keywords: Speech recognition, hidden Markov models, SPHINX-II

1. INTRODUCTION

At Carnegie Mellon, we have made significant progress in large-vocabulary speaker-independent continuous speech recognition during the past years [1, 2, 3]. SPHINX is the first accurate large-vocabulary continuous speaker-independent speech recognition system [1]. Recently, the performance of the SPHINX system was significantly improved. This paper describes the SPHINX-II speech recognition system and summarizes our recent speech recognition efforts.

In order for speech recognizers to deal with increased task perplexity, speaker variation, and environment variation, improved speech recognition is critical. Steady progress has been made along those three dimensions. Some of our recent contributions include use of additional dynamic features, speaker-normalized features, semi-continuous hidden Markov models, subphonetic modeling, vocabulary-independent and -adaptive speech recognition, speaker adaptation, efficient search, and language modeling. We are currently refining and extending these and related technologies to develop practical unlimited-vocabulary dictation systems, and spoken language systems for more general application domains with larger vocabularies and reduced linguistic constraint.

One of the most important contributor has been our training data. When the amount of training data is increased, the modeling error can be dramatically decreased. However, more data require different models so that more detailed acoustic-phonetic phenomena can be well characterized. Towards this end, recent progress can be broadly classified into feature extraction, detailed representation through parameter sharing, application-related issues, search, and language modeling.

2. FEATURE EXTRACTION

How to extract reliable features is one of the most important issues in speech recognition. The training data also play a key role in this research. We need to remember the curse of dimensionality. Since the amount of training data is always limited, incorporation of additional features may not lead to any error reduction, which does not necessarily mean that added features are not good, but we don't have sufficient data to reliably model those added features. Many environment-robust [4] and speaker-robust [5] models have similar constraints.

2.1. Dynamic Features

Temporal changes in the spectra are believed to play an important role in human perception. One way to capture this information is to use delta coefficients or differenced coefficients that measure the change in coefficients over time. Temporal information is particularly suitable for HMMs, since HMMs assume each frame is independent of the past, and these dynamic features broaden the scope of a frame.

In the past, the SPHINX system has utilized three codebooks containing [6]: (1) 12 LPC cepstrum coefficients $x_t(k)$, $1 <= k <= 12$; (2) 12 differenced LPC cepstrum coefficients (40 msec. difference) $\Delta x_t(k)$, $1 <= k <= 12$; (3) Power and differenced power (40 msec.) $x_t(0)$ and $\Delta x_t(0)$. Since we are using the multiple-codebook-based hidden Markov model, it is easy to incorporate new features into an additional codebook. We experimented with a number of new measures of spectral dynamics, including: (1) second order differential cepstrum and power ($\Delta\Delta x_t(k)$, $1 <= k <= 12$, and $\Delta\Delta x_t(0)$) and third order differential cepstrum and power. The first set of coefficients is incorporated into a new codebook, whose parameters are second order differences of the cepstrum. The second order difference for frame t, $\Delta\Delta x_t(k)$, where t is in units of 10ms, is the difference between $t+1$ and $t-1$ first order differential coefficients, or

$$\Delta\Delta x_t(k) = \Delta x_{t-1}(k) - \Delta x_{t+1}(k) \tag{1}$$

Next, we incorporated both 40 msec. and 80 msec. differences, which represent short-term and long-term spectral dynamics, respectively. The 80 msec. differenced cepstrum $\Delta x'_t(k)$ is computed as:

$$\Delta x'_t(k) = x_{t-4}(k) - x_{t+4}(k) \tag{2}$$

We hoped that these two sources of information are more complementary than redundant. We incorporated both Δx_t and $\Delta x'_t$ into one codebook (packing the two into one frame), weighted by their variances. We attempted to compute optimal linear combination of cepstral segment, where weights are computed from linear discriminants. But we found that performance deteriorated slightly. This may be due to limited training data (curse of dimensionality) or there may be little information beyond second-order differences.

Thus, the final configuration involves four codebooks, each with 256 entries, that use (1) 12 LPC cepstrum coefficients; (2) 12 40-msec differenced LPC cepstrum coefficients and 12 80-msec differenced LPC cepstrum coefficients; (3) 12 second-order differenced cepstrum; and (4) Power, 40-msec differenced power, second-order differenced power. The new combination reduced errors by 15% over the baseline results.

2.2. Speaker-Normalized Features

The error rate of a well-trained speaker-dependent speech recognition system is typically two to three times less than that of a speaker-independent system [7]. Since between-speaker variability is one of the major error sources, we investigated the advantages and problems associated with speaker normalization [8]. We use neural networks to transform data of different speakers to the golden speaker cluster. The codeword-dependent networks are used to enhance the mapping quality [8].

As frame to frame normalization lacks use of dynamic information, the architecture of normalization network is chosen to incorporate multiple neighboring frames. Here, the current frame and its left and right neighboring frame are fed to the multi-layer neural network as inputs. The network output is a normalized frame corresponding to the current input frame. By using multiple input frames for the network, the important dynamic information can be effectively used in estimating network parameters and in normalizing the speech frames. When presented with a large amount of training data, a single network is often unable to produce satisfactory results during training as the network is only suitable to a relatively small task. To improve the mapping performance, decomposition and modular construction are usually required. One solution is to partition the mapping space into several smaller regions, and to construct a neural network for each region. As each neural network is trained on a separate region in the acoustic space, the complexity of the mapping required of each network is thus reduced. Functionally, an assembly of modular neural networks is similar to a huge neural network. However, each network in the assembly is learned independently with training data for the corresponding regions. This reduces the complexity of finding a good solution in a huge space of possible network configurations since strong constraints are introduced in performing complex constraint satisfaction in a massively interconnected network. Vector quantization partitions the original acoustic space into different prototypes (codewords). This partition can be regarded as a procedure to perform broad-acoustic pattern classification.

The basic unit used in many neural networks computes the weighted sum of its inputs and passes this sum through a nonlinear function such as a threshold or sigmoid function. In the standard sigmoid function, the output dynamic range is between 0 and 1. If the input/output of the mapping function is out of this range, then feature conditioning is generally required. However, for speaker adaptation, the original acoustic data may not be conditioned. Because of this, none the output units in the network are associated with any sigmoid function, but rather a linear function. In addition, the sigmoid function is generalized as $SIGMOID(x) = \frac{\alpha}{1+e^{-\beta * x}} - \gamma$. Using the generalized $SIGMOID$ function, the dynamic range and the shape can be easily controlled according to the training data. In fact, these parameters can be learned automatically during backpropagation.

For speaker-independent speech recognition, we need to have a speaker-independent normalization network. We first constructed sex-dependent speaker clusters. Two golden speaker clusters were chosen for both male and female speakers respectively. The cluster that has the maximum number of speakers is designated as golden cluster. Other clusters were mapped to the golden speaker cluster using our codeword-dependent neural networks. For the DARPA resource management task, the speaker-independent speech recognition word error rate was reduced by another 15%.

3. DETAILED MODELING THROUGH PARAMETER SHARING

We need to model a wide range of acoustic-phonetic phenomena, but this requires a large amount of training data. Since we will never have sufficient training data, one of the central issues becomes that of how to achieve detailed modeling through parameter sharing. Our successful examples include semi-continuous hidden Markov models, senone, and tree-based allophonic models.

3.1. Semi-continuous Models

The semi-continuous hidden Markov model (SCHMM) (also known as a tied-mixture model), first proposed by [9], is an excellent example of detailed modeling through parameter sharing. Intuitively, from the continuous mixture HMM point of view, SCHMMs employ a shared mixture of continuous output probability densities for each individual HMM. Shared mixtures substantially reduce the number of free parameters and computational complexity in comparison with the continuous mixture HMM, while maintaining, reasonably, its modeling power. From the discrete HMM point of view, SCHMMs integrate quantization accuracy into the HMM, and robustly estimate the discrete output probabilities by considering multiple codeword candidates in the VQ procedure. It mutually optimizes the VQ codebook and HMM parameters under a unified probabilistic framework [10], where each VQ codeword is regarded as a continuous probability density function.

For the semi-continuous model, appropriate acoustic representation with given probability density functions is crucial to the recognition accuracy. With appropriately chosen acoustic parameters and probability density functions, the SCHMM can greatly enhance robustness in comparison with the discrete HMM [11]. We first performed exploratory semi-continuous experiments on our three-codebook system. The SCHMM was extended to accommodate a multiple feature front-end [11, 10]. All codebook means and covariance matrices were reestimated together with the HMM parameters except the power covariance matrices, which were fixed. When three codebooks were used, the diagonal semi-continuous model reduced the error rate of the discrete HMM by 10-15%.

Another advantage of using the SCHMM is that it requires less training data in comparison with the discrete HMM. Therefore, given current training data set, more detailed models can be employed to improve the recognition accuracy. One way to increase the number of parameters is to use speaker-clustered models. Due to the smoothing abilities of the semi-continuous model, we were able to train multiple sets of models for different speakers. We investigated automatic speaker clustering and explicit male/female clustered models [3]. By using multiple model sets with the SCHMM, the error rate is further reduced by 10%.

3.2. Senones

To share parameters among different word models, context-dependent subword models have been used successfully in many state-of-the-art speech recognition systems [12, 13]. The principle of parameter sharing can also be extended to subphonetic models. For subphonetic modeling, fenones [14] have been used as the front end output of the IBM acoustic processor. We believe that codeword-dependent fenones may be insufficient for large-vocabulary continuous speech recognition. In addition, determination of a single fenonic baseform by the vector quantizer can not consider multiple examples simultaneously. We introduced the modeling of subphonetic events with Markov states [15]. We treat the state in phonetic hidden Markov models as the basic subphonetic unit — *senone*, which is a state-related modeling unit. Senones are constructed by clustering the state-dependent output distributions across different phonetic models. The total number of senones can be determined by clustering all the triphone states as shared-distribution models [16]. States of different phonetic models may thus be tied to the same senone if they are close according to the distance measure.

The advantages of senones include better parameter sharing and improved pronunciation optimization. Clustering at the granularity of the state rather than the entire model (like generalized triphones [13]) can keep the dissimilar states of two models apart while the other corresponding states are merged, and thus lead to better parameter sharing. In addition to finer parameter sharing, senones also give us the freedom to use a larger number of states for each phonetic model. Although an increase in the number of states will increase the total number of free parameters, with senone sharing we can essentially eliminate those redundant states and have the luxury of maintaining necessary ones. With regard to pronunciation optimization, we can use the forward-backward algorithm to iteratively optimize a senone sequence appropriate for modeling multiple utterances of a word. That is, given the multiple examples, we can train a word HMM with the forward-backward algorithm. When the reestimation reaches its optimum, each estimated state can be *quantized* with the senone codebook (clustered output distributions). The closest one can be used to label the states of the word HMM. This sequence of senones becomes the senonic baseform of the word. Here arbitrary sequences of senones are allowed, providing added flexibility for the automatically learned pronunciation. When the senone sequence of every word is determined, the parameters may be re-trained using the forward-backward algorithm. Although each word model generally has more states than the traditional phoneme-concatenated word model, the number of parameters remains the same since the size of the senone codebook is intact. By constructing senone codebook (output-distribution clustering) we were able to reduce the word error rate of the speaker-independent Resource Management task by 20% in comparison with the generalized triphone. When senones were used for pronunciation optimization in a preliminary experiment, we achieved another 10-15% error reduction in a speaker-independent continuous spelling task.

4. APPLICATION ISSUES

Vocabulary- or speaker-independent speech recognition systems are desirable in many applications where vocabulary- or speaker-specific data do not exist. However, if vocabulary- or speaker-dependent data become available, a vocabulary- or speaker-independent system could be adapted to the specific vocabulary and speaker to further reduce the error rate.

4.1. Vocabulary-Independence and -Adaptation

In general, we need *vocabulary-dependent* (VD) training on a large population of speakers for each vocabulary; such training demands much time for data collection (weeks to months), dictionary generation (days to weeks), and data processing (hours to days). As speech recognition flourishes and new applications emerge, the demand for vocabulary-specific training will become the bottleneck in building speech recognizers, and the difficulty of adding new words will become the Achilles' heal when customizing speech recognizers. Our vocabulary-independent system trained on a large *vocabulary-independent* database is trained only once and can be used for any task [17]. In order to achieve this, we need to find subword models which are detailed, consistent, trainable and especially generalizable. The biggest challenge for VI subword modeling is to find good generalized models when exact coverage is lacking. we proposed a new decision-tree based subword clustering algorithm to find more suitable models for the subword units not covered in the training set [18]. The decision tree classifies subword units by asking questions in hierarchical manner. These questions were first created using speech knowledge from human experts. The tree was automatically constructed by searching for simple as well as composite questions. Finally, the tree was pruned using cross validation. When the algorithm terminated, the leaf nodes of the tree represented the *generalized allophones* to be used. In an experiment with Resource Management task, the decision-tree based generalized allophonic models enable the vocabulary-independent system to outperform the vocabulary-dependent system.

As mentioned earlier, it is desirable to implement vocabulary learning to tailor the VI system to the target vocabulary (task). Our first vocabulary learning algorithm is to build vocabulary-adapted allophonic clustering decision trees for the target vocabulary based on only the relevant allophones (those which occur in the target task). The adapted trees would only focus on the relevant contexts to separate the relevant allophones, thus give the resulting allophonic clusters more discriminative power for the target vocabulary. In an experiment of adaptive allophonic clustering for the Resource Management task, this algorithm achieved an 8% error reduction. Our second vocabulary learning algorithm is to focus on the relevant allophones during training of generalized allophonic models, instead of focusing on them during generation of allophone clustering decision trees. To achieve that, we give the relevant allophones more prominence by assigning more weight to the relevant allophones during Baum-Welch training of generalized allophonic models. With *vocabulary-bias training* we are able to reduce the VI error rate by 15% for the Resource Management task. If vocabulary-specific data is available, we can further adapt the VI system to the target task. Basically, we use the VI models to initialize the training of vocabulary-specific data. Once the new vocabulary-dependent models are ready, we interpolate them with the original VI models. In a pilot experiment, we achieve 20% improvement over vocabulary-dependent system by performing vocabulary adaptation with vocabulary-specific data [17].

4.2. Speaker Adaptation

To bridge the gap between speaker-dependent and speaker-independent speech recognition, we would like to modify the two most important parameter sets for each speaker, i.e. the vector quantization codebooks (or the SCHMM mixture components) and the output distributions (or the SCHMM mixing coefficients) in the framework of either discrete or semi-continuous models. We are interested in developing adaptation algorithms that are consistent with the estimation criterion used in either speaker-independent or speaker-dependent systems. In addition, the algorithm should adapt parameters that are less sensitive to limited training. The codebook can represent the essential characteristics of different speakers. Since the codebook mean vector can also be rapidly estimated with only a limited amount of training data, we consider it to be the most important parameter set. Based on speaker-independent models, we can modify the codebook so that the likelihood can be maximized for a given speaker. Another important parameter set is the output distribution of the SCHMM. In a manner analogous to Bayesian learning, we can interpolate speaker-dependent output distribution with speaker-independent estimates. A nice feature of our adaptation algorithm is that it converges well to speaker-dependent speech recognition [7]. Although combination of codebook and output distribution adaptation gives us the best performance, it is interesting to note that when the amount of adaptation data is limited, adapting the codebook leads to substantial improvement; however, when the amount of available adaptation data is large, adapting output distributions becomes more important.

When we incrementally increased the number of adaptation sentences from 1 to 2400, the error rate decreased gradually. It was found that the error rate of speaker-adaptive speech recognition was always better than or equal to that of speaker-dependent recognition trained with the same amount of data [5] as illustrated in in Figure 1. With only 300 adaptation sentences, the error rate was measurably lower than that of the speaker-dependent system trained with 600 sentences. This shows that speaker-adaptive recognition utilizes training data more effectively than speaker-dependent recognition.

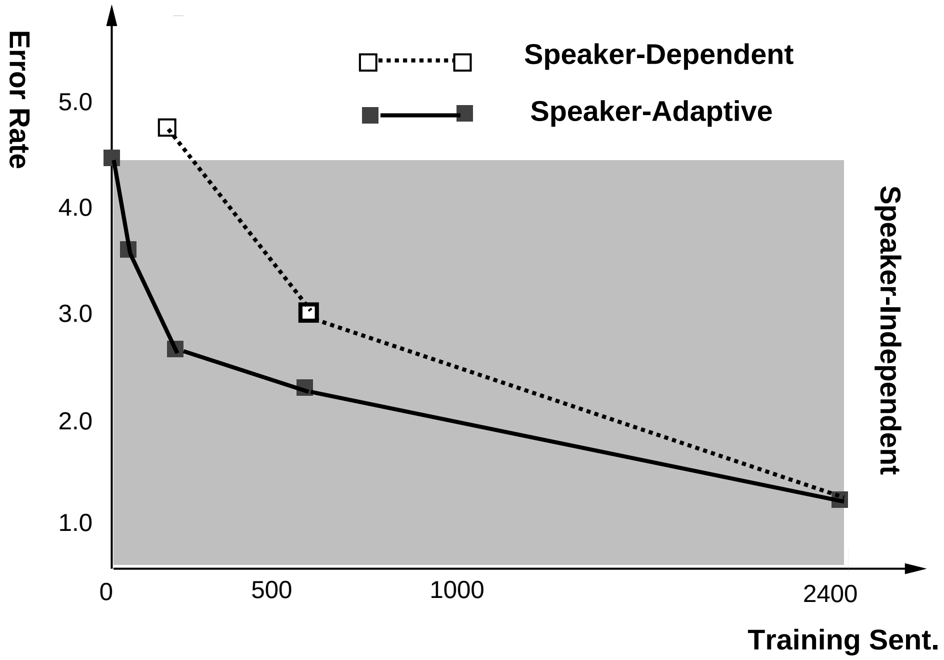

Figure 1: Performance of speaker-independent, -dependent, and -adaptive speech recognition.

5. SEARCH

Efficient search in speech recognition systems is necessary for interactive applications and for the developmental cycle of speech recognition systems. Improving search efficiency is important because recent trends (fluent speech, co-articulation modeling, complex language models) in automatic speech recognition, have exacerbated the demands that the search component places on memory bandwidth, memory size, and computational power. The challenge here is to design a search that makes the appropriate compromises among memory bandwidth, memory size, and computational power [19].

5.1. Search Tree Pruning

The use of pruning in the CSR (continuous speech recognition) search is the procedure that transforms an optimal but exponential time algorithm into a sub-optimal, but linear time algorithm. Therefore the goal of the pruning procedure is to maintain the tightest constraint that still allows the correct utterance to be found in the search tree.

In order to improve the performance of the beam search pruning procedure the single global pruning threshold has been replaced with three pruning thresholds. The three pruning thresholds correspond to, in order of increasing pruning, the Markov state level, the phone level, and the word level. At the Markov state level we have the least evidence with respect to its participation in the final outcome of the search and so it has the loosest pruning threshold. At the phone level and then at the word level more information is available and so the pruning threshold can be made tighter. This type of multiple threshold pruning has improved the performance of the recognition component by a factor of two.

5.2. Memory Organization

In the SPHINX-II system the topology of the HMM is the same for all phonetic models, so it is reasonable to consider embedding the topology in the model evaluation function rather than using explicit state pointers. This is completely analogous to the way dynamic time warping implicitly encodes its graph traversal constraints on the warping space. By implicitly encoding the topology of the HMM the expense of pointer following is saved. Pointer following requires one memory read for each arc plus a test and a branch to control the loop over the outgoing arcs of each state.

This still doesn't solve the order of evaluation problem since some states in a model may not be active. Our solution therefore is to require that either *all* of the states of a model be classified as active or *none* of the states be classified as active.

We now have a hybrid search strategy that uses Viterbi search for the model state graph and the Viterbi beam search for the word level model graph. In addition to the memory bandwidth and and CPU savings achieved, an additional benefit is the improved memory locality since for each active model all of its states will be evaluated consecutively. This hybrid strategy provides another factor of two in performance improvement.

5.3. State Management

The overall efficiency of the search is improved because, the state expansion is now twice as efficient at the model level, the cost of state management is reduced by a factor of five since states are now managed at the model level rather than at the state level (there are five states in the current SPHINX-II model) and memory locality is preserved during state evaluation since all of the states of a model are evaluated at the same time.

6. LANGUAGE MODELING

Linguistic constraints are an important factor in human comprehension of speech. Their effect on automatic speech recognition is similar, in that they provide both a pruning method and a means of ordering likely candidates. As vocabularies for speech recognition increase in size, more accurate modeling of linguistic constraints becomes essential.

6.1. Long Distance Bigrams

In a traditional stochastic language model, the current word is predicted based on the preceding word (bigram) or the preceding $n-1$ words (n-gram). This is because most of the relevant syntactic information can reasonably be expected to lie in the immediate past. But some information, syntactic as well as semantic, may still exist in the more distant past, albeit use of n-gram for $n > 3$ will increase the number of free parameters exponentially. To reduce the number of free parameters and maintain the modeling capacity, we experimented with long distance bigrams (the same principle can be extended to long distance n-grams). We define a *distance-d bigram* to be a bigram that predicts a word W_i based on the word W_{i-d}. With k long-distanced bigrams, $Pr(W_i|W_{i-k}...W_{i-1})$ can be approximated as $\sum_{d=1}^{k} \lambda_d Pr(W_i|W_{i-d})$, where λ can be optimized based on heldout data.

We constructed a long distance backoff bigram for $d = 1, \ldots, 10, 1000$, using the Brown Corpus as our training data. The distance-1000 case was used as a control, since at that distance we expect no mutual information whatsoever. For each such bigram, we computed the perplexity of the *training data*. For a large enough training set, the latter is a measure of the average mutual information between word W_i and word W_{i-d}. As expected, we found the perplexity to be low for $d = 1$, and to increase significantly as we moved through $d = 2, 3, 4$, and 5. For $d = 6, \ldots, 10$, the training-set perplexity remained at about the same level. But interestingly, that level was slightly but consistently below the perplexity of the $d = 1000$ case. We conclude that some information indeed exists in the distant past, but it is spread thinly across the entire history, and will require more sensitive methods to extract. Work along these lines is currently under way.

6.2. Modification of the Backoff Model

The backoff language model[20] is a compact yet powerful way of modeling the dependence of the current word on its immediate history. A known weakness of the backoff model[21, p.457] is that backed-off N-gram probabilities are proportional to the probabilities of N-1–grams. For frequent $N-1$–grams, there may exist sufficient statistical evidence to suggest that the backed-off probabilities should in fact be much lower. This deficiency of the backoff model thus results in serious *overestimation* of some probabilities. But since overestimation degrades perplexity only indirectly (by affecting a slight underestimation of all the other probabilities), we did not expect this weakness to have a strong impact on the overall perplexity. Nevertheless, we wanted to try to correct this problem and to measure the effect on perplexity and on recognition rate.

Using a simple Bayesian argument, we establish a confidence interval for the true value of each of the overestimated probabilities. The conventional backoff estimate is then forced to lie within that interval. This *capping off* of the estimates requires renormalization, leading to an iterative reestimation of the backoff weights. The procedure was found to converge rapidly in all cases. Using this modification, we achieved a reduction of 6% in the bigram perplexity and of 7% in the trigram perplexity of the Brown Corpus. Although the reduction is modest, as expected, it should be noted that it is achieved with hardly any increase in the complexity of the model. As can be predicted from the statistical analysis, when the ratio of training-set size to vocabulary size was increased, the improvement in perplexity was found to diminish. Therefore, this modification may be more suitable for cases where training data is relatively sparse. We are now in the process of measuring the effect on the overall recognition rate.

7. SUMMARY

In comparison with the SPHINX speech recognition system, the word error rate has been reduced significantly. For the DARPA Resource Management task, the reduced error rate can be summarized in Figure 2. We are confident that improved modeling technology will make speech recognition a reality in the near future.

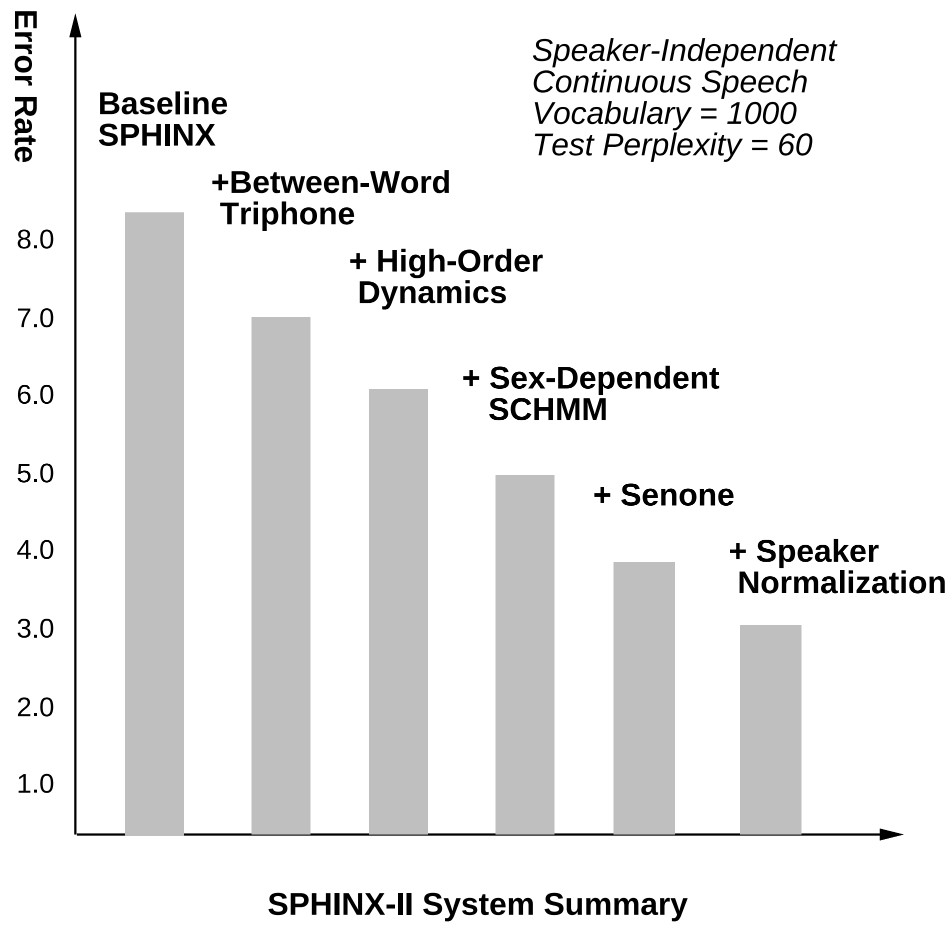

Figure 2: Summary of the SPHINX-II speech recognition system.

8. Acknowledgements

This research was sponsored in part by US West and in part by the Defense Advanced Research Projects Agency (DOD), Arpa Order No. 5167, under contract number N00039-85-C-0163. The authors would like to express their gratitude to Raj Reddy, Kai-Fu Lee, and other members of CMU speech group for their help.

References

[1] Lee, K., Hon, H., Hwang, M., Mahajan, S., and Reddy, R. *The SPHINX Speech Recognition System*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. Glasgow, Scotland, UK, 1989.

[2] Lee, K., Hon, H., Hwang, M., and Huang, X. *Speech Recognition Using Hidden Markov Models: A CMU Perspective*. **Speech Communications**, vol. 9 (1990).

[3] Huang, X., Lee, K., Hon, H., and Hwang, M. *Improved Acoustic Modeling for the SPHINX Speech Recognition System*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. Toronto, Ontario, CANADA, 1991, pp. 345–348.

[4] Acero, A. and Stern, R. *Environmental Robustness in Automatic Speech Recognition*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1990, pp. 849–852.

[5] Huang, X. *A Study on Speaker-Adaptive Speech Recognition*. in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[6] Lee, K., Hon, H., and Reddy, R. *An Overview of the SPHINX Speech Recognition System*. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, January 1990, pp. 35–45.

[7] Huang, X. and Lee, K. *On Speaker-Independent, Speaker-Dependent, and Speaker-Adaptive Speech Recognition*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1991, pp. 877–880.

[8] Huang, X. *Speaker Adaptation Using Codeword-Dependent Neural Networks*. in: **IEEE Workshop on Speech Recognition, Arden House**. 1991.

[9] Huang, X. and Jack, M. *Semi-Continuous Hidden Markov Models for Speech Signals*. **Computer Speech and Language**, vol. 3 (1989), pp. 239–252.

[10] Huang, X., Ariki, Y., and Jack, M. **Hidden Markov Models for Speech Recognition**. Edinburgh University Press, Edinburgh, U.K., 1990.

[11] Huang, X., Lee, K., and Hon, H. *On Semi-Continuous Hidden Markov Modeling*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. Albuquerque, NM, 1990, pp. 689–692.

[12] Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and Makhoul, J. *Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1985, pp. 1205–1208.

[13] Lee, K. *Context-Dependent Phonetic Hidden Markov Models for Continuous Speech Recognition*. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, April 1990, pp. 599–609.

[14] Bahl, L., Brown, P., De Souza, P., and Mercer, R. *Acoustic Markov Models Used in the Tangora Speech Recognition System*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1988.

[15] Hwang, M. and Huang, X. *Subphonetic Modeling with Markov States - Senone*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1992.

[16] Hwang, M. and Huang, X. *Acoustic Classification of Phonetic Hidden Markov Models*. in: **Proceedings of Eurospeech**. 1991.

[17] Hon, H. and Lee, K. *On Vocabulary-Independent Speech Modeling*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. Albuquerque, NM, 1990, pp. 725–728.

[18] Hon, H. and Lee, K. *CMU Robust Vocabulary-Independent Speech Recognition System*. in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. Toronto, Ontario, CANADA, 1991, pp. 889–892.

[19] Alleva, F., Hon, H., Huang, X., Hwang, M., Rosenfeld, R., and Weide, R. *Applying SPHINX-II to the DARPA Wall Street Journal CSR Task*. in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[20] Katz, S. *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer*. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, vol. ASSP-35 (1987), pp. 400–401.

[21] Jelinek, F. *Self-Organized Language Modeling for Speech Recognition*. in: **Self-Organized Language Modeling for Speech Recognition**, by F. Jelinek, edited by S. Furui and M. Sondhi. Marcel Dekker, Inc., 1990.