The 1997 CMU Sphinx-3 English Broadcast News Transcription System

K. Seymore, S. Chen, S. Doh, M. Eskenazi, E. Gouvêa, B. Raj, M. Ravishankar, R. Rosenfeld, M. Siegler, R. Stern, and E. Thayer

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

This paper describes the 1997 Hub-4 Broadcast News Sphinx-3 speech recognition system. This year's system includes full-bandwidth acoustic models trained on Broadcast News and Wall Street Journal acoustic training data, an expanded vocabulary, and a 4-gram language model for N-best list rescoring. The system structure, acoustic and language models, and adaptation components are described in detail, and results are presented to establish the contributions of multiple recognition passes. Additionally, experimental results are presented for several different acoustic and language model configurations.

1. INTRODUCTION

This year's Hub-4 task consisted of transcribing broadcast news shows in a completely unpartitioned manner, meaning that the broadcast news audio was not accompanied by any types of markers indicating speaker or show changes. Recognition systems had to rely on completely automatic methods of segmenting the audio into manageable pieces. Additionally, no information was provided about channel conditions, speaker gender or accent, the presence of noise or music, or speaking style, as was done in 1996. Therefore, this year's recognition task represented a more realistic scenario in which a speech recognizer needed to intelligently and automatically cope with a variety of acoustic and linguistic conditions.

In the following sections, we present an overview of the Sphinx-3 evaluation system. In Section 2, the stages of the recognition system are introduced. The details of the specific evaluation configuration chosen are discussed in Section 3. A variety of experimental results on acoustic model and language model variations are presented in Section 4. Evaluation results for each stage of processing are given in Section 5.

2. SYSTEM OVERVIEW

The Sphinx-3 system is a fully-continuous Hidden Markov Model-based speech recognizer that uses senonically-tied states [1]. Each state is a mixture of a number of diagonal-covariance Gaussian densities. The 1997 Sphinx-3 configuration is similar in many ways to the 1996 system [5]. The recognition process consists of acoustic segmentation, classification and clustering [8], followed by three recognition passes. Each pass consists of a Viterbi decoding using beam search and a best path search of the Viterbi word lattice. The final two passes include N-best list generation and rescoring. Between each pass, acoustic adaptation using a transformation of the mean vectors based on linear regression (MLLR) [4] is performed. These steps are summarized in the following list:

1. Automatic data segmentation, classification, and clustering
2. Pass 1
    a. Viterbi decoding using beam search
    b. Best path search of Viterbi word lattice
3. Acoustic adaptation
4. Pass 2
    a. Viterbi decoding using beam search
    b. Best path search of Viterbi word lattice
    c. N-best generation and rescoring
5. Acoustic adaptation
6. Pass 3
    a. Viterbi decoding using beam search
    b. Best path search of Viterbi word lattice
    c. N-best generation and rescoring

2.1. Front End Processing

Before recognition, the unannotated broadcast news audio is automatically segmented at acoustic boundaries. Each segment is classified as either full-bandwidth or narrow-bandwidth in order that the correct acoustic models may be applied. Segments are then clustered together into acoustically-similar groups, which is useful for acoustic adaptation. Finally, all segments that encompass more than 30 seconds of data are subsegmented into smaller utterances. These techniques are summarized below; details are available in [8].

Automatic Segmentation: The goal of automatic segmentation is to break the audio stream into acoustically homogenous sections. Ideally, segment boundaries should occur in silence regions so that a word is not split in two. To accomplish this, a symmetric relative cross entropy distance metric compares the statistics of 250 frames (2.5 sec) of cepstra before and after each frame. When the distance is at a local maximum and is also greater than a predefined threshold, an acoustic boundary is hypothesized. Instead of the boundary being placed right at the location of the local maximum, two seconds of audio before and after the hypothesized break are searched for silences. A silence is located at frame x when the following criteria are met (1 frame equals 10 ms):

1. The average power over the frames [x-7,x+7] is more than 8 dB lower than the power over the frames [x-200,x+200].
2. The range of the power over the frames [x-7,x+7] is less than 10 dB.

If a silence is found within the search window, an acoustic boundary is placed at that location. If no silence is found, no acoustic boundary is assigned.

Classification: Each segment is then classified as either full-bandwidth (non-telephone) or narrow-bandwidth (telephone) using Gaussian mixture models. The full-bandwidth Gaussian mixture model contains 16 Gaussian densities and was trained from the data labelled as F0, F1, F3, and F4 in the Hub-4 1996 acoustic training corpus. The narrow-bandwidth Gaussian mixture model contains 8 densities and was trained using hand-labeled telephone segments from the 1995 Hub-4 training data.

Clustering: Segments are clustered into acoustically-similar groups using the same symmetric relative cross entropy distance metric mentioned for acoustic segmentation. First, the maximum likelihood estimation of single density Gaussian parameters for each utterance is obtained. Then, utterances are clustered together if the symmetric relative cross entropy between them is smaller than an empirically-derived threshold. Full- and narrow-bandwidth segments are not clustered together.

Sub-segmentation: To reduce the length of the automatically generated segments to 30 seconds, additional silences in each segment are located, and the segments are broken at those points. The resulting subsegments are given to the decoder for recognition.

2.2. Recognition Stages

Viterbi Decoding Using Beam Search: The first stage of recognition consists of a straight-forward Viterbi beam search using continuous density acoustic models. This search produces a word lattice for each subsegment, as well as a best-scoring hypothesis transcription.

Best Path Search: A word graph is constructed from the Viterbi word lattice and then searched for the global best path according to a trigram language model and an empirically determined optimal language weight using a shortest path graph search algorithm [6]. The only acoustic scores used in this search are the ones stored in the lattice from the Viterbi recognition. As a result, this search is much quicker than the Viterbi search. A new best-scoring hypothesis transcription is produced.

N-best Generation and Rescoring: N-best lists are generated for each subsegment using an A* search on the word lattices produced by the Viterbi beam search. For this evaluation, $N = 500$. The N-best rescorer takes as input the N-best lists, which are augmented with the single best hypothesis generated by the Viterbi decoder and the single best hypothesis generated by the best path search. The N-best lists are rescored using the acoustic scores provided by the Viterbi decoder, a new language model score, and a word insertion penalty. Given the rescoring, the new highest scoring hypothesis is output for the subsequent adaptation step or for the final system output.

2.3. Acoustic Adaptation

Unsupervised adaptation of Gaussian density means in the acoustic model is performed, given the output of the best path or N-best search. In order to obtain larger sample sizes, the test set is clustered as described in Section 2.1.

The maximum likelihood linear regression (MLLR) [4] approach to mean adaptation is used. A 1-class MLLR transform is obtained for each cluster using the baseline acoustic models and the selected hypotheses. The means of the baseline acoustic models are transformed for each cluster and the adapted models are used during the next recognition pass.

3. EVALUATION SYSTEM

3.1. Acoustic Models

The acoustic models used in the evaluation system are fully-continuous, diagonal-covariance mixture Gaussian models with approximately 6000 senonically-tied [1] states. A five-state Bakis model topology is used throughout.

Two sets of acoustic models are used: non-telephone (full-bandwidth) models and telephone (narrow-bandwidth) models. The non-telephone models are trained over the Wall Street Journal SI-284 corpus concatenated with the Hub-4 Broadcast News training corpus. Mixture splitting is used to obtain an initial set of acoustic models. Further exploration of the acoustic parameter space is performed using the state labels generated from a forced alignment of the initial models. These labels are used to classify the training data for K-means followed by an E-M reestimation of the output density parameters. One or more passes of Baum-Welch reestimation is then performed to correct the Viterbi assumption underlying the state classification. A final configuration of 6000 tied states and 20 mixture components per state is obtained using this approach.

The telephone models are trained on WSJ SI-321 with reduced bandwidth. This acoustic model is structured as 6000 senonically-tied states mapped into triphones, plus 52 context independent phones and 3 noise phones (including silence). Each tied state is a mixture of 16 densities.

3.2. Dictionary

The recognizer's vocabulary consists of the most frequent 62,549 words of the Broadcast News language model training corpus, supplemented with the 8,309 words from the 1995 Hub-4 Marketplace training data and 355 names from the Broadcast News acoustic training data speaker database. The final number of unique words in the vocabulary is 62,927, which results in a dictionary size of 68,623 pronunciations. We refer to this vocabulary as our 64k vocabulary.

3.3. Language Models

The language model used in the recognizer is a Good-Turing discounted trigram backoff language model. It is trained on the Broadcast News language model training data and the 1995 Hub-4 Marketplace training data. The model is built using a 64k vocabulary, and excludes all singleton trigrams. The out-of-vocabulary rate (OOV) and perplexity (PP) of this model on the development and evaluation data is shown in Table 1.

	OOV	PP
DEV	0.63%	170
EVAL	0.54%	171

Table 1: Out-of-vocabulary rate and perplexity of the evaluation language model on the development and evaluation test sets.

A 4-gram language model smoothed with a variation of Kneser-Ney smoothing is used for N-best rescoring. This model uses the same training data and 64k vocabulary as the Good-Turing discounted model, but does not exclude any n-grams. The smoothing parameters, language weight, and word insertion penalty are optimized using Powell's algorithm on the entire development test set.

Filled pauses are predicted with unigram probabilities that are estimated from the acoustic training data [7]. This year, acoustic models

were built from scratch for each filled pause event.

3.4. Improvements

This year's evaluation system incorporates several improvements over last year's system. The acoustic models are trained on an improved lexicon, and the filler word set introduced last year is trained from scratch. The acoustic models are also trained from scratch, on both the SI-284 Wall Street Journal data and the Broadcast News acoustic training data. The language model is built from an enlarged vocabulary, and does not exclude singleton bigrams as was done last year. This year, phrases and acronyms are not included in the vocabulary, since their inclusion did not significantly improve recognition performance in development experiments (see Section 4.4). Also, a 4-gram language model is used for N-best list rescoring, instead of the trigram model from last year.

4. EXPERIMENTS

The 1997 development test set consists of four hours of broadcast speech representative of the different acoustic conditions and styles typical of the broadcast news domain. In order to speed up experiment turn-around time, two shortened development test sets were defined as subsets of the complete 4-hour set. `SET1` represents a 1-hour selection of acoustic segments taken from last year's PE segmentation of different F-conditions. Segments were selected so that the test set is acoustically balanced, containing data from all F-conditions in the same proportion that these conditions occur in the entire 4-hour development set. The selected segments provide adequate speech from a number of speakers for speaker adaptation experiments, and cover each development set show. The chosen segments are not necessarily adjacent in time and are based on the original PE segmentations. All segments are further subsegmented automatically so that they are not longer than 30 seconds.

The second test set, `SET2`, is representative of completely automatic segmentation. It is also 1 hour in length, but is not acoustically balanced. Instead, entire portions of shows were selected so that the segments would be time adjacent and so that the reference transcript could be easily assembled. This test set was used to quickly run experiments on automatic segmentation. Table 2 shows how many words occur for each acoustic condition in each of the short test sets.

	SET1	SET2
All	11408	10520
F0	2875	2976
F1	3133	3559
F2	1363	961
F3	904	527
F4	1358	1195
F5	443	299
FX	1332	1003

Table 2: Number of words per acoustic condition for short development test sets.

4.1. Mixture Variation

The evaluation system uses fully-continuous acoustic models with approximately 6000 senonically-tied states. Each state is a mixture of a number of diagonal-covariance Gaussian densities. The number of Gaussian components was varied from 16 to 20 per state for the full-bandwidth acoustic models. The Sphinx-3 decoder was run on `SET1` with each set of acoustic models, holding all other parameters constant. The word error rate results from both the Viterbi decoder stage (vit) and the best path search of the word lattices (dag) are shown in Table 3. Since only the full-bandwidth models were used, the F2 results are not optimal. However, we see that across all conditions, the models with 20 mixture-components per state provide superior results.

	16		20	
	vit	dag	vit	dag
All	30.4	29.1	29.4	28.0
F0	18.4	16.7	18.4	15.5
F1	27.5	25.9	26.3	25.3
F2	45.7	43.7	45.1	43.6
F3	32.6	31.4	30.4	30.4
F4	27.3	28.1	25.9	27.8
F5	34.5	33.6	33.4	30.2
FX	47.3	46.7	45.9	43.5

Table 3: Word error rate (%) on `SET1` for different numbers of Gaussian densities per state.

4.2. Vocabulary Optimization

Three Good-Turing discounted trigram backoff language models were built with 40k, 51k and 64k vocabularies. In each case, the vocabulary was chosen from the most frequently occurring words in the Broadcast News language model training data, as well as all of the words from the 1995 Marketplace training data and 355 names from the acoustic training data speaker database. The Sphinx-3 decoder was run on `SET1` with each language model, holding all other parameters constant. Word error rate results are shown in Table 4. Overall, the 64k language model provided a slightly better result than the 51k or 40k language models.

	40k	51k	64k
All	29.5	29.3	29.2
F0	18.5	18.5	18.7
F1	26.3	26.5	26.3
F2	41.7	41.2	40.9
F3	30.3	30.0	29.3
F4	28.5	27.2	27.3
F5	36.1	35.7	35.4
FX	46.8	46.2	46.5

Table 4: Word error rate (%) on `SET1` for different language model vocabularies.

4.3. Language Model Smoothing

Two language models were built using different smoothing techniques. The first model was a 51k Good-Turing discounted trigram backoff language model[2], and the second a 51k Kneser-Ney smoothed trigram language model[3]. The Sphinx-3 decoder was run on `SET1` with each language model, holding all other parameters constant. Word error rate results are shown in Table 5. The Good-Turing discounted backoff model provided superior performance on this test set.

4.4. Compound words

In an effort to establish how the modeling of compound words, which are phrases and acronyms considered as one unit, affects

	G-T	K-N
All	29.3	29.8
F0	18.5	19.2
F1	26.5	27.2
F2	41.2	41.5
F3	30.0	31.0
F4	27.2	27.4
F5	35.7	36.8
FX	46.2	46.2

Table 5: Word error rate (%) on SET1 for different language model smoothing strategies.

recognition performance, four different compound word scenarios were investigated. First, the decoder was run with no compound words in the dictionary or language model (NO). Next, the decoder was run with a list of 355 phrases and acronyms in the dictionary only (DT). The decoder was altered to retrieve the necessary language model scores for each word in the compound word phrase, but only one acoustic score was applied. Then, the decoder was run with the list of compound words in the dictionary and in the language model (LM). In this case, the compound words were modeled as one unit throughout the entire recognition process. Finally, the decoder was run with a shortened list of compound words (DT2) in the dictionary only. This short list was made up of 30 phrases that were believed to be the most acoustically different when occurring together than when occuring in separate, different contexts.

Word error rate results for two different tests are shown in Table 6. The first test was run on the full 4-hour development test set with a 40k language model. The second test was run with a 51k language model on SET1 with a different set of acoustic models than the first test. Therefore, the results are not directly comparable across tests. Additionally, in some cases narrowband acoustic models were used for the automatically-labeled telephone utterances, while in other cases the full-bandwidth models were used. As a result, no F2 results are reported, and the *All* row does not include the F2 condition. Overall, it does not appear that modeling the long set of phrases in the dictionary or in the language model helped recognition. Having the short list of phrases present in the dictionary may help recognition slightly. No compound words were used in the final evaluation system.

	Test1			Test2	
	NO	DT	DT2	DT	LM
All, no F2	33.1	33.2	32.9	30.7	30.6
F0	21.2	21.3	21.2	19.9	19.2
F1	30.5	30.3	30.1	28.6	29.4
F3	40.3	41.2	40.3	35.0	34.7
F4	34.5	34.4	34.6	30.7	30.8
F5	38.7	38.7	38.6	38.6	38.8
FX	65.7	66.6	65.8	53.0	52.2

Table 6: Word error rate (%) for different compound word modeling strategies.

4.5. Segmentation and Context

Automatic segmentation of the broadcast news audio does not guarantee that break points will be chosen at linguistic boundaries. An automatically-segmented utterance may begin or end anywhere within a sentence, or occasionally within a word. Likewise, an utterance may contain a sentence boundary internally.

In order to investigate the effects of automatic segmentation and language model sentence-boundary modeling on word error rate, three different 51k-vocabulary language models were tested with and without hypothesized context. The first language model, noted by S, is a trigram backoff language model trained on language model training text annotated with sentence-boundary tokens. The second language model, XB, contains the sentence-boundary tokens as well as cross-boundary trigrams [7], which are meant to help model the case where sentence boundaries occur inside of an utterance. The third model, NS, is built from the training text without sentence-boundary tokens.

Each model is used to decode SET2 using an automatically generated segmentation. In the standard case, the beginning of each utterance is assumed to transition out of the begin-of-sentence token <s> and transition into the end-of-sentence token </s> at the end of the utterance. In the *context* case, noted by +C, the last two hypothesized words of a preceding utterance are given as trigram context to the current utterance if the preceding utterance occurs just before the current utterance in time. If no utterance immediately precedes the current utterance in time, then the <s> token is given as the context. In either case, no end-of-sentence transition is assumed.

The word error rate results of decoding SET2 with these different configurations are shown in Table 7. Overall, the standard technique of modeling the begin-of-sentence token and assuming the end-of-sentence token provided the lowest word error rate. Introducing two words of context instead of transitioning out of the begin-of-sentence token did not significantly affect word error rate.

	S	S+C	XB	XB+C	NS	NS+C
All	32.0	32.1	32.6	32.7	32.3	32.3
F0	24.6	24.7	25.5	25.7	24.5	24.4
F1	29.2	29.1	29.3	29.3	29.9	29.8
F2	35.9	35.5	34.9	34.5	35.7	35.5
F3	54.3	57.3	57.5	58.3	57.9	57.9
F4	28.2	28.6	28.5	29.2	27.5	27.7
F5	36.5	37.5	36.5	37.1	38.5	38.1
FX	51.4	51.2	53.6	53.4	51.4	51.5

Table 7: Word error rate (%) for different sentence-boundary modeling techniques.

4.6. N-best Rescoring

The N-best rescoring stage of the recognition process involves generating the 500 most-likely hypotheses for each utterance from the Viterbi word lattice. The hypotheses are rescored using the acoustic score from the lattice, a new language model score, and a word insertion penalty. A series of experiments was conducted to determine the best language model to use during rescoring.

Good-Turing discounted trigram and 4-gram models, and Kneser-Ney smoothed trigram and 4-gram models were built from the Broadcast News training data and the Marketplace training data, including all bigrams and trigrams. All four models were used to rescore 500-best lists from the 1-hour SET1 and the entire 4-hour DEV97 test sets. The word error rate results after rescoring are shown in Table 9. The first line of the table shows the rescoring results using the language model scores present in the lattices, which were generated from a Good-Turing discounted trigram language model

Pass	All	F0	F1	F2	F3	F4	F5	FX
pass1, vit	26.9	17.6	25.3	36.7	35.4	35.9	38.0	54.8
pass1, dag	25.8	17.0	23.8	35.0	35.1	35.3	37.2	53.0
pass2, vit	25.8	17.0	24.9	34.1	34.8	33.2	33.4	54.1
pass2, dag	24.9	16.0	23.8	33.1	35.5	33.0	34.3	52.5
N-best rescore	24.1	15.5	22.9	32.5	33.3	31.2	33.0	51.6
pass3, vit	25.4	16.7	24.7	33.9	34.1	32.6	33.2	52.3
pass3, dag	24.6	15.9	24.0	32.4	34.4	32.4	34.0	51.4
N-best rescore2	24.0	15.5	22.8	32.2	33.4	30.8	33.0	50.3

Table 8: Summary of evaluation word error rates (%) by stage.

that excluded singleton trigrams. For both test sets, the Kneser-Ney smoothed 4-gram model performs the best.

Model	SET1	DEV97
Original score	29.7	35.1
G-T 3-gram	29.7	34.9
G-T 4-gram	29.0	34.5
K-N 3-gram	29.4	34.8
K-N 4-gram	28.6	34.2

Table 9: N-best rescoring word error rates (%) for different language models.

Individual Kneser-Ney trigram and 4-gram language models were then built from language model training data from a variety of sources: 130 MW of Broadcast News, 1MW of Broadcast News acoustic training data, 3MW of Switchboard data, 115MW of Hub-3 AP data, 100MW of Hub-3 Wall Street Journal data and 30MW of 1995-only data from Hub-3 excluding Wall Street Journal. Each of these models was interpolated either at the word or sentence level, and the new language scores were used to rescore the 500-best lists. Interpolation weights were chosen to optimize the perplexity of held-out data. Results are shown in Table 10. In this case, word-level interpolation slightly outperforms sentence-level interpolation. A comparison of these results with the Kneser-Ney results from Table 9 shows that using multiple language models does improve performance when rescoring with trigrams, but there is little difference between using just the Broadcast News 4-gram and interpolating the scores from the six different 4-gram language models.

Model	SET1	DEV97
3-gram, word	29.0	34.4
4-gram, word	28.5	34.0
3-gram, sent	29.1	34.6
4-gram, sent	28.6	34.2

Table 10: N-best rescoring word error rates (%) when interpolating language models from different sources.

5. EVALUATION RESULTS SUMMARY

The Sphinx-3 evaluation results at each stage of processing are shown in Table 8. The final system word error rate was 24.0%. The intermediate word error rates were 25.8% at the end of the first pass and 24.1% at the end of the second pass. The third pass of the recognition system did not significantly decrease the word error rate; two passes of the recognizer would have been sufficient.

6. ACKNOWLEDGEMENTS

This research was sponsored by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005 and by the National Security Agency under Grant numbers MDA904-96-1-0113 and MDA904-97-1-0006. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The first author is additionally supported under a National Science Foundation Graduate Research Fellowship.

References

1. M. Y. Hwang, "Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition", PhD. thesis, Carnegie Mellon University, *Computer Science Department tech report CMU-CS-93-230*, 1993.
2. S. M. Katz, "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, no. 3, pp. 400-401, March 1987.
3. R. Kneser and H. Ney, "Improved Backing-off for M-Gram Language Modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 181-184, 1995.
4. C. J. Leggetter, and P. C. Woodland, "Speaker Adaptation of HMMS using Linear Regression", *Cambridge University Engg. Dept., F-INFENG, Tech Report 181*, June 1994.
5. P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer "The 1996 Hub-4 Sphinx-3 System", *Proceedings of the 1997 ARPA Speech Recognition Workshop*, pp. 85-89, Feb. 1997.
6. M. Ravishankar, "Efficient Algorithms for Speech Recognition", PhD. thesis, Carnegie Mellon University, *Computer Science Department tech report CMU-CS-96-143*, 1996.
7. K. Seymore, S. Chen, M. Eskenazi and R. Rosenfeld, "Language and Pronunciation Modeling in the CMU 1996 Hub-4 Evaluation", *Proceedings of the 1997 ARPA Speech Recognition Workshop*, 1997.
8. M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic Segmentation, Classification and Clustering of Broadcast News Audio", *Proceedings of the 1997 ARPA Speech Recognition Workshop*, pp. 97-99, Feb. 1997.