FAST DECODING FOR INDEXATION OF BROADCAST DATA

Jean-Luc Gauvain and Lori Lamel

Spoken Language Processing Group (http://www.limsi.fr/tlp)
LIMSI-CNRS, B.P. 133, 91403 Orsay cedex, France
{gauvain,lamel}@limsi.fr

ABSTRACT

Processing time is an important factor in making a speech transcription system viable for automatic indexation of radio and television broadcasts. When only concerned by the word error rate, it is common to design systems that run in 100 times real-time or more. This paper addresses issues in reducing the speech recognition time for automatic indexation of radio and TV broadcasts with the aim of obtaining reasonable performance for close to real-time operation. We investigated computational resources in the range 1 to 10xRT on commonly available platforms. Constraints on the computational resources led us to reconsider design issues, particularly those concerning the acoustic models and the decoding strategy. A new decoder was implemented which transcribes broadcast data in few times real-time with only a slight increase in word error rate when compared to our best system. Experiments with spoken document retrieval show that comparable IR results are obtained with a 10xRT automatic transcription or with manual transcription, and that reasonable performamce is still obtained with a 1.4xRT transcription system.

1. INTRODUCTION

A major advance in speech recognition technology is the ability of todays systems to deal with non-homogeneous data as is exemplified by broadcast news: changing speakers, languages, backgrounds, topics. However transcribing such data requires significantly higher processing power than what is needed to transcribe read speech data in a controlled environment, such as for speaker adapted dictation. With the rapid expansion of different media sources for information dissemination, there is a pressing need for automatic processing of the audio data stream. A variety of near-term applications are possible such as audio data mining, selective dissemination of information, media monitoring services, disclosure of the information content and content-based indexation for digital libraries, etc. Although it is usually assumed that processing time is not a major issue since computer power has been increasing continuously, it is also known that the amount of data appearing on information channels is increasing at a close rate. Therefore processing time is an important factor in making a speech transcription system viable for audio data mining and other related applications. In this paper we address issues in reducing the computation time for automatic indexation of radio and television broadcasts. Current state-of-the-art laboratory systems can transcribe unrestricted broadcast news data with word error rates under 20%. These systems are often designed to minimize the word error rate, without paying too much attention to the computing resources as long as experiments can be carried out in a reasonable time frame. So it is common practice to develop systems that run in 100xRT, especially to evaluate the absolute quality of the acoustic and language models. Constraints on the computational resources force us to reconsider design issues, in particular concerning the acoustic models and the decoding strategies [2, 14]. In designing a broadcast news indexation system with computational resources in the range of 10xRT, we gathered experimental results to answer the following questions: Is it better to use a single pass or multiple pass decoding strategy? Do the best models developed for a system without resource constraints still perform the best when resource constraints are imposed? Which language model order provides the best performance given cpu time constraints? What level of word transcription accuracy is needed to achieve reasonable indexation?

In the next section we give an overview of the LIMSI broadcast news indexation system, followed by a comparison of decoding strategies (single pass versus multiple pass), and the experiments carried out to address the influence acoustic model size and of language model order on performance. We then discuss the impact of the word error rate on the information retrieval process.

2. SYSTEM OVERVIEW

The LIMSI broadcast news automatic indexation system [3] consists of an audio partitioner [6], a speech recognizer [7, 8] and an indexation module [5].

The goal of audio partitioning is to divide the acoustic signal into homogeneous segments, labeling and structuring the acoustic content of the data. Partitioning consists of identifying and removing non-speech segments, and then clustering the speech segments and assigning bandwidth and gender labels to each segment. The result of the partitioning process is a set of speech segments with cluster, gender and telephone/wideband labels, which can be used to generate metadata annotations. While it is possible to transcribe the continuous stream of audio data without any prior segmentation, partitioning offers several advantages over this straight-forward solution. First, in addition to the transcription of what was said, other interesting information can be extracted such as the division into speaker turns and the speaker identities, and background acoustic conditions. Second, by clustering segments from the same speaker, acoustic model adaptation can be carried out on a per cluster basis, as opposed to on a single segment basis, thus providing more adaptation data. Third, prior segmentation can avoid problems caused by linguistic discontinuity at speaker changes. Fourth, by using acoustic models trained on particular acoustic conditions (such as wide-band or telephone band), overall performance can be significantly improved. Finally, eliminating non-speech segments and dividing the data into shorter segments (which can still be several minutes long), substantially reduces the computation time and simplifies decoding. Note that processing the non-speech segments as if they were speech does not significantly increase the word error rate, but does considerably increase the processing time.

The partitioning approach used in the LIMSI BN transcription system relies on an audio stream mixture model [6]. Each component audio source, representing a speaker in a particular background and channel condition, is in turn modeled by a GMM. The segment boundaries and labels are jointly identified by an iterative maximum likelihood segmentation/clustering procedure using GMMs and agglomerative clustering.

For each speech segment, the word recognizer determines the sequence of words in the segment, associating start and end times and an optional confidence measure with each word. The speaker-independent large vocabulary, continuous speech recognizer makes use of n-gram statistics for language modeling and of continuous density HMMs with Gaussian mixtures for acoustic modeling. Word recognition is usually performed in three steps: 1) initial hypothesis generation, 2) word graph generation, 3) final hypothesis generation. The hypothesis are used in cluster-based acoustic model adaptation using the MLLR technique [9] prior to word graph generation, and all subsequent decoding passes. The final hypothesis is generated using a 4-gram language model.

For all the experimental results given in this paper, the following training conditions were used. The acoustic models were trained on about 150 hours of American English broadcast news data. The phone models are position-dependent triphones, with about 11500 tied-states for the largest model set. The state-tying is obtained via a divisive, decision tree based clustering algorithm. Wideband and telephone band sets of gender-dependent acoustic models were built using MAP adaptation of SI seed models. Fixed language models were obtained by interpolation of n-gram backoff language models trained on 3 different data sets: 203 M words of BN transcripts; 343 M words of NAB newspaper texts and AP Wordstream texts; 1.6 M words corresponding to the transcriptions of the acoustic training data. The interpolation coefficients of these LMs were chosen so as to minimize the perplexity on the Hub4 Nov98 evaluation data. The 4-gram LM contains 7M bigrams, 14M trigrams and 11M fourgrams.

The recognition word list contains 65122 words, and has a lexical coverage of 99.7% and 99.5% on the Hub4-Nov98 and the eval99 (set 2) test sets, respectively. The pronunciations are based on a 48 phone set (3 of them are used for silence, filler words, and breath noises). A pronunciation graph is associated with each word so as to allow for alternate pronunciations, including optional phones. Frequent inflected forms have been verified to provide more systematic pronunciations. As done in the past, compound words for about 300 frequent word sequences subject to reduced pronunciations were included in the lexicon as well as the representation of the most frequent acronyms as words.

The information retrieval system relies on a unigram model (vector space model) per story. In order to reduce the number of lexical items for a given word sense, each word is mapped to its stem (as defined in [13]) or, more generally, into a form that is chosen as being representative of its semantic family. The score of a story is obtained by summing the query term weights which are simply the log probabilities of the terms given the story model once interpolated with a general English model. This term weigthing has been shown to perform as well as the popular TF$*$IDF weigthing scheme [10].

All the reported runs were done on a Compaq XP1000 500MHz machine with Digital Unix.

3. SINGLE PASS DECODER

A 4-gram single pass dynamic network decoder has been developed. It is a time-synchronous Viterbi decoder with dynamic expansion of LM state conditioned lexical trees [1, 11, 12] with acoustic and language model lookaheads. The decoder can handle position-dependent, cross-word triphones and lexicons with contextual pronunciations. It makes use of various pruning techniques to reduce the search space and computation time, including three HMM-state pruning beams and fast Gaussian likelihood computations. It can also generate word graphs and rescore them with different acoustic and language models. Faster than realtime decoding can be obtained using this decoder with a word error under 30%, running in less than 100 Mb of memory on widely available platforms such Pentium III or Alpha machines.

The decoder by itself does not solve by itself the problem of reducing the recognition time as proper models have to be used in order to optimize the recognizer accuracy at a given decoding speed. In general, better models have more parameters, and therefore require more computation. However, since the models are more accurate, it is often possible to use a tighter pruning level (thus reducing the computational load) without any loss in accuracy. Thus, limitations on the available computational resources affect the design of the acoustic and language models. For each operating point, the right balance between model complexity and pruning level must be found.

Acoustic models

Processing time constraints significantly affect the way the acoustic models are selected. For instance, using word-position dependent triphone models, enables more accurate acoustic modeling at word boundaries as the contexts are limited to those triphones actually occurring in cross-word position. The states of the triphone models are tied by means of a decision tree, with 90 questions about the phonetic features of the phone and state positions. The number of triphone contexts and the amount of parameter sharing (state tying) influence the total model size (number of Gaussians) and consequently the decoding speed. To illustrate this point, Figure 1 plots the word error rate as a function of processing time for 3 sets of acoustic models, which taken together minimize the word error rate over a wide range of processing times (from 0.3xRT to 20xRT) for broadcast news data. Transcribing such inhomogeneous data requires significantly higher processing power than for speaker adapted dictation systems, due to the lack of control of the recordings and linguistic content, which on average results in a poorer fit of the acoustic and language models to the data, and as a consequence, the need for larger models. These results on a representative portion of the Hub4-98 eval test data are obtained using a 3-gram language model, and without acoustic model adaptation. The largest model set (350k Gaussians, 11k tied states, 30k phone contexts) provides the best performance/speed ratio for processing times over 4xRT. The 92k model set (92k Gaussians, 6k tied states, 5k phone contexts) performs better in the range of 0.6xRT to 3xRT, whereas a much smaller model set (16k Gaussians) gives a small gain for less than 0.5xRT.

Language models

For a decoder based on lexical tree copies, the potential search space is proportional to the number of LM contexts, i.e., the number of n-1-grams in the backoff component of the n-gram LM. As observed for the acoustic models, there is a tradeoff between model complexity and search space, i.e. the best model without computational constraints may not be the best when such con-

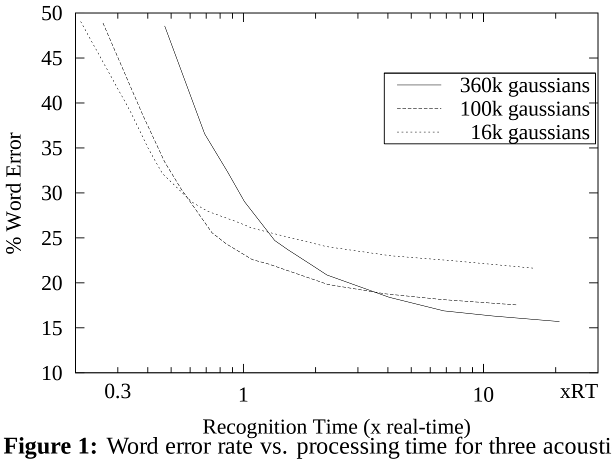

Figure 1: Word error rate vs. processing time for three acoustic model sets with 350k, 92k and 16k Gaussians on a subset of the Hub4-98 test data. (Single pass decoding with a trigram LM and no acoustic model adaptation.)

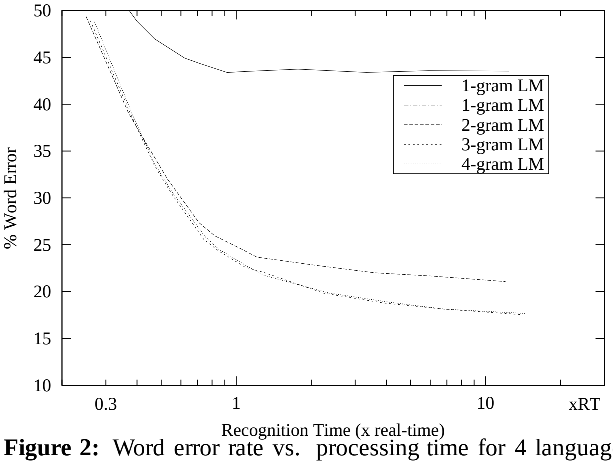

Figure 2: Word error rate vs. processing time for 4 language models (1-gram to 4-gram LM) on a subset of the Hub4-98 data. (Single pass decoding with the 92k acoustic model set and no adaptation.)

straints are imposed. Figure 2 gives the word error rate as a function of the recognition time for four language models (1-gram to 4-gram LM) on the same representative subset of the Hub4-98 eval test data set. The same acoustic model set (6k states, 92k Gaussians) is used for all runs. It can be seen that the trigram LM is the best comprise for computation times in the range of interest (0.5 to 10xRT). In this range the 4-gram LM gives the same results, but requires about 50% more parameters than the 3-gram language model. (The difference is even larger if the required memory space needed to store the models is compared.) To observe a significant difference in favor of the 4-gram LM, the computation time needs to be over 20xRT with this single pass decoding. For computation times under 0.5xRT it does not matter which LM order is used, as long as it is greater than 1.

4. MULTIPLE PASS DECODER

Many systems use a multiple pass decoding strategy to reduce the computational requirements. In multipass decoding, additional knowledge sources are progressively used in the decoding process, which allows the complexity of each individual decoding pass to be reduced and often results in a faster overall decoder. One of the main advantages of multiple pass decoding is the possibility to carry out acoustic model adaptation, such as unsupervised MLLR, between passes by making use of the current best hypotheses. Our targeted speed being lower than 10xRT, we need to pay attention to the computing resources required to perform the adaptation. In these experiments we use a single block diagonal regression matrix and run only one iteration of MLLR reestimation. Table 1 gives the computation time and word error rates for various decoding strategies. The pruning thresholds have been set to try to match the computing time of the most interesting setups. All passes perform a full decode, except the last decoding pass (labelled E) which is a word graph rescoring using a graph generated in the second 3-gram pass. Only two of the acoustic model sets compared in Figure 1 are used: the 350k Gaussian set and 92k Gaussian set used only in the first decoding pass.

These results clearly demonstrate the interest of using a multiple pass decoder. Comparing the setups A (1 pass, 16.8%) and D (2 passes, 15.4%), or comparing setups B (1 pass, 15.4%) and C (2 passes, 14.6%), we see that the extra computing time needed for the first decode and the MLLR adaptation is largely compensated by the reduction in word error rate. Using adapted acoustic models allows us to use a tighter pruning threshold and have the same overall computing time but with a significantly lower word error rate. Also comparing setups C (2 passes, 10.7xRT, 14.6%) and E (3 passes, 8.4xRT, 14.2%) demonstrate the advantage of using an extra decoding pass to take advantage of the 4-gram LM and hypotheses for the MLLR adaptation.

	Pass	*AM*	*LM*	*Time*	*Total time*	*Werr*
A	1	92k	3g	6.8xRT	6.8xRT	16.8%
B	1	350k	4g	10.8xRT	10.8xRT	15.9%
	1	92k	3g	0.8xRT		24.7%
C	2	350k+mllr	4g	9.9xRT	10.7xRT	14.6%
	1	92k	3g	0.8xRT		24.7%
D	2	350k+mllr	3g	6.1xRT	6.9xRT	15.4%
E	3	350k+mllr	4g	1.5xRT	8.4xRT	14.2%

Table 1: Comparison of decoding strategies on the NIST Hub4 eval98 set (partitioning and coding times are not included).

In Table 2 the word error rates and the total computation time (including partitionning) are given for both the development test set (Hub4 eval98) and the Hub4 eval99 test set. For reference, the official result on the eval98 test set using our Nov98 system was 13.6%, with a decoding time around 200xRT [7]. Using only the first decoding pass, unrestricted BN data can be decoded in less than 1.4xRT (including partitioning) with a word error rate around 30%. The same decoding strategy has been successively applied to the BN transcription in other languages (French, German and Mandarin) with comparable word error rates.

5. IMPACT OF WERR ON RETRIEVAL

In order to assess the effect of the recognition time on the information retrieval results we transcribed the 500h of broadcast news data (the TREC SDR99 data set – epoch Feb98 to Jun98) using two decoder configurations: a single pass 1.4xRT system and a three pass 10xRT system. The SDR99 test data consists of 21750 stories and an associated set of 50 queries with on average 14 words. Although for IR purposes the story boundaries are assumed to be known, this information is not used by the speech recognizer. The information retrieval results are given in term of mean average precison (MAP), as is done for the TREC benchmarks. Word error rates are measured on a 10h test subset [4]. For

	Dev data (eval98)		*Test data (eval99)*	
Step	*CPU time*	*Werr*	*CPU time*	*Werr*
Coding and Partitioning:	0.5xRT		0.5xRT	
Word decoding:				
pass#1 (generate 3-gram hyp):	0.8xRT	24.7%	0.9xRT	29.3%
pass#2 (MLLR, 3-gram):	6.1xRT	15.4%	6.5xRT	18.5%
pass#3 (MLLR, 4-gram):	1.5xRT	14.2%	1.5xRT	17.1%
Overall:	8.9xRT	14.2%	9.4xRT	17.1%

Table 2: 10xRT results in word error rate for the NIST BN 1998 and 1999 test sets.

comparison, results are also given for manually produced closed captions. In order for the same IR system to be applied to different text data types (automatic transcriptions, closed captions, additional texts from newspapers or newswires), all of the documents are preprocessed in a homogeneous manner. This preprocessing, or tokenization, is the same as the text source preparation for training the speech recognizer language models.

Table 3 gives the word error rates and IR results for the three sets of transcriptions with and without query expansion. Query expansion uses blind relevance feedback (BRF) on both the audio document collection and some commercially available broadcast news transcripts predating the audio corpus (Jun-Dec 1997 vs Feb-Jun 1998). With query expansion comparable IR results are obtained using the closed captions and the 10xRT transcriptions, and a small degradation (4% absolute) is observed using the 1.4xRT transcriptions.

Transcriptions	*Werr*	*Base-MAP*	*BRF-MAP*
Closed-captions	-	0.4691	0.5430
10xRT	20.5%	0.4528	0.5385
1.4xRT	32.6%	0.4090	0.4938

Table 3: IR results on the 500h SDR99 data set.

6. CONCLUSIONS

In this paper we have described our efforts in developing a fast decoder for indexation of broadcast data. This new decoder transcribes broadcast data in several (6 to 10) times real-time with only a slight increase in word error rate when compared to our best system [7], and with a word error of about 30% for essentially real-time decoding. Our development work with this decoder showed us how processing time constraints can significantly change the way we build our models. For each operating point, the right balance between model complexity and search pruning level must be found. For moderate decoding times (in the range 0.6xRT to 3xRT) a model set containing 92k Gaussians, 6k tied states, 5k phone contexts performs substantially better than smaller or larger models. For processing times over 5xRT, an a larger model set (350k Gaussians, 11k tied states, 30k phone contexts) provides the best performance/speed tradeoff.

Experiments with Spoken Document Retrieval [4] illustrate that only a moderate IR performance degradation is obtained with the real-time system, and that generally speaking, the transcription quality of our system is not a limiting factor given todays IR techniques.

ACKNOWLEDGEMENTS

This work has been partially financed by the European Commission and the French Ministry of Defense. The authors wish to thank Gilles Adda for providing the language models used in these experiments.

REFERENCES

[1] X. Aubert, “One Pass Cross Word Decoding for Large Vocabularies Based on a Lexical Tree Search Organization,” *Proc. ESCA Eurospeech’99*, **4**, pp. 1559-1562, Budapest, Hungary, September 1999.

[2] J. Davenport, L. Nguyen, S. Matsoukas, R. Schwartz, J. Makhoul, “The 1998 BBN Byblos 10x Real Time System,” *Proc. DARPA Broadcast News Workshop*, Feb.-Mar. 1999.

[3] J.L. Gauvain, L. Lamel, and G. Adda, “Transcribing broadcast news for audio and video indexing,” Communications of the ACM, 43(2), Feb 2000.

[4] J.S. Garofolo et al., “1999 Trec-8 Spoken Document Retrieval Track Overview and Results,” Proc. 8th Text Retrieval Conference TREC-8, Nov. 1999.

[5] J.L. Gauvain, Y. de Kercadio, L.F. Lamel, G. Adda “The LIMSI SDR system for TREC-8,” *Proc. of the 8th Text Retrieval Conference TREC-8*, pp. 405-412, Gaithersburg, MD, November 1999.

[6] J.L. Gauvain, L. Lamel, G. Adda, “Partitioning and Transcription of Broadcast News Data,” *ICSLP’98*, **5**, pp. 1335-1338, Dec. 1998.

[7] J.L. Gauvain, L. Lamel, G. Adda and M. Jardino, ”The LIMSI 1998 Hub-4E Transcription System”, *Proc.* DARPA *Broadcast News Workshop*, pp. 99-104, Herndon, VA, February 1999.

[8] J.L. Gauvain, L. Lamel, G. Adda, “Recent Advances in Transcribing Television and Radio Broadcasts,” *Proc. Eurospeech’99*, **2**, pp. 655-658, Budapest, Sept. 1999.

[9] C.J. Leggetter, P.C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech & Language*, **9**(2), pp. 171-185, 1995.

[10] D. Miller, T. Leek, R. Schwartz, “Using Hidden Markov Models for Information Retrieval”, Proceedings of the TREC-7 conference, 1998.

[11] H. Ney, R. Haeb-Umbach, B.H. Tran and M. Oerder, “Improvements in Beam Search for 10000-Word Continuous Speech Recognition,” *Proc. IEEE ICASSP-92*, **I**, pp. 9-12, San Francisco, CA, March 1992.

[12] J.J. Odell, V. Valtchev, P.C. Woodland and S.J. Young, “A One Pass Decoder Design for Large Vocabulary Recognition,” *Proc. ARPA Human Language Technology Workshop*, pp. 405-410, Princeton, NJ, March 1994.

[13] M. F. Porter, “An algorithm for suffix stripping”, *Program*, **14**, pp. 130–137, 1980.

[14] P.C. Woodland, J.J. Odell, T. Hain, G.L. Moore, T.R. Niesler, A. Tuerk, E.W.D. Whittaker, “Improvements in Accuracy and Speed in the HTK Broadcast News Transcription System,” *Eurospeech’99*, pp. 1043-1046, September 1999.