

The LIA speech recognition system: from 10xRT to 1xRT

G. Linarès, P. Nocera, D. Massonié, and D. Matrouf

Laboratoire Informatique d'Avignon, LIA, Avignon, France
{georges.linares, pascal.nocera, dominique.massonie,
driss.matrouf}@lia.univ-avignon.fr

Abstract. The LIA developed a speech recognition toolkit providing most of the components required by speech-to-text systems. This toolbox allowed to build a Broadcast News (BN) transcription system which was involved in the ESTER evaluation campaign ([3]), on *unconstrained transcription* and *real-time transcription* tasks. In this paper, we describe the techniques we used to reach the real-time, starting from our baseline 10xRT system. We focus on some aspects of the A* search algorithm which are critical for both efficiency and accuracy. Then, we evaluate the impact of the different system components (lexicon, language models and acoustic models) to the trade-off between efficiency and accuracy. Experiments are carried out in framework of the ESTER evaluation campaign. Our results show that the real time system reaches performance on about 5.6% absolute WER whereas the standard 10xRT system, with an absolute WER (Word Error Rate) of about 26.8%.

1 Introduction

The LIA developed a full set of software components for speech-to-text system building, including tools for speech segmentation, speaker tracking and diarization, HMM training and adaptation... The aim of the toolkit is to provide software for transcription system design and implementation. It is composed of two main packages addressing a large part of speech-to-text related tasks. The first one contains software components for segmentation and speaker recognition. It is based on ALIZE toolkit ([2]). The second is dedicated to HMM-based acoustic modeling and decoding. This software environment allowed us to build a Broadcast News (BN) transcription system which was involved in ESTER evaluation campaign. In this paper, we describe the efforts we produced to reach the real time, starting from this baseline BN system.

The core of the transcription toolkit is constituted of a recognition engine (Speeral [8]), which is a stack decoder derived from the A* algorithm. Most of the real-time speech recognition systems used a beam-search approach, since A* performs a depth-first exploration of the search graph. Usually, the main motivation for using A* decoder relies in its well known capacity in integrating new information sources into the search. In the first part of this paper, we investigate some methods for reaching the real time by using such an asynchronous engine.

We propose an architecture for a very fast access to linguistic and acoustic resources, and we show how it can be taken advantage of the specificity of the A* decoder to improve the efficiency of pruning.

In the second part, we present our BN system and we discuss about efficiency issues related to the global transcription strategy. Then, starting from the 10xRT system, we evaluate the system configuration which leads to an optimal trade-off between accuracy and decoding duration. Finally, section 4 provides some conclusion on this work.

2 The Speeral decoder

2.1 Search strategy

A* is an algorithm dedicated to the search of the best path in a graph. It has been used in several speech recognition engines, generally for word-graph decoding. In Speeral, the search algorithm operates on a phoneme lattice, which are estimated by using cross-word and context-dependent HMM.

The exploration of the graph is supervised by an estimate function $F(h_n)$ which evaluates the probability of the hypothesis h_n crossing the node n :

$$F(h_n) = g(h_n) + p(h_n) \quad (1)$$

where $g(h_n)$ is the probability of the current hypothesis which results from the partial exploration of the search graph (from the starting point to the current node n); $p(h_n)$ is the probe which estimates the probability of the best hypothesis from the current node n to the ending node.

The graph exploration is based on the function of estimate $F()$. Indeed, the stack of hypothesis is ordered on each node according to $F()$. The best paths are then explored firstly. This deep search refines the evaluation of the current hypothesis and low-probability paths are cutted-off, leading to search backtrack. It is clear that precision of the probe function is a key point for search efficiency. We have produced substantial efforts to improve the probe by integrating, as soon as possible, all available information. The Speeral look-ahead strategy is described in the next section.

2.2 Acoustic-Linguistic look-ahead

As explained previously, the probe function aims to evaluate the probability of each path which have to be developed. The more this approximation is close to the exact one, the soon a decision of leaving or developping a path is taken. Moreover, the CPU-cost of this function is critical while it is used at each node of the search graph. We use a long-term acoustic probe combined with a short-term linguistic look-ahead. The acoustic term is computed from a Viterbi-back algorithm based on context-free acoustic models. This algorithm evaluates the best acoustic probabilities from the end-point to the current one. Of course, the evaluation of all partial paths are performed once, in a first pass. Nevertheless,

as explained in the last section, the probe must provide an upper limit of path probabilities. So, the best phoneme sequences are rescored by using *upper-models*. Upper models are context-free models resulting from the aggregation in a large HMM, of all context-dependant states associated to a context-free one, remaining left-Right transition constraints. Hence, the probability of emission given the upper-model is an upper-limit of path-probabilities, given any context-dependent model.

Anticipating the linguistic information (known as LMLA - Language Model Look-Ahead) enables the comparison of competing hypotheses before reaching a word boundary. The probability of a partial word corresponds to the best probability in the list of words sharing the same prefix. The probability of a partial word corresponds to the best probability in the list of words sharing the same prefix :

$$P(W^*|h) = \max_i P(W_i|h)$$

where W^* is the best possible continuation word and h the word history (partially present in $g(h_n)$). The lexicon is stored as a PPT (Pronunciation Prefix Tree), each node containing the list of reachable words. To ensure the consistency between linguistically well formed hypotheses and pending ones, linguistic probabilities have to be computed at the same n-gram order. This means doing LMLA also at the 3-gram level. We developed a fast computation and approximation method based on a divide and conquer strategy ([5]). Our approach consists in first comparing the list W^* with the list of available trained 3-grams stored in the LM. The LM is an on-disk tree structure containing lists of word probabilities at each n-gram level. Comparing lists at runtime spares most of the LM back-off computation with low overhead. The LMLA approximation does not affect the results. Moreover we introduced precomputed LMLA probabilities to speed-up the computation of the biggest lists.

2.3 Optimizing the computation of acoustic likelihoods

The *a priori* estimation of the contribution of each component in the decoding duration is difficult, as it depends to the search strategy and to the models complexity. Nevertheless, considering the complexity of acoustic models involved in LVCSR systems, the computation of acoustic probabilities may take a large part of the decoding computational cost. [4] estimates this ratio ranges between 30% and 70% in a large vocabulary system. More recent systems use models of millions of parameters; such complexity should not be tractable without any fast calculation method. We produced substantial efforts in optimizing the management of acoustic scoring, by using an efficient caching scheme and an original method for fast-likelihood computation.

Likelihood access and caching. A* decoding and state tying lead to an asynchronous use of acoustic probabilities, at both the HMM and GMM levels:

- probabilities $P(X_{t,t+d}|H_i)$ of observation sequences $X_{t,t+d}$ given a HMM H_i are firstly computed during first pass of acoustic-phonetic decoding. Rescoring with upper-models require the evaluation of $P(X_t|S_i)$, for each GMM S_i matching to the best phonetic sequence;
- as the search develops a part of the exploration graph, various concurrent hypothesis are evaluated. Each of them corresponds to a phonetic sequence. It is clear that competing word-hypotheses could share some phonetic subsequences;
- state tying leads to involve the same state in computation of different HMM probabilities; the architecture of acoustic handler should take advantage of this state sharing.

In order to avoid multiple computation of a likelihoods, we separate clearly the search algorithm and an acoustic handler which is in charge of acoustic probabilities computing and caching. This handler is based on a two-level caching mechanism; as the search algorithm has to score an hypothesis, it requires a probability $P(X_{t,t+d}|H_i)$. The acoustic handler search this score in the level-1 cache (L1); if it is not found, this score is computed by using the Viterbi algorithm and the probabilities of emission $P(X_{t,t+d}|H_i)$. These last ones are searched in the level-2 cache (L2). When the targeted values are not found, they are computed by using a fast likelihood methods which are describe below. L1 and L2 caches are implemented as circular buffers. Moreover, likelihood computation function is written in assembly language, by using SIMD instruction set. Finally, likelihoods are computed on-demand, allowing to limit the computed scores to the ones effectively required by the search and to take benefit from the lexical and linguistic constraints. The figure 1 describes the global architecture of the acoustic handler.

Fast likelihood computation. Numerous methods for fast likelihood computation have been evaluated the last decades. Most of them rely on Gaussian selection techniques which identify, in the full set of Gaussian, the ones which contribute significantly to the frame likelihood estimate. We developed an original method which guaranties a constant precision ϵ of the likelihood approximation. This method consists in off-line clustering of Gaussian and in on-line selection of Gaussian clusters.

As proposed in some papers ([1],[4]), Gaussian are clustered by a classical k-means algorithm on the full set of Gaussian, using a minimum-likelihood loss distance. Each center of cluster is a mono-Gaussian model G_i resulting of the merge of all members of the cluster.

The on-line selection process consists in selecting a set of clusters which model the observation neighborhood. It is important to note that, at contrary of classical Gaussian selection methods, the number of selected cluster is variable, according to the considered frame and to the expected precision ϵ .

The clusters are selected by computing the likelihood of the frame knowing each cluster center G_i ; these likelihoods are used for partitioning clusters into two subsets (tagged *selected* and *unselected* clusters) respecting the rules: (1) each

frame likelihood knowing a selected cluster center is greater than each unselected one and (2) the sum of unselected clusters likelihood is lower than an *a priori* fixed precision threshold

Lastly, *a posteriori* probabilities are computed using only Gaussian belonging to selected clusters. Probabilities of unselected Gaussian are estimated by backing off to the cluster probabilities.

Our experiments have shown that this method allows to decrease the number of computed likelihood by a factor 10 without impacting the WER (Word Error Rate). In adverse acoustic conditions, this method allows to remain a good acoustic precision, since computational cost is increasing.

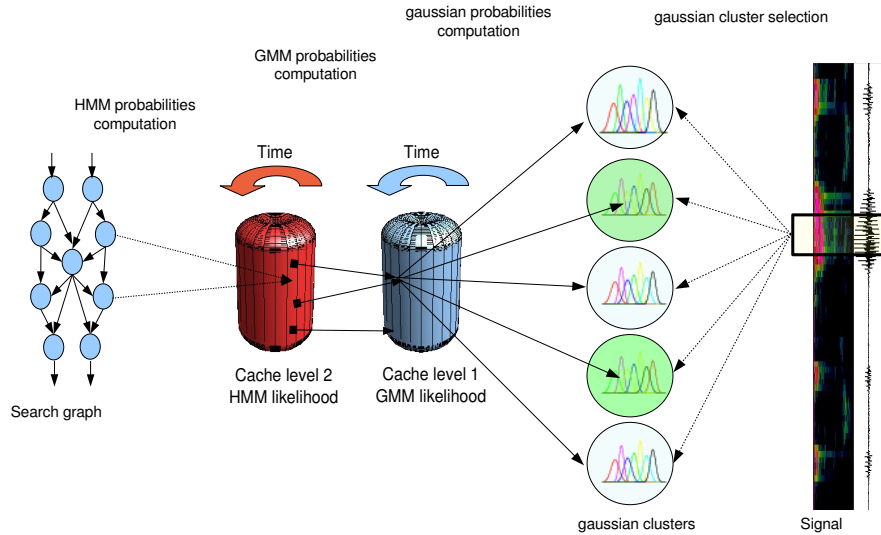


Fig. 1. Architecture of Speeral acoustic handler; likelihood are computed on-demand, depending to the path which are effectively developed by the search algorithm; the cache L1 stores acoustic probabilities of an observation sequence given a HMM; the L2 cache stores probabilities of an observation X_t knowing the considered state S_i . Finally, the Gaussian probabilities are computed, or approximated by the ones of Gaussian-clusters.

3 Overview of the broadcast news transcription system

In addition to intrinsic difficulty of speech recognition, broadcast news transcription adds specific problems related to the signal flow continuity as well as the diversity of the acoustic conditions. Recognizers require tractable speech segments

and high level acoustic information about the nature of segments (speaker identity, recording conditions, etc.). In our system, 2 successive segmentation passes are performed. Speech segments are initially isolated from audio flow; then, a wide/narrow band segmentation identifies telephone segments. We use a method based on a hierarchical classification based of GMM classifiers and morphological rules ([7]). Speaker segmentation is achieved by a fast method ($\simeq 0.05xRT$) based on mono-Gaussian models and the BIC-based criterion ([6]).

The 10xRT system runs two decoding passes; the first one provides transcript which is used for MLLR-based adaptation. While the same models are involved in the 2 passes, the pruning scheme changes: the first pass takes about 3xRT for about 6xRT for the final one. As we aim to reach the real time, the RT system runs only 1 pass, without any speaker adaptation. In the following, we study how we can reach real time by tuning the acoustic and linguistic models involved in the system. All tests reported in this section were performed on 3 hours extracted from the ESTER development corpus.

3.1 Transcription

3.2 Acoustic modelling

We use a classical PLP parametrization; feature vectors are composed of 12 coefficients, plus energy, and first and seconds derivative of these 13 coefficients. At last, we perform a cepstral normalization (mean removal and variance reduction) in a 500ms sliding window. The system uses context-dependent models trained on the 90 hours of Ester transcribed data. State tying is performed by a decision tree algorithm, using acoustic context related questions.

Two sets of speaker-independent acoustic models are used: a large band model and a narrow band model, both gender-dependent. Ester corpus provides a small amount of narrow-bandwidth data; so, narrow-bandwidth models were first trained using filtered large bandwidth data (using a low-pass filter); finally, telephone models were mapped to real narrow-bandwidth data extracted from the Ester train corpus.

The 10xRT system is based on acoustic models set containing 10000 HMM for 3600 emitting states, 64 Gaussian each. Here, we use only the first pass of this system, which is about 3xRT. This acoustic model contains about 230000 Gaussian components. In spite of efficient Gaussian selection, this model is too large for real time decoding. We built two smaller model sets, composed respectively by 3600 states 24 Gaussian each (90k Gaussian, noted 90Kg) and 936 emitting states (60k Gaussian, noted 60Kg). Tests are carried out by using the pruning scheme of the 3xRT system (3xRT); CPU-time is computed on a small server (2.2GHz opteron, only one process dedicated to the test). Results show that the 90k model allows a very significant gain in term of efficiency, since the WER is increased of about 0.5% absolute. The model 330k, composed by 5200 emitting states is too large considering the amount of training data.

Acoustic model	60Kg	90Kg	230Kg	330Kg
WER	25.8%	24.7%	24.2%	24.5%
RealTime Factor	1.6	1.9	2.9	3.4

Table 1. WER of systems according to the number of Gaussian components

3.3 Lexical and language models

The linguistic resources are extracted from two corpora: newspaper Le Monde from 1987 to 2003 (330 Million words) and ESTER (960K words). The trigram language model was learned on the corpus Le Monde and ESTER training set. It is obtained by a linear combination of three models; the first two were learned on the data of Le Monde 1987-2002 and on Le Monde 2002-2003, and the last on the ESTER corpus. Lastly, these models are mixed into an unique model with interpolation coefficients determined by the ESTER development corpus entropy. The language model used by the 10xRT system is based on a lexicon of 65000 word (named 65Kw) and language model including 16.7 Million of bigrams and about 20M of trigrams. In order to reduce the computational cost due to the size of lexicon, we built a 20000 word dictionary (named 20Kw) for which the out of vocabulary rate is about 1.2% (0.5% for 65Kw lexicon) results are compared to the ones obtained by using the 65000 word lexicon. The table 2 compares the results obtained by using this two LM, for 60Kg and 90Kg acoustic models. For these tests, the system is configured in one drastic pruning scheme (noted 1xRT), according to the targeted real-time decoding. Results of the system based on 65Kw lexicon and 90Kg acoustic model obtain very good results (24.7%WER), while being under the 2xRT; moreover, the configuration 90Kg and 65Kw represents a very good trade-off which could be reached by using a more recent processor. Finally, by using acoustic model of 90K Gaussian and a lexicon 20000 word, the system runs in about 1.0xRT and reach a WER of 26.8% (cf. table 2).

System	20Kw 60Kg 1xRT	65Kw 60Kg 1xRT	20Kw 90Kg 1xRT	65Kw 90Kg 1xRT	65Kw 90Kg 3xRT
WER	28.5%	27.5%	26.8%	25.6	24.7%
Real time factor	0.7	0.9	1.0	1.3	1.9

Table 2. WER and real-time factors for Speeral decoding according to the lexicon size, the size of acoustic models, and the pruning scheme.

4 Conclusion and perspectives

We presented the main aspects of the LIA real time transcription system. An efficient architecture is proposed and we provide a full methodological framework

for fast A* decoding. Our results shows that real-time can be reached while remaining the functional model of our baseline system. This real-time system obtained an absolute WER of 26.8% WER. This system ranked 2 in *real time transcription task* of the ESTER evaluation campaign.

References

1. E. Bocchieri. Vector quantization for the efficient computation of continuous density likelihood. In IEEE, editor, *Proc ICASSP'93*, volume 2, pages 692–696, Speech Research Dept., AT&T Lab., Murray Hill, 1993. IEEE.
2. J.-F. Bonastre, F. Wils, and S. Meignier. ALIZE, a free toolkit for speaker recognition. In *ICASSP'05*, Philadelphia, USA, March 2005.
3. S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier. The ESTER Phase II evaluation campaign for the rich transcription of French broadcast news. In *Proc. of the ECSCT*, 2005.
4. K.M. Knill, M.J. Gales, and S.J. Young. Use of gaussian selection in large vocabulary continuous speech recognition using HMMS. In *Proc. ICSLP'96*, volume 1, pages 470–474, Philadelphia, PA, USA, 1996. Cambridge University.
5. D. Massonié, P. Nocéra, and G. Linarès. Scalable language model look-ahead for LVCSR. *InterSpeech'05, Lisboa, Portugal*, 2005.
6. S. Meignier, D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier. Step-by-step and integrated approaches in broadcast news speaker diarization. In *In: Odyssey'04*, volume 20, pages 303–330. Toledo University, 2004.
7. P. Nocera, C. Fredouille, G. Linares, D. Matrouf, S. Meignier, J.-F. Bonastre, D. Massonié, and F. Béchet. The LIA's French broadcast news transcription system. In *SWIM: Lectures by Masters in Speech Processing*, Maui, Hawaii, 2004.
8. Pascal Nocera, Georges Linarès, and Dominique Massonié. Phoneme lattice based a* search algorithm for speech recognition. *Text, Speech and Dialogue : 5th International Conference, TSD 2002, Brno, Czech Republic*, 2002.