ACOUSTIC AND SYNTACTICAL MODELING IN THE ATROS SYSTEM

D. Llorens†, F. Casacuberta, E. Segarra, J. A. Sánchez, P. Aibar†, and M. J. Castro

Dpto. Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, España.
†Unitat Predepartamental d'Informàtica, Universitat Jaume I, Castelló, España.
{fcn,esegarra,jandreu,mcastro}@dsic.upv.es, †{dllorens,aibar}@inf.uji.es

ABSTRACT

Current speech technology allows us to build efficient speech recognition systems. However, model learning of knowledge sources in a speech recognition system is not a closed problem. In addition, lower demand of computational requirements are crucial to building real-time systems.

ATROS is an automatic speech recognition system whose acoustic, lexical, and syntactical models can be learnt automatically from training data by using similar techniques. In this paper, an improved version of ATROS which can deal with large smoothed language models and with large vocabularies is presented. This version supports acoustic and syntactical models trained with advanced grammatical inference techniques. It also incorporates new data structures and improved search algorithms to reduce the computational requirements for decoding. The system has been tested on a Spanish task of queries to a geographical database (with a vocabulary of 1,208 words).

1. INTRODUCTION

Nowadays, good speech recognition systems can be built thanks to the current status of speech technology. However, there are a lot of aspects that can be improved. Some of these aspects can affect the knowledge sources themselves (acoustical, lexical, and syntactical sources) and the way in which the corresponding models can be learnt from speech and/or text data. Other improvements deal with computational problems, such as internal data representation and the search algorithms to handle them in order to decode an input utterance.

ATROS (Automatically Trainable Recognizer Of Speech) is an automatic speech recognizer in which all its knowledge sources can be learnt automatically from real data [7]. The ATROS version presented in this paper includes improvements in the aspects mentioned above. This version is the speech input interface of a speech understanding system developed under the Spanish project SENGLAR.

ATROS supports new acoustic models and language models. Continuous Density Hidden Markov Models (CDHMMs), which have been trained with the HTK toolkit [14], and hybrid connectionist-structural models [3, 4] have been used as acoustic models. In addition, stochastic finite state grammars interpolated with unigrams [12] are used as language models.

With regard to computational issues, new original and well-known techniques have been incorporated in order to achieve greater computational efficiency. Hashing techniques have also been introduced to efficiently handle the search space (trellis). The vocabulary has been represented as a prefix tree (tree lexicon), and language model look-ahead techniques [10] have been also used. Fast phoneme look-ahead has been introduced in ATROS to reduce the search space [10]. The smoothed language model is not fully expanded thanks to the introduction of empty transitions between the stochastic finite state grammar and a unigram. This implementation allows for a reduced amount of memory without an increase in computation time.

The following section is devoted to an ATROS system overview. The representation and training of each knowledge source are described in sections 3 and 4. The search procedure and related aspects are presented in section 5. Section 6 shows the experiments that have been carried out. Finally, some conclusions are mentioned in section 7.

2. SYSTEM OVERVIEW

ATROS is composed of two parts: the *feature extraction module* and the *decoding module*. The first module computes a sequence of feature vectors from the input speech signal. From this sequence, the second module computes a string of words as a hypothesis of the words that have been uttered.

The processes involved in the feature extraction module are: a) Acquisition and sampling of the speech signal (typically at 16 kHz.) b) Computation of the outputs of a mel-scale filter bank at 10 msec. (typically 22 filters) c) Computation of the cepstrum coefficients of the outputs of the filter bank (typically at 10 coefficients plus the energy) d) Computation of the first and second derivatives of the cepstrum coefficients.

Typically, the feature extraction module produces a feature vector of 33 components (cepstrum coefficients and their first and second derivatives) every 10 msec.

The decoding module is based on the statistical approach [6]. In ATROS, the language model is a stochastic regular grammar or an n-gram model represented by stochastic finite state networks.

The acoustic processor is composed of two parts: the sublexical one and the lexical one. The sublexical part consists of phone-like models represented as CDHMMs or hybrid connectionist-structural models. The lexical part consists of word acoustic models which are obtained by the concatenation of sublexical models according to orthographic-phonetic rules. These models are represented by stochastic finite state networks whose transitions are labelled with phone-like models.

For decoding, the word acoustic models are integrated dynamically in the language model: the transitions in the language model are substituted by the corresponding word acoustic models. The

This work was supported by the Spanish CICYT under grant TIC95-0884-CO4-01.

decoding process through the integrated network is performed with a beam-search Viterbi algorithm that is described in section 5.

3. ACOUSTIC, LEXICAL AND LANGUAGE MODELING

3.1. Acoustic and Lexical Models

Different types of sublexical units were modeled through CDHMMs. The emission probability of each state is represented by a Gaussian mixture density with diagonal covariance matrix. ATROS can compute the emission probability density values at each state in two ways. The first one adds the contribution of each Gaussian from the mixture to the total mass of emission probability, while the second takes the highest probability density value from all of the Gaussian emission of the mixtures. This type of computation allows us to use minus-log values of the probabilities and probability densities, and consequently, the computation of a maximum operator presents a lower computational cost than the addition operator. Different numbers of component densities per state were tested in order to study the relation between the word-error rate achieved and the recognition speed. For the experiments reported in section 6, each model had three states without skip transitions.

Hybrid connectionist-structural models composed of hidden Markov chains and Multilayer Perceptrons (MLPs) to estimate the emission probabilities are also proposed for acoustic modeling [3, 4]. Different topologies of the Markov chains and MLPs have been tested.

The lexical models are composed by the concatenation of sublexical models to form word acoustic models.

3.2. Language Models

Two language models were tested in the system. On the one hand, a trigram model was estimated with the first version of the Stochastic Language Model (SLM) Toolkit [11]. On the other hand, a language model was estimated through a grammatical inference technique known as MGGI [12]. This technique, which is conceptually similar to bigrams and incorporates a smoothing method based on the back-off, estimates more accurate language models than bigrams.

4. ACOUSTIC AND LANGUAGE TRAINING

4.1. Acoustic models training

The acoustic models were trained with the acoustic material defined in the SENGLAR project: the overall training database gathers 1,529 utterances from 57 speakers (which accounts for nearly 470,000 acoustic frames and 55,000 phonetic units).

The acoustic CDHMM were trained by using the Baum-Welch algorithm of the HTK toolkit [14] from training data parametrized into sequences of cepstral coefficients by the ATROS system. Two sorts of sublexical units were tested as illustrated in Table 1. The first sort were 27 context-independent phone-like units (including initial, middle and final silences) which were defined in the SENGLAR project. The second sort of sublexical units were triphones (those that appeared at least 100 times in the training data). This set of units was composed of 67 triphones and 27 monophones. Models with 1, 2, 4, 8, 16, 32 and 64 mixtures per state were evaluated. State clustering was used when context-dependent models were trained.

Table 1: Number of total mixtures for phone-like and triphone sublexical units.

Sublexical units	Number of units	Maximum mixtures per state	Total number of mixtures
phones	27	32	2,687
phones	27	64	5,362
triphones	67+27	8	4,284
triphones	67+27	16	5,894

Hybrid models of context-independent units were also trained. The underlying Markov chains had three states or were durational (with a number of states equal to the average duration of the phone-unit). MLPs with 27 output units (one for each phone) and an input layer formed by the actual frame plus four frames of left and right context (nine frames in total). Different sizes of the hidden layer (100, 500 and 1,000 hidden units and two layers of 100 units each) have been tested. We also tried committees of MLPs (CMLPs) [2] in order to obtain better estimation of the emission probabilities of the models. We created several committees of MLPs with two or three MLPs, and the output of the committees was defined as the average of the outputs of each MLP. Finally, the best performance was obtained with a committee of three MLPs and Markov chains of three states. The number of weights of that committee of MLPs was 237,372.

4.2. Language model training

The training set used for the estimation of both language models consisted of 8,262 written sentences (81,700 Words) of Queries to a Spanish Geographic information Database (GDQ) [5], with a vocabulary of 1,208 words. A test set of 1,147 different written sentences (11,845 Words) was used to measure the perplexity of the obtained models.

The perplexity of the test set with the trigram model was 9.44. In order to estimate a model using the smoothed MGGI language model, a labeling function had to be defined (for details see [12]). For the language model used in the system presented in this paper, a relative position renaming function of 5 intervals was defined. The test set perplexity presented by this language model was 12.00 while a bigram model presented a test set perplexity of 16.13.

5. SEARCH

The search for the most likely word sequence is approximated by the most likely state sequence in a network that integrates the acoustic, lexical and syntactic models.

To build the integrated network, the transitions of the language model are dynamically substituted by the corresponding acoustic models. All word models that correspond to transitions which leave from a particular state of the language model are represented as a prefix-tree (tree lexicon) [8].

Figure 1 shows part of an integrated network that corresponds to a smoothed trigram model. The paths that take into account the word k from state ij are illustrated in this figure. Note that $\Pr(k \mid ij)$ and $\Pr(k \mid j)$ may not be in the model, but there always is a path through the unigram model for word k.

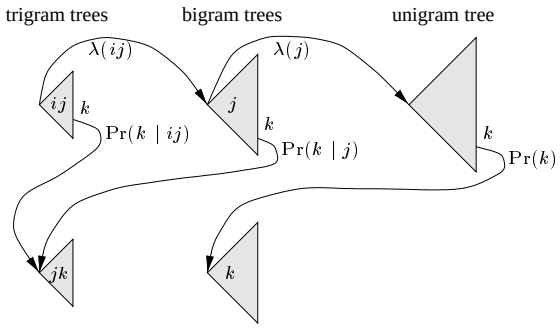

Figure 1: Partial view of the representation of the integrated network corresponding to the smoothed trigram model. $\lambda(ij)$ and $\lambda(j)$ are the interpolation parameters [1] (in practice, these values are the back-off normalizing factors).

5.1. Beam Search

The search for the most likely state sequence in the integrated network can be performed by using the Viterbi algorithm. This search strategy is known as *Synchronous Search* [8].

This strategy is time-consuming; moreover, many paths correspond to low feasible hypotheses that will probably not contribute to the desired solution. *Synchronous Beam Search* [9] is a technique commonly used to overcome this drawback. This technique consists of maintaining only those hypotheses that are more likely to survive in the search process. It is a well-known fact that adequate choices of beam search parameters dramatically reduce the computational search time required without decreasing the system performance.

5.2. Language Model Look-Ahead

One of the main drawbacks of the tree lexicon technique is that the word probabilities of the language model must be applied to the leaves of the tree vocabulary. This fact requires wide beams and, therefore, a great number of hypotheses are generated.

A solution to this problem consists of using the language model probabilities in the tree as soon as possible. This can be done by putting an upper bound of the probabilities associated to the corresponding leaves that are a descendant of a node into each node of the tree [13]. In ATROS, these upper bounds are computed only once and stored in memory. The memory requirements to store these bounds for a complete bigram can be very high. In practice, the use of smoothed languages and their corresponding implementation allows us to minimize these memory requirements.

5.3. Fast Phoneme Look-Ahead

The fast phoneme look-ahead technique [10] has been incorporated in ATROS in order to reduce the number of hypotheses which are considered in the search process and, consequently, to reduce the search time. The main idea of the fast phoneme look-ahead consists of determining whether every new phoneme model which is being started is likely to survive pruning steps in the future. This is decided by computing an approximate score (look-ahead score) using a simpler phoneme model (look-ahead phoneme model) and some future time frames (look-ahead buffer). Several hypotheses could continue with the same phoneme model and therefore this computation should only have to be carried out once. The look-ahead score is combined with the exact score of the predecessor phoneme model and the phoneme is started if this new value is over a certain threshold (in way similar to the beam search). If the phoneme model is started, then the exact score is computed. This means that the optimal path can be pruned and only a suboptimal solution may be achieved.

The fast phoneme look-ahead has been incorporated to the system in a way similar to the one in [10]. The fast look-ahead score is computed every time frame not by using the exact phoneme models, but rather by using a simpler one in order to reduce the amount of computation. In ATROS, the look-ahead phoneme models had three states (the same as the exact phoneme models) and two densities in each state. Look-ahead phoneme models with a greater number of densities were studied, but the recognition results obtained were not offset by the increase in computation. These models were trained by using the HTK toolkit. Several sizes of the look-ahead buffer were tested and the optimal value was three frames.

6. EVALUATION OF THE SYSTEM

In this section we present some experiments that were carried out to both evaluate the performance of the system and to establish the influence of each modeling technique. Several configurations of the system were defined and evaluated in terms of performance as well as complexity. The task was the GDQ (described previously) with a vocabulary of 1,208 words.

The performance of the system was measured on a test set which consisted of 600 utterances from 12 speakers (200 different sentences, 5,655 words) out of the GDQ application task [5]. Note that the GDQ database and the utterances used to train the acoustic models (SENGLAR corpus) were independent with respect to the speakers, the text and the task.

To evaluate the performance of the system, we matched each decoded utterance against the correct transcription of the sentence (in terms of a sequence of words). Then, the word-error rate (wer) was calculated.

6.1. Experimental results

For each experiment we show the obtained word-error rate and a measure of the consumed time, given by the number of seconds which were necessary to process one hundred frames (equivalent to approximately real time). All the experiments were performed on a SGI2 workstation R10000 with 384 MB of RAM.

Different beam-search and grammar-scale factors were proved and the best results for each type of acoustic unit and language model are shown in Table 2. These experiments were carried out without using fast phoneme look-ahead. The best performances were obtained using the trigram-CDHMM64 phone system producing a wer of 10.7% and using trigram-HMM/CMLP phone system giving a wer of 9.6%. However this last system ran faster. On the other hand, the results indicate that the smoothed trigram model estimated with the SLM toolkit performed better than the MGGI language model.

In order to test the efficiency of the fast phoneme look-ahead, we repeated the trigram-CDHMM64 experiment of Table 2 with this technique. The wer obtained in this case was 12.1 and the real time factor was 1.9. This means an increase of 1.5 points in

Table 2: Word-error rate (wer) obtained for the test set along with the real time factor (t). The language model used is shown in the first column, and the sublexical units and the maximum number of mixtures per state in the second one.

Lang. model	Sublexical units		wer	t
MGGI	phones	(32)	14.3	10.4
trigram	phones	(32)	12.7	1.9
MGGI	phones	(64)	12.6	20.5
trigram	phones	(64)	10.7	8.6
MGGI	triphones	(8)	13.6	8.5
trigram	triphones	(8)	12.1	1.7
MGGI	triphones	(16)	13.6	9.3
trigram	triphones	(16)	12.0	2.2
MGGI	phones	HMM/CMLP	10.8	3.1
trigram	phones	HMM/CMLP	9.6	2.1

the wer, but a savings of 78% in time. This result is similar to the one obtained without fast phoneme look-ahead using trigram as language model and triphones with 8 mixtures per state.

7. SUMMARY

An efficient version of the continuous speech recognition system ATROS has been presented. All the models can be learnt automatically from training data. Lexical and syntactic models are represented in a similar format. Acoustic models can be CDHMM or HMM/CMLP. New, well-known search techniques have been introduced to improve the computational efficiency of ATROS.

Experiments on a task of medium complexity (a vocabulary of 1,208 words and a perplexity of 9.44 with a trigram language model) were carried out to assess ATROS. Better performance was achieved using trigram language models instead of MGGI language models. Similar results were achieved with CDHMM and with HMM/CMLP. Triphones did not improve the results achieved with phone-like units.

For future work we will continue the research of grammatical inference methods for language modeling. And we will continue studying the improvement of contextual acoustic models.

8. REFERENCES

[1] G. Antoniol, F. Brugnara, M. Cettolo, and M. Federico. Language Model Representations for Beam-Search Decoding. In *Proceedings of the ICASSP'95*, pages 588–591, Detroit, MI (USA), May 1995.

[2] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1996.

[3] H. Bourlard and N. Morgan. *Connectionist speech recognition—A hybrid approach*. Kluwer Academic, 1994.

[4] M. J. Castro, F. Prat, P. Aibar, and F. Casacuberta. Geometric pattern recognition techniques for acoustic-phonetic decoding of Spanish continuous speech. In *Proceedings of the Eurospeech'95*, pages 1495–1498, Madrid (Spain), 1995.

[5] J. E. Díaz-Verdejo, A. M. Peinado, A. J. Rubio, E. Segarra, N. Prieto, and F. Casacuberta. ALBAYZIN: a Task-Oriented Spanish Speech Corpus. In *First International Conference on Language Resources and Evaluation*, pages 497–591, Granada (Spain), 1998.

[6] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[7] D. Llorens, V.M. Jimenez, J.A. Snchez, E. Vidal, and H. Rulot. ATROS, an Automatically Trainable Continuous-Speech Recognition System for Limited-Domain Tasks. In *VI Spanish Symposium on Pattern Recognition and Image Analysis*, pages 478–483, Córdoba (Spain), 1995.

[8] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder. Improvements in beam search for 10000-word continuous speech recognition. In *Proceedings of the ICASSP'92*, volume 1, pages 9–12, San Francisco, California (USA), March 1992.

[9] H. Ney, D. Mergel, A. Noll, and A. Paeseler. Data Driven Search Organization for continuous Speech Recognition. *IEEE Transactions on Signal Processing*, 40(2):272–281, 1992.

[10] S. Ortmanns, A. Eiden, H. Ney, and N. Coenen. Look-ahead techniques for fast beam search. In *Proceedings of the ICASSP'97*, pages 1783–1786, Munich (Germany), 1997.

[11] R. Rosenfeld. The CMU Statistical Language Modeling Toolkit and its use in the 1994 ARPA CSR Evaluation. In *ARPA Spoken Language Technology Workshop*, Austin, Texas (USA), 1995.

[12] E. Segarra and L. Hurtado. Construction of Language Models using the Morphic Generator Grammatical Inference (MGGI) Methodology. In *Proceedings of the Eurospeech'97*, pages 2695–2698, Rhodes (Greece), 1997.

[13] V. Steinbiss, B.-H. Tran, and H. Ney. Improvements in Beam Search. In *Proceedings of the ICSLP'94*, pages 2143–2146, Yokohama (Japan), September 1994.

[14] S.J. Young, P. C. Woodland, and W.J. Byrne. HTK: Hidden Markov Model Toolkit V1.5. Technical report, Cambridge University Engineering Department Speech Group and Entropic Research Laboratories Inc, 1993.