A NEW INTEGRATED SYSTEM FOR THE CONTINUOUS SPEECH RECOGNITION OF SPANISH

L.J. Rodríguez, M.I. Torres, J.M. Alcaide, A. Varona, K. López de Ipiña, M. Peñagarikano, G. Bordel

Departamento de Electricidad y Electrónica. Facultad de Ciencias.

Universidad del País Vasco / Euskal Herriko Unibertsitatea (UPV/EHU).

Apartado 644. 48080 Bilbao. Spain.

e-mail: luisja@we.lc.ehu.es

Abstract

This paper presents a new system for the continuous speech recognition of Spanish, integrating previous works in the fields of acoustic-phonetic decoding and language modelling. Acoustic and language models -separately trained with speech and text samples, respectively- are integrated into one single automaton, and their probabilities combined according to a standard beam search procedure. Two key issues were to adequately adjust the beam parameter and the weight affecting the language model probabilities. For the implementation, a client-server arquitecture was selected, due to the desirable working scene where one or more simple machines in the client side make the speech analysis task, and a more powerful workstation in the server side looks for the best sentence hypotheses. Preliminary experimentation gives promising results with around 90% word recognition rates in a medium size word speech recognition task[1].

Keywords: Continuos Speech Recognition, Client-Server Arquitecture

1. INTRODUCTION.

This paper describes a new Continous Speech Recognition (CSR) System for medium size Spanish tasks integrated into a client/server architecture. The main original features of the system are:

- Context Dependent Units obtained by applying decision trees for Spanish recognition tasks.
- A syntactic approach of the well-known n-gram models, the K-TLSS Language Models, fully integrated in the recognition system.
- A linear organization of the lexicon fully compatible with the search strategy.
- A client/server architecture where the main search network is implemented as server and the acoustic front-end works independently in one or more clients.

[1] This work has been partially supported by the Spanish CICYT, under project TIC95-0884-C04-03.

Other features of the system include some state-of-the-art speech recognition technologies: robust speech signal analysis, HMM acoustic models, beam-search and (optionally) fast phoneme look ahead algorithms.

The CSR System was evaluated over a task oriented speech corpus representing a set of queries to a Spanish geography database. This is a specific task designed to test integrated systems (involving acoustic, syntactic and semantic models) in automatic speech understanding.

The paper is organized as follows: Section 2 summarizes the speech analysis procedure and the acoustic models. In Section 3 language modelling and search strategy are outlined. Section 4 describes the client/server system architecture. Finally, some preliminary results for the evaluation of the CSR system are presented in Section 5.

2. ACOUSTIC-PHONETIC MODELLING.

Speech analysis -as suggested by [1]- is made by following conventions of the Entropic Hidden Markov Modeling Toolkit (HTK) [2], with some minor changes. Speech -acquired typically through a headset condenser microphone- is sampled at 16 kHz, each sample having 16 bit precision. Speech samples are grouped into successive frames and passed through a Hamming window. Frame length is 25 ms and inter-frame distance 10 ms. A filter bank formed by 24 triangular filters with increasing widths according to mel-frequency scale, is applied to a 512-point FFT, producing 24 spectral weighted mean values. A discrete cosine transform applied to these coefficients decorrelates the spectral envelope from other characteristics not relevant for recognition, producing 12 mel-frequency cepstral coefficients (MFCC). Cepstral mean normalization and liftering are then applied to compensate for channel distortion and speaker characteristics. The normalized logarithm of the energy is also computed. Finally, dynamic characteristics (first and second derivatives) are added to the MFCC and log-energy parameters, obtaining a 39-component acoustic observation vector.

As usual when dealing with medium to large size vocabularies, sublexical units were used as the basic speech units. Three different sets were defined. The first and simpler one contained 24 phone-like units, including a single model for silence. The second one was a mixture of 103 trainable (e.g. with enough training samples) monophones, diphones and triphones, as described in [3]. The well known technique of decision tree clustering [4] was applied to obtain a third optimal set of 101 trainable, discriminative and generalized context dependent units. An additional set of border units was specifically trained to generate lexical baseforms, covering all possible intraword contexts and being context independent to the outside. We are currently working in generating a more general set of inter and intraword context dependent units, and a first approach was made to evaluate their contribution to the recognition process (see Section 5). To deal with multiple codebook observations, as was the case in our baseline system, a simple

and not very expensive discriminative function, combining probabilities from all the codebooks, was used. The set of decision tree based context dependent units gave the best results in acoustic-phonetic decoding experiments and also when building word models for a more reliable test, as will be shown in Section 5.

Discrete Left-to-Right Hidden Markov Models with 3 looped states and 4 discrete observation distributions per state, were used as acoustic models. Emision and transition probabilities were estimated using both the Baum-Welch and Viterbi procedures, applying the Maximum Likelihood criterion.

To deal with discrete acoustic models, a standard Vector Quantization procedure was applied, obtaining four different codebooks, each containing 256 centroids, corresponding to MFCCs, first derivatives of MFCCs, second derivatives of MFCCs and a 3-component vector formed by log-energy, first and second derivatives of log-energy. During recognition, each parameter subvector is assigned the nearest centroid index and a four index tag is passed as acoustic observation to the search automaton.

3. LANGUAGE MODELLING AND SEARCH STRATEGY.

A syntactic approach of the well known n-grams models, the K-Testable Languages in the Strict Sense (K-TLSS) was used. The use of K-TLSS regular grammars allowed to obtain a deterministic, and hence unambigous, Stochastic Finite State Automaton (SFSA) integrating a number of K-TLSS models in a self-contained model [5]. Then a syntactic back-off smoothing technique was applied to the SFSA to consider unseen events [6]. This formulation strongly reduced the number of parameters to be handled and led to a very compact representation of the model parameters learned at training time. Thus the Smoothed SFSA was efficiently allocated in a simple array of an adequate size.

Although words have been used as basic units in our baseline system, some other alternative lexical units, statistically derived from the training text, have been studied, yielding sifgnificative improvements in terms of model complexity and performance, and could replace words in a later implementation.

Lexical baseforms are linearly represented. Each word is transcribed as the concatenation of sublexical units, taking into account only intraword contexts. Word nodes are expanded with the corresponding concatenation of previously trained acoustic models, and one single automaton results with both acoustic and syntactic probabilities.

The time-synchronous Viterbi decoding algorithm is used at parsing time: a simple search function through the array representing the Language Model allows to directly obtain the next state of the SFSA needed at each decoding time. After some preliminary experimentation, a kind

of balance between acoustic and syntactic probabilities was found necessary in the search automaton. A weight α affecting the Language Model probabilities was heuristically optimized. Then a beam threshold is established and (optionally) a fast phoneme look ahead algorithm is included to reduce the average size of the search network [7].

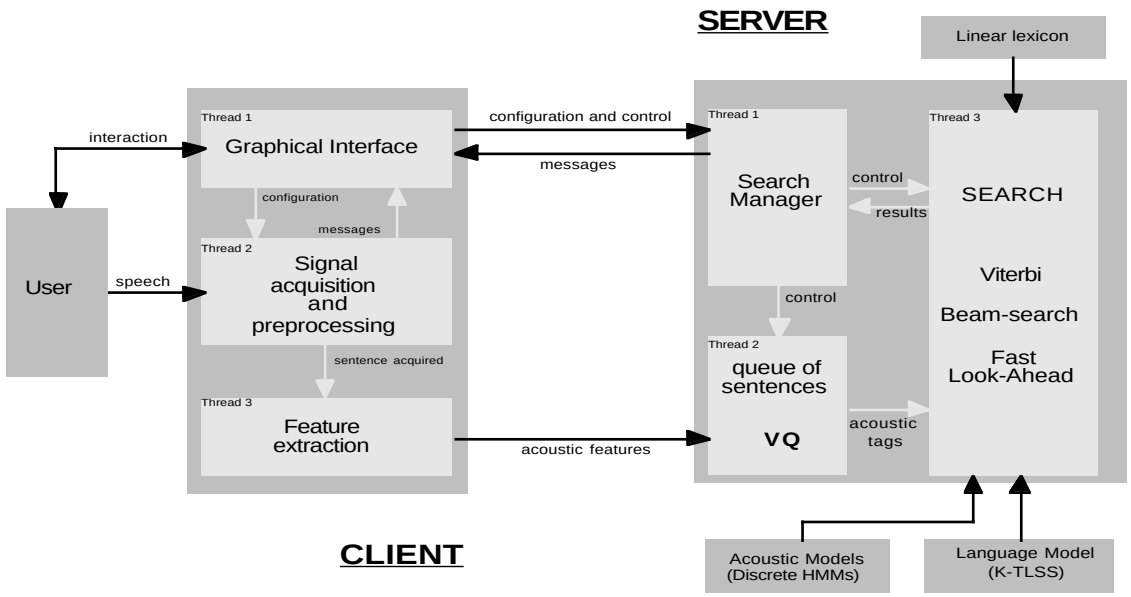

Figure 1. System arquitecture showing both the client and server sides communicating through a local network.

4. SYSTEM ARQUITECTURE.

The system design is based on a distributed, client-server arquitecture. The server is the main program, as it includes the search module. On the other hand, although the more apparent part of the client application is the graphical user interface, it also acquires the audio signal and does the preprocessing (i.e. the edge detection) and the acoustic feature extraction. By means of the graphical interface, which is based on Tcl/Tk [8], the user controls the whole system and gets feedback in the form of detailed messages. The client and server programs may run on different computers attached to a network. Both are highly parallelized through POSIX threads: for example, GUI, audio acquisition and feature extraction are three separate threads of the client program. Figure 1 outlines the main characteristics of the system arquitecture.

This distributed, parallelized arquitecture makes possible to get the maximum performance from modern computer hardware such as networked multiprocessor systems. Clearly, the client

application must run on a computer provided with a graphic display and audio hardware; the CPU power is not important. However, the server application should run on a computer with a faster CPU or more CPUs; by contrast, graphics and audio capabilities are not needed in this case. Obviously, both client and server applications may also run on the same computer.

While the system is running, the audio signal is being acquired continuously but not processed when signal energy is low (silence condition). Whenever a spoken sentence (or some long emphatic sound) is detected, audio data are given to the acoustic feature extraction module. Then a vector composed of the signal features is sent to the server, which adds it to a queue. Another server thread is continuously processing this queue, doing the main job. The server is also provided with an optional VQ module working only when discrete acoustic models are especified, as in our baseline system. Control and configuration messages may be sent at any time from the client to the server using an independent bidirectional communication channel. The same channel may be used by the server for sending messages to the client, which in turn displays them to the user.

5. EXPERIMENTAL EVALUATION.

All the parts of the system were carefully tested and optimized before integrating them. From the audio acquisition module to the search module, separate evaluations were made. Here we present only two of them, reflecting our current research interests: the selection of an adequate set of sublexical units for acoustic modelling, and the integration of acoustic and language models into one single search automaton.

5.1. Evaluating sublexical units.

As mentioned above, three different sets of sublexical units were tested: 24 context independent phone-like units, a mixture of 103 monophones, diphones and triphones, and 101 decision tree based generalized context dependent units. An acoustic-phonetic decoding task was used as benchmark, having a balanced phonetic database both for training and testing purposes. The training corpus was composed of 1529 sentences, involving around 60000 phones. A speaker independent test was applied, containing 700 sentences. Discrete HMMs with 4 codebooks were used as acoustic models. No phonological model was used in these experiments. Table I shows phone recognition rates. Context independent phone-like units gave significative lower rates (around two points) than the two other sets of context dependent units, which in turn showed similar performances.

However, a more reliable test seems necessary to decide which set of units is more suitable, by building lexical baseforms and obtaining word recognition rates. Two different procedures

have been used to obtain lexical baseforms, both considering words as isolated. In the first one, called TR1, border units are selected to be context independent both sides. In the second one, called TR2, border units are selected to be context dependent in the word side and context independent in the outside. Another transcription procedure, called TR3, was used to test the contribution of modelling interword contexts to speech recognition. TR3 takes into account the interword contexts appearing in the test database to obtain lexical baseforms of words. Table II shows word recognition rates using these transcription procedures, in a speech recognition task with no language model. Only the baseforms of the 203 words found in the test database were used to run the alignment procedure, so a closed test was carried out. Word recognition rates show that TR2 works better than TR1 -as may be expected. However, TR3 gave the best performance, showing the importance of modelling interword contexts. Finally, note that DT-based context dependent units outperform the two other sets of units.

Table I. Phone recognition rates for a phonetic-decoding task in Spanish, using three different sets of sublexical units.

Type of unit	# units	% REC
CI phone-like	24	63.97
Mixture (Mph, Dph, Tph)	103	65.90
DT-based	101	66.44

Table II. Word recognition rates for a speech recognition task in Spanish, using three different sets of sublexical units and three different transcription procedures.

	% Word Recognition		
Type of unit	TR1	TR2	TR3
CI phone-like	49.83	-	
Mixture (Mph, Dph, Tph)	-	51.16	56.73
DT-based	52.86	53.26	58.01

5.2. Evaluating the search procedure.

Not only the parameter α weighting syntactic probabilities over acoustic probabilities must be optimized, but also the beam parameter. Actually, both should be optimized jointly, but we first heuristically found an optimum a and then an optimum beam. The beam is a key issue to reach real-time operation, because a balance must be found between system accuracy and computation time.

At any time, each possible transition from each active node in the search network has an acoustic probability and possibly also a syntactic probability, which are multiplied by the

accumulated probability at the departure node, and assigned as accumulated probability to the arrival node. To obtain the set of active nodes at that time, the beam parameter must be applied to discard all nodes whose accumulated probability falls below certain threshold, thus drastically reducing the size of the search network. This threshold is usually obtained by multiplying the beam parameter by the maximum accumulated probability at that time.

Once again discrete HMMs with 4 codebooks were used as acoustic models. Context independent phone-like units were used as sublexical units, and lexical baseforms were generated by applying TR1. K-TLSS models were used for Language Modelling, trained with a task-oriented text containing 8262 sentences, for several values of k. The task, a set of queries to a Spanish geography database, had a vocabulary of 1213 words. For testing purposes, an independent but fully covered text containing 600 sentences was used. Table III shows system performance for values of k=2,3,4, beam=0.67, 0.55, 0.45, and fixed α =6.

Table III. System performance for different values of k and beam, and fixed α=6. Significative measures are showed: average number of active nodes (#AN average), average processing time per frame (PTF average, in milliseconds), word recognition rate (%W) and sentence recognition rate (%S).

k	beam	#AN average	PTF average (ms)	%W	%S
	0.67	40.13	3.80	68.58	29.17
2	0.55	218.21	11.75	84.05	41.67
	0.45	858.51	33.42	85.19	44.17
	0.67	32.24	3.30	70.48	36.33
3	0.55	179.01	10.96	89.15	56.00
	0.45	800.52	36.67	90.35	57.50
	0.67	31.50	3.38	71.36	45.00
4	0.55	177.99	11.24	89.78	59.00
	0.45	808.30	38.19	91.42	61.50

Computational resources required by the system (processing time and memory, i. e. the number of active nodes) reduce drastically as the beam narrows, while system accuracy remains quite good, especially for k=4. Note that only the more constrained search (with beam=0.67) fullfils the condition of real-time operation (i.e. average processing time per frame less than 10 milliseconds).

6. CONCLUDING REMARKS.

In this paper a new CSR System for medium size Spanish tasks was presented. State-of-the-art speech recognition technologies, including Discrete HMMs and K-TLSS models, were integrated into a client/server architecture, where the main search module -a time-synchronous

Viterbi alignment provided with beam search and fast phoneme look ahead algorithms- was implemented as a server, and the acoustic front-end was part of the client application.

Future work includes programming a client application for PCs, expanding the currently available client, designed for Silicon Graphics and Sun workstations, to more common platforms; the use of decision tree based context dependent sublexical units, instead of context independent phone-like units, to form lexical baseforms incorporating interword context modelling; the use of alternative lexical units replacing words; the use of more accurate acoustic models, and finally improving the search module to fully acomplish real-time operation with good system accuracy.

This work is part of a greater CSR project jointly developed with three other spanish universities: Universidad Politécnica de Valencia, Universidad Politécnica de Cataluña and Universidad de Zaragoza. Here we would like to thank all the help and advise obtained from them.

7. REFERENCES.

[1] J.B. Mariño, J. Hernando. "Notes on feature extraction for speech recognition". *Personal communication*. March 1998.

[2] S. Young, J. Odell, D. Ollason, V. Valtchev, P. Woodland. "The HTK Book. Hidden Markov Model Toolkit V. 2.1". *Entropic Cambridge Research Laboratory*. March 1997.

[3] A. Bonafonte, R. Estany, E. Vives. "Study of subword units for Spanish speech recognition". *Proc. EUROSPEECH'95*, pp.1607-1610.

[4] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, M.A. Picheny. "Decision Trees for Phonological Rules in Continuous Speech". *Proc. IEEE ICASSP-91*, pp. 185-188.

[5] G. Bordel, A. Varona, I. Torres. "K-TLSS(S) Language Models for Speech Recognition". *Proc. IEEE ICASSP-97*, pp 819-822.

[6] G. Bordel, I. Torres, E. Vidal. "Back-off Smoothing in a syntactic approach to Language Modelling". *Proc. ICSLP-94*, pp. 851-854.

[7] A. Varona, I. Torres. "Using smoothed K-TSS Language Models in Continuous Speech Recognition". *Accepted, to appear in Proc. IEEE ICASSP-99*.

[8] J.K. Ousterhout. "Tcl and Tk toolkit". Addison-Wesley. Reading, Massachussets, 1994.