COMPARISON OF THREE APPROACHES TO PHONETIC STRING GENERATION FOR LARGE VOCABULARY SPEECH RECOGNITION*

Javier Macías-Guarasa, Manuel A. Leandro, Xavier Menéndez-Pidal, José Colás, Ascensión Gallardo, José M. Pardo and Santiago Aguilera
Grupo de Tecnología del Habla. Departamento de Ingeniería Electrónica. E.T.S.I. de Telecomunicación
Ciudad Universitaria s/n | 28040 - MADRID | SPAIN
macias@die.upm.es

ABSTRACT

We are building a large vocabulary, isolated word preselection system according to a bottom-up design strategy. It will be used in the development of a dictation machine for Spanish and it is composed of three main modules: feature extraction, phonetic string build up and lexical access. In the second one, we are considering three different technological approaches based on static modeling (SM), Hidden Markov Models (HMM) and Neural Networks (NN). This paper will compare these three alternatives in terms of recognition performance, training complexity and computational load, and will conclude with the results of the comparison in order to adopt the most suitable approach depending on the task.

I. INTRODUCTION

The study we are presenting was done to help the decision process of adopting a certain technology for future developments in very large vocabulary speech recognition systems based in the hypothesis-verification paradigm. The stage analyzed here constitutes the first step (hypothetization) towards this goal. The non-integrated (bottom-up) design [8][9] selected, allowed us finer degree of experimentation and detail while tuning and testing the system, and was chosen even at the cost of losing the better accuracy achievable in top-down methodologies, in which a global guiding mechanism limits and restricts the search space.

The performance of a recognition system is not the only feature to consider when addressing the problem of developing a real-world, real-time application. The key factor is the *cost* of the system, taken in a broad sense. This implies finding a balanced trade-off among all the terms included in this broad-concept of cost.

Besides the recognition rate achieved, other attributes of the algorithms such as the training complexity, computational load and memory requirements become important factors to bear in mind. The features of the preselection stage must be low computational cost, ease of training (specially for speaker-dependent or speaker adaptive systems) and high inclusion rate (not to bound the performance of the detailed analysis modules).

In this paper we compare three different systems in the terms mentioned above. The system architecture is presented in section II. Three different algorithms have been implemented for the phonetic string build up module and are described in section III. Section IV compares the three approaches in terms of training complexity, performance and computational cost. Finally, section V presents our conclusions.

II. SYSTEM OVERVIEW

Our large vocabulary, isolated word preselection system is composed of three different modules, as depicted in Figure 1.

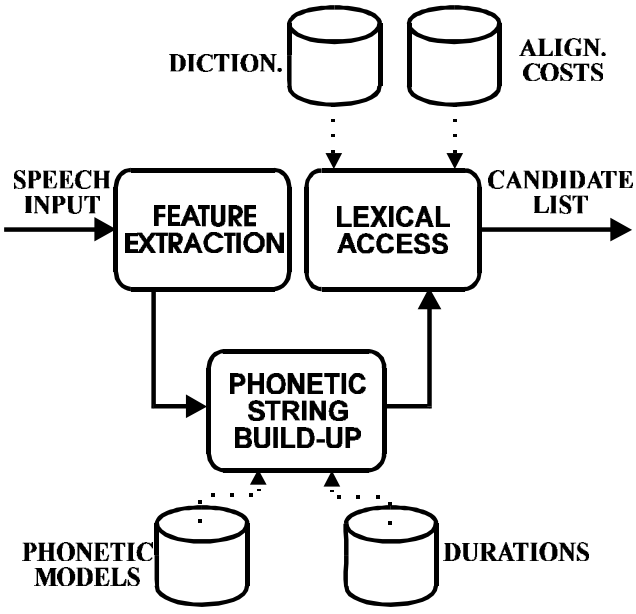

Figure 1.- System architecture

* This work has been partially supported by INSERSO (Ministerio de Asuntos Sociales) and CYCIT grant.

Speech signal is low-pass filtered at 6.25 KHz and sampled at 16 KHz with a resolution of 16 bits. Hamming windowing is then applied to a 16 milliseconds. frame.

The feature extraction module will be discussed later as minor modifications are introduced for the different technologies applied.

The phonetic string build up is the core of the preselection system, as the quality of the output strings will bound the recognition accuracy of the lexical access. As mentioned above the three different approaches considered are based, respectively. in static modeling (SM), Hidden Markov Models (HMM) and neural networks (NN).

The lexical access module computes similarities between the phonetic string and each item of the dictionary [6]. The search is implemented as a tree to save computational effort. Reductions from 50 to 70 % are achieved compared with a linear search. The algorithm is based on a dynamic programming procedure. In each point of the search space, substitution, insertion and deletion alignment errors are considered. For each of them a cost is calculated in the training stage.

III. ALTERNATIVES FOR PHONETIC STRING BUILD UP

III.1. Static Modeling Based Approach

For this approach the feature extraction module calculates a vector composed of 10 mel frequency cepstrum coefficients (MFCC) plus log-energy every 10 milliseconds.

The static modeling exploits the phonetic-acoustic characteristics of Spanish language, that can be reasonably modeled considering only stable portions of speech. This system only models the stable Spanish phonemes: vowels (a e i o u), main unvoiced fricatives (f T s X), voiced fricatives (B D G), main nasals (m n) and main liquid (l). Each model (called prototype in this context) consists of a vector of 10 MFCC parameters obtained by averaging several frames in which the phoneme to be modeled appears.

The recognition process makes a distance measure between each frame and each prototype, and keeps the closest prototype symbol, generating a symbol sequence (one symbol per frame). The phonetic sequence information is combined with a spectral stability function and heuristic rules to obtain the final phonetic string [2].

In the training stage, the 15 prototypes are extracted from a 50 word database and the procedure is fully automatic.

III.2. Discrete Hidden Markov Model Based Approach

In this case, the feature extraction is similar to the SM system, but frame shift is reduced to 6.25 milliseconds, and vector quantization is performed prior to recognition using a 128-centroids codebook. Mahalonobis distance with diagonal covariance matrix is used.

The generation of a phonetic string is done using a low-cost, frame-synchronous, one-pass algorithm [4], based on discrete Hidden Markov Models (DHMM) and modified in order to allow the use of phoneme durations, phoneme-pair statistics and beam-search. A backtracking procedure recovers the most likely sequence of recognized phonemes at the end of the utterance, but a module based on partial traceback [5] techniques has been designed to allow parallel execution of the lexical access algorithm.

In the implementation we are comparing, 23 phoneme-like units (3 states per DHMM with single and double transitions allowed) are used, plus another two models for initial and final silence in order to increase robustness against errors/deviations in the endpoint detection. The number of units has been chosen to keep computational load as low as possible while maintaining acoustic modeling accuracy and to increase the reliability of parameter estimation with a small training database.

The training database is composed of 500 words for each speaker for both codebook design and HMM parameters estimation.

III.3. Neural Network Based approach

For this approach a vector composed of 17 parameters (16 log-energy in mel frequency scale plus log-energy over the whole frame) is computed every 10 milliseconds.

A Time State Neural Network (TSNN) [4] is trained to discriminate 28 allophones plus silence using a hand-labeled 500 words database. A Time Delay Neural Network was also studied and it proved to be more expensive in training time because it has to handle more neurons on the 1st layer to have competitive results. In the 1st layer for the TSNN we need 25 neurons at least to achieve good performance. The major drawback is the huge training time required for the task.

The recognition process extracts the most-likely allophone in each frame, generating a phonetic sequence. This sequence is then filtered using a 7-points majority filter in order to eliminate the spurious symbols produced by the net. Finally, a merging procedure guided with time duration constrains, extracts the final phonetic string. With this high data compression the system does not lose accuracy, while suppressing the redundancy of the phonetic sequence, the cost in the time alignment process is reduced by a factor of 9 [12].

IV.- COMPARISON

Systems comparison will be done based upon three different measures: complexity of the training stage, recognition performance (for different tasks) and computational requirements (CPU time and memory).

IV.1.- Training Complexity

Training complexity has been measured according to the following points:

- Size of the training database (number of words to be spoken in the training script for each new speaker).
- Convergence (number of iterations of the training algorithm for optimal performance).
- Time consumed in the training procedure (measured in a SUN-3 workstation under the same conditions of computational load).
- Number of parameters to estimate.
- Use of manual phonetic-labeling (referring to the table below, the NN approach in the current implementation is the only one which uses this feature, although automatic labeling could also be applied in this case).

In Table 1 below, we show the figures for each of the terms defined:

	SM	DHMM	NN
Size	50	500	500
Convergence	4-5 iter.	10-20 iter.	400 iter.
Time	0.5 hours	20-30 hours	15-20 days
Parameters	150	9057	10250
Manual labeling	NO	NO	YES

Table 1.- Comparison of training complexity

IV.2.- Performance

We are showing the recognition rate achieved by each system assuming the following points:

- Each system used the strategy defined in section III to train the phoneme-like models.
- The recognition set is composed of 500 isolated words, different from the training set.
- Comparison for one speaker will be shown for the three alternatives (table 2 below) and two speakers for the SM and HMM approaches (table 3 below, where results for the speaker in table 2 are replicated for clarity).
- The vocabulary used is 2000 words, extracted according to frequency of use in a newspaper text corpus database. No language model is used (perplexity equals the vocabulary size).The lexical access module is equivalent in the three approaches. The only difference is the strategy used to train the alignment costs: In the SM modeling, a multi-speaker matrix is calculated from a set of 10 speakers x 500 words per speaker (5000 words) training database (so that a new speaker does not need additional training). In the other two approaches, the matrix is speaker dependent and 500 words are used for training.

Results are shown as a function of the position in the preselection list in which the word was correctly recognized, as we are specially interested in the inclusion rate achieved by each of the methods proposed:

Results for speaker SAN (%)			
Candidate position	SM	DHMM	NN
1	77,1	86,5	84,5
2	86,5	93,6	92,5
5	94,0	97,7	96,5
10	96,7	98,6	98,2
40	98,4	100.0	99.4

Table 2.- Comparison of recognition performance I

Results for speakers SAN and MOR (%)						
	SM			DHMM		
Cand.	SAN	MOR	Avg.	SAN	MOR	Avg.
1	77,1	61,1	69,0	86,5	75,8	81,2
2	86,5	73,8	80,1	93,6	85,0	89,3
5	94,0	86,2	89,8	97,7	90,8	94,3
10	96,8	91,5	94,2	98,6	93,6	96,1
40	98,4	97,8	98,1	100,0	98,8	99,4

Table 3.- Comparison of recognition performance II

The results shown are also competitive compared to the POLYGLOT isolated word recognition system described in [10], which is based in the same static modeling described above.

The statistical relevance of the differences between both DHMM and NN compared with SM was confirmed applying McNemar´s test [11].

Computational cost, measured as memory requirements and computational load is a key factor involving the adoption of a certain technology in the development of an usable real-world application. Values for memory requirements (size of main data structures used, as code size is assumed to be irrelevant in comparison) and computational load (effective CPU time of the algorithms using a SUN-3 workstation under the same load conditions) are shown in table 4 below:

	SM	HMM	NN
Memory requirements (Kbytes)	1.0	36.2	41.0
Execution time (average seconds per word)	7.0	13.7	29.5

Table 4.- Comparison of computational requirements

IV.3.- Performance for a cerebral-palsy speech database

In this section, we present additional comparisons on a cerebral palsy speech recognition task (in the same conditions than the experiments described in section IV.2.). The speaker was an adult with moderate dysarthria.

This study gives us another point of comparison in a much more difficult task, to confirm the tendency of the results shown above. Experiments were carried out only for the SM and DHMM systems and its results can be seen in Table 5 below (in this case, training of the alignment costs was done using 500 words for both alternatives).

Candidate position	SM	DHMM
1	45,9 %	55,7 %
2	57,6 %	66,8 %
5	73,9 %	78,6 %
10	81,7 %	85,8 %
40	93,6 %	93,8 %

Table 5.- Comparison of recognition performance for a Cerebral Palsy speech task

V.- CONCLUSIONS

The figures in the preceding section show the advantages and drawbacks of the three different approaches used in the development of a phonetic string build up module for isolated word recognition.

While the static modeling technology is a cheap alternative both in training cost and computational complexity, it had worse recognition performance, although the differences are less important if we consider the rate achieved for the whole preselection list.

On the other hand, HMM and NN proved to be comparable in performance and more expensive in resources demands. Of these two, the NN approach showed the highest resources demand, so that it is less suitable for developing a real-time application.

The HMM approach will be improved adding additional refinements in the acoustic modeling (use of differential parameters, increasing number of models and size of codebook to gain accuracy, etc.). Further research is to be done on robust speaker adaptation techniques to shorten the size of the training script, which is one of the major drawbacks of this alternative.

Both HMM and NN are suitable for large and very large vocabulary tasks, while SM is indicated when saving resources is a must and medium to large-sized vocabularies are needed.

In this moment the SM and DHMM approaches are the base of the first prototype of a real-time, large vocabulary isolated word dictation machine for Spanish. Details on the evolution and implementation status of the system can be found in [7] in this same conference.

VI.- BIBLIOGRAPHY

[1] R. Billi et al. “Word preselection for large vocabulary speech recognition”. ICASSP, pp. 65-68, 1986

[2] M. Leandro and J.M. Pardo. “Low cost speaker dependent isolated word speech preselection system using static phoneme pattern recognition”. Eurospeech, vol. 1, pp. 117-120, 1993.

[3] Y. Komori. “Time state neural network (TSNN) for phoneme identification by considering temporal structure of phonetic features”. ICASSP, vol. 1, pp. 123-125, 1991.

[4] H. Ney. “The use of a one-stage dynamic programming algorithm for connected word recognition”. IEEE Transactions on ASSP, vol. 32, n. 2, 1984

[5] P.F. Brown et al. “Parcial traceback and dynamic programming”. ICASSP, pp. 1629-1632, 1982.

[6] L. Fissore et al. “Lexical access to large vocabularies for speech recognition”. IEEE Transactions on ASSP, vol. 17, n. 8, pp. 1197-1213, 1989.

[7] J. Macías-Guarasa et al. “On the development of a dictation machine for Spanish: DIVO”, In this conference, 1994.

[8] H. Ney and R. Billi. “Prototype systems for large vocabulary speech recognition: Polyglot and Spicos”. Eurospeech, pp. 193-200, 1991.

[9] P. Buttafara et al. “Architecture and implementation of the Olivetti PC-based very large vocabulary isolated word speech recognition system”. Eurospeech, pp. 90-93, 1990.

[10] ESPRIT PROJECT 2104 - POLYGLOT-1 (1992): Sixth Semester Report. February-August 1992.

[11] L. Gillic and S. J. Cox. “Some statistical issues in the comparison of speech recognition algorithms”. ICASSP, pp. 532-535, 1989.

[12] X. Menéndez-Pidal, J. Macías-Guarasa, M A. Leandro et al. “Experiments with Neural Networks in Isolated Word Recognition Tasks”. To appear in EUSIPCO 1994.