DEMISYLLABLE-BASED HMM SPOTTING FOR CONTINUOUS SPEECH RECOGNITION

Eduardo Lleida, José B. Mariño, Climent Nadeu, Joan Salavedra
Dept. of Signal Theory and Communications
Universidad Politécnica de Catalunya
08034,Barcelona, Spain
lleida@tsc.upc.es

ABSTRACT

This paper describes the acoustic processor of a Spanish Continuous Speech Recognition System based on Demisyllable units. The acoustic processor is based on a spotting algorithm which takes as input the unknown utterance, the HMM of the reference demisyllables and the lexical knowledge in terms of a finite state network. The spotting algorithm is a modified version of the one-step Viterbi algorithm with multiple hypothesis [1]. The output of the system is a lattice of word hypothesis suitable to be parsed by a linguistic analyzer. The proposed acoustic processor was tested using the integers from 0 to 1000 and the telephonic numbers in a speaker independent approach. The results show the good performance of the demisyllable as recognition unit for the Spanish language and the efficiency of the spotting algorithm.

1. INTRODUCTION

One of the most important problems in speech recognition is to build a robust acoustic processor. The definition of an acoustic processor for continuous speech recognition involves some questions related with the language to be recognized and the architecture of the system. The phonetic unit, the lexical knowledge, the speech model and the mechanism of recognition are some of those questions.

During the last years, many answers have been proposed to those questions and suitable systems have been built [2,3,4]. Most of those systems have a common factor, they are based on phonetic HMM modeling of the speech. This approach implies the definition of the phonetic unit which is highly depended on the language and vocabulary to be recognized. The Spanish language has a syllabic character which suggest to use the demisyllable as phonetic unit. The inventory of Spanish demisyllables is relatively small: less than 750 units. Thus, demisyllables afford a convenient phonetic coding of Spanish utterances. The lexical knowledge describes words in terms of demisyllables. This information is compiled in a finite state network infered from the word vocabulary [5]. This approach provides a compact representation of the lexical knowledge in terms of predecessors and successors of the phonetic units. The last question is how to locate the words of the vocabulary in the speech signal to give a lattice of word hypothesis. Our proposal is to use a word spotting algorithm driven by the lexical knowledge. It takes as input the unknown utterance, the HMM of the demisyllables and the lexical knowledge. The output is a lattice of word candidates. The spotting algorithm proposed in this paper has the following features:

1) The spotting algorithm is a modified version of the time-synchronous Viterbi algorithm with two levels: the demisyllable and the word levels.
2) Since the lexical knowledge is given in a compact finite state network, several words can share some phonetic units which makes necessary to generate multiple hypothesis in the demisyllable level of the Viterbi algorithm [1].

The paper is organized in the following way: Section 2 describes an overview of the system, in Section 3 the spotting algorithm is presented, Section 4 provides the experimental results, and finally, Section 5 contains the main conclusions.

2. SYSTEM OVERVIEW

Figure 1 shows a general block-diagram of the system architecture. The heart of the system is the spotting algorithm which is driven by the lexical knowledge compiled in a finite-state network.

2.1. Signal Processing

The speech signal is band-pass (100 Hz - 3400 Hz) filtered by an antialiasing filter and sampled at 8 kHz. The utterance is isolated by an end-point detection algorithm and pre-emphasized. A linear prediction based parameterization follows: the signal is segmented into frames of 30 miliseconds by a Hamming window at a rate of 15 miliseconds, and every frame is characterized by a LP-fiter with 8 coefficients. Afterwards, 12 band-pass lifted cepstrum coefficients are computed [6]; the energy of the frame completes the parameterization. Before entering the recognition algorithm, the system evaluates the spectral difference with a time-average of 90 miliseconds [8]. In a similar way, the energy difference is calculated. The spectral vector and the spectral and energy differences are vector-quantized separately; in that way, every frame of the speech signal is represented by three symbols.

According to the most recent proposals, our system considers energy and time evolution information. However, the energy is not used directly as a parameter of the signal. This is because the energy depends on the prosody of the sentence and the intensity of the utterance, two very fluctuant features of speech. On the contrary, if the energy is expressed by a logarithmic measure, its difference does not vary with a change in the intensity of the overall sentence, and the variation due to prosodic effects is greatly alleviated.

This work was supported by the PRONTIC grant number 105/88

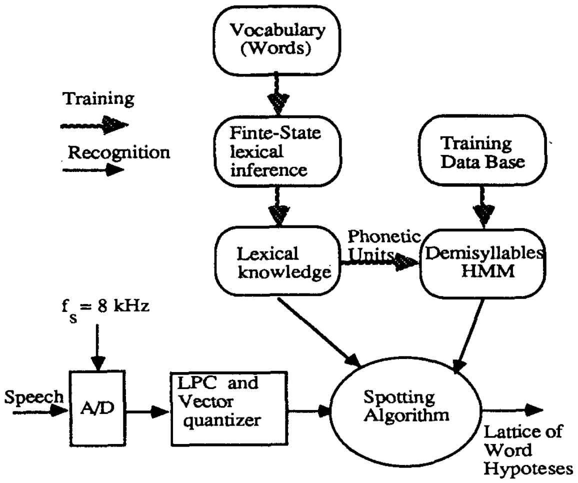

Figure 1. Acoustic processor architecture

2.2. Phonetic unit

Demisyllables afford a convenient phonetic coding of Spanish utterances, according to the syllabic character of this language. In order to define the demisyllable set, every possible syllable was divided by the strong vowel into an initial demisyllable and a final demisyllable; accordingly, we distinguished between stressed final demisyllables and unstressed final demisyllables. The main cues of prosodic stress in Spanish are pitch, loudness and syllable length; as pitch and loudness information are not considered in our system, the main difference between stressed and unstressed final demisyllable is the length of their references.

2.3. HMM demisyllable units

The structure used for the HMM is the typical left-to-right structure, that allows to skip one state when the model makes a transition between states. The emission of symbols is associated to the states, that issue three independent symbols (spectrum, spectrum difference and energy difference) when they are visited.The number N of states was determined [7] as a function of the average length of the demisyllable, according with table 1.

Finally, each demisyllable reference is composed by a HMM and the mean and variance of the length of the demisyllable.

Average length in frames	≤4	5,6	7,8	9,10	>10
Number of states	2	3	4	5	6

Table 1. Criterion to select the number of states of HMM

2.4. Data bases

Three data bases have been used for testing the system:

DB1) 40 strings of integers uttered by ten speakers (S0 to S9, 5 male and 5 female), for example, 25011/96, 1019/05/70. This data base was segmented by hand into demisyllables and used for training the HMM of the demisyllable units. The articulation rate of speech spanned from 5 to 7 syllables per second.

DB2) Telephone numbers uttered by nine speakers (S0 to S1 and S10 to S16, 5 male and 4 female). The telephone numbers were uttered as chains of numbers from 0 to 99, for example, 3/12/36/54, 58/66/15/9 and so on. This data base was used for testing the system. The vocabulary is composed by 25 words with 61 demisyllables.The articulation rate of speech spanned from 5 to 7 syllables per second.

DB3) 44 Integers from cero to one thousand uttered by ten speakers (S0 to S1 and S10 to S17, 6 male and 4 female), for example, 495 /four hundred and ninety five/. This data base was used for testing the system. The vocabulary is composed by 32 words with 66 demisyllables. The articulation rate of speech spanned from 4 to 7 syllables per second.

2.5. Discrete HMM training.

Each model was trained independently of the others. Once the samples of every demisyllable were collected from de utterances of DB1, the Baum-Welch estimation algorithm was applied. At the same time, the mean and the variance of the length of the demisyllable was computed.

We use three independent codebooks of 64 codewords for the two codebooks dedicated to spectral information and 32 codewords for the codebook devoted to energy differences.

2.6. The lexical knowledge.

The lexical knowledge compiles all expected phonetic realizations of the vocabulary words in a network. Classically, this network is a tree where all words having the same first N phonetic units share the same initial nodes of the tree. The last node of the pronunciation has a pointer to the word. However, this representation is not convenient for applying to a time-synchronous Viterbi algorithm which is the base of our spotting algorithm. Thus, our approach is based on the use of a finite-state lexical network as the used for the finite-state grammars. In this case, the lexical knowledge is described in terms of lexical units (states of the network) and the predecessor or successor states of all of them. Defining the phonetic unit as every demisyllable used to consider the different sounds in the language, a phonetic unit can have associated several states in the lexical network which form the lexical units. We have developed an inference algorithm [5] for finite states grammars that can be used to build the finite-state lexicon. The algorithm operates by simple enumeration of the words and gives in two steps a finite-state network with a minimum number of states. Thus, our lexical knowledge is composed by a dictionary tree with the pronunciation of the words in terms of demisyllables and a compiled version of this dictionary in terms of a finite-state network suitable for driving the spotting algorithm. Figure 2 shows an example of a dictionary tree and their corresponding finite-state lexicon network.

The three data bases used in our experiments are based on the vocabulary of numbers. We have to take into account that can be some variations in the pronunciation of the words with regard to a standard phonetic transcription. Thus, we have to expand the dictionary tree and the finite-state lexical network with the most frequent allophonic variations. Thus, the same word can have more than one end node in the dictionary tree. This expansion of the dictionary tree makes more efficient the representation in a finite-state network. For instance, the words needed to cope with the integers from 0 to 1000 (DB3) are represented with a network of 85 lexical units when the number of final nodes in the dictionary tree is 82.

Finally, we distinguish two levels in the lexical

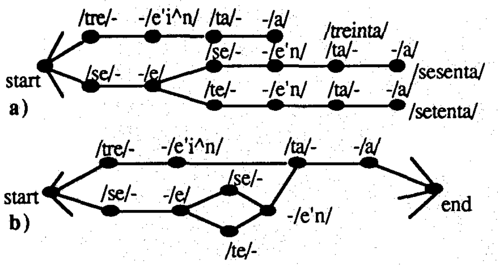

Figure 2. a) Basic dictionary tree for the Spanish words /treinta/ (thirty), /sesenta/ (sixty) and /setenta/ (seventy). b) Finite-state lexical network for these words.

knowledge; the demisyllable level where the demisyllables are classified in two classes: initial demisyllables and final demisyllables, and the word level where the lexical units are classified in three classes: start units, inside units and end units. The start units are the subset of initial demisyllables that can be the first demisyllable of a word, the inside units are both initial and final demisyllables and the end units are the subset of final demisyllables that can be the last demisyllable of a word.

3. THE SPOTTING ALGORITHM

The heart of the system is the spotting algorithm. It takes as input the unknown utterance, the HMM of the demisyllables and the lexical knowledge in terms of a finite- state network. The spotting algorithm is a one-step time-synchronous Viterbi algorithm which gives for each input frame the likelihood that each word of the vocabulary ends in that frame. Each input frame could be a starting point of a path in the Viterbi decoding, that is, the starting constraint of the time-synchronous algorithm is relaxed. We relax the starting constraint using the following criterion: "a frame will be a starting point of a path when its observation probability in the first state of the HMM is greater than the observation probability of the path followed until that frame". Thus, in the first state (HMM) of a start unit, the average of the observation probabilities of the actual path is compared with the observation probability of the input frame in this state. When the difference between the observation probability and the average probabilities is greater than a threshold, that frame is marked as a starting point of a path. The threshold may vary from 0 to 1. When the threshold is equal to 1 only the first frame can be a starting point. In our experiment, a threshold of 0 was used. As each frame could be a starting point of a path, we need to normalize the probabilities by the length of the path to compete all the path in the same conditions. One unit of length is defined by an observation probability and the transition probability. Thus, the length of the path is the difference between the actual path point and the starting point of the path. The probabilities of the states are updated time-synchronously by comparing the normalized probability of each path. Figure 3 shows an example of this criterion.

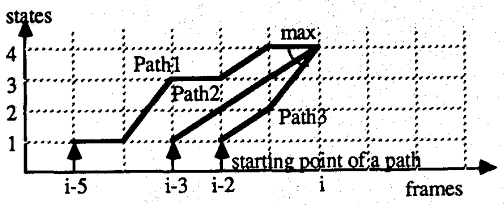

Figure 3. The probability of each path must be normalized by the length of the path prior to compare.

Then, the update probability for the frame *i*, state *s* will be:

$$P(i,s)=b_s(o_i)\ a_{ks}\ P(i-1,k) \qquad (1)$$

where

$$k=\underset{s,s-1,s-2}{\text{argmax}}\left|\begin{array}{l}P(i-1,s)\ a_{s,s}/Path1\\ P(i-1,s-1)\ a_{s-1,s}/Path2\\ P(i-1,s-2)\ a_{s-2,s}/Path3\end{array}\right. \qquad (2)$$

To spot a word, the Viterbi path has to go from a start unit to an end unit. That means that we have to define a between-unit transitions which are controled by the lexical network. The last state of each HMM has associated a duration probability which determines the transition probability between units. The duration probability of a demisyllable is parameterized by the mean and the variance of a Gaussian distribution.

Due to the fact that a lexical unit can be shared by several words, we have to modify the time-synchronous algorithm to generate multiple hypothesis in the between-unit transitions [1]. That modification implies to keep the N-best sequence of lexical units in each transition.

Finally, for each input frame, a probability measure can be obtained in the last state of each end unit which gives the probability that each word of the vocabulary ends in that frame. A prunning strategy is used to keep only the M-best word probabilities and a backtrack procedure over the lexical units is done to find the M-best words that end in each frame. Once, all the frames of the unknown utterance have been processed, a merging procedure is actived to select the P-best words which will compose the lattice of word hypothesis which is the output of the acoustic processor. The merging procedure select the most probable location of a word when it has been detected in successive starting and ending frames. Figure 4 shows an example of the spotting results that provides the merging procedure.

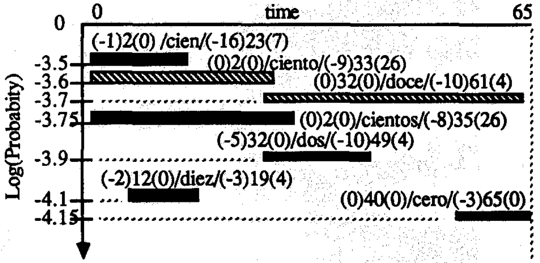

Figure 4. Spotting results analizing the number /ciento doce/ (one hundred and twelve). For each word the system gives the following information: word recognized, the best location with its probability and the variation in the starting and ending point. For instance, "(-1)2(0)/cien/(-16)23(7)" means: the word recognized was /cien/, the best location (-3.5 of probability) was between the frames 2 and 23, but the same word can begin 1 frame before the best location and can end 16 frames before the best ending point and 7 after the best ending point.

4. EXPERIMENTAL RESULTS

4.1. Demisyllable spotting results.

A first set of experiments were carried to test the spotting algorithm. We compare the one-step spotting algorithm with an exhaustive I-steps spotting algorithm where the starting point for the i-th step is the i-th input frame. We use as recognition units the demisyllables. The training data base was DB1 and the test data base was a subset of 5 utterances of each speaker of the DB3 data base. The number of demisyllables to be located was

66. The spotting algorithm gives for each frame the best demisyllable that ends in that frame. Thus, we decide that a demisyllable has been located if it appears around the rigth position. A demisyllable is located in a 1 hypothesis if the right demisyllable is the one with maximum probability of the set of posible demisyllables which are around the right position, it is the 2 hypothesis if it has the second maximum probability and so on. Table 2 shows a summary of the results. From this results, we can conclude that our spotting algorithm has as good performace as an exhaustive method like the I-steps spotting algorithm. Furthermore, the segmentation given by both algorithms is very similar (equal or with differences of 1 or 2 frames).

	1 Hyp.	2 Hyp.	3 Hyp.	>4 Hyp.
1-step	61.8%	69.5%	74.7%	84.3%
I-steps	63.0%	71.0%	75.5%	85.1%

Table 2. Results of demisyllable spotting.

4.2. Acoustic processor results.

The performance of the system in a speaker independent approach was tested with the DB2 and DB3 data bases. The system was trained with the DB1 data base which has a different articulation rate, different sentences and different speakers than DB2 and DB3. Two experiments were carried with each data base. The first experiment use the finite-state lexical network without multiple-hypothesis in the between-unit transitions (1 choice) and the second experiment use the finite-state lexical network with multiple-hypotesis (N choice). Over the lattice of words, we define the hypothesis levels as the position, in probability order, of the correct word in its correct position in the utterance. Figure 4 shows the recognition rates for the data base of numbers (DB3). The accuracy of word spotting was about 82% for the first hypotesis level, 95% for the top five hypotesis levels without multiple hypothesis in the demisyllable level (1 choice) and 99% for the top five hypotesis levels with multiple hypotesis in the demisyllable level (N choice, where N depends on the units with more predecessors, in this experiment N=4). The average number of words in each sentence (integers from 0 to 1000) was 2.56.

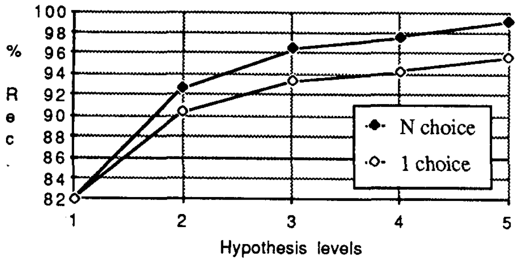

Figure 4. Recognition rates of words for the data base of numbers (DB3) (N=4).

Figure 5 shows the recognition rates for the data base of telephone numbers (DB2). It can be noted a small degradation of the recognition performance. One reason is the fact that the average number of words per sentence is 6.2 and then the words are more coarticulated. Nevertheless, the accuracy of word spotting is still high, 74 % for the first hypothesis level and 93 % for the top five hypothesis levels without multiple hypothesis and 98 % for the top five hypothesis levels with multiple hypotesis in the demisyllable level.

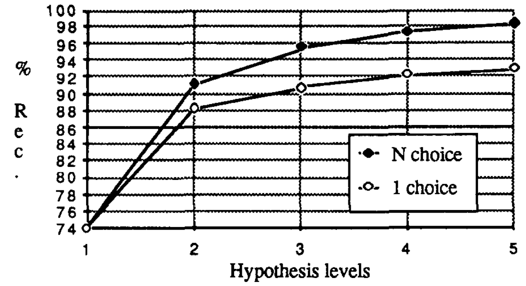

Figure 5. Recognition rates of words for the data base of telephone numbers (DB2) (N=4).

5. CONCLUSIONS

We have developed an acoustic processor for Spanish continuous speech recognition based on the use of a HMM spotting algorithm and demisyllables as phonetic units. The integration of the one-step spotting algorithm with multiple hypothesis and the lexical knowledge compiled in a finite-state network gives an efficient and accurate acoustic processor to generate a word lattice. A demisyllable spotting accuracy of more than 70 % and a word spotting accuracy of more than 90 % in the recognition of the integers from 0 to 1000 and the telephonic numbers, show the good performance of the demisyllable as recognition unit for the Spanish language and the efficiency of the spotting algorithm.

We are currently working to improve the training of the HMM models and to develope a linguistic processor.

AKNOWLEDGMENTS

The authors would like to thank Mr. Antonio Bonafonte, Research Engineer, for his useful discussion and help in the software development.

REFERENCES

[1] J.B. Mariño, E. Monte, "Generation of multiple hypothesis in connected phonetic-unit recognition by a modified one-stage dynamic programming algorithm", EUROSPEECH-89, 408-411, Paris 1989.

[2] Y.L Chow et al. "BYBLOS: The BBN continuous speech recognition system", Proc. ICASSP-87, 89-92, 1987.

[3] K.F. Lee et al. "An overview of the SPHINX speech recognition system", Trans on ASSP-38, 35-45, Jan. 1990.

[4] S. Nakagawa, "Speaker-independent continuous-speech recognition by phoneme-based word spotting and time-synchronous context-free parsing", Computer Speech and Language, vol 3, 227-299, 1989.

[5] J.B. Mariño et al. "Finite state grammar inference for connected word recognition", EUSIPCO-88, 1059-1062, 1988.

[6] B.H. Juang, "On the use of bandpass liftering in speech recognition". IEEE trans ASSP-35, 947-954, July 1987.

[7] J.B. Mariño et al., "Recognition of numbers by using demisyllables and Hidden Markov Models", Proc. EUSIPCO-90, 1363-1366, Sept. 1990.

[8] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum", IEEE trans ASSP-34, 52-59, Feb. 1986.