2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

# TWO LEVEL CONTINUOUS SPEECH RECOGNITION USING DEMISYLLABLE-BASED HMM WORD SPOTTING

Eduardo Lleida, José B. Mariño, Climent Nadeu, Albert Oliveras*
Dept. of Signal Theory and Communications
*Dept. of Automatics and Systems Engineering
Universidad Politécnica de Catalunya
08034,Barcelona, Spain
lleida@tsc.upc.es

## ABSTRACT

This paper describes a two level Spanish Continuous Speech Recognition System based on Demisyllable HMM modelling, word-spotting and finite-state lexical and syntactic knowledge. The first level, the word level, is based on a spotting algorithm which takes as input the unknown utterance, the HMM of the reference demisyllable and the lexical knowledge in terms of a finite-state network. The output of the word level is a lattice of word hypothesis [1]. The second level, the phrase level, searches in a time-synchronous procedure the best sentence that end at each time instant. It takes as input the word lattice and the syntactic knowledge in terms of a finite-state network, giving as output the best legal sentence. The proposal two-level system was tested recognizing the integers from 0 to 1000 in a speaker independent approach. We get a word accuracy of 93,2% with a sentence accuracy of 84.5%.

Keywords: Speech Recognition, Hidden Markov Model, Fuzzy Training, Demisyllable, Word-spotting, Multiple Hypothesis, Finite State Networks.

## 1. INTRODUCTION

During the last years, many continuous speech recognition systems have been proposed giving good results in different task and vocabulary size. Attending to the use of the acoustic, lexical and syntactic knowledge, the systems can be divided in two groups, the integrated systems, where all the knowledges are integrated in a large network to represent all possible sentences in the task [2,3,4] or the multi-level systems, where each knowledge source works with the results of the others [5,6]. Up to now, the integrated systems get better results, but the multi-level systems let apply the natural language advances easier, being a good approach to understand the spontaneous speech. Our current approach goes in that direction.

In this paper, we describes a two-level continuous speech recognition systems based on separating the acoustic and lexical knowledge from the syntactic knowledge. The first level makes use of the acoustic and lexical knowledge to translate the speech signal into a lattice of word hypothesis. We call this level ' word level ', which works as an acoustic processor. The definition of the word level for continuous speech recognition involves some questions related with the language to be recognized and the architecture of the system. The Spanish language has a syllabic character which suggests to use the demisyllable as phonetic recognition 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. This approach provides a compact representation of the lexical knowledge in terms of predecessors and successors of the phonetic units. To locate the words of the vocabulary in the speech signal, we make use of 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 second level of our approach is the phase-level which takes as input the word lattice and the syntactic knowledge. In this paper, the syntactic knowledge is given by a finite-state network as in the lexical knowledge. The parser is a Viterbi search algorithm which is controled by the syntactic knowledge and it is time-syncronous with the ending time of the spotted words.

As the lexical and syntactic knowledges are compiled in a finite-state network and the search procedures are based on the Viterbi algorithm (word-spotting and parser), several words or sentences can share some states of the network, thus , if we want to generate the best word hypothesis or sentence hypothesis, it is necessary to use a multiple hypothesis algorithm in the Viterbi search[1,7].

The paper is organized in the following way: Section 2 describes an overview of the system, in section 3 the word level is described, section 4 describes the phrase level, section 5 provides the experimental results, and finally, sections 6 contains the main conclusions.

## 2. SYSTEM OVERVIEW

Figure 1 shows a general block-diagram of the system architecture. The system consists of a two-level process around the spotting and the parser algorithm. In the training step, the lexical and syntactic knowledge are infered, and the HMM are estimated from the training data base.

### 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

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

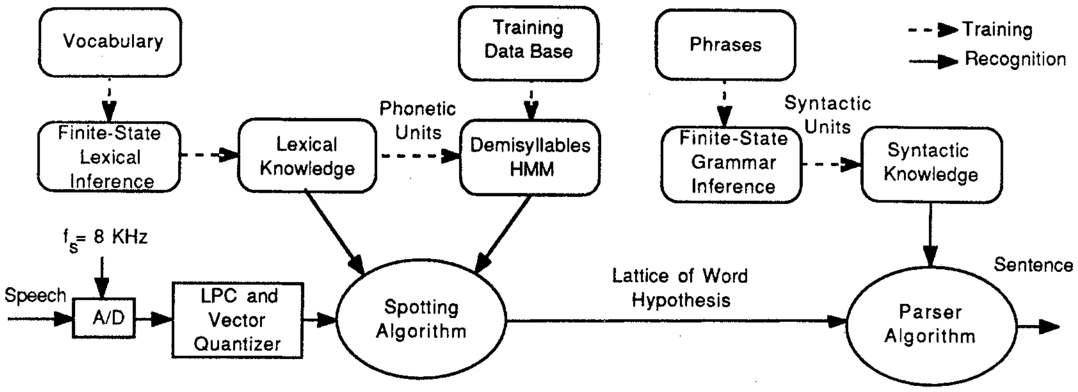

**Figure 1**. Two-level architecture

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; 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.

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

2.4. Data bases

Two data bases have been used for testing and training 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) 44 integers from zero 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 (Table 1). The articulation rate of speech spanned from 4 to 7 syllables per second.

| | |
|---|---|
| 0<br>9 | cero (0), uno (1), dos (2), tres (3), cuatro (4), cinco (5), seis (6), siete (7), ocho (8), nueve (9) |
| 10<br>99 | diez (10), once (11),doce (12), trece (13), catorce (14), quince (15), dieci (1*), veinte (2*), treinta (3*), cuarenta (4*), cincuenta (5*), sesenta (6*), setenta (7*), ochenta (8*), noventa (9*) |
| 100<br>1000 | cien (100), ciento (1**), cientos (1**), quinientos (5**), sete (7**), nove (9**), mil (1000) |

**Table 1**. Vocabulary words

2.5. Discrete Fuzzy 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.

Every frame of speech was vector-quantized with the three nearest codewords, during the training phase; so, for one frame of speech the probabilities of three codewords could be trained. The contribution of a codeword appearance to the probability estimate was weighted inversely with respect to the distance between the frame and the codeword. Thereby, the model estimation and the model smoothing were carried out simultaneously. During the recognition phase, the speech frames were vector-quantized with the nearest codeword only.

2.6. The lexical and syntactic knowledge.

The lexical knowledge inference compiles all expected phonetic realizations of the vocabulary words in a network. The syntactic knowledge inference compiles all correct sentences in a network. Thus, our approach is based on the use of finite-state networks to represent the lexical and syntactic knowledge. The lexical knowledge is described in terms of lexical units (states of the network) and the predecessor 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. The syntactic knowledge is describe in terms of syntactic units (states of the network) and the predecessors states of all of them. We infer both finite-state networks by means of an automatic inference algorithm [10]. Thus, the word level makes use of the lexical knowledge in terms of a lexical tree with the pronunciation of the words for retrieving words and a compiled version of this tree in a finite-state network suitable for driving the spotting algorithm. The phrase level makes use of the syntactic knowledge in terms of a finite-state grammar.

We classify the lexical units in start units, inside units and end units. The start units are the subset of lexical units that can be the first demisyllable of a word, the end units are the subset of lexical units that can be the last demisyllable of a word and the inside units are the rest of lexical units.

## 3. WORD LEVEL: SPOTTING ALGORITHM

The heart of the word level is the spotting algorithm. It takes as input the unknown utterance, the HMM of the demisyllables and the lexical knowledge. 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 [1].

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. 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 [7]. That modification implies to keep the N-best sequence of lexical units in each transition.

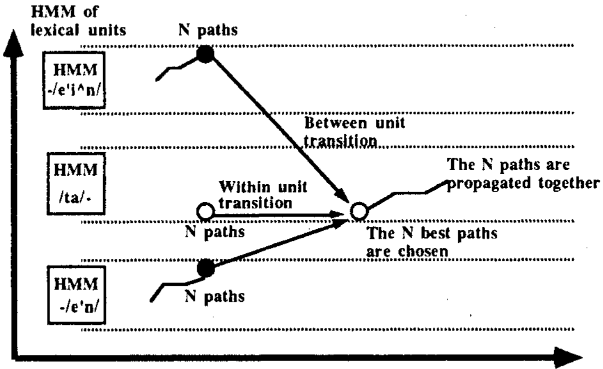

Figure 2. Multiple hypothesis algorithm.

We use a simplified implementation of the general problem of generating the N best hypothesis (figure 2). Our multiple hypothesis algorithm is based on chosen the N best paths in each between unit transition. Thus, in the first state of an unit, we take the N best paths which can make the transitions between units. If the highest probability of these N paths is greater than the highest probability of the within-unit path, we decide that a transition is made and then we propagate the probabilities of the N paths together. Thus, the best path decides when a between-unit transition takes place and the rest of the algorithm decisions are made under the best path hypothesis.

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 pruning 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 build the lattice of word hypothesis. The merging procedure compacts the output information of the word spotting algorithm selecting the most probable location of a word when it has been detected in successive starting and ending frames. Figure 3 shows an example of the spotting results that provides the merging procedure. 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.

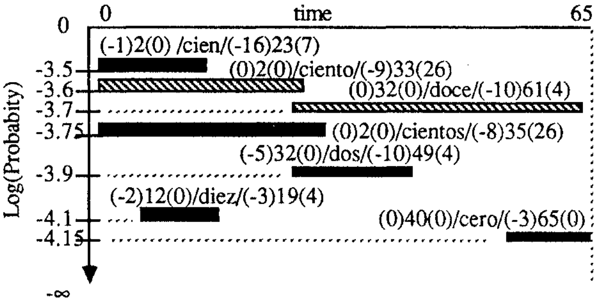

Figure 3. Spotting results analizing the number /ciento doce/ (one hundred and twelve).

## 4. PHRASE LEVEL: PARSER ALGORITHM

Taking as input the word lattice and the finite-state grammar, the parser algorithm gives the best legal sentence. To achieve this purpose, we process time-synchronous the word lattice with the Viterbi algorithm, thus, for each time instant and state of the grammar we get the best sequence of words from the beginning of the utterance.

First, the parser identifies the words of the lattice with the syntactic units, giving a search space defined by the time axe and the syntactic unit axe. Once the search space has been built, the lattice is parsed time-synchronous. The Viterbi algorithm works time-synchronous with the ending time of the spotted words and the search is driven by the predecessors of each syntactic unit. If CP(s,m) is the cumulative probability for the sequence of words ending at time **m** in the syntactic unit **s** with a set of predecessors **p**, the cumulative probability is computed as

$$CP(s,m) = \max_{l,p} [P(s,l,m) + CP(p,l)]$$

where P(s,l,m) is the log-probability that the syntactic unit **s** starts at time **l** and ends at time **m**. Once the word lattice has been processed, a backtracking process retrieves the sentence.

To deal with the concatenation of the words of the lattice, we relax the beginning and ending time of each word, penalizing the temporal difference by a factor. The beginning and ending time of the sentence is relaxed to the 10 % of the sentence length. The search space could be reduced by using a beam search heuristic and driven the Viterbi algorithm with the successors of each syntactic unit.

## 5. EXPERIMENTAL RESULTS

### 5.1. word level results.

The performance of the word level in a speaker independent approach was tested with the DB2 data base. Two experiments were carried. 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-hypothesis (N choice). Over the lattice of words, we define the top 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. The accuracy of word spotting was about 82% for the first top hypothesis level, 95% for the five top hypothesis levels without multiple hypothesis (1 choice) and 99% for the five top hypothesis levels with multiple hypothesis (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) is 2.56.

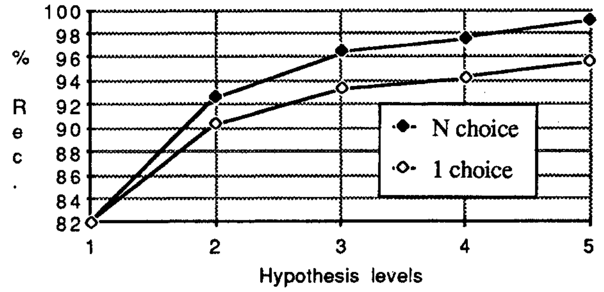

**Figure 4.** Recognition rates of words for the data base of numbers (DB2) (N=4).

### 5.2. Two-level results.

In this experiment, we test the performance of the two-level system under the following conditions. At the word level, the multiple hypothesis spotting algorithm is used. The pruning procedure selects the 5-best words at each frame and the merging procedure compact the word lattice with a maximum of 30 word hypothesis. Usually, the words in the lattice have a fixed starting point and a great variability at the ending time. Thus, at the phrase level, the starting time of each word is relaxed by a 20 % of the best location length, penalizing the temporal difference between the relaxed starting time and the best starting time by a factor of 0.1. The temporal difference between the ending time and the best ending time is penalized by a factor of 0.05 when it is greater than the 10 % of the best location length. Table 2 summarizes the results under these conditions for the ten speakers of DB2 data base (speakers S0 and S1 are in the training data base). The final word error is 7.54% and the word recognition is 93.17%. Nevertheless, in almost all cases (99.5%), the right word was in the word lattice. We want to point out that although we have to recognize only 32 words, they are highly confusables. The final sentence accuracy is 84.5%. An analysis of the errors shows that it is necessary to improve the processing at the phrase level of the concatenation of words (starting and ending time relax).

| | Word Recognition Performance | | | | | Sentence Accuaracy |
|---|---|---|---|---|---|---|
| | correct | Subs | Dels | Ins | error | |
| S0 | 98,80% | 1,19% | - | 0,80% | 1,99% | 97,08% |
| S1 | 96,46% | 3,53% | - | - | 3,54% | 91,30% |
| S10 | 93,46% | 5,60% | 0,93% | - | 6,54% | 84,09% |
| S11 | 87,85% | 10,28% | 1,87% | 1,87% | 14,02% | 70,45% |
| S12 | 97,20% | 2,80% | - | - | 2,80% | 93,18% |
| S13 | 91,16% | 8,84% | - | 1,36% | 10,20% | 75,86% |
| S14 | 94,39% | 4,67% | 0,93% | - | 5,61% | 88,64% |
| S15 | 79,44% | 19,62% | 0,93% | 1,87% | 22,43% | 56,82% |
| S16 | 92,52% | 7,48% | - | - | 7,48% | 81,82% |
| S17 | 93,46% | 6,54% | - | 0,93% | 7,48% | 81,82% |
| TOTAL | 93,17% | 6,42% | 0,39% | 0,71% | 7,54% | 84,52% |

**Table 2.** Two-level results for each speaker (S0-S17)

## 6. CONCLUSIONS

We have developed a two-level Spanish continuous speech recognition system based on the separation of the acoustic-lexical knowledge ( word level) from the syntactic knowledge ( Phrase level). The word level is based on a HMM multiple hypothesis spotting algorithm and demisyllables as phonetic units. The lexical knowledge is given by a finite-state network for driving the spotting algorithm and a lexical tree for retrieving the words. The phrase level parsers the word lattice time-synchronous by means of the Viterbi algorithm and a finite-state grammar. The results show a good performance of the word level, 99 % for the five top hypothesis levels and 93.17% of word accuracy with a finite-state grammar in the Phrase level. The sentence recognition rate is 84.5%. Further studies will include an improvement on the criterion to relax the concatenation of words.

## REFERENCES

[1] E. Lleida, J.B. Mariño, C. Nadeu, J. Salavedra, " Demisyllable-based HMM spotting for continuous speech recognition", Proc. ICASSP-91, Toronto 1991.

[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] A. Paeseler, H. Ney, "Continuous speech recognition using a stochastic language model", Proc. ICASSP-89, 1989.

[5] G. Pirani, "Advances Algorithms and Architectures for Speech Understanding", Research Reports ESPRIT, Project 26, SIP 1, Springer-Verlag, 1990.

[6] 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.

[7] 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.

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

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

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