ANCHOR POINT DETECTION FOR CONTINUOUS SPEECH RECOGNITION IN SPANISH: THE SPOTTING OF PHONETIC EVENTS

Manuel A. Leandro †‡, Jose M. Pardo †

† E.T.S. Ingenieros de Telecomunicación. Universidad Politécnica de Madrid.
‡ Universidad Alfonso X el Sabio. Villanueva de la Cañada. Madrid.

ABSTRACT

Several techniques have been used to constraint the search space in an HMM based continuous speech recognition system (grammar, beam search, etc.) in order to reduce computation without significant lose in performance [1].

The use of anchor points is a state-of-the-art technique that has already been used in some systems [2][3][4]. This approach has lead us to a bottom-up strategy in a continuous speech recognition system in which the first module performs the spotting of pre-defined phonetic events and the second module uses them as anchor points in order to guide the HMM-based recognition task.

This paper describes the definition of phonetic events for Spanish (based on expert knowledge of language) and the algorithms used for their detection and classification. Figures of performance are presented.

INTRODUCTION

Since previous word segmentation of a continuous utterance is not a simple task and coarticulation between words occurs, most current algorithms for continuous speech recognition use a top-down strategy, that is, utterance segmentation (in words, for example) and recognition are performed in a single step.

On the other hand, if given an utterance, several pre-defined phonetic events could be easily detected, a language model of the application could be built based on the expectation of these events, and this model would guide recognition restricting search. Then, an easy bottom-up strategy could be implemented, in which the first step is the detection of these events, and the second step is the recognition of the speech portions between them according to an appropriate language model.

Although the event "transition between words" is not easy to detect [5][6], there might be other phonetic events that could be used for segmentation. But in this case the utterance cannot be managed as a "concatenation of words", but a "concatenation of segments between pre-defined phonetic events". For recognition purposes, this forces us to express the language of the application as a graph in which the expectations of the pre-defined phonetic events must appear. The difficulty here is to acquire enough expert knowledge about these expectations.

The easy detection of some phonetic events in Spanish may suggest the possibility of using them as anchor points to guide recognition. A full continuous speech recognition system for connected digit recognition, that exhibits competitive performance, has been built according to this idea. In this paper we will address only the first step: the spotting of phonetic events.

EVENT DEFINITION

Three phonetic events were considered: unvoiced plosives, unvoiced fricatives and nasals.

Unvoiced plosives: There are only three phonemes of this kind in Spanish: [p], [t], [k]. They show three acoustic moments: stop, burst, and transition to the following phoneme. The first one is so deep in our language that can be easily detected using a sensitive silence detector; the second one produces, in most cases, the rising of a zero crossing rate plot during a few milliseconds.

Unvoiced fricatives: They are easy to detect because a zero crossing rate detector always gives high values for them during long segments (50-100 milliseconds, typically). Phonemes that clearly exhibit this feature are [f], [s], [T], [x].

Nasals: They can be detected recalling that energy for these phonemes is very concentrated in the band 0-500 Hz, and energy in band 1000-2000 Hz is very low. Nasal phonemes are [n], [m], [~N].

It can be seen that, to detect the events we just defined, it is enough to consider three features: energy, zero crossing rate (zcr), and "nasal energy" (the difference between energy in band 0-500 Hz and 1000-2000 Hz). Nevertheless, expert observation shows several problems during spotting. One example: sometimes it is difficult to watch a zcr rise in a plosive burst and, in these conditions, only the energy dip during the stop can help, but this dip can also be produced by a short pause in the utterance, or by the transition between some phonemes (fricatives and vowels, for example), so confusion may appear among plosives, pauses or some phoneme transitions. Another example: nasals are not easy to detect: "nasal energy" does not have a strong pattern of activation and, besides, some phonemes, like vowel [u] can activate this feature.

Affricate phonemes (that could be considered as plosive + fricative) will have also to be brought into consideration, because they will exhibit a long zcr rising after an energy dip. Affricate phonemes are [ks] and [T/].

EVENT DETECTION

Event detection is done in three steps:

1. *Continuous features extraction*: three basic features are extracted for each frame of speech: energy, zero crossing rate, and nasal energy.
2. *Binary features extraction*: the preceding features are processed to generate other five: strong silence (SS), weak silence (WS), strong frication (SF), weak frication (WF) and nasalization (N). They are called binary because for each frame, they have a binary activation (0 or 1). To achieve this, a state machine is built. SL and SF are said to be continuous activated, and WS, WF and NS, discrete activated, because the former, when they activate, they do in a set of one or more consecutive frames, while the later do it always in one frame.
3. *Event isolation*: temporal islands, in which binary features activation has place, are isolated. This time segments, along with their feature activation, are called events.

EVENT LABELING

After we have isolated the events, they must be classified. We define five classes:

PL	For plosives
FR	For fricatives
NS	For nasals
FR_PL	For fricative followed by plosive
IN	For insertions

There are several constraints that we have to consider when designing the labeling module:

1. As event detection is based on the isolation of temporal segments with binary features activation, two contiguous events will be detected as one. Then, we have to consider the cases where this may happen, and manage them correctly. There are mainly two cases: fricative followed by plosive, and affricate (that can be seen as plosive followed by fricative). For the first, we have defined the new event type (FR_PL); for the second, we have generalized the definition of the event fricative (FR), so that it includes affricate phonemes.
2. Event labeling and theoretical event expectation must be thought together. Recall that event expectation is used to build a language model that guides the recognition process. One example: if we have defined affricate phonemes to be a generalization of fricatives, then the expectation for a word in which we have an affricate (like “hacha”) is to find the event fricative (FR). Another example: if event nasal (NS) tends to appear whenever there is vowel [u], we can manage this in two ways: either we build the language model with expectation of a nasal event whenever there is [u], or we tune the labeling module in such a way that this case is labeled as insertion (IN).
3. For the following recognition process, in most cases, a labeling error would be unrecoverable. This is the reason why we try to minimize the error rate, at the cost of **ambiguous labeling**. The only problem is that, for the recognition module, the search space will be not as much constrained as with deterministic labeling, yielding to a more time consuming process. Anyway, considering maximum ambiguity (what is to say that an event could be of any class), what yields to a null labeling error rate, the search space reduction is still very important. Theoretical calculations demonstrate that, for the ambiguity factor we manage in our system, there will be a minimum reduction in computational load of about 83% (1/7 original load) in an HMM-based recognition algorithm.

In summary, on one hand, the cost of a labeling error is so big that ambiguous classification is preferred to deterministic classification if there is an error rate reduction, at the cost of some increase in computational load. Also, insertions are preferred to deletions (the recognition system will be designed to recover from the former but not from the later). On the other hand the intimate relation between the classification process and the generation of the language model with event expectations imposes a parallel design of both, together with precise expert knowledge of the language.

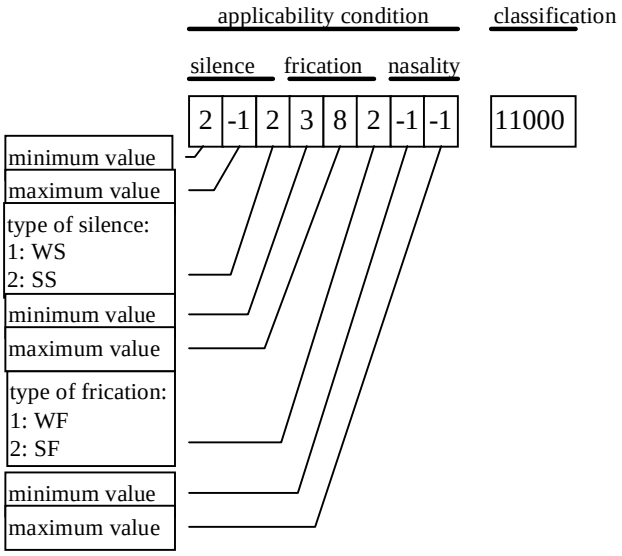
Two classifications methods were tested: heuristic rules and neural net.

Heuristic labeling

A number of heuristic rules performs the classification of the events. Each rule has two parts: condition and classification. Condition is defined with a vector of 8 components: minimum and maximum duration of silence, type of silence (SS or WS), minimum and maximum duration of frication, type of frication (WF or SF), and minimum and maximum duration of nasality. Classification is defined with a vector of 5 binary components (0 or 1), one per class. In this vector, there might be more than one component with value 1 and, in this case, it is an ambiguous classification.

When classifying an event, the rules are processed in a sequential manner, and the first rule that has a condition satisfied by the event is applied.

Next figure shows an example of rule. Value -1 stands for no restriction. Then, this rule applies when the event exhibits an activation of feature SS with duration bigger or equal than 2, an activation of feature SF with duration between 3 and 8, both included, and no restriction for feature N. If condition fires, classification vector applies. The vector in the example indicates activation of PL and FR events, that is, event could be either a plosive or a fricative (example of ambiguous classification).



Neural net labeling

We said that an event is a temporal segment together with its set of feature activation. Before classification, we need to code the event as a vector that can be fed to the neural net. It would be too long to describe the method here. We will only say that it is done in two steps: in the first one, we try to consider only relevant information and we code it in a vector of 24 integer numbers. In the second step we perform a non linear normalization of the component values (considering a statistical study obtained from an histogram), in such a way that each component of the new vector is a real number ranging from -1 to 1. We also redistribute the number of components. The new vector has 23 components.

The neural net used is a multilayer perceptron with the following characteristics:

Input: 23 inputs with continuous activation in [-1, 1].
Output: 5 outputs with continuous activation in [0, 1].
Number of layers: 2
Neurons in first layer: 30
Neurons in second layer: 5
Connections: full connection, forward.
Number of weights: 840

During the recognition process, a non linear function is applied to the activation value of the outputs (ranging in the interval [0, 1]), so as to obtain a vector of five binary components. In this function, a parameter called F controls the "ambiguity" of the classification, in the sense that, for low values of F, the ambiguity is small (only one output tends to be with value 1 and the rest will be 0), and as F increases, the average number of outputs activated in a classification will increase.

PERFORMANCE

Detection performance

A good behavior implies a detection of all expected events in the frame where they are expected. A quantitative measure for this would imply to have a reference for comparison that must be done by an expert. This have not been done, but a very complete expert observation of results yields to the following conclusions:

1. Though there are insertions, all events are detected (no deletions). Insertions are not important as we said that the following recognition process can recover from them.
2. We do not find errors in temporal precision, except for the case of nasal detection: sometimes the point where start of nasal is detected is actually shifted towards the stable part of the phoneme.

Classification performance

We have applied the system to a connected digit recognition task. Database was recorded by 107 speakers, and the average number of digits per utterance was 6,8. Two subsets were extracted from this database: the training set, consisting of 260 utterances, and the recognition set, consisting of 104 utterances. In the heuristic rules case, training set was used to acquire enough expert knowledge to write the rules. In the neural net case, events in the training set were previously detected by the detection module, and then hand labeled so as to train the net in a supervised way.

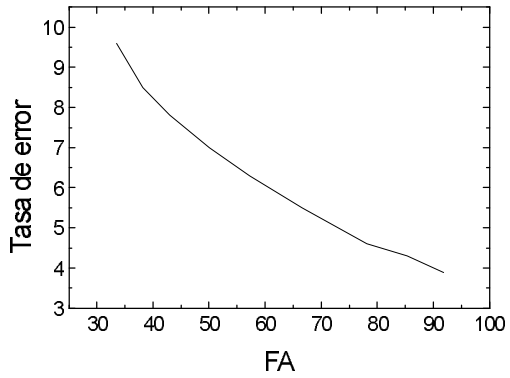
Classification performance is given by two figures:

1. *Relevant error rate (RER)* is the percentage of events that do not exhibit activation of the correct class in their classification vector, and this error can produce a non recoverable error in the following recognition process.
2. *Ambiguity factor (AF)* is calculated with the following formula:

$$AF = \frac{NA}{NE} - 1$$

where NA is the number of activations at the output of the classifier, in the whole experiment, and NE is the number of events classified. It can be seen that this factor is actually the average number of extra activated outputs per event, so that, if there is no ambiguity (one sole output activated per event) AF= 0, and if ambiguity is maximum, AF= 4 (if there are 5 outputs, 4 extra outputs can be activated besides the correct one).

In the case of neural net labeling, the function applied to the output of the net can be adjusted (through constant F) to set the ambiguity factor. As we said, ambiguity and error rate are tied, so constant F also sets the error rate. Then, we can present a plot of error rate vs. ambiguity factor (see figure).



Error rate versus ambiguity factor in the neural net system for event classification

In the case of heuristic rules classification, setting error rate and ambiguity factor at the desired value is not so easy, because we must change the rules (writing more deterministic or more ambiguous rules). Once we have decided the rules, ambiguity factor and error rate is fixed and is:

Relevant error rate	*3,0%*
Ambiguity factor	*1,12%*

CONCLUSIONS

Our research has rescued an old approach for continuous speech recognition: the segmentation + recognition strategy. Discarded because of the difficulty to detect transition between words, we have shifted to a new vision in which other kind of events are detected, producing a segmentation not in words, but that can be successfully managed if we build a language graph according to the expectations of the defined events. The theoretical benefits of this approach has to do with important computational load saving.

We have focused the paper in the first step: the definition, detection and classification of events.

Expert phonetic knowledge of Spanish has been used to define the events: they must be easy to detect and their “density” in any utterance must be enough for a posterior guidance of the recognition process.

For detection, an algorithm based on features, with a bottom-up strategy, was developed. The algorithm is strongly based on the expert knowledge we have about the acoustic characteristics of some Spanish phonemes. Qualitative performance observed is very good.

For classification, we have seen that the recognition process imposes severe restrictions on the way it must be done: due to the elevate cost of errors and deletions, a system must be designed with low error rate, at the cost of ambiguous classification, if necessary; and with no deletions, at the cost of high insertion rate, if necessary. Two approaches (heuristic rules and neural net) were tested. Both exhibit a performance good enough to address the recognition process and, in the neural net case, it shows obvious the tying between ambiguity and error rate, so that we can reduce the later if we increase the former. This will affect the recognition process in a moderate increase in computational load.

REFERENCES

[1] Hermann Ney, “Modeling and search in continuous speech recognition”, in Proceedings Eurospeech 93, pp. 491-494. 1993.

[2] David Lubensky, "Continuous digit recognition using coarse phonetic segmentation", in Proceedings ICASSP 87, pp. 817-820, 1987.

[3] Hy Murveit, John Butzberger, Vassilios Digalakis, Mitchel Weintraub, "Large-vocabulary dictation using DECIPHER speech recognition system: progressive search techniques", in Proceedings ICASSP 93, pp. II-319 II-322, 1993.

[4] Martine Adda-Decker, "Continuous speech recognition using phone-based anchor point detection and diphone-based dp-matching", in Proceeding Eurospeech 89, vol. I, pp. 94-97, 1989.

[5] J.M. McQueen and E.J. Briscoe, “A computational tool for examining lexical segmentation in continuous speech”, in Proceedings Eurospeech 91, pp. 697-700, 1991.

[6] J. Harrington, I. Johnson and M. Cooper, “The application of phoneme sequence constraints to word boundary identification in automatic continuous speech recognition”. J. Laver and M. Jack (Eds.) European Conference on Speech Technology, vol. 1, pp. 163-166, 1987.