Improved Performance Evaluation of Speech Event Detectors

Carla Lopes[1,2], *Fernando Perdigão*[1,3]

[1]Instituto de Telecomunicações, Coimbra, Portugal
[2]Instituto Politécnico de Leiria-ESTG, [3]Universidade de Coimbra-DEEC
{calopes, fp}@co.it.pt

Abstract

The goal of event-based (EB) systems is the detection of the occurrence of important elements in the speech signal for different sound classes. In a speech recognition system, events can be combined to detect phones, words or sentences, or to identify landmarks to which a classifier or a decoder could be synchronized. The time boundaries of the events are then as important as the events themselves. Accordingly, the assessment of EB systems must take into account not only the correct identified sequence of events but also their correct time localization. Usually, only the token sequence or its boundaries are taken for evaluation. In this paper we propose an extension to standard recognition evaluation procedure, which combines recognition and segmentation performance. In our proposal, a modified *Levensthein* algorithm is used in the alignment between labeled and recognized events, where the degree of overlapping between them is taken in the local distance definition. We evaluate our approach on a rule based event detector, using the TIMIT corpus and compare the results of the new evaluation procedure with standard metrics. The results show that *accuracy* drops if alignment is made as a function of the overlapping between labels; nevertheless the *agreement* with the labeled boundaries is significantly improved.

Index Terms: diarization, event detection, speech segmentation

1. Introduction

Despite the continuous nature of speech, standard automatic speech recognition systems describe it as a sequence of discrete units, usually phonemes. Since speech is a result of changes on both the excitation source and the vocal tract system, it may be described as a sequence of events. These events may be related with the signal acoustics, the signal production, the language, the speaker, etc, because any significant change may, itself, be treated as an event. In the literature, event-based (EB) systems are described in several contexts, namely: on the classification of the signal into broad classes according to the presence of some specific features on the acoustic structure of the signal, [1-3]; on the detection of landmarks where some specific changes as syllabic dips, glottal closures or vowel onset points occur [4, 5]; in finding structural events like sentence boundaries, filled pauses, discourse markers, and edit disfluencies, [6]; on the detection of word boundaries and voice activity, [7]; applied to speaker recognition [8]; attempting to find gestural events, [9] and representing auditory events, [10]. Notwithstanding this fuzzy concept of speech events, all event-based systems have the same goal: to detect the occurrence of important elements (events) as well as the instant when they occur, which means that both recognition and segmentation are necessary.

Automatic speech recognition (ASR) and automatic speech segmentation (ASS) are quite different in their main purpose. ASR systems should provide the best sequence of labels that correspond to the input signal but, when the system is evaluated, no attention is given to the boundaries of the labels. On the contrary, ASS systems aim at decomposing a signal into acoustically different adjacent segments, but there are no concerns with the labels of the segments. In this case the goal is to find the maximum number of boundaries that match those from a manual or a reference annotation. Since the main goal of the two referred fields is quite different, the evaluation measures employed also differ.

As mentioned above, some previous works have explicitly addressed the problem of event detection; nevertheless there is no well-established measure for evaluating such systems when both labels and boundaries are important to measure the performance of the system. They have been evaluated using speech recognition or speech segmentation evaluation procedures.

In this paper we propose an extension to the traditional speech recognition evaluation procedure, making the alignment between reference and recognized events as a function of the degree of overlapping between labeled and recognized events.

This paper is organized as follows. In Section 2, the standard evaluation metrics used in ASR and ASS are described. Section 3 outlines the proposed evaluation measure, while in Section 4 some experimental results are presented. In Section 5 we discuss the performance of our proposal.

2. Standard Evaluation Metrics

Typically, event detection systems are evaluated using speech recognition, [1-3, 5, 8] and speech segmentation, [6, 11] metrics. Speech segmentation systems are, in general, evaluated by comparing automatic with manual alignment, [11-13] and the question is: *what percentage of the boundaries found by the segmentation system, is correct*? The response to this question depends on the system design. Amongst segmentation methods, it is usual to take as input both the signal and its phonetic transcription, and a force alignment technique is employed, [6, 11]. In this case the task is not exactly segmentation, but an alignment between the phoneme sequence and the acoustic signal. The performance evaluation measure used is usually called *agreement*. *Agreement* is used instead of *accuracy*, because manual alignments are prone to subjectivity, [11]. Let N_C be the number of boundaries in agreement with the ones of the manual alignment and N_T the total number of boundaries, then,

$$Agreement = N_C \,/\, N_T \ . \qquad (1)$$

This measure is particularly appropriated when forced alignment is employed, because in this situation the number of limits found by the automatic aligner agrees with the number of limits of the reference. Otherwise the number of limits found by the method is not likely to agree with the number of limits of the reference utterance and insertion/deletion errors come out. That is the case of systems where the input is only the acoustic signal, [12]. In this case the *agreement* measure is not suitable to evaluate the quality of the segmentation and so other performance measures are commonly used, [5, 12]; these include the portion of segment boundaries placed correctly, $Precision{=}N_C/\left(N_C{+}I\right)$, and the ratio of correctly placed boundaries to all manually placed segment boundaries, $Recall{=}N_C/\left(N_C{+}D\right)$, where D and I refers to the number of deletion and insertion errors, respectively and N_C the number of boundaries in agreement with those from manual alignment. Another usual measure takes the (weighted) harmonic average of precision and recall and leads to the F score measure, [14].

In ASR systems, the most common measure is word error rate (WER), or the related performance metric word *accuracy* rate. This last one is defined by the following expression:

$$Accuracy = \left(N_T - S - D - I\right) / N_T , \qquad (1)$$

where N_T is the total number of labels in the reference utterance and S, D and I are the substitution, deletion and insertion errors, respectively. Another measure is *correctness*, which is similar to *accuracy*, but where insertion errors have no influence. This measure is defined by

$$Correct = \left(N_T - S - D\right) / N_T \ . \qquad (3)$$

The number of insertion, deletion and substitution errors is computed using the best alignment between two token sequences: the manually aligned (reference) and the recognized (test). An alignment resulting from search strategies based on dynamic programming are currently being used in a successful way for a large number of speech recognition tasks, [15]. Speech recognition toolkits, such as HTK, [16], include tools to calculate *accuracy* and related measures on the basis of the transcribed data and recognition outputs using this dynamic programming algorithm.

Measures such as *accuracy* and *correctness* are useful when the task is specifically speech recognition, but it is a poor measure from the point of view of applications, where higher-level information is needed. To demonstrate the unsuitability of the *accuracy* measure for EB systems, an example is shown in Figure 1. The top signal corresponds to a TIMIT [17] utterance with the corresponding manual alignment and the other one is an example of the output of an ASR system.

Because the label sequence in *a*) and *b*) is the same, the alignment between them is perfect, what will correspond to an optimal recognition. However, for an event detection this result is, obviously, not correct; some segments (labels) do not even overlap! So, if the alignment is not made as a function of the overlapping, *accuracy* and *correctness* rates will not tell much about the performance of the system. The next section focuses on this problem and on the suggestion of an improved evaluation procedure for EB systems.

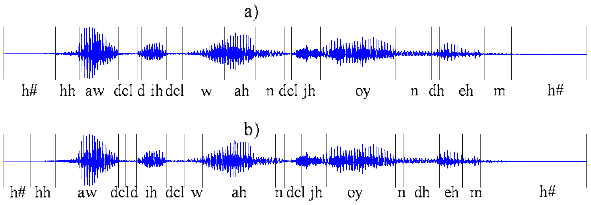

Figure 1. *a) TIMIT utterance* (“*how did one join them*”) *manually aligned, b) Output example of the ASR system.*

3. Proposed Evaluation Measure

No well-established measure has yet been proposed to evaluate speech event detectors. A possible solution consists in defining an alignment function between two label sequences (reference and test) in which a penalty is applied to a pair of labels, as a function of the boundary misalignment.

The usual procedure aligns the label sequences according to the *Levensthein* algorithm, [18]. This algorithm finds the best alignment between two strings inserting a penalty if an error occurs (insertion, deletion and substitution), but no penalty is applied if the labels match. In our proposal we include an additional penalty that is proportional to the average of the left and right misalignments. If the labels do not overlap ($T_{OV} \leq 0$ in Figure 2), this penalty is set to a maximum value (p_{max}), such that an insertion or a deletion will be preferred to a misaligned substitution.

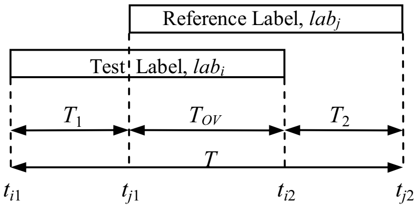

Figure 2: *Measurement of the time misalignment between two labels.*

Considering t_{i1} , t_{i2} and t_{j1} , t_{j2} as the boundaries of the test and reference labels, respectively, as indicated in Figure 2, then

$$T = \max(t_{i2}, t_{j2}) - \min(t_{i1}, t_{j1}) = T_1 + T_2 + T_{OV} \ ,$$

the overlapping time is

$$T_{OV} = \min(t_{i2}, t_{j2}) - \max(t_{i1}, t_{j1})$$

and the left and right misalignments are $T_1 = \left|t_{j1} - t_{i1}\right|$ and $T_2 = \left|t_{j2} - t_{i2}\right|$. If the labels lab_i and lab_j match but are not perfectly aligned, then we introduce an additional association penalty, $p_A(i,j)$, inversely proportional to the overlap between the labels, according to the following expression:

$$p_A(i,j) = \frac{(T_1+T_2)/2}{T_{OV}} = \frac{1}{2}\left(\frac{T}{T_{OV}} - 1\right). \qquad (3)$$

If the labels overlap more that 50%, p_A is smaller than 0.5. As far as the overlapping decreases, this distance increases and is clipped to p_{max}=15, which corresponds to 3.2% of overlapping. According to the *Levensthein* algorithm there are four types of alignments each of them with different penalties (hit, substitution, insertion and deletion). Table 1 displays these penalties as used in the HTK evaluation tool (`HResults`) and in our proposal.

Table 1. *Types of alignment and corresponding penalties.*

Type of Alignment	Penalties in HTK	Proposed Penalties
Hit	$p_{HIT} = 0$	$p = p_A$
Substitution	$p_{SUB} = 10$	$p = p_A + p_{SUB}$; $p_{SUB} = 7$
Insertion	$p_{INS} = 7$	$p_{INS} = 4$
Deletion	$p_{DEL} = 7$	$p_{DEL} = 4$

The dynamic programming algorithm is then defined according to the equation

$$D(i,j) = \min\begin{cases} D(i-1,j) + p_{INS} \\ D(i,j-1) + p_{DEL} \\ D(i-1,j-1) + p(i,j) \end{cases} \qquad (4)$$

where

$$p(i,j) = p_A(i,j) + \begin{cases} p_{SUB} & , lab_i \neq lab_j \\ 0 & , lab_i = lab_j \end{cases} \qquad (5)$$

and where $D(i, j)$ is the accumulated distance to a node (i, j) of the alignment space. The optimal alignment is found by tracing back the path from $D(m,n)$ to the origin, where m and n denote, respectively, the lengths of the test and reference strings.

Notice that in this case substitutions are more penalized than in the normal case. If the two labels are not considerably overlapped, then it is worth to consider a deletion or insertion rather than a substitution.

To show the performance of the proposed method, we present in Figure 3 an example corresponding to the output of a fricative detector. To measure the detection results, we need to align the sentences depicted in figure 4.

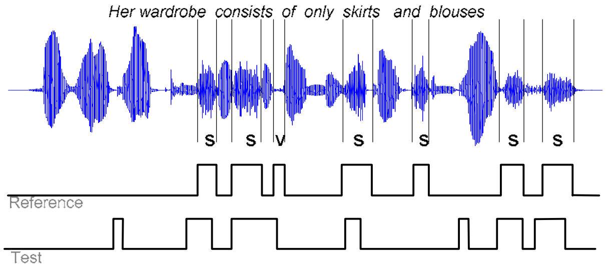

Figure 3. Example of a fricative detector.

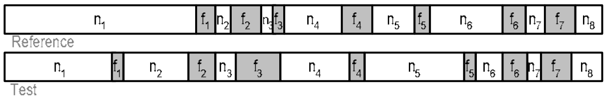

Figure 4. The reference and test sentences to be aligned.

If we compute *accuracy* using the HTK tool taking this example, a 100% rate will be obtained, because the recognized sentence was perfectly mapped with the reference sentence. This is obviously a wrong performance evaluation. There are some alignments that can not be possible: the test event f_1 may be aligned with the reference n_1 or may result on an insertion, but should never be aligned with the reference f_1; the test event f_3 may be recognized as the reference f_2, n_3 or f_3, but not with events with which do not overlap. In this example the test events f_1, n_2, f_5 should be considered insertions and the reference events n_3, f_3 and f_5 should be considered missing.

4. Experimental Results

To compare the proposed evaluation procedure with the traditional one, event detection was carried out using the TIMIT database, [17]. The complete test set was used (excluding the SA utterances) in a total of 1344 utterances. The event to detect is the presence of a set of fricative phonemes. The 61 phone classes in TIMIT were reduced to a set of 2 labels (‘*fri*’, ‘*nfri*’) and adjacent events of the same class were merged. The {’f’, ’th’, ‘z’, ’s’, ’zh’, ’sh’, ’jh’, ’ch’} phonemes were labeled as ‘*fri*’, and all the others as ‘*nfri*’.

Event detection has been carried out by means of four acoustic-level features at a frame rate of 5 ms using a Hamming window of 15 ms. The features are: 1 - Energy; 2 - Spectral Flatness Measure; 3 – Spectral Centroid and 4 – the difference between log-energy at high and low frequencies. We also used major variations of energy and spectral flatness in the rules that classify each set of frames as ‘*fri*’ or ‘*nfri*’. This rule-based event detector was used mainly to test the proposed performance evaluation algorithm; a more robust detector is under investigation.

To evaluate the quality of the segmentation, we computed *correctness* and *accuracy* according to equations (2) and (3)and also the *agreement* (of the hits) with manual boundaries within 10, 20 and 30 ms, according to equation (1). Table 2 lists the performance rates obtained with the HTK tool and with the proposed algorithm. Comparing the results given by HTK with those from the new proposed method we conclude that *accuracy* drops if alignment is made as a function of the overlapping between labels; nevertheless the improvement on the quality of the boundaries of the events is quite noticeable in this case. We achieved an improvement of about 26%in *agreement* within a window of ±20 ms, sacrificing *accuracy* which drops 4.3%. With our proposal there are much more insertions (23.8%) and deletions (38.2%) and there are also substitutions errors, unlike HTK tool where this last kind of error never occurs. Despite the drop in *accuracy* we consider that for the assessment of a speech event detector the correct alignment of the events is of most importance.

Table 2. *Results for the proposed evaluation method and of HTK method, in terms of correctness, accuracy and agreement rates.*

(%)	Corr.	Acc.	Agree. (10ms)	Agree. (20ms)	Agree. (30ms)
HTK	94.44	**85.52**	50.27	**59.17**	61.85
Proposed method	92.26	**81.21**	72.54	**85.2**	88.98

In order to evaluate the significance of the proposed method we computed the left (T_1) and right (T_2) misalignments of the hits as defined in Figure 2. Figure 5 shows a histogram of the distances between the boundaries of test and reference labels aligned with the two methods under consideration. It is clear that with no time information the resulting distances are high, because the alignment method can associate tokens that may be in total temporal disagreement. In this case 36% of the boundary distances are farther than 50 ms from the reference ones. It is interesting to note that with our alignment method more than 50% of the distances are less than 5 ms and 85% less than 20ms from manual alignment. These results indicate that the proposed alignment method is appropriate to evaluate both the *accuracy* and the *agreement* of an event based system.

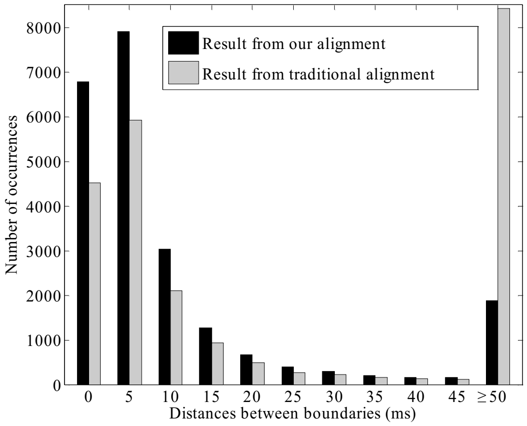

Figure 5. *Histogram of the distance between the boundaries of the well recognized and reference events.*

5. Conclusions

Typically, event detection systems are not evaluated using segmentation information, which may induce to imprecise performance measures. This problem motivated us to investigate a method to evaluate event detection algorithms. It involves the enhancement of the alignment process, which is driven by the recognized positions of the boundaries. The proposed method overcomes the problem of aligning labels of the reference and test utterances with boundaries misaligned, setting a penalty inversely proportional to the degree of overlapping between the pair of labels.

The results with a fricative event detector show that a drop of 4.3% in *accuracy* conduct to an improvement of 26% in the alignment of well recognized events (within a window of ±20 ms). 85% of *agreement* was achieved with our proposal against 59% of common evaluation procedures. This indicates that the described evaluation methodology perfectly fit our goal: to measure the performance of an event based system in terms of recognized events as well as their corresponding boundaries.

6. References

[1] A. Juneja and C. Espy-Wilson, "Speech segmentation using probabilistic phonetic feature hierarchy and support vector machines," International Joint Conference on Neural Networks, Portland, 2003.

[2] A. Juneja, "Speech Recognition Based On Phonetic Features And Acoustic Landmarks," Ph.D. Thesis: University of Maryland, 2004.

[3] J. Li and C. H. Lee, "On Designing and Evaluating Speech Event Detectors," Interspeech2005, Lisbon, 2005.

[4] S. Prasanna, "Event based analysis of speech," in Dept. of Computer Science and Engineering, Ph.D. Thesis: Indian Institute of Technology Madras, India, 2004.

[5] M. Hasegawa-Johnson, *et all*, "Landmark-Based Speech Recognition: Report of the 2004 Johns Hopkins Summer Workshop," 2005.

[6] Y. Liu, "Structural Event Detection for Rich Transcription of Speech," Ph.D. Thesis: Purdue University, 2004.

[7] G. Evangelopoulos and P. Maragos, "Speech Event Detection using Multiband Modulation Energy," Interspeech2005, Lisbon, 2005.

[8] N. Scheffer and J.-F. Bonastre, "Speaker Detection using Acoustic Event Sequences," Interspeech2005, Lisbon, 2005.

[9] A. Gutkin and S. King, "Detection of Symbolic Gestural Events in Articulatory Data for use in Strutural Representations of Continuous Speech," ICASSP'05, 2005.

[10] G. Hu and D.L.Wang, "Auditory segmentation based on event detection," ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, 2004.

[11] J.-P. Hosom, "Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information," Ph.D. Thesis: Oregon Graduate Inst. of Science and Technology, 2000.

[12] P. Korhonen and U. Laine, "Unsupervised Segmentation of Continuous Speech Using Vector Autoregressive Time-Frequency Modeling Errors," Interspeech2005, Lisbon, 2005.

[13] J. Keshet and e. all, "Phoneme Alignment Based on Discriminative Learning," Interspeech2005, Lisbon, 2005.

[14] C. J. v. Rijsbergen, Information Retrieval: Department of Computing Science - University of Glasgow, 1979.

[15] H. Ney and S. Ortmanns, "Progress in Dynamic Programming Search for LVCSR," IEEE, vol. 88, pp. 1224-1240, 2000.

[16] S. J. Young, *et all*, "The HTK Book," Cambridge University Engineering Department, Cambridge, UK 2005.

[17] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT Acoustic-Phonetic Continuous Speech Corpus," NIST, 1990.

[18] V. I. Levensthein, "Binary codes capable of correcting deletions, insertions and reversals," Sov. Phys.-Dokl, vol. 10, pp. 707-710, 1966.