Cooperation between global and local methods for the automatic segmentation of speech synthesis corpora

Safaa Jarifi[1], Dominique Pastor[1], Olivier Rosec[2]

[1] École Nat. Sup. des Télécommunications de Bretagne,
Département Signal et Communication,
Technopôle Brest-Iroise, CS 83818, 29285 Brest Cedex, France.
{safaa.jarifi,dominique.pastor}@enst-bretagne.fr

[2] France Télécom, Division R&D TECH/SSTP/VMI,
2, avenue Pierre Marzin, 22307 Lannion Cedex, France
olivier.rosec@francetelecom.com

Abstract

This paper introduces a new approach for the automatic segmentation of corpora dedicated to speech synthesis. The main idea behind this approach is to merge the outputs of three segmentation algorithms. The first one is the standard HMM-based (Hidden Markov Model) approach. The second algorithm uses a phone boundary model, namely a GMM (Gaussian Mixture Model). The third method is based on Brandt's GLR (Generalized Likelihood Ratio) and aims to detect signal discontinuities in the vicinity of the HMM boundaries. Different combination strategies are considered for each phonetic class. The experiments presented in this paper show that the proposed approach yields better accuracy than existing methods.

Index Terms : automatic segmentation, hard combination, soft combination, refinement by boundary model, Brandt's GLR method, speech synthesis.

1. Introduction

This paper deals with the problem of automatic segmentation of speech corpora for concatenative TTS synthesis systems. In the development process of such systems, the segmentation of large databases constitutes a key task. Obviously, the optimal segmentation to use in these systems is the manual one. Nevertheless, an accurate automatic segmentation saves a lot of human effort and time in creating new synthesized voices and thus drastically simplifies the personalization of a TTS speech synthesis.

Up to now, the HMM approach [1, 2] is the most widely used for automatic segmentation and it is considered as the most reliable. This approach is linguistically constrained because it needs the true phonetic sequence associated to the recorded utterances in order to estimate the HMM sequence. Then it applies a forced alignment between this HMM sequence and the speech signal. However, this approach has still some limitations for building voices for TTS systems based on the principles of unit-selection and concatenative synthesis. The main limitation is that HMMs model well steady areas but are not really suited to detect locally the transitions between phonemes in a speech signal. For this reason, in order to guarantee a good quality of synthesized voices, a manual checking is applied to the HMM segmentation before synthesis.

Brandt's GLR algorithm [7] is another suitable approach for segmenting speech signals. Nevertheless, it produces insertions and omissions because it is linguistically unconstrained.

With respect to the foregoing, the purpose of this paper is to combine global and local automatic segmentation algorithms. To achieve this, we choose in this paper three automatic segmentation algorithms. The first is the HMM segmentation. The second uses a boundary model which is estimated on a small database and which is used to refine the HMM segmentation marks. The third one is Brandt's GLR method which was modified in order to avoid omissions and insertions. These algorithms are described in section 2. In section 3, two combination methods are proposed and evaluated on French and English corpora dedicated to speech synthesis.

2. Segmentation algorithms

2.1. Segmentation by HMM

This approach generally consists of two steps. The first step is the training phase that aims at estimating the acoustic models. In the second step, these models are used to segment the speech signal by the Viterbi algorithm. This one applies a forced alignment between the models associated to the known phonetic sequence and the speech signal.

Note that the training is a decisive step because the accuracy of the obtained segmentation heavily depends on the quality of the estimated models. One solution to perform well the training step is to resort to an iterative training [3]. The phone labels resulting from the previous iteration are used for initializating and re-estimating the HMMs via the Baum-Welch algorithm. After a few iterations, mismatches between segmentation marks produced by an HMM approach and marks obtained manually are considerably reduced as shown in [4]. Another method to train the models is to use a representative small speech database manually labeled and segmented [5]. We estimate first the models using this small database. Then we segment the whole corpus with the models. As the estimation of the models on the small corpus is accurate, the processing offers better results than the iterative training. For that reason this strategy is used in this paper.

This study is supported by France Telecom

2.2. Refinement by boundary model

The main idea of this method is to train models for boundaries on the basis of a small database segmented and labeled manually. Then, these models are used to refine an initial segmentation [8].

For each boundary of the training database, we create a super vector as mentioned in figure 1 by concatenating the acoustic vectors of size N_c associated to the $(2N+1)$ frames around the boundary. Because the number of labeled data is limited in practice, the boundaries are clustered into classes using a classification and regression tree (CART). Then a Gaussian model is estimated for each class.

The second step aims at refining each boundary of every segment given a labeled sentence and its initial segmentation. For that purpose, we seek, in a certain vicinity of each boundary, the time instant that maximizes the likelihood of its super vector in comparison with the Gaussain model of this transition.

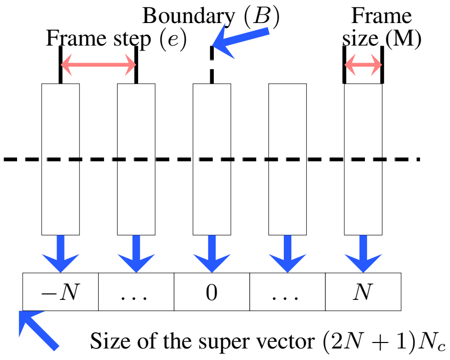

FIG. 1 – Elements of a super vector

2.3. Brandt's GLR algorithm

The aim of this method is to detect discontinuities in speech signals. Speech signals are assumed to be sequences of homogeneous units. Each unit or window w is a finite sequence $w = (y_n)$ of samples that are assumed to obey an AR model : $y_n = \sum_{i=1}^{p} a_i y_{n-i} + e_n$. In this equation, p is the model order, which is assumed to be constant for all units and e_n is a zero mean white Gaussian noise with variance equal to σ^2. Such a unit is thus characterized by the parameter vector $\Theta = (a_1, \ldots, a_p, \sigma)$. Let w_0 be some window of n samples and Θ_0 the corresponding parameter vector. The authors of [6, 7] attempt to decide whether w_0 should be split into two subsegments w_1 and w_2 or not. In fact, a possible splitting derives from the detection of some jump between the parameter vectors Θ_1 and Θ_2 of w_1 and w_2 respectively. Brandt's GLR method decides that such a jump has occurred by comparing : $D_n(r) = nlog\hat{\sigma}_0 - rlog\hat{\sigma}_1 - (n-r)log\hat{\sigma}_2$ to a predefined threshold λ. Note that D_n is merely the GLR. In the equation above, r is the size of the time interval covered by w_1, whereas $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the noise standard deviation estimates of the models characterized respectively by the parameter vectors Θ_1 and Θ_2. Thus, the change instant corresponds to $arg(max_r(D_n(r)) \geq \lambda)$.

As mentioned before, the basic Brandt's GLR method is an algorithm capable of detecting discontinuities of speech signals without any further knowledge upon the phonetic sequence. As this algorithm is linguistically unconstrained, it makes insertions and omissions. However, for TTS synthesis, the phonetic sequence of every utterance is known. For this reason, we decide to take into account this information by using the boundaries produced by a segmentation algorithm that uses the phonetic sequences as the HMM segmentation. More precisely, let $(U_0, U_1, \ldots, U_L)$ be the boundaries obtained by such initial algorithm. For i in $\{1, \ldots, L-1\}$, we seek a speech discontinuity between $V_i = \frac{(U_{i-1}+U_i)}{2}$ and $V_{i+1} = \frac{(U_i+U_{i+1})}{2}$ by applying a modified Brandt's GLR method : to avoid omissions and insertions, the use of the threshold is replaced here by the maximization of the GLR on $[V_i, V_{i+1}]$.

3. Evaluation of the three algorithms

In this section, we present the experimental results obtained by the three segmentation algorithms described above on French and English corpora. Both acoustic databases were recorded by a professional native female speaker and sampled at 16 kHz. The French corpus "*corpusFR*" and the English corpus "*corpusEN*" respectively contain 7300 and 8900 sentences.

The segmentation by HMM uses the HTK toolkit [9] for the acoustic analysis, the training and the segmentation steps. It considers mixtures of 2 Gaussian density and the acoustic vector contains 39 coefficients which are the 12 MFCCs (Mel Frequency Cepstral Coefficients), the normalized energy, and their first and second derivatives. The HMM segmentation obtained by the use of a small training database is called hereafter *HMMSeg*. Twenty iterations of the Baum-Welch algorithm are applied to train the HMM. The refinement by boundary model is applied to the HMM segmentation and the parameters N, M and e of figure 1 are fixed to 2, 20 ms and 30 ms respectively. These parameters were adjusted on the French corpus in [10]. The segmentation thus obtained is called *RefineSeg*. The segmentation obtained with Brandt's GLR method initialized by the HMM segmentation is denoted *BrSeg*. The model order is set to 12 and the minimal length of w_1 and w_2 are equal to 10 ms.

The training phases of *HMMSeg* and *RefineSeg* were carried out with the number *SizeAlg* of training sets equal to 100, 300 and 700. For each set of learning sentences randomly chosen, the test set was built by considering the remaining sentences in the corpus under study. Moreover, to illustrate the consistency of the results a cross-validation procedure was used where 3 trials were done for each value of *SizeAlg*. All the accuracies are calculated at a tolerance equal to 20 ms. This value is commonly considered as an acceptable limit to guarantee a good quality of a synthesized voice. The results presented here are obtained by averaging the accuracies using this cross-validation procedure.

Table 1 presents the accuracies of each algorithm with respect to *SizeAlg* for each corpus. Table 2 shows the limit of performance of each algorithm. This limit corresponds to using the whole database for the training of *HMMSeg* and *RefineSeg*. According to these results, we can make the following remarks :

- the refinement by boundary model gives the best results when the corpus size equal 300 or 700 for "corpusFR" and 700 for "corpusEN". This is normal because the boundary models are well trained ;
- the modified Brandt's GLR method is inaccurate at 20 ms in comparison with the other algorithms ;

Because the important measure in TTS systems is the accuracy at 20 ms, it seems reasonable to say from table 1 that the refinement by boundary model is the most accurate algorithm. Nevertheless, we should not forget that the algorithms are suited to different phonetic classes. In fact, during these tests, it turned out that Brandt's GLR method detects well some boundaries like silence/speech and voiced/unvoiced transitions. Thus, depending on the classes to the right and the left of a transition to detect, it seems relevant to take into account that the algorithms do not perform equally. This is the purpose of combination methods.

TAB. 1 – Accuracies for each algorithm

	SizeAlg	*HMMSeg*	*RefineSeg*	*BrSeg*
corpusFR	100	91.71%	91.08%	83.22%
corpusEN		91.98%	89.58%	86.78%
corpusFR	300	92.51%	93.26%	83.39%
corpusEN		92.95%	92.46%	87.10%
corpusFR	700	92.47%	94.00%	83.38%
corpusEN		93.00%	93.50%	87.09%

TAB. 2 – The limit of performance for each algorithm

	HMMSeg	*RefineSeg*	*BrSeg*
corpusFR	92.68%	95.00%	83.22%
corpusEN	93.17%	94.30%	87.19%

4. Combination of several segmentations

4.1. Principles of the merging processes

Segmentation algorithms behave differently according to the transitions they are asked to detect. The main idea here is to take into account the different behaviors of segmentation algorithms so as to favor more some segmentation marks than others given a certain type of transition to detect. We thus propose a processing that merges K boundaries produced by K different algorithms.

Let $\{c_1, \dots, c_T\}$ be a set of T phonetic classes. By using a small database segmented manually, we start by estimating the segmentation accuracy $\alpha_k(c_i, c_j)$ at a tolerance of 20 ms for every algorithm indexed by k, $k = 1, 2, \dots, K$ and every pair (c_i, c_j) of classes, $(i, j) \in \{1, \dots, T\}^2$.

Then, let s be an unknown transition time instant to estimate given two classes $c_\ell(s)$ and $c_r(s)$. Given $t_k(s)$, $k = 1, 2, \dots, K$, the K estimates of s returned by the K available algorithms, the merging phase consists in estimating s on the basis of these K estimates and the accuracies $\alpha_k(c_\ell(s), c_r(s))$.

The first solution that we propose consists in choosing the algorithm that offers the best accuracy computed during the training for the type of transition under consideration. Following the terminology used in [11], we adopt a linear hard combination given by :

$$\hat{t}_{hard}(s) = \frac{\sum_{k \in A} t_k(s)}{Card(A)} \tag{1}$$

where A is the set of algorithms k that maximize $\alpha_k(c_g(s), c_d(s))$, $k \in \{1, \dots, K\}$. Note that A is not restricted to one element. For example, if the transitions between two classes i and j are absent from the training database, the segmentation accuracy $\alpha_k(i, j)$ is not defined and thus, we impose $\alpha_k(i, j)$ to be equal to 1 for each k. In this case, $Card(A) = K$ and the equation (1) becomes a simple average of the K time instants produced by the K algorithms.

Still following [11], another solution is to apply a soft combination for the K boundaries performed by the K algorithms. Thus the estimated boundary for the transition s is computed as the barycenter of the K time instants as :

$$\hat{t}_{soft}(s) = \frac{\sum_{k=1}^{K} \alpha_k(c_l(s), c_r(s)) t_k(s)}{\sum_{k=1}^{K} \alpha_k(c_l(s), c_r(s))}. \tag{2}$$

Similarly to the hard combination, $\alpha_k(i, j)$ is set to 1 for each k if the transitions between the classes i and j are absent from the training database. Obviously, this case is equivalent to a hard combination.

4.2. Experimental results

In this section, we present the results obtained by the application of the two combination methods (see the equations (1) and (2)) to the triplet $($*HMMSeg*, *RefineSeg*, *BrSeg*$)$.

For the French corpus, the combination was achieved by using 12 classes : unvoiced plosives, voiced plosives, unvoiced fricatives, voiced fricatives, oral vowels, nasal vowels, diphtongues, nasal consonants, liquid consonants, semi vowels, pauses and silences. For the English corpus, 10 classes were used : vowels, voiced/unvoiced plosives, voiced/unvoiced fricatives, nasal consonants, liquid consonants, semi vowels, pauses and silences.

TAB. 3 – Accuracies at 20 ms for different combination methods for the French corpus

SizeComb	*SizeAlg*	*hard fusion*	*isobary-center*	*soft fusion*	*optimal soft fusion*
100	*100*	93.04%	93.67%	94.20%	94.24%
	300	93.81%	94.38%	94.82%	94.90%
	700	94.14%	94.58%	94.97%	95.10%
300	*100*	92.89%	93.68%	94.23%	94.26%
	300	93.77%	94.39%	94.88%	94.91%
	700	94.18%	94.58%	95.07%	95.10%

TAB. 4 – Accuracies at 20 ms for different combination methods for the English corpus

SizeComb	*SizeAlg*	*hard fusion*	*isobary-center*	*soft fusion*	*optimal soft fusion*
100	*100*	93.02%	93.68%	93.96%	94.00%
	300	93.74%	94.36%	94.69%	94.70%
	700	94.10%	94.58%	94.91%	94.93%
300	*100*	93.08%	93.66%	93.98%	94.00%
	300	93.80%	94.37%	94.70%	94.70%
	700	94.25%	94.58%	94.92%	94.93%

TAB. 5 – The limit performance of the combination methods

	hard combination	*isobary-center*	*soft combination*
corpusFR	95.11%	94.86%	95.39%
corpusEN	94.70%	94.85%	95.19%

TAB. 6 – Corrective ability of the three combination methods for two typical configurations

Mark position in case at least one algorithm produces an error	*Corpus*	*Frequency of occurrence*	*Correction after hard combination*	*Correction after isobarycenter combination*	*Correction after soft combination*
3 marks on the same side	*corpusFR*	20.35%	51.25%	48.50%	50.57%
	corpusEN	16.28%	43.15%	41.56%	42.66%
2 marks on the same side	*corpusFR*	8.11%	79.14%	95.07%	95.71%
	corpusEN	7.23%	84.27%	95.44%	96.30%

Let *SizeComb* denote the number of sentences in the training database used for the merging process. The estimation of the accuracies for the soft and hard combinations is achieved by using two different values of *SizeComb* : 100 and 300. The chosen corpora are different from those used for the training of *HMMSeg* and *RefineSeg*. Thus, the accuracies given in this section are computed at a tolerance of 20 ms and evaluated on all the sentences of the database except those employed for training *HMMSeg*, *RefineSeg* and the soft and hard combinations. As in section 3, the results presented here are obtained by averaging the accuracies using a cross-validation procedure.

The hard and soft combinations are compared to the so-called isobarycenter method and the optimal soft combination. The isobarycenter method averages the three time instants obtained by *HMMSeg*, *RefineSeg* and *BrSeg*. The optimal soft combination is the soft combination when the accuracies $\alpha_k(c_i, c_j)$ are estimated on the whole corpus.

The results of the combination methods are given in tables 3 and 4. For every pair *(SizeComb,SizeAlg)*, the accuracies obtained by the four combination methods are larger than the best result of the line associated to *SizeAlg* in table 1.

Similarly to table 2, table 5 presents the results obtained by using the whole database for training *HMMSeg* and *RefineSeg* and estimating the accuracies for the combination methods. These results can be regarded as the limit performance of the merging methods. The difference between this ideal case and the accuracies presented in Tables 3 and 4 is not very important, which illustrates that using more learning data hardly improves the results.

In order to get further insight into the behavior of the three combination methods, we analyzed the ability of each combination method to correct errors made by *HMMSeg*, *RefineSeg* and *BrSeg*. By error, we mean a segmentation mark further that 20 ms from the manual boundary. For that purpose, two configurations were studied : the first one where the 3 marks are located on the same side relatively to the manual boundary and the second one where these marks are on both sides of the manual boundary. The results obtained on *corpusFR* and *corpusEN* are presented in table 6 and show that in the first configuration, all the combination methods lead to similar performance. However, in the second configuration, the accuracies yielded by the soft and isobarycenter combination methods are respectively 95.07% and 95.71% for *corpusFR* and 95.44% and 96.30% for *corpusEN*. These results are significantly higher than the rate obtained by the hard combination method (79.14% for *corpusFR* and 84.27% for *corpusEN*).

5. Conclusion

In this paper we have analyzed the performance of three automatic segmentation algorithms for French and English corpora : a global method based on HMM and two local methods that aim at detecting a transition in the vicinity of a boundary (Refinement by boundary model and Brandt's GLR method).

We have also proposed two methods capable of merging the boundaries produced by the different segmentation algorithms. The experimental results of these two methods applied to two languages show a clear improvement of the accuracy at 20 ms. Furthermore, these methods are simple and not computationally expensive. They seem to be a good prospect regarding the segmentation problem for TTS synthesis applications.

6. References

[1] Brugnara, F. and Falavigna, D. and Omologo, M., "Automatic segmentation and labelling of speech based on Hidden Markov Models", Speech Communications, V(12), pp. 357-370, 1993.

[2] Nefti, S., "Segmentation automatique de la parole en phones. Correction d'étiquetage par l'introduction de mesures de confiance", Université de Rennes I, 2004.

[3] Kim, Y. J. and Conkie, A., "Automatic segmentation combining an HMM-based approach and spectral boundary correction", ICSLP 2002,Colorado, september, 2002.

[4] Jarifi, S. and Pastor, D. and Rosec, O., "Coopération entre méthodes locales et globales pour la segmentation automatique de corpus dédiés à la synthèse vocale", JEP, June, 2006.

[5] Matousek, J. and Tihelka, D. and Psutka, J., "Automatic Segmentation for Czech Concatenative Speech Synthesis Using Statistical Approach with Boundary-Specific Correction", EUROSPEECH, Geneva, 2003.

[6] Brandt, A. V., "Detecting and estimating parameters jumps using ladder algorithms and likelihood ratio test", Proc.ICASSP, pp. 1017-1020, November, 1983.

[7] Obrecht, R. A., "A New Statistical Approach for the Automatic Segmentation of Continuous Speech Signals ", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.36(1), pp. 29-40, 1988.

[8] Wang, L. and Zhao, Y. and Chu, M. and Zhou, J. and Cao, Z., "Refining Segmental Boundaries for TTS Database Using Fine Contextual-Dependent Boudary Models", Proc. ICASSP, pp. 641-644, Montreal, Canada, 2004.

[9] Young, S. and Evermann, G. and Hain, T. and Kershow, D. and Moore, G. and Odell, J., "The HTK Book for HTK V3.2.1", Cambridge University Press, Cambridge, 2002.

[10] Jarifi, S. and Pastor, D. and Rosec, O., "Brandt's GLR method & refined HMM segmentation for TTS synthesis application", 13th European Signal Processing Conference (EUSIPCO),2005.

[11] Pigeon, S., "Authentification multimodale d'identité", PhD thesis, Université Catholique de Louvain, 1999.