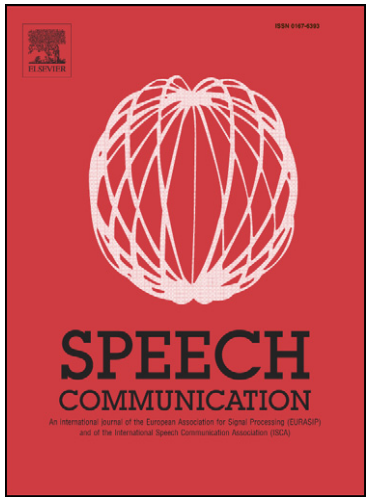
A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis

Safaa Jarifi, Dominique Pastor, Olivier Rosec



A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis

Safaa Jarifi, Dominique Pastor, Olivier Rosec

PII: S0167-6393(07)00121-5
DOI: 10.1016/j.specom.2007.07.001
Reference: SPECOM 1656

To appear in: *Speech Communication*

Received Date: 13 November 2006
Revised Date: 18 June 2007
Accepted Date: 2 July 2007

Please cite this article as: Jarifi, S., Pastor, D., Rosec, O., A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis, *Speech Communication* (2007), doi: 10.1016/j.specom.2007.07.001

A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis ⋆

Safaa Jarifi [a] , Dominique Pastor [a] , Olivier Rosec [b]

[a] *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*

[b] *France Telecom, R&D Division, TECH/SSTP/VMI, 2, avenue Pierre Marzin, 22307 Lannion Cedex, France.*
olivier.rosec@orange-ftgroup.com

Abstract

This paper deals with the automatic segmentation of large speech corpora in the case when the phonetic sequence corresponding to the speech signal is known. A direct and typical application is corpus-based Text-To-Speech (TTS) synthesis.

We start by proposing a general approach for combining several segmentations produced by different algorithms. Then, we describe and analyse three automatic segmentation algorithms that will be used to evaluate our fusion approach. The first algorithm is segmentation by Hidden Markov Models (HMM). The second one, called refinement by boundary model, aims at improving the segmentation performed by HMM via a Gaussian Mixture Model (GMM) of each boundary. The third one is a slightly modified version of Brandt's Generalized Likelihood Ratio (GLR) method; its goal is to detect signal discontinuities in the vicinity of the HMM boundaries.

Objective performance measurements show that refinement by boundary model is the most accurate of the three algorithms in the sense that the estimated segmentation marks are the closest to the manual ones. When applied to the different output segmentations obtained by the three algorithms mentioned above, any of the fusion methods proposed in this paper is more accurate than refinement by boundary model. With respect to the corpora considered in this paper, the most accurate fusion method, called *optimal fusion by soft supervision*, reduces by 25.5%, 60% and 75%, the number of segmentation errors made by refinement by boundary model, standard HMM segmentation and Brandt's GLR method, respectively. Subjective listening tests are carried out in the context of corpus-based speech synthesis. They show that the quality of the synthetic speech obtained when the speech corpus is segmented by *optimal fusion by soft supervision* approaches that obtained when the same corpus is manually segmented.

Key words: Automatic speech segmentation, speech synthesis, HMM, Brandt's GLR algorithm, boundary model, soft supervision, hard supervision.

1 Introduction

For several years, corpus-based speech synthesis has become increasingly popular for the high quality of synthetic voice it provides. By selecting and concatenating speech segments or units stored in a large database, such synthesizers can select a sequence of units that corresponds to the context of the entry text. By so proceeding, modification of the speech signals is avoided or at least drastically limited and thus the naturalness of the original speech can be preserved. However, as always in concatenative speech synthesis, the quality of the output speech is highly dependent on the corpus and on the processing operated on this corpus. More precisely, the phonetic transcription and segmentation tasks are of prime importance. In the scope of this paper, the transcription is assumed to contain no or very few errors. This assumption is reasonable in the context of corpus-based speech synthesis since, during the recording of each corpus, the speaker's uttering is strictly monitored so as to guarantee an almost perfect reading. Moreover, further checking of the phonetic transcription is also performed. The most tedious task in the voice creation process is by far the segmentation. This is due to the fact that automatic segmentation methods are not reliable enough and thus manual checking remains mandatory, a task which is extremely costly both in terms of time and development costs. This need for manual intervention is considered as a limiting factor for building new voices for corpus-based synthesis. Given the increasing demand in terms of voice diversification for speech synthesis, there is therefore a need to improve the automation and accuracy of the segmentation process for TTS applications.

So far, the HMM approach [4,10] has been the most widely used for automatic segmentation and it is considered as the most reliable. This approach is linguistically constrained because it requires the knowledge of the true phonetic sequence associated with the recorded utterances in order to estimate the HMM sequence. However, this approach has some limitations for building voices for TTS systems. The main limitation is that HMM model steady areas well but are not really suited to locally detecting the transitions between phonemes in a speech signal.

It should also be noted that more local approaches have been proposed for

⋆ This study is supported by France Telecom

speech segmentation. For instance boundary models were introduced in [17] to refine the marks produced by a classic HMM based segmentation algorithm. Another strategy is to detect discontinuities in the speech signal as proposed by Brandt in [3]. Nevertheless, this approach is not linguistically constrained and therefore leads to the omission and insertion of segmentation marks.

With respect to the foregoing, the purpose of this paper is to combine global and local automatic segmentation algorithms in order to improve the accuracy of the resulting automatic segmentation. This is the aim of section 2. Several fusion methods are proposed. They are based on a general scheme presented in section 2 for the weighted average of segmentation marks.

In order to evaluate the performance of these fusion methods, we study and justify the choice of three automatic segmentation algorithms in section 3. The first one is HMM segmentation. It applies a forced alignment between the HMM sequence and the speech signal. The second segmentation algorithm is referred to as refinement by boundary model. It was originally proposed in [17] to segment a Chinese corpus. It uses a boundary model, which is estimated on a small database, to refine the HMM segmentation marks. The third algorithm is Brandt's GLR method whose aim is to detect discontinuities in speech signals. Unlike segmentation by HMM and refinement by boundary model, this method needs no prior knowledge of the transcription. However, when the transcription is available, this method can easily be adapted so as to take this information into account.

The accuracy rates of these automatic segmentation methods are then evaluated in section 4 on a French and on an English corpus. These accuracy rates are computed with respect to manual segmentations of the corpora. By manual segmentation, we mean the segmentation resulting from the manual checking of a standard HMM segmentation. The accuracy rates are computed at a tolerance of 20 ms with respect to the manual segmentations, the accuracy rate at 20 ms of a given segmentation being the percentage of those segmentation marks that are at more than 20 ms away from their corresponding manual marks. This tolerance is considered in [9,17] as an acceptable limit to produce a synthetic speech of good quality. According to this criterion, the three algorithms turn out to be complementary in the sense that they are adapted to detecting different types of boundaries. Note that a tolerance of 20 ms is certainly too strict for a vowel-semivowel (or liquid) boundary and that the notion of boundary between two vowels is of an elusive type, whereas plosive bursts are characterized by clear acoustic features. It follows that the criterion should be context-dependent. However, to the authors' knowledge, no exhaustive study making it possible to choose the tolerance as a function of the type of transition to detect has yet been achieved.

Section 5.1 evaluates the accuracy of the fusion methods and section 5.2

presents the results of subjective tests that measure the speech quality when the best fusion method is used to segment the French and the English corpora. The last section concludes this paper and proposes some extensions.

2 A general fusion approach for combining segmentations

Generally, segmentation algorithms behave differently according to the phonetic transition to be detected. The main idea of the approach proposed below is to take into account these different behaviours so as to favour some segmentation marks rather than others, given a certain type of transition.

More specifically, let s be a transition to be detected between two phonemes and assume that the phonetic class of the phoneme to the right (resp. to the left) of s is c_r (resp. c_ℓ). The principle of the proposed method is to compute a new estimate $\widehat{t}(s)$ of the transition instant on the basis of K time instants $t_1(s), \ldots, t_K(s)$ produced by K segmentation algorithms.

This can be regarded as a problem of fusion. The solution we propose is to compute an estimate $\widehat{t}(s)$ based on a weighted average of selected segmentation marks. This estimate is given by

$$\widehat{t}(s) = \sum_{k \in A} \beta_k(c_\ell, c_r) t_k(s), \tag{1}$$

where A is the index set of the selected marks and the coefficients $\beta_k(c_\ell, c_r)$ satisfy the relation

$$\sum_{k \in A} \beta_k(c_\ell, c_r) = 1.$$

The estimate given by equation (1) corresponds to the case of algorithms that make no systematic error. If any algorithm, say the k^{th}, made a known systematic error, it would suffice to replace in equation (1) the corresponding estimate $t_k(s)$ by $t_k(s) - m_k$ where m_k is the value of this error.

Figure 1 illustrates the computation of $\widehat{t}(s)$. We now detail the computation of this estimate by describing the different components of the fusion scheme. To the authors' best knowledge, this fusion scheme is not usual for combining segmentation marks and is complementary to approaches such as those proposed in [13,14].

By introducing the *mark selection* whose outcome is the index set A used to compute $\widehat{t}(s)$, we take into account that selecting marks independently of the coefficients $\beta_k(c_\ell, c_r)$ may prove sensible. Consider the following example. Suppose that 6 different algorithms are used to estimate the segmentation mark.

Assume that 5 of these algorithms detect the time instant of the transition s within the same interval and that the sixth algorithm gives an estimate of this time instant significantly further away from the others. In this case, it is likely that the time instant performed by the sixth algorithm is not correct and, thus, a simple average of the 6 estimations will be less accurate than the average of those located in the same interval. Thus, it can be relevant to select only some of the available estimates in order to compute $\widehat{t}(s)$. The distance between marks can be a natural criterion to achieve this selection. In subsection 2.1, two possible types of mark selection are described, the second one being based on the distance between marks.

The coefficients $\beta_1(c_\ell, c_r), \beta_2(c_\ell, c_r), \ldots, \beta_K(c_\ell, c_r)$ are obtained as follows. With the notations introduced above, we start by *scoring* the K algorithms on the basis of a training database. The scores $\gamma_k(c_\ell, c_r), k = 1, \ldots, K$, must quantify the respective behaviour of the algorithms for detecting the transition between the classes c_ℓ and c_r. For example, a large value for $\gamma_k(c_\ell, c_r)$ should be assigned to the k^{th} algorithm, if this algorithm performs well on the pair of classes (c_ℓ, c_r). This scoring phase is performed once and for all. We thus obtain a set of scores for all the algorithms and for all the pairs of phonetic classes present in the training corpus. Several types of scores can be proposed. In this paper, we will consider the accuracy rate at 20 ms. This choice is motivated for the following reasons. On the one hand, we are interested in the precision of the segmentation at a tolerance of 20 ms; on the other hand, this score is a reliable measure of the ability of a given algorithm to detect a type of transition.

The next step is the *score supervision* that transforms the sequence $\gamma_k(c_\ell, c_r)$ of scores, where $k = 1, \ldots, K$, into a sequence $\omega_1(c_\ell, c_r), \omega_2(c_\ell, c_r), \ldots, \omega_K(c_\ell, c_r)$ of weights. The weights indicate the quality of each algorithm in comparison with the others. The role of this phase is similar to that of a supervisor who decides to favour some algorithms rather than others on the basis of his experience, his prior knowledge, some heuristics and so forth. This transformation is essentially a matter of choice and various possibilities are considered later. Note that the computation of the weights can be achieved regardless of the scores. In particular, the score supervision can favour no algorithm by simply assigning the same weight to every algorithm (see section 2.2.1). Note that if the transitions between two classes c_i and c_j are absent from the training database, $\omega_k(c_i, c_j)$ is not defined and, thus, we force $\omega_k(c_i, c_j)$ to 1 for every k. In subsection 2.2, we describe three possible types of supervision.

During the segmentation task, only the weights corresponding to the selected marks are normalized to produce the coefficients $\beta_k(c_\ell, c_r)$ defined by:

$$\beta_k(c_\ell, c_r) = \frac{\omega_k(c_\ell, c_r)}{\sum_{j \in A} \omega_j(c_\ell, c_r)}, k \in A.$$

To perform the combination, we must choose the type of score, the mark selection and the supervision. Many choices are possible. In what follows, we propose and discuss some simple and efficient choices.

2.1 Mark selection

The mark selection involves choosing, for each transition, the marks of the algorithms that will be used to estimate the transition time instant. It is thus achieved by a function that assigns a K-uple $(\delta_1, \ldots, \delta_K)$ in $\{0,1\}^K$ to every K-uple $(t_1, \ldots, t_K)$ of segmentation marks.

2.1.1 Total selection

This is the basic case where we keep the K marks produced by the K algorithms. Therefore, we have $\delta_k = 1$ for each k and $A = \{1, 2, \ldots, K\}$.

2.1.2 Partial selection

Partial selection involves choosing a subset of the K marks we have. This selection is achieved in two steps. The first step is to determine clusters of marks located within the same zone. Here, we use a distance to find these clusters. The second step is to choose one or more clusters on the basis of a criterion. For example, we can choose the cluster that contains the largest number of marks.

The separation of the marks into clusters is a complicated problem in the general case. Relatively sophisticated algorithms, such as k-NN (k-Nearest Neighbours) and genetic algorithms, can be used. When $K = 3$ and since segmentation marks are real numbers, which is our case, the marks can be easily determined as follows. We choose the two marks that are closest together. In cases where they are equidistant, we keep the three marks. In what follows, this selection will be called *partial selection by distance criterion*.

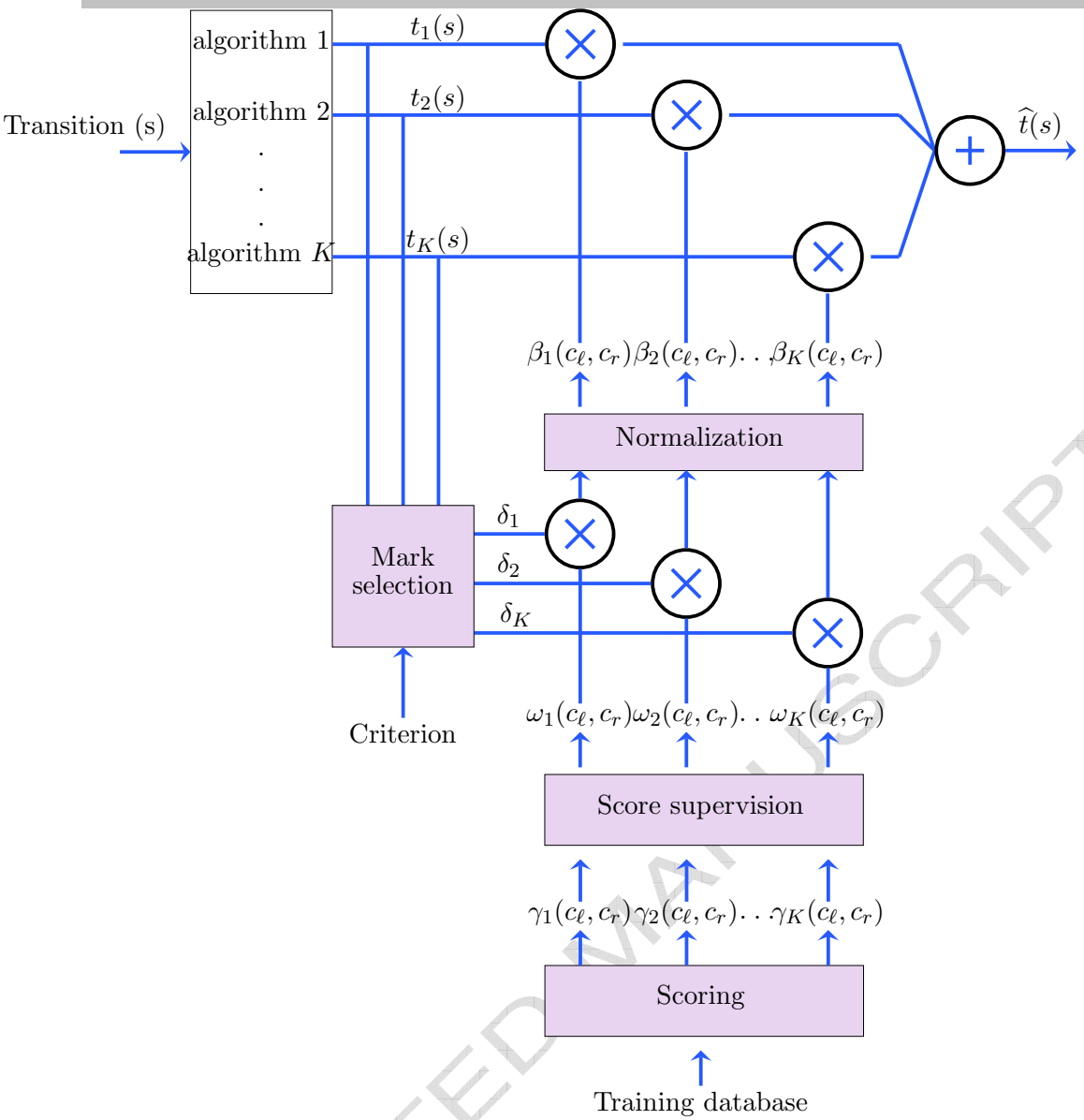


Fig. 1. For a given transition s to detect between two phonemes when the phoneme to the left (resp. to the right) belongs to class c_l (resp. c_r), we present a general scheme for computing $\widehat{t}(s)$ by the weighted average of segmentation marks. We have δ_k equal to 1 if $k \in A$, where A is the index set of the selected marks, and 0 otherwise.

2.2 Score supervision

The score supervision is basically a function that assigns the weights $\omega_1(c_\ell, c_r)$, $\ldots, \omega_K(c_\ell, c_r)$ to the scores $\gamma_1(c_\ell, c_r), \ldots, \gamma_K(c_\ell, c_r)$. In this paper we consider the particular case where the computation of the weights is achieved by using one single function f, called a *weighting function*, such that $\omega_k(c_\ell, c_r) = f(\gamma_k(c_\ell, c_r))$ for $k = 1, \ldots, K$.

Equation (1) becomes:

$$\widehat{t}(s) = \frac{\sum_{k \in A} f(\gamma_k(c_\ell, c_r)) t_k(s)}{\sum_{k \in A} f(\gamma_k(c_\ell, c_r))}. \tag{2}$$

The supervision must be adapted to the type of score. If the larger the score $\gamma_k(c_\ell, c_r)$, the more accurate the k^{th} algorithm, the weighting function f must be non-decreasing. Otherwise, if the larger the score $\gamma_k(c_\ell, c_r)$, the less accurate the k^{th} algorithm, the weighting function f must be non-increasing.

2.2.1 Uniform supervision

This is the simplest supervision that we can suggest: $f(\gamma_k(c_\ell, c_r))$ is equal to 1, for every type of score, every algorithm and every type of transition. In other words, the supervisor favours no algorithm. The outcome of the score supervision is thus the average value of the selected marks, that is the time-instants $t_k(s)$ such that $k \in A$. We then have

$$\widehat{t}(s) = \frac{1}{\mathrm{card}(A)} \sum_{k \in A} t_k(s), \tag{3}$$

where $\mathrm{card}(A)$ denotes the number of elements of A.

2.2.2 Hard supervision

The weights assigned by the supervision are 0 or 1, hence the name *hard supervision*. These binary weights are computed as follows. Let $\gamma_{\max}$ be the maximum value of the scores $\gamma_k(c_\ell, c_r)$ of the selected algorithms, that is, for $k \in A$. The elements of the set $I = \{k \in A : \gamma_k(c_\ell, c_r) = \gamma_{\max}\}$ are the most appropriate algorithms for detecting transition s. In this case, the weighting function f is defined by:

$$f(\gamma_k(c_\ell, c_r)) = \begin{cases} 1 & \text{if} \quad k \in I \\ 0 & \text{otherwise} \end{cases}.$$

The estimate $\widehat{t}(s)$ is then given by:

$$\widehat{t}(s) = \frac{1}{\mathrm{card}(I)} \sum_{k \in I} t_k(s). \tag{4}$$

2.2.3 Soft supervision

In contrast to hard supervision, soft supervision assigns a non binary value. In this paper, we propose two different weighting functions valued in $\mathbb{R}$. These functions are increasing: since the score we consider is the accuracy rate at 20 ms, the larger the score, the larger the weight must be.

The two weighting functions studied in this paper are:

$$f(x) = x$$

and

$$f(x) = \frac{1}{1-x},$$

where x is the accuracy rate at 20 ms.

Many other functions can be proposed. With the first function, we consider that the accuracy rate is a sufficiently good confidence measure. Since x is the accuracy rate at 20 ms, $1-x$ is the error rate at 20 ms; therefore, the value of the second function at x is the inverse of this error rate. Similarly to the accuracy, the inverse error rate at 20 ms is also a good confidence measure. With this second function, we discriminate more between the different algorithms. For instance, given a pair of classes (c_ℓ, c_r), suppose that the accuracy rates at 20 ms of 2 algorithms are 80% and 90% respectively. The corresponding inverse error rates are then 0.05 and 0.1. The weight of the second algorithm is thus twice as large as that of the first one when soft supervision is performed on the basis of the inverse error rates, whereas the weights in terms of accuracy rates are of the same order.

3 The three automatic segmentation algorithms

We describe the three segmentation algorithms that will be combined via the general fusion approach proposed above. The three algorithms are: segmentation by HMM, refinement by boundary model and Brandt's GLR method. This choice is motivated by the fact that the algorithms behave differently depending on the classes of the transitions to be detected. In this sense, we can say that these algorithms are complementary.

3.1 Segmentation by HMM

This approach is considered as the standard method for speech segmentation and basically consists of two main steps. The first step is training that aims at

estimating the acoustic models. The second step uses these models to segment the speech signal by means of the Viterbi algorithm. The latter applies a forced alignment between the models associated with the known phonetic sequence and the speech signal.

The training phase is crucial because the accuracy of the segmentation by HMM closely depends on the quality of the estimated models and thus on the initialization of these models. To initialize the models, several methods exist.

For example, we can use iterative training [18] on the whole corpus. The boundaries resulting from the previous iteration are used to initialize and re-estimate the models via the Baum-Welch algorithm. After a few iterations of the training process, mismatches between the manual segmentation marks and the boundaries produced by the HMM approach are significantly reduced. The HMM-based approach using this "flat start" training will hereafter be referred to as standard HMM segmentation.

Another method that can be considered is illustrated in figure 2. It uses a small speech database segmented and labelled manually to estimate the models [9]. Then, we segment the whole corpus with these models. The initialization of these models is the same as in the first method. If the small corpus contains several realizations of each phone of the database, the initialization of the models on this small corpus is good and this latter processing performs better than the method described in the preceding paragraph [7]. For this reason, we prefer to apply the general fusion approach to the HMM segmentation that uses this training. In what follows, we call *HMMSeg* the segmentation performed by using this training procedure.

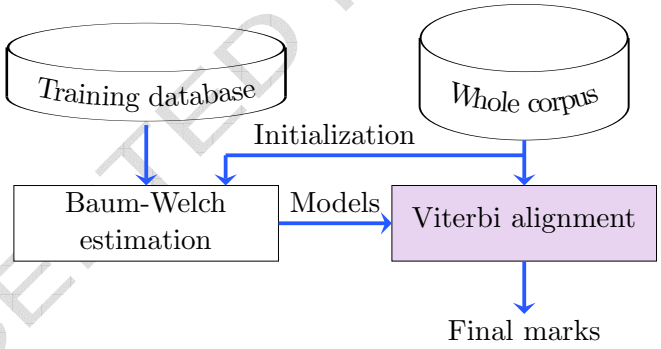


Fig. 2. Segmentation by HMM based on a training corpus manually segmented.

3.2 Refinement by boundary model [17]

The main idea of this method is to train a set of boundary models by using a small database manually labelled and segmented. Then, these models serve

to refine an initial segmentation as in [17]. More specifically, this method is carried out in two steps as shown in figure 3.

For each boundary of the training database, we create a super vector by concatenating the acoustic vectors of the $(2N+1)$ frames that are around the manual boundary (see figure 4). Since each boundary B depends on the phoneme X to its left and on the phoneme Y to its right, boundary B is henceforth called the pseudo-triphone $X-B+Y$ as proposed in [17] (see figure 5). Because the number of labelled data is limited in practice, the pseudo-triphones are clustered into a reduced number of classes via a Classification And Regression Tree (CART). A Gaussian Mixture Model (GMM) is then estimated for each class. The questions put during the construction of the CART concern the phonetic classes and phonemic identity.

The second step aims at refining each boundary of an initial segmentation. Given a labelled sentence and its initial segmentation, we seek in a certain vicinity of each boundary the time instant that maximizes the likelihood of the super vector corresponding to this instant. This likelihood is computed as follows. For each possible time instant around the initial boundary, we form a super vector centred on the current frame as in the training step; since this super vector is assumed to represent a given pseudo-triphone, we use the CART [11] to determine the class corresponding to this pseudo-triphone; the likelihood is finally calculated given the GMM associated with the obtained class and the super vector.

This algorithm is linguistically constrained because it needs prior knowledge of the phonetic sequence in order to create the boundary models. However, it can be applied to any segmentation that contains no omission and no insertion. For example, in [17], refinement by boundary model was applied to HMM segmentation obtained by forced alignment.

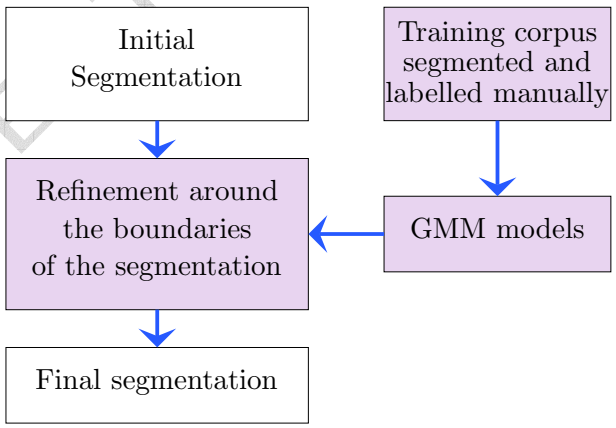


Fig. 3. The different steps of refinement by boundary model

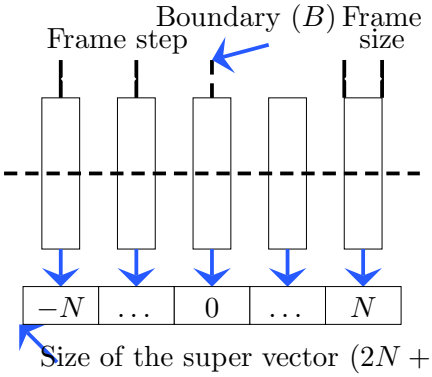


Fig. 4. Construction of a super vector. We consider N non overlapping frames to the right and N non overlapping frames to the left of a boundary. In addition, we take into account the frame centred on the boundary. The $(2N+1)$ acoustic vectors of these $(2N+1)$ frames form the super vector.

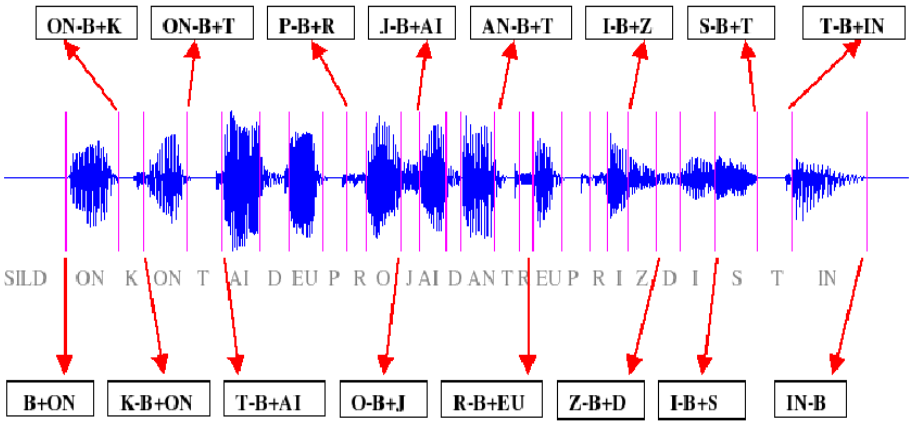


Fig. 5. The pseudo triphones of the French sentence "On comptait deux projets d'entreprise distincts"

3.3 Brandt's GLR algorithm

3.3.1 The basic algorithm

The aim of this method is to detect discontinuities in speech signals. Speech signals are assumed to be sequences of homogeneous segments. Each segment w is a finite sequence $w = (y_n)$ of samples that are assumed to obey an autoregressive (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 the segments, and e_n is a zero mean white Gaussian noise with variance equal to σ^2. Such a segment is thus characterized by the parameter vector $\Theta = (a_1, \ldots, a_p, \sigma)$. Let w_0 be some segment of N samples and Θ_0 be the corresponding parameter vector. Brandt attempts in [3] to decide whether w_0 should be split into two subsegments w_1 and w_2 or not. A possible split results from the detection of a jump between the parameter vectors Θ_1 and Θ_2 of w_1 and w_2 respectively. For that purpose, Brandt's algorithm uses the generalized likelihood ratio, which, under the assumption that the samples $y_1, \ldots, y_n$ are Gaussian, can be written as

$$D_N(r) = N \ln \hat{\sigma}_0 - r \ln \hat{\sigma}_1 - (N - r) \ln \hat{\sigma}_2,$$

where r is the size of the time interval covered by w_1, while $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the noise standard deviation estimates of the models characterized by the parameter vectors Θ_1 and Θ_2 respectively. Brandt's GLR method decides that a jump between the parameter vectors Θ_1 and Θ_2 of w_1 and w_2 has occurred by comparing $max_r(D_N(r))$ to a predefined threshold λ. The change instant is the value $\hat{r} = arg(max_r(D_N(r)) \geq \lambda)$.

A direct implementation of this method is computationally expensive. A sub-optimal version is recommended in [2]. In particular, the length of w_2 is fixed to a predefined value L. For further details, the reader can refer to [2].

3.3.2 Brandt's GLR algorithm with known phonetic transcription

As mentioned above, the purpose of Brandt's GLR method is to detect discontinuities of speech signals without any further knowledge of the phonetic sequence. This algorithm is linguistically unconstrained and makes insertions and omissions.

We propose here an adaptation of Brandt's GLR method when the pronounced phonetic sequence is available as often assumed for the segmentation of speech synthesis corpora. In such a case, an initial segmentation can be obtained, for example, by an HMM-based method. For each initial segmentation mark, we define a time interval over which a modified version of Brandt's GLR method is applied so as to provide one single segmentation mark.

More specifically, let $(U_0, U_1, \ldots, U_L)$ be the boundaries of the initial segmentation. 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 determining the time instant that maximizes the GLR. By removing the thresholding, we make no omission and no insertion. This is the method used below and despite the modification proposed, we still call it Brandt's GLR method.

4 Evaluation of the three segmentation algorithms

The general fusion approach proposed in section 2 is basically aimed at achieving a more accurate segmentation than that produced by the different segmentation algorithms that are combined. Hence, we evaluate the performance of each of the three algorithms proposed above and we will verify that they are complementary.

4.1 Description of the corpora

The performance of each algorithm is evaluated on a French and on an English corpus. The French corpus, hereafter called *FRcorpus*, contains 7300 sentences uttered by a female speaker and sampled at 16 kHz. The English corpus, called *ENcorpus*, corrsponds to the recording of 8900 sentences uttered by a female speaker and also sampled at 16 kHz.

The training phase of refinement by boundary model and that required to compute *HMMSeg* (see section 3.1) are carried out successively on common databases containing 100, 300 and 700 sentences. Each database is chosen randomly within the speech corpora. This random choice is made up to a minimum number of realizations per phone. We choose this minimum equal to 3. Our analysis is based on a cross-validation procedure that includes three different training databases. These training databases were built so that they do not overlap for a given training database size.

4.2 Parameters

The segmentation *HMMSeg* is performed by using the HTK toolkit [18] for the acoustic analysis, the model training and the segmentation. For each phone, we consider a left-to-right three-state model; the observation probabilities are modelled by the mixture of two Gaussian distributions. The acoustic vectors contain 39 coefficients: 12 Mel Frequency Cepstral Coefficients (MFCCs), the normalized energy, and the first and second derivatives of these 13 coefficients. Twenty iterations of the Baum-Welch algorithm are applied to train the HMM.

The segmentation obtained by applying refinement by boundary model to *HMMSeg* is called *RefinedHMMSeg*. For every boundary of the HMM segmentation, the refined boundary is searched for within an interval of 60 ms centred on this boundary with a search step fixed to 5 ms. The super vector is computed with $N = 2$, a frame step equal to 30 ms and a frame size equal to 20 ms.

Thus, each super vector contains $39(2\times2+1) = 165$ coefficients. The parameter values given above were originally determined for a Chinese corpus in [17]. In [6], it is shown that these values remain suitable for a French corpus.

For Brandt's GLR method, the input segmentation is *HMMSeg*. Therefore, the performance of Brandt's GLR method depends on the training set used to achieve *HMMSeg*. With Brandt's GLR method, we search for a discontinuity around each HMM boundary. The resulting segmentation is called *BrandtSeg*. The AR model order is set to 12 and the minimal length of w_1 and w_2 is equal to 10 ms.

4.3 Results and discussion

For the evaluation of *HMMSeg*, *RefinedHMMSeg* and *BrandtSeg*, the accuracy rates at a tolerance equal to 20 ms with respect to the manual segmentation are computed on the whole corpus except the sentences used for the training processes. As we are using a cross-validation procedure, with three different training sets, the results depicted in Table 1 are the average accuracy rates obtained on the three corresponding test sets. Table 2 shows the accuracy rates of the standard HMM segmentation and the performance limit of the other algorithms. The performance limit of a given algorithm is the accuracy obtained by training this algorithm on the whole database. As far as the standard HMM segmentation is concerned, the entire database is used both for the estimation of the HMM models and the segmentation task.

Table 1
Accuracies of *HMMSeg*, *RefinedHMMSeg* and *BrandtSeg*

	AlgSize	*HMMSeg*	*RefinedHMMSeg*	*BrandtSeg*
FRcorpus	100	91.71%	91.08%	83.22%
ENcorpus		91.98%	89.58%	86.78%
FRcorpus	300	92.51%	93.26%	83.39%
ENcorpus		92.95%	92.46%	87.10%
FRcorpus	700	92.47%	94.00%	83.38%
ENcorpus		93.00%	93.50%	87.09%

According to these tables, we can make the following remarks.

- *HMMSeg* is overall more accurate than the standard HMM segmentation. This shows that an initialization of the models via a small manually segmented database yields better results than a standard HMM initialization

Table 2
Accuracies of the standard HMM segmentation with flat start (hereafter, Standard HMM) and performance limit of *HMMSeg*, *RefinedHMMSeg* and *BrandtSeg*

	HMMSeg	*RefinedHMMSeg*	*BrandtSeg*	*Standard HMM*
FRcorpus	92.68%	95.00%	83.22%	88.53%
ENcorpus	93.17%	94.30%	87.19%	87.77%

based on the whole corpus.

- *RefinedHMMSeg* is more accurate than *HMMSeg* provided that the boundary models are well trained i.e. if the number of boundaries available in the training database is large enough. 300 sentences for the French corpus and 700 for the English corpus are sufficient for *RefinedHMMSeg* to outperform the others; 300 sentences of *FRcorpus* correspond approximately to 10000 boundaries and 700 sentences of *ENcorpus* contain around 30000 boundaries.
- The accuracy obtained with Brandt's GLR method is significantly lower than that obtained by *HMMSeg* or *RefinedHMMSeg*. However, for the English corpus, Brandt's GLR method is comparable with the standard HMM segmentation.
- With a training database of 700 sentences, the HMM approach, the refinement by boundary model and Brandt's GLR method yield accuracy rates close to their performance limits.

Because the accuracy rate at 20 ms is regarded as an important objective criterion in TTS applications, it seems reasonable to conclude from table 1 that refinement by boundary model is the most accurate algorithm. Nevertheless, we should not forget that the algorithms are not suitable for the same phonetic classes. On the one hand, it turned out that Brandt's GLR method detects some boundaries well, for example between silence and speech or between voiced and unvoiced phones. On the other hand, the refinement by boundary model and the HMM approach behave better than Brandt's GLR method for transitions between voiced phones.

To convince the reader that the three algorithms behave differently for the French corpus, we determined (see table 3) the best algorithm for each pair of phonetic classes. To construct this table, by using the same test corpus, we computed the error rate at 20 ms for each algorithm and each class of transition. We observe from this table that each algorithm is useful. For a given algorithm, there is a number of transition classes for which this algorithm gives the most accurate marks. For example, Brandt's GLR method is the best algorithm for detecting the boundaries between voiced plosives and unvoiced plosives while the refinement method by boundary model is the best algorithm to find marks between nasal vowels and unvoiced plosives. Finally, the HMM approach is the most adapted to detecting transition marks between nasal

vowels and voiced plosives. Thus, the three proposed methods seem to behave in a complementary way.

Table 3
The best algorithm for each pair of phonetic classes and for the French corpus. The terms "H", "R" et "B" refer to HMM segmentation, refinement by boundary-model and Brandt's GLR method respectively. The French phonetic classes are: oral vowels (OV), nasal vowels (NV), unvoiced plosives (UVP), voiced plosives (VP), unvoiced fricatives (UVF), voiced fricatives (VF), diphthongs (DIPH), nasal consonants (NC), liquid consonants (LC), semivowels (SV), pauses (SP) and silences (SIL). − − − means that no transition between the pair of classes is available in the corpus.

	OV	*NV*	*DIPH*	*VP*	*UVP*	*VF*	*UVF*	*NC*	*LC*	*SV*	*SP*	*SIL*
OV	B	R	B	B	R	H	H	B	R	B	B	B
NV	R	R	B	H	R	B	B	B	H	B	H	H
DIPH	H	H/R	B	B	B	H	H	B	R	R	B	H
VP	H	R	H	B	B	H	R	H	R	H	B	B
UVP	H	H	H	B	R	H	H/R	R	R	H	B	R
VF	R	R	H	B	B	H	B	H	B	H	B	B
UVF	H	H/B	R	H/B	R	H	B	H/R	B	H	R	B
NC	R	R	H	H	R	H/R/B	R/B	B	H	R	R	B
LC	R	H	H	B	B	R/B	B	R	B	H/R	R	R
SV	R	R	B	B	B	B	B	B	B	R/B	R	B
SP	B	R/B	H/R/B	R	R	R	H	R	R	B	− − −	− − −
SIL	B	B	H/R/B	R	R	B	R	R/B	R	R/B	− − −	− − −

5 Experimental results for the general fusion approach

In what follows, by fusion method, we mean a weighted average defined by a scoring mechanism, a mark selection, a score supervision and a weighting function. For hard and uniform supervisions, the weighting functions are fixed, while, for soft supervision, two possible weighting functions were proposed. Thus, for each selection method we have 4 weighting functions. Therefore, considering the two selection methods proposed in section 2 we have a total of 8 fusion methods to compare.

In this section, we start by identifying the best fusion method among the 8 chosen. This is achieved in section 5.1 by comparing the segmentation accuracy

rates computed when the fusion methods are applied to the triplet (*HMMSeg*, *RefinedHMMSeg*, *BrandtSeg*). Then, in section 5.2, we compare the quality of the synthetic speech obtained by using the segmentation produced by the best fusion method to that achieved when the HMM and manual segmentations are used.

5.1 Objective tests

Let *CombSize* denote the number of sentences of the training database used to score *HMMSeg*, *RefinedHMMSeg* and *BrandtSeg*. Three different values for *CombSize* are considered: 100, 300 and 700. The sentences of the training databases used for the scoring are chosen randomly within the whole corpus and are different and non-overlapping from those used for the training required to compute *HMMSeg* and *RefinedHMMSeg*. For *FRcorpus*, the fusion was achieved by using 12 classes: unvoiced plosives, voiced plosives, unvoiced fricatives, voiced fricatives, oral vowels, nasal vowels, diphthongs, nasal consonants, liquid consonants, semivowels, pauses and silences. For *ENcorpus*, 11 classes were considered: vowels, voiced/unvoiced plosives, voiced/unvoiced fricatives, affricates, nasal consonants, liquid consonants, semivowels, pauses and silences. The accuracy rates given in this section are computed at a tolerance of 20 ms and evaluated on all the sentences of the database except those needed to train the models for the computation of *HMMSeg*, *RefinedHMMSeg* and the different fusion methods we use to combine them. As in section 3, the results presented here are obtained by averaging the accuracy rates using a cross-validation procedure. The accuracy rates achieved by the fusion methods on *FRcorpus* and *ENcorpus* are given in tables 4 and 5 respectively.

For every pair (*CombSize, AlgSize*), any fusion method yields a segmentation more accurate than *HMMSeg*, *RefinedHMMSeg* and *BrandtSeg*. These tables also clearly show that hard fusion is not a good method as it always leads to accuracy rates lower than those obtained with uniform fusion. Concerning the mark selection method, the results show that the total selection strategy is preferable to the proposed selection based on a distance criterion. The best choice seems to be in favour of fusion with total selection and soft supervision of the inverse error rates (when $f(x) = \frac{1}{1-x}$), which will be referred to as *optimal fusion by soft supervision* in the rest of the paper. For instance, according to table 4 and when $(CombSize, AlgSize) = (300, 300)$, *optimal fusion by soft supervision* achieves an accuracy rate of 94.98% for *FRcorpus*. If we compare this accuracy rate to the values given in table 1 for the same corpus and $AlgSize = 300$, we observe a reduction of 25.50% for the error rate compared to *RefinedHMMSeg*.

Similarly to table 2, table 6 displays the results obtained by using the whole

corpus for the training phases needed to compute *HMMSeg*, *RefinedHMMSeg* and estimate the scores for the fusion methods. These results are the maximum accuracy rates that the fusion methods can reach and thus can be regarded as the upper limits in terms of performance achieved by these fusion methods. The accuracy rates given in tables 4 and 5 are reasonably close to these limits when $AlgSize = 700$. In fact, the maximum accuracy rates obtained by *optimal fusion by soft supervision* are 95.72% and 95.53% for *FRcorpus* and *ENcorpus* respectively. For $(CombSize, AlgSize) = (700, 700)$, the accuracy rates of the same method are 95.22% and 95.23% for *FRcorpus* and *ENcorpus* respectively. When $AlgSize = 300$, the results are also good and this size seems reasonable for practical applications without too much performance loss in comparison with the case $AlgSize = 700$.

The training database used to tune the fusion methods and estimate the weighting functions is different from that used to train the models for HMM segmentation and refinement by boundary model. Of course, in practice, it is more appropriate to choose the same database so as to reduce the number of sentences to segment manually. This is possible without any significant performance loss. Table 7 shows the accuracy rates at 20 ms when the database is the same both for the scoring and for the training of the models in the case of *HMMSeg* and *RefinedHMMSeg*. To compute these accuracy rates, we use the 4 fusion methods that are derived from the use of the total mark selection and the three types of supervision. For a training database containing 700 sentences, the accuracy rates obtained by *optimal fusion by soft supervision* are 95.26% and 95.17% for *FRcorpus* and *ENcorpus* respectively. However, with uniform supervision, we obtain accuracy rates equal to 94.59% and 94.58%. This means that in comparison with the uniform supervision, the number of segmentation errors is reduced by 12.38% and 10.89% respectively when *optimal fusion by soft supervision* is used. It is worth noting that this optimal fusion process does not require more manually segmented data and does not introduce any significant increase in the computational load of the segmentation process. Therefore we can conclude that the estimation of the scores via the proposed training phase is useful.

The results presented in this section show that *optimal fusion by soft supervision* significantly improves the accuracy rate at 20 ms in comparison with standard HMM segmentation. It is now interesting to see if *optimal fusion by soft supervision* is capable of removing most of the coarse errors. By coarse error, we mean a segmentation error larger than 50 ms. In this respect, table 8 presents, for different tolerances, the accuracy rates obtained with standard HMM segmentation and with the segmentation achieved by *optimal fusion by soft supervision*, when the same database of size 300 is used for the scoring and the computation of *HMMSeg* and *RefinedHMMSeg*. From this table, we can observe that the number of coarse errors made by standard HMM segmentation is reduced by a fifth via *optimal fusion by soft supervision*.

Table 4
Accuracies at 20 ms for *FRcorpus* when linear fusion is achieved with different score supervisions and mark selections

CombSize	*AlgSize*	*Total selection*				*Selection by distance criterion*			
		uniform	*hard*	*soft*		*uniform*	*hard*	*soft*	
				$f(x) = x$	$f(x) = \frac{1}{1-x}$			$f(x) = x$	$f(x) = \frac{1}{1-x}$
100	*100*	93.67%	93.04%	94.20%	94.13%	93.13%	93.02%	93.16%	93.08%
	300	94.38%	93.81%	94.82%	94.75%	94.06%	93.99%	94.07%	94.02%
	700	94.58%	94.14%	94.97%	94.84%	94.32%	94.28%	94.33%	94.29%
300	*100*	93.68%	92.89%	94.23%	94.34%	94.14%	93.02%	93.15%	93.16%
	300	94.39%	93.77%	94.88%	94.98%	94.07%	94.01%	94.10%	94.14%
	700	94.58%	94.18%	95.07%	95.17%	94.32%	94.28%	94.35%	94.36%
700	*100*	93.66%	93.10%	94.22%	94.45%	93.12%	93.01%	93.14%	93.18%
	300	94.40%	93.88%	94.91%	95.10%	94.07%	94.00%	94.09%	94.15%
	700	94.58%	94.32%	95.08%	95.22%	94.33%	94.28%	94.34%	94.40%

Table 5
Accuracies at 20 ms for *ENcorpus* when linear fusion is achieved with different score supervisions and mark selections

CombSize	*AlgSize*	*Total selection*				*Selection by distance criterion*			
		uniform	*hard*	*soft*		*uniform*	*hard*	*soft*	
				$f(x) = x$	$f(x) = \frac{1}{1-x}$			$f(x) = x$	$f(x) = \frac{1}{1-x}$
100	*100*	93.68%	93.02%	93.96%	93.98%	93.26%	93.21%	93.29%	93.15%
	300	94.36%	93.74%	94.69%	94.64%	94.11%	94.10%	94.13%	94.03%
	700	94.58%	94.10%	94.91%	94.97%	94.41%	94.41%	94.42%	94.36%
300	*100*	93.66%	93.08%	93.98%	94.17%	93.24%	93.18%	93.27%	93.24%
	300	94.37%	93.80%	94.70%	94.89%	94.12%	94.11%	94.13%	94.13%
	700	94.58%	94.25%	94.92%	95.14%	94.40%	94.40%	94.42%	94.43%
700	*100*	93.66%	93.21%	93.97%	94.25%	93.25%	93.19%	93.27%	93.33%
	300	94.37%	93.97%	94.69%	94.98%	94.11%	94.11%	94.14%	94.17%
	700	94.60%	94.23%	94.93%	95.23%	94.41%	94.41%	94.43%	94.46%

5.2 Subjective tests

In the previous section, *optimal fusion by soft supervision* turned out to be the most accurate method among those studied. For TTS applications, does this

Table 6
The limit performance of the fusion methods with different score supervisions and mark selections

	Total selection				*Selection by distance criterion*			
	uniform	*hard*	*soft*		*uniform*	*hard*	*soft*	
			$f(x) = x$	$f(x) = \frac{1}{1-x}$			$f(x) = x$	$f(x) = \frac{1}{1-x}$
FRcorpus	94.86%	95.11%	95.39%	95.72%	94.75%	94.75%	94.77%	94.88%
ENcorpus	94.85%	94.70%	95.19%	95.53%	94.77%	94.77%	94.78%	94.82%

Table 7
Accuracies of the segmentation obtained by fusion by soft supervision when the same database is used for the scoring and the computation of *HMMSeg*, *RefinedHMMSeg*

		uniform	*hard*	*soft*	
				$f(x) = x$	$f(x) = \frac{1}{1-x}$
100	*FRcorpus*	93.68%	92.50%	94.08%	93.77%
	ENcorpus	93.67%	92.35%	93.92%	93.77%
300	*FRcorpus*	94.39%	93.83%	94.87%	94.92%
	ENcorpus	94.36%	93.10%	94.67%	94.77%
700	*FRcorpus*	94.59%	94.31%	95.09%	95.26%
	ENcorpus	94.58%	93.81%	94.93%	95.17%

Table 8
Accuracies, for different tolerances, of the standard HMM segmentation and the segmentation obtained by *optimal fusion by soft supervision* when the same training database of size 300 is used for the scoring and the training of the models needed to create *HMMSeg*, *RefinedHMMSeg* : $AlgSize = CombSize = 300$. The different tolerances used to compute the segmentation accuracy are 10, 20, 50 and 80 ms.

		10 ms	20 ms	50 ms	80 ms
Optimal fusion	*FRcorpus*	79.90%	94.92%	99.47%	99.90%
	ENcorpus	81.71%	94.77%	99.43%	99.87%
HMM segmentation	*FRcorpus*	67.12%	88.53%	97.21%	98.92%
	ENcorpus	66.16%	87.77%	97.44%	99.43%

better accuracy improve the overall quality of synthetic speech? This question can be answered by performing objective or, preferably, subjective tests. Subjective tests are preferable because they are based on direct ratings by human listeners and, thus, are often regarded as more reliable than objective ones. For synthesis systems, several subjective tests are available. In such tests, human

subjects are asked to listen to speech signals and rate them according to the categories chosen for the subjective test. The Mean Opinion Score (MOS) [12] is the most widely used subjective method. It is an Absolute Category Rating (ACR) procedure in which listeners are asked to rate the quality of each utterance with a score belonging to $\{1, 2, 3, 4, 5\}$, where 1 and 5 correspond to bad and excellent speech quality respectively.

In this paper, the MOS test aims to assess the impact of the segmentation on the synthetic speech quality. The principle is to build acoustic inventories using the segmentations obtained by the different candidate algorithms and then to use these acoustic inventories to produce test utterances to be compared. For this test, we will use French and English synthesized utterances produced by the corpus-based speech synthesis system *Baratinoo* developed by France Telecom.

This system needs a large database of segmented and labelled diphones. A diphone is defined as a unit that extends from the middle of one phone steady zone to the middle of the next phone steady zone. At this stage, the reader may wonder whether, instead of segmenting the database into phonemes before building the diphones, seeking the middle-states of the phonemes so as to directly construct the diphones would not be more efficient. However, to our current knowledge, determining the steady-state of a phoneme is not an easier task than finding a boundary between two phonemes. This is why the major providers of speech synthesis systems approximate a diphone as the unit that begins from the middle of a phoneme to the middle of the following one. An important exception to this practice concerns plosives, for which it is usual to put the diphone mark before the burst occurs.

Ideally, it would be interesting to make an exhaustive comparison of the algorithms developed in this paper. This would imply comparing the following segmentations: manual segmentation, standard HMM segmentation, *HMMSeg*, *RefinedHMMSeg*, *BrandtSeg*, as well as the eight segmentations produced by the proposed fusion methods. However, bearing in mind that a minimum of 20 test utterrances is necessary for a MOS test, evaluating the whole set of segmentation results would be impractical due to the too large number of stimuli needed. Instead, we restrict ourselves to the evaluation of the three following segmentation methods: manual segmentation, standard HMM segmentation, which is a completely automatic method, and the proposed *fusion by optimal supervision* method, which can be seen as a semi-automatic segmentation procedure. Thus, the purpose of our test is to evaluate the best proposed fusion method (with respect to the objective results presented in section 5.1) in comparison to the manual and the fully automatic segmentation methods.

In order to evaluate the impact of the selected segmentation algorithms on the quality of synthetic speech, we must select the test stimuli very carefully.

Indeed, given the limited set of utterances that will be used for this test, we must make sure that the stimuli used contain speech units for which significantly different segmentations have been obtained. Stated another way, we must select synthetic utterances containing units for which artefacts due to segmentation errors are likely to occur. For that purpose, we propose the following scheme for the generation of the test stimuli.

First we collect 2000 sentences from books of the "Gutenberg" project. The Gutenberg project was started in 1971 and consists of a large electronic library of nearly 17000 books that are freely downloadable. These 2000 sentences are synthesized via the acoustic dictionary derived from standard HMM segmentation. We then count the number of segmentation errors for each synthesized sentence and select the 20 sentences with the largest numbers of errors. It must be noted that during this selection process, the utterances for which a segmentation error occurs on pauses and silences are excluded for two reasons. On the one hand, standard HMM segmentation performs poorly on silences and pauses. On the other hand, our purpose is to consider the largest variety of HMM segmentation errors. Thus, if we took into account the errors on silences and pauses, we might select sentences where most errors are due to silences.

Of course, this experimental protocol introduces some bias. In particular, it concerns only contexts, except pauses and silences within phonemes, where standard HMM segmentation performs poorly. Therefore, it would also be interesting to analyse the behaviour of *optimal fusion by soft supervision* when standard HMM segmentation performs well. This complementary experiment has not yet been carried out. It should be part of the general (and, thus, more exhaustive) MOS test to carry out in future work. Hence, the following experimental results must be regarded as preliminary ones.

By successively using each of the three diphone acoustic dictionaries described above, the synthesis of these 20 sentences via *Baratinoo* provides the 60 sentences that the human listeners will be asked to score. Note that all the listeners are native speakers and naive and a training phase with 5 sentences is performed before the test. This training phase allows the listeners to have a good idea about the quality of the synthetic voice in order to use the whole range of marks appropriately. The results for the French and the English voices and each segmentation are given in table 9. Each value in the fourth column of this table represents the average mark of the synthesized voice quality calculated on the whole set of sentences and listeners.

Considering first the results for the French voice, the differences in score are statistically significant in the sense that, for a significance level of 0.05, an analysis of variance (ANOVA) test on the 16 sequences of 20 scores gives a rather small p-value equal to 0.005. For the French voice, the quality of the

Table 9
Results of the MOS test for the French and the English voices

		Number of subjects	*Score*	*Standard deviation*
French	HMM segmentation	16	2.86	0.41
	Soft fusion		3.15	0.37
	Manual segmentation		3.35	0.4
English	HMM segmentation	11	3.04	0.37
	Soft fusion		3.13	0.41
	Manual segmentation		3.06	0.44

synthetic voice obtained with a database segmented by the *optimal fusion by soft supervision* method can actually be regarded as better than the quality obtained by using standard HMM segmentation and as very close to the quality obtained with manual segmentation.

If we now consider the results concerning the English voice, an ANOVA test on the available 11 sequences of 20 scores leads to a p-value equal to 0.9431. Therefore, the means of our 11 sequences cannot be regarded as statistically different. A possible explanation is the following. The so-called manual segmentation results from the manual correction of the segmentation errors made by the standard HMM algorithm. Although this manual correction is performed by several native experts, the corrected segmentation of the English corpus might still contain errors that bias the experimental results.

6 Conclusion and extensions

In this paper, we have proposed a general fusion approach which makes it possible to combine the output of several automatic segmentation algorithms. The idea of this approach was based on the fact that the segmentation algorithms used behave differently according to the phonetic transition considered. To evaluate the performance of this approach, we have proposed to combine the segmentations marks produced by three methods: HMM segmentation, refinement by boundary model and Brandt's GLR method. In this respect, we have proposed and tested several fusion methods.

From a more general point of view, combining several segmentations seems to be a good solution for segmenting large corpora. The accuracy improves: for instance, *optimal fusion by soft supervision* reduces by 60% the number of errors made by standard HMM segmentation. However, as far as TTS synthesis

applications are concerned, more complete and general subjective tests than those proposed above must be carried out. If general subjective tests showed that the synthetic speech quality obtained with a corpus segmented by the *optimal fusion by soft supervision* method is actually as good as the quality obtained with a manually-segmented corpus, *optimal fusion by soft supervision* would represent a real alternative to the manual segmentation process.

The authors' feeling is that *optimal fusion by soft supervision* is very promising since this approach is flexible and can certainly be improved. Indeed, the approach proposed in section 2 and summarized in figure 1 is a general framework and many other types of score, many other weighting functions and different criteria for the mark selection can be proposed and tested. For example, the use of polynomials as weighting functions could be studied. In this paper, the algorithms are scored by their accuracy rates at 20 ms because a deviation of at most 20 ms is considered to be an acceptable upper limit for guaranteeing good quality of synthesized voice; however, the standard deviation of the segmentation error at 20 ms could also be a relevant type of score. This approach can also involve other segmentations in addition to or instead of those studied above. In order to obtain good performance measurements, we recommend the following: the segmentations that are to be combined should contain no insertion and no omission; the segmentation methods should perform differently depending on the type of transition classes so as to guarantee some complementarity. For instance, *optimal fusion by soft supervision* does not use the standard HMM segmentation since the latter is not really complementary to and does not perform as well as the refinement method by boundary model. On the other hand, the fusion approach might apply to algorithms such as those presented in [1,8,15,16].

In order to obtain better accuracy, attention should be given to the types of boundary that still cause many segmentation errors so as to develop some processing dedicated to them. We highlight such types of boundary as follows. Given a pair of phonetic classes and thus a type of boundary, we compute the number of segmentation errors at a tolerance of 20 ms; we also compute the ratio between this number of errors and the total number of boundaries of this type. The results are presented in tables 10 and 11 concerning *FRcorpus* and *ENcorpus* respectively. These results were obtained on the basis of the segmentation achieved by *optimal fusion by soft supervision* when $(CombSize, AlgSize) = (300, 300)$. From these figures, we note that most errors are made in detecting a transition between a phonetic class and the classes SIL (silence) and SP (pause). To convince the reader that decreasing the number of errors for these pairs of classes is important, we compute the accuracy rate at 20 ms by using manual segmentation to correct all the errors between any class and SIL. We reach an accuracy rate of 95.70% at 20 ms for *FRcorpus* and of 95.47% for *ENcorpus*. By removing all the errors between any class and SP, we arrive at 96.20% for *FRcorpus* and 95.73% for

ENcorpus. In terms of accuracy, these values clearly suggest using very accurate speech/silence detection. In terms of synthetic speech quality, it would be relevant to evaluate the gain provided by such speech/silence detection.

As clearly specified in the introduction, this study has been performed in the case of corpora whose phonetic transcriptions contain very few or no errors. However, for many applications, the exactness of the transcription cannot be guaranteed and can be a crucial problem. Another focus of interest is thus to evaluate *optimal fusion by soft supervision* when the phonetic transcription contains errors. Some work is in progress on this topic. The reader can also refer to [5] and [10], where several solutions are proposed to iteratively correct erroneous transcriptions.

It would also be interesting to assess *optimal fusion by soft supervision* for the segmentation of large corpora dedicated to other applications such as speech recognition. This would make it possible to verify the robustness of the fusion approach to uncontrolled or variable recording conditions and to noisy signals.

Acknowledgement

The authors are very grateful to the reviewers for their numerous and insightful remarks, comments and suggestions. They also thank Toufic Chmayssani for his contribution during his summer study.

References

[1] J. Adell and A. Bonafonte. Towards phone segmentation for concatenative speech synthesis. *Proceedings of the 5th ISCA Workshop on Speech Synthesis*, pages 139–144, June 2004.

[2] R. André-Obrecht. A new statistical approach for the automatic segmentation of continous speech signals. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 36, pages 29–40, January 1988.

[3] A.V. Brandt. Detecting and estimating parameters jumps using ladder algorithms and likelihood ratio test. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1983)*, pages 1017–1020, November 1983.

[4] F. Brugnara, D. Falavigna, and M. Omologo. Automatic segmentation and labelling of speech based on Hidden Markov Models. In *Speech Communication*, volume 12, pages 357–370, August 1993.

[5] S. Jarifi. *Segmentation automatique de corpus de parole continue dédiés à la synthèse vocale.* PhD thesis, École Nationale Supérieure Des Télécommunications de Bretagne and University of Rennes I, 2007.

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

[7] S. Jarifi, D. Pastor, and O. Rosec. Cooperation between global and local methods for the automatic segmentation of speech synthesis corpora. *9th International Conference on Spoken Language Processing (ICSLP 2006)*, September 2006.

[8] Y. J. Kim and A. Conkie. Automatic segmentation combining an HMM-based approach and spectral boundary correction. *7th International Conference on Spoken Language Processing (ICSLP 2002)*, pages 145–148, September 2002.

[9] J. Matousek, D. Tihelka, and J. Psutka. Automatic segmentation for czech concatenative speech synthesis using statistical approach with boundary-specific correction. *8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pages 301–304, September 2003.

[10] S. Nefti. *Segmentation automatique de la parole en phones. Correction d'étiquetage par l'introduction de mesures de confiance.* PhD thesis, University of Rennes I, 2004.

[11] J. J. Odell. *The use of context in large vocabulary speech recognition.* PhD thesis, The university of Cambridge, 1995.

[12] ITU-T Recommendation P.800.1. Mean opinion score (MOS) terminology. 2003.

[13] S. S. Park and N. S. Kim. Automatic speech segmentation based on boundary-type candidate selection. In *IEEE Signal Processing Letters*, volume 13, pages 640–643, September 2006.

[14] S. S. Park, J. W. Shin, and N. S. Kim. Automatic speech segmentation with multiple statistical models. *9th International Conference on Spoken Language Processing (ICSLP 2006)*, September 2006.

[15] D. Torre Toledano, M. A. Rodríguez Crespo, and J.G. Escalada Sardina. Trying to mimic human segmentation of speech using HMM and fuzzy logic post-correction rules. *Third ESCA/COSCOSDA International Workshop on Speech Synthesis*, pages 26–29, November 1998.

[16] D. Torre Toledano, L. A. Hernández Gómez, and L. Villarubia Grande. Automatic phonetic segmentation. In *IEEE Transactions on Speech and Audio processing*, volume 11, pages 617–625, November 2003.

[17] L. Wang, Y. Zhao, M. Chu, J. Zhou, and Z. Cao. Refining Segmental Boundaries for TTS Database Using Fine Contextual-Dependent Boundary Models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, volume I, pages 641–644, May 2004.

[18] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, and J. Odell. The HTK Book for HTK V 3.2.1. 2002.

Table 10

Numbers of segmentation errors at a tolerance of 20 ms and corresponding error rates for each given pair of French phonetic classes. The error rate is defined as the ratio between the number of segmentation errors and the total number of boundaries available for the pair of classes under consideration. The French phonetic classes are: oral vowels (OV), nasal vowels (NV), unvoiced plosives (UVP), voiced plosives (VP), unvoiced fricatives (UVF), voiced fricatives (VF), diphthongs (DIPH), nasal consonants (NC), liquid consonants (LC), semivowels (SV), pauses (SP) and silences (SIL). A class in the first column represents the phonetic class of the phoneme located to the left of a boundary and a class in the first line represents the class of the phoneme which is at the right of this boundary. For instance, if we consider the pair (OV, UVP) of phonetic classes, there were 676 errors in detecting the transitions between these two classes when the phonemes to the left are oral vowels and the phonemes to the right are unvoiced stops. The error rate for this pair of classes equals 4.10%. In this table, and in the subsequent one, we emphasize the pairs of classes with large numbers of errors and large error rates. $---$ means that no transition between the pair of classes is available in the corpus.

	OV	*NV*	*DIPH*	*VP*	*UVP*	*VF*	*UVF*	*NC*	*LC*	*SV*	*SP*	*SIL*
OV	639/23.9	102/13.8	12/32.4	58/0.79	676/4.10	70/0.83	34/0.33	35/0.41	1277/6.32	137/16.5	416/29.1	653/29.8
NV	96/15.6	76/42.7	6/25.0	172/6.14	183/3.96	3/0.31	2/0.07	29/2.50	44/5.19	5/29.4	111/14.8	278/26.1
DIPH	17/35.4	2/10.0	1/100	0/0.00	65/26.0	4/3.92	0/0.00	1/2.17	23/25.0	17/44.7	34/64.15	32/46.3
VP	135/1.41	22/1.97	14/10.3	31/28.9	38/39.1	10/5.32	13/12.2	6/5.26	121/6.36	51/10.9	69/82.1	118/75.1
UVP	408/2.60	5/0.18	11/5.58	54/15.34	66/7.50	20/19.2	47/4.97	18/5.64	155/2.88	7/1.20	94/18.8	154/25.3
VF	80/1.11	5/0.31	2/3.70	3/1.38	30/20.0	4/7.14	12/12.90	1/0.39	9/2.33	52/5.86	61/24.90	68/21.6
UVF	2/0.02	0/0.00	0/0.00	14/4.59	85/4.19	7/12.96	43/15.8	12/6.38	34/5.14	17/0.84	51/14.4	51/11.3
NC	201/2.48	62/3.37	14/34.1	43/10.1	40/10.9	5/4.27	4/1.42	38/20.1	21/11.4	141/13.8	50/18.1	82/27.3
LC	482/2.57	52/2.35	68/26.3	38/2.45	63/3.22	15/2.05	22/1.41	43/2.94	112/11.5	80/8.86	237/24.8	401/26.5
SV	785/18.1	284/14.1	2/100	6/8.33	7/14.8	0/0.00	2/5.00	4/10.0	8/21.6	3/100	14/24.6	25/20.5
SP	1/0.07	1/0.29	0/0.00	20/4.59	378/51.4	5/1.77	17/2.64	0/0.00	2/0.40	0/0.00	$---$	$---$
SIL	3/0.17	3/0.60	0/0.00	15/3.93	283/43.5	4/0.93	26/2.69	3/0.73	7/0.42	1/12.5	$---$	$---$

Table 11
Numbers of segmentation errors at a tolerance of 20 ms and corresponding error rates for each given pair of French phonetic classes. The English phonetic classes are: vowels (V), voiced plosives (VP), unvoiced plosives (UVP), voiced fricatives (VF), unvoiced fricatives (UVF), affricates (AF), nasal consonants (NC), liquid consonants (LC), semivowels (SV), pauses (SP) and silences (SIL). This table reads like table 10.

	V	*VP*	*UVP*	*VF*	*UVF*	*AF*	*NC*	*LC*	*SV*	*SP*	*SIL*
V	1325/20.7	116/0.81	366/1.54	99/0.56	215/1.15	58/2.16	319/1.06	2137/16.1	240/7.75	288/25.6	544/26.6
VP	101/0.74	124/18.5	86/12.9	111/8.23	81/4.75	31/23.8	29/3.30	40/1.20	73/6.71	71/19.1	142/17.1
UVP	109/0.48	186/23.4	433/20.2	244/23.9	255/5.89	70/41.1	38/4.74	69/1.34	264/13.6	111/20.6	331/23.8
VF	167/1.04	92/7.41	85/5.64	90/5.32	243/10.9	17/11.9	58/3.26	9/0.86	60/6.83	83/18.5	144/14.6
UVF	682/3.25	56/11.9	408/7.13	42/9.61	117/12.2	17/12.1	45/2.26	21/0.96	197/23.4	100/25.5	174/19.2
AF	23/0.80	12/7.69	14/5.88	5/6.10	46/22.6	7/23.3	11/13.4	1/1.09	11/11.7	14/25.0	25/24.04
NC	397/2.76	85/1.79	66/1.27	34/1.07	32/0.84	3/0.60	97/12.4	112/10.2	137/10.3	160/24.2	472/29.1
LC	1728/8.15	35/2.99	30/3.73	18/2.49	14/1.33	8/8.79	38/13.1	53/24.7	52/14.0	89/33.8	168/32.5
SV	1232/11.3	– – –	– – –	– – –	– – –	– – –	0/0.00	4/44.4	– – –	– – –	– – –
SP	46/3.22	148/39.9	46/16.6	66/26.1	63/8.39	0/0.00	15/6.76	21/14.3	24/6.67	– – –	– – –
SIL	18/0.71	183/21.8	86/20.8	138/8.48	128/9.36	3/3.45	23/5.03	13/8.72	32/3.59	– – –	– – –