BRANDT'S GLR METHOD & REFINED HMM SEGMENTATION FOR TTS SYNTHESIS APPLICATION

Safaa Jarifi, Dominique Pastor*, Olivier Rosec***

* Ecole Nationale Supérieure des
Télécommunications de Bretagne,
Technopôle Brest Iroise, 29238 Brest Cedex, France
safaa.jarifi, dominique.pastor@enst-bretagne.fr

** France Telecom, R&D Division TECH/SSTP/VMI
2, avenue Pierre Marzin, 22307 Lannion Cedex
olivier.rosec@rd.francetelecom.com

ABSTRACT

In comparison with standard HMM (Hidden Markov Model) with forced alignment, this paper discusses two automatic segmentation algorithms from different points of view: the probabilities of insertion and omission, and the accuracy. The first algorithm, hereafter named the refined HMM algorithm, aims at refining the segmentation performed by standard HMM via a GMM (Gaussian Mixture Model) of each boundary. The second is the Brandt's GLR (Generalized Likelihood Ratio) method. Its goal is to detect signal discontinuities. Provided that the sequence of speech units is known, the experimental results presented in this paper suggest in combining the refined HMM algorithm with Brandt's GLR method and other algorithms adapted to the detection of boundaries between known acoustic classes.

1. INTRODUCTION

The objective of this paper is the segmentation of large acoustic databases for application to corpus-based speech synthesis. Segmenting spontaneous and continuous speech signals is still an issue. Because the sequence of speech units is known, automatic segmentation is usually performed by HMM's with forced alignment. This procedure shows reasonable results. However, in order to guarantee a good quality of a synthetic voice, the outcome of the segmentation stage must still be verified manually, which is a difficult task. Thus, it is still important to design a segmentation algorithm capable of providing segmentation marks close to handmade ones with the smallest possible number of errors.

With respect to the foregoing, the purpose of this paper is to analyze and compare the accuracies of three segmentation algorithms on a large French corpus. The first one is the standard HMM with forced alignment as described above. The second one is the refined HMM, originally proposed in [7] for segmenting a Chinese corpus. These two methods lead to a number of segmentation marks that corresponds to the number of phonemes in the input speech unit sequence. Thus they do not yield omissions nor insertions. The third algorithm is Brandt's GLR method ([2]) which aims at detecting discontinuities of speech signals without any further knowledge upon the phonetic sequence. As this algorithm is linguistically unconstrained, it makes insertions and omissions.

From this comparison, we derive an approach aimed at improving the accuracy of the refined HMM so as to tend to some reasonable performance objective introduced below.

The paper is organized as follows. In the next two sections, we briefly present the refined HMM and Brandt's GLR method. After introducing how performance measurements are computed, section 4 presents and discusses experimental results. On the basis of these results, we suggest in section 5 an approach for improving the refined HMM.

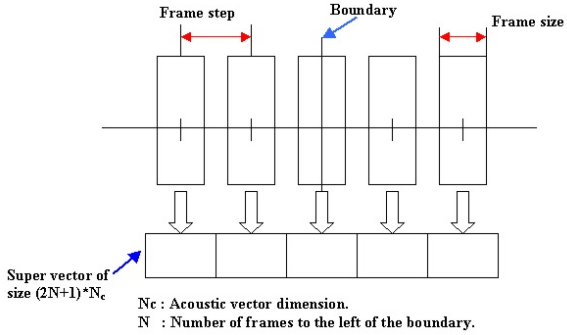

Figure 1: Example of a super vector for a particular boundary

2. THE REFINED HMM ALGORITHM

The main idea of this method is to train a GMM model for each HMM boundary with super vectors that are extracted using frames around the boundaries ([7, 5]). This method is carried out in three steps:

1. Achieve an HMM segmentation with forced alignment. Thus, the number of segmentation marks equals that of the handmade segmentation.
2. For a small database that is manually segmented, create a super vector for each boundary of this database. This vector is obtained by putting together acoustic vectors for frames near the boundary. In Fig. 1, we illustrate a super vector with $(2N+1)$ frames.
   Each boundary B depends on the phoneme X to the left of it and on the phoneme Y to its right. We denote this boundary by $X-B+Y$ as proposed in [8] (see Fig. 2). If we model each possible pseudo triphone by a GMM, the models are badly trained. This is because the number of labeled data is limited in practice. Therefore, a classification and regression tree (CART) is used in order to cluster pseudo-triphones into a reduced number of classes; this cart is a recursive binary tree where each node corresponds to a phonemic or linguistic question. Then a GMM is trained for each leaf node on the CART.
3. Given a labeled sentence and its segmentation, try to refine each boundary of every segment. In this respect, for each frame in a certain vicinity of a given HMM boundary, compute the likelihood that this frame contains the actual boundary. The likelihood that the true boundary

THIS STUDY IS SUPPORTED BY THE R&D DIVISION OF FRANCE TELECOM.

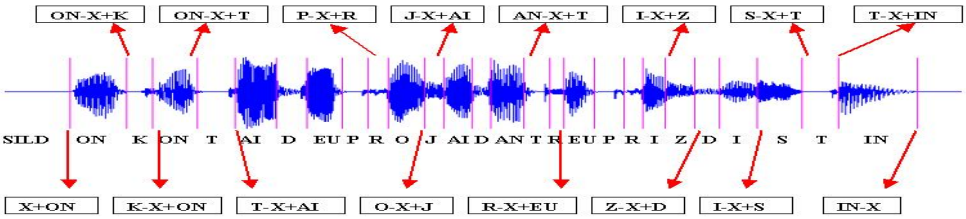

Figure 2: Examples of pseudo-triphones

lies within a given frame is then computed as follows. We form a super vector centered on the current frame. Since this super vector is assumed to represent a pseudo-triphone, we determine the corresponding leaf node in the CART. We then compute the likelihood of the GMM model associated with this leaf node. Finally, the optimal boundary is then assumed to be in the frame that has the maximum likelihood.

3. BRANDT'S GLR METHOD

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 $= (a_1, \ldots, a_p, \sigma)$. Let w_0 be some window of n samples and $_0$ the corresponding parameter vector. Brandt attempts in [2] to decide whether w_0 should be split in 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 $_1$ and $_2$ of w_1 and w_2 respectively. Brandt's GLR method decides that such a jump has occurred by comparing: $D_n(r) = n log\hat{\sigma}_0 - r log\hat{\sigma}_1 - (n-r) log\hat{\sigma}_2$ 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 $_1$ and $_2$. Thus, the change instant corresponds to $arg(max_r(D_n(r)) \geq \lambda)$.

A direct implementation of this method is computationally expensive. Thus, we use the sub-optimal version recommended in [7]. In particular, the length of w_2 is fixed to a predefined value L. For further details, the reader can refer to [2, 3].

4. EXPERIMENTAL RESULTS AND DISCUSSION

Our purpose is to analyze the behavior of the algorithms presented above. We do so by computing performance measurements on the basis of experiments. We start by describing the criteria used to evaluate the performance of each algorithm. These performance criteria are the probabilities of insertion and omission and the accuracy. On the basis of [7], we then adjust the refined HMM parameters for application to a French corpus. For this tuning, only the accuracy is needed because the refined HMM yields no insertions and no omissions. For Brandt's GLR method, we compute the probabilities of insertion and omission. The last subsection compares the accuracies of the refined HMM, the standard HMM and Brandt's GLR method.

4.1 Probabilities of insertion and omission, accuracies

The quantities described in this section allow us to measure the performance of any segmentation algorithm. We propose the subsequent definitions especially in order to take into account insertions and omissions of algorithms such as Brandt's GLR method.

We start by briefly describing how to locate insertions and omissions of a segmentation algorithm so as to compute the probabilities of insertion and omission of this same algorithm.

Let $U = \{U_1, U_2, \ldots, U_n\}$ and $V = \{V_1, V_2, \ldots, V_p\}$ be the time instants of the segmentation marks obtained respectively by an automatic algorithm and by a manual procedure (hereon referred to as the reference segmentation). For each U_j, a correspondence is done with the reference segmentation by determining the time instant V_{k_j} which is closest to U_j. This way, a sequence $V_U = \{V_{k_1}, \ldots, V_{k_n}\}$ is built in order to compare both segmentation.

The reader can easily verify the following facts: the omissions are the marks V_ℓ, $\ell \in \{1, \ldots, p\}$, that are not in the list V_U; if V_U contains m times the same mark, the number of insertions n_i equals $m-1$. In the latter case, if V_ℓ is the mark contained m times in the list V_U, the nearest mark to V_ℓ in U is considered as a non-insertion and, thence, the $m-1$ other marks are regarded as insertions.

We define the ratios $P_i = \frac{n_i}{p+n_i}$ and $P_o = \frac{n_o}{n+n_o}$. The former can be regarded as the probability of insertion of the segmentation algorithm, whereas the latter can be considered as the probability of omission of this algorithm. Note that $n + n_o = p + n_i$.

The accuracy of any given segmentation is computed as follows. First, locate the insertions as described above and remove them from the list U. For each mark U_j of the resulting list of non-insertions, consider the closest reference mark V_{k_j}. If the distance $|V_{k_j} - U_j|$ is less than or equal to a given tolerance value ε, the mark segmentation U_j is said to be correct. Otherwise, it is called an error. The accuracy is then defined as the ratio in percentage of the number of correct marks to $p + n_i$:

$$accuracy = \frac{100}{p+n_i} \sum_{j=1}^{n} I_{[0,\varepsilon]}(|V_{k_j} - U_j|)$$

where $I_{[0,\varepsilon]}(x)$ equals 1 if $x \in [0, \varepsilon]$ and 0 otherwise. The accuracy depends on the number of insertions, the number of omissions and ε. Therefore, an accurate segmentation must make a good compromise between these three parameters.

For instance, a segmentation with many insertions will clearly be characterized by a P_i close to 1 and an accuracy close to 0; similarly, in the case of many omissions, P_o is close to 1 and the accuracy remains small; in contrast with

Table 1: HMM segmentation accuracies

ε	5	10	20	30
accuracy	33.54	59.77	85.24	92.83

the foregoing, an accurate segmentation is such that P_o and P_i are small and the accuracy is close to 1. Further mathematical details regarding these criteria will be given in a forthcoming work.

4.2 Application of the refined HMM algorithm to a French corpus

As mentioned in [7], the method "makes no inherent assumptions for the language or speaker type". So, our purpose is to validate the parameter values exhibited in [7] when we apply the method to a French corpus containing 7350 sentences. We also adjust some parameters in addition to the reference. We proceed by using HTK toolkits [8]. Note that the notion of accuracy used in [7] is a particular case of that employed in this present paper to measure the performance of segmentation algorithms.

The training parameters of this algorithm are: the training size, the number of frames $(2N+1)$, the frame step e, the frame size (set to 20 ms), the number of GMM components (equal to 1, see [7] for details), the acoustic vector dimension N_c (set to 39 : 12 MFCCs+ energy+ 13 first order deviations+ 13 secondary deviations), the stopping criteria of the CART. These are the minimum number of leaf node instances, denoted by MTI, and the value T of the log likelihood to exceed, in order to consider a question as an actual node of the CART. For every HMM mark of the database, the refined HMM boundary is searched within a 60 ms interval centered on this mark with a search step fixed to 5 ms. The accuracies presented below are calculated for ε equal to $5, 10, 20$ and 30 ms. They must be compared to those obtained by standard HMM and given in table 1.

In table 2, we point out a suitable pair (T, MTI) that yields a good level of accuracy when the tolerance varies. The results of table 2 are obtained by following [7] and thus fixing the pair (N, e) to $(2, 30)$ and the training size to 300. We observe that T hardly influences the accuracy. We fix T to 100. The results also show that the smaller the MTI, the better the accuracy.

Table 3 illustrates the contribution of the training database size. We again fix the pair (N, e) to $(2, 30)$. The pair (T, MTI) is set to $(100, 10)$ according to the foregoing. On the basis of these results, we choose a size of 300 sentences in order to limit the number of manually labeled boundaries. This value is close to that obtained for a Chinese corpus in [7]

With table 4, we verify that $(N, e) = (2, 30)$ still remains a suitable choice for dealing with our French corpus. In fact, this choice seems to be a reasonable trade-off between the following two facts. On the one hand, the total duration corresponding to the super vector must be long enough to involve as much information as possible concerning the transition between the two phonemes around the boundary. On the other hand, if this duration is too long, the super vector will take into account information not directly linked to the boundary itself.

Summarizing, the refined HMM algorithm performs well when applied to a French corpus (6% accuracy gain when the tolerance is 10 ms) with $MTI = 10$, $N = 2$, $e = 30$ ms and a training size equal to 300.

Table 2: Accuracy vs (T, MTI)

T	ε	$MTI = 10$	$MTI = 20$	$MTI = 40$	$MTI = 100$
20	5	38.49	38.08	37.48	35.13
	10	64.55	64.29	62.94	60.13
	20	87.91	87.74	87.05	85.16
	30	94.46	94.44	93.93	93.45
100	5	38.55	38.12	37.48	35.13
	10	64.63	64.28	62.94	60.13
	20	87.98	87.73	87.04	85.16
	30	94.50	94.46	93.93	93.45
350	5	38.38	37.90	37.49	35.13
	10	64.32	64.01	62.94	60.13
	20	87.93	87.51	87.11	85.16
	30	94.35	94.42	93.93	93.45

Table 3: Accuracy vs training set size

Training size	$\varepsilon = 5$	$\varepsilon = 10$	$\varepsilon = 20$	$\varepsilon = 30$
200	38.27	63.97	87.63	94.31
300	38.55	64.63	87.98	94.49
600	41.26	67.10	88.78	95.03
800	41.42	67.76	88.87	95.15

4.3 Insertions and omissions with Brandt's GLR method

For Brandt's GLR method, the model order p and the window length L are chosen equal to 16 and 20 ms respectively. The threshold λ is set to 30 as in [3].

We found that the probabilities of insertion and omission are equal to 0.6 and 0.1 respectively. This high insertion probability is explained by the oscillatory behavior of the GLR function. Note that the insertion probability is divided by two when the Brandt'GLR method is merged with a silence/speech detection [1]. Indeed, many insertions are located in silences (see Figure 3).

4.4 Comparing the accuracies of the refined HMM and Brandt's GLR methods

The results presented in this section are obtained by averaging the accuracies using a cross-validation procedure where each test set contains 1200 sentences. For the refined HMM, the training set contains 300 sentences for each test. Figure 4 depicts the results obtained with the three methods. Note that for Brandt's GLR method three curves are plotted. The "Brandt's GLR" curve represents the accuracies of this algorithm taking into account its insertions and omissions; the "Brandt's GLR and speech/silence detection" curve displays the accuracies when Brandt's GLR method is merged with the speech/silence detection algorithm already used in [1]; the "ideal Brandt's GLR" curve is obtained by computing accuracies after removing the insertions and omissions via the procedure of subsection 4.1.

According to the results illustrated by these curves, we can make the following remarks: the refined HMM is more accurate than the standard HMM segmentation; the Brandt's GLR method is not accurate because of its large number

Table 4: Accuracies for different values of (N, e)

e	N	$\varepsilon = 5$	$\varepsilon = 10$	$\varepsilon = 20$	$\varepsilon = 30$
0	0	33.95	58.70	83.70	92.27
10	2	39.01	64.42	86.13	93.41
30	2	38.55	64.63	87.98	94.49
30	3	37.16	63.83	88.06	94.58

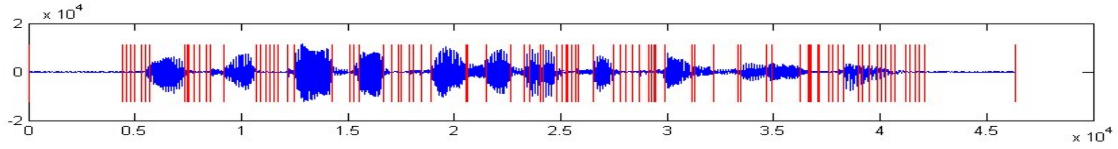

Figure 3: Brandt's GLR segmentation

of insertions even when this number of insertions is significantly reduced; the accuracy of the "ideal" Brandt segmentation is itself significantly better than that of the refined HMM. Therefore, the marks produced by Brandt's GLR method that are not insertions are more accurate than those of the refined HMM.

5. SUGGESTIONS FOR IMPROVING THE REFINED HMM

As mentioned above, Brandt's GLR method is not accurate with respect to the accuracy measure given in this paper. It is however worth noticing that our assessment is too strict regarding Brandt's GLR method because the definition of accuracy does not take into account the distribution of segmentation marks. In fact, insertions of Brandt's GLR method remain located in silence regions and in the vicinity of handmade marks. It also turns out that high values of the GLR correspond to actual discontinuities of speech signals. Hence, we suggest to improve the accuracy of the refined HMM by proceeding as follows.

Given a segmentation mark performed by the refined HMM algorithm, compute the GLR of this initial mark. If the GLR exceeds some threshold (to define), keep this mark. Otherwise, consider that neither the refined HMM nor Brandt's GLR method are reliable. Therefore, choose a segmentation algorithm adapted to the acoustic classes of the segments to identify. These classes are known thanks to the sequence of speech units we have. Basic examples of such algorithms are speech/silence detection and voiced/unvoiced detection. More sophisticated techniques are described in the literature on the topic (see for instance [6]).

Since Brandt's algorithm segmentation marks that are not insertions are accurate, it can easily be thought up that a reasonable performance objective for the approach described above is to achieve accuracies better than those of the refined HMM and closer to those of the "ideal" Brandt segmentation.

6. CONCLUSION

This paper analyzed the performance measurements of two automatic segmentation algorithms: the refined HMM algorithm and Brandt's GLR method. To carry out our assessment, we used three performance criteria, namely, the probability of insertion, the probability of omission and the accuracy. From the experimental results, we derived an approach for improving the refined HMM segmentation algorithm as well as a reasonable performance target (the accuracy of the "ideal" Brandt's GLR method). With respect to the contents of this paper, the purpose of our forthcoming work will be twofold. On the one hand, we will complete our analysis concerning the performance criteria. In particular, we will study how the distribution of marks can be taken into account or merged with the criteria employed in this paper. On the other hand, we will detail, implement and test the method proposed

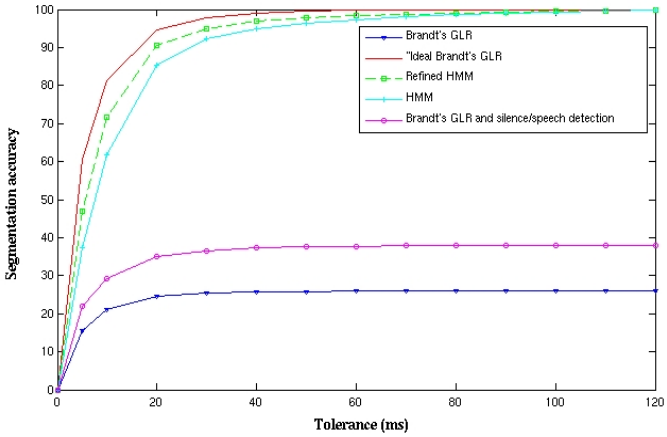

Figure 4: Segmentation accuracy for 900 sentences

in the foregoing section. Our intention is to approach the accuracy of the "ideal" Brandt segmentation.

REFERENCES

[1] S. Jarifi, D. Pastor, O. Rosec, *Jump and silence/speech detection for automatic continuous speech segmentation*, International Symposium on Image/Video Communications, Brest, France, 2004.

[2] R.A. Obrecht, *Automatic segmentation of continuous speech signals*, Proc, ICASSP, pp.2275-2278, Tokyo, Japan, 1986.

[3] R.A. Obrecht, *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.

[4] M. Seck, *Détection de ruptures et suivi de classe de sons pour l'indexation sonore*, Phd thesis, Université de Rennes 1, France, 2001.

[5] A. Sethy and S. Narayanan, *Refined Speech Segmentation for Concatenative Speech Synthesis*, Proc, ICSLP, pp.149-152, Denver, USA, 2002.

[6] D.T. Toledano and L.A. Hernández Gómez and L. Villarubia Grande, *Automatic Phonetic Segmentation*, IEEE Transactions on Speech and Audio Processing, Vol.11(6), pp.617-625, 2003.

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

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