ARCHIVES OF ACOUSTICS
32, 1, 127–134 (2007)

DYNAMIC PROGRAMMING METHOD FOR FINE-TUNING THE BOUNDARY POINTS IN AUTOMATIC SEGMENTATION OF SPEECH

Marcin SZYMAŃSKI, Stefan GROCHOLEWSKI

Poznań University of Technology
Institute of Computing Science
Piotrowo 3a, 60-965 Poznań, Poland
e-mail: mszymanski@cs.put.poznan.pl

(*received October 16, 2006; accepted December 15, 2006*)

The important element of today's speech systems is the set of recorded wavefiles annotated by a sequence of phonemes and boundary time-points. As the manual segmentation of speech is a very laborious task, there is the need for automatic segmentation algorithms. However, it was observed that common HMM-based methods are prone to systematical errors. Thus, some boundary refinement approaches were introduced. In this paper we combine two sources of information: boundary error distribution and an acoustic observation distribution, in a single dynamic programming approach.

Keywords: speech segmentation, dynamic programming.

1. Introduction

In the process of constructing speech recognition and synthesis systems it is essential that the proper set of prerecorded utterances should be available. It should additionally contain full annotation, including the sequence of phoneme labels and subsequent unit durations (which identify the transition time points). The accuracy of phoneme boundaries may not be crucial in case of recognition systems, however errors in segmentation seriously affect the quality of the obtained *synthesis* systems.

Manual segmentation of speech is a very labor-intensive process, moreover it should be performed by an expert (usually in phonetics) and it is prone to inconsistencies. The simplest idea is to implement an algorithm which will do this task automatically. We assume that the phone sequence is known in this task (in contrast to the phoneme recognition problem), either directly or in the form of a convertible orthographic transcription. Obviously, the obtained automatic boundary points will not be faultless.

The basic algorithmic solution of the segmentation is to run a HMM recognizer in *forced alignment* mode. The segmentation can now be considered as a special case of

recognition, where the word- and model-net are simple concatenation of units, corresponding to the imposed phonetic transcription of an utterance. The standard recognition for HHMs involves finding the most likely state sequence via the dynamic Viterbi decoding:

$$\hat{s}_1^T = \arg\max_{s_1^T} p\left(y_1^T | s_1^T\right) p\left(s_1^T\right),$$

where s_1^T denotes a state sequence and T is a length of the observation sequence y_1^T.

For segmentation, where the sequence s_1^T is known *a priori*, the model boundaries are returned as a supplemental result of the Viterbi decoding or can be obtained in a *backward-pass*. This approach has been already well studied (see e.g. [8]).

The important limitation of HMM's application to speech processing is its ignoring the probability densities (pdf's) of phoneme duration. Since the state transition probability in standard HMM is represented by one constant value, the state duration has an implicit geometric probability density, which most probably is inadequate as the duration model. For this reason, the observed phoneme duration is often modeled. In general, this can be viewed as a conversion from the Markov chain approach into the *segment models* [5]. Segment-based recognition involves finding

$$\left(\hat{N}, \hat{a}_1^N\right) = \arg\max_{N, a_1^N} \left\{ \max_{l_1^N} p\left(y_1^T | l_1^N, a_1^N\right) p\left(l_1^N | a_1^N\right) p\left(a_1^N\right) \right\},$$

where a_1^N is a N-length phone sequence and l_1^N denotes a vector of respective segment lengths. As far as the segmentation is concerned, only the model duration sequence (l_1^N) is searched for. We implement this in [6].

The above state-chain based methods are primarily designed for speech recognition, not segmentation, and thus have the following drawbacks:

- the results they return are discretized according to the given frame-rate (usually 10 ms), while in the segmentation task we expect higher precision;
- statistical models of acoustic observations (represented by MFCC vectors) contained in each HMM state are trained on longer fragments of phoneme segments, while models trained on near-boundary observations only seem to be more appropriate for the segmentation task.

Moreover, it was observed that the Viterbi decoding tends to make systematical errors for certain boundary classes, e.g. transitions from speech to silence are often located ca. 20 ms before reference annotation.

For those reasons a refinement stage is required, in which "coarse" boundary points obtained from state-chain method are further fine-tuned. The simple solution, boundary-specific correction (BSC), as proposed in [4], consists in calculating the mean error for each of the 100 boundary classes (the phonetic alphabet was split into 10 clusters) and subtracting such estimate from every transition case in a test set. In [1], the regression-trees (CART) were used instead of an enforced partition into 100 classes; this method was reimplemented in this paper as the baseline of our work.

The BSC/CART method, however, applies the fine-tuning "blindly", i.e. it does not consider the underlying acoustic contents of a wavefile in place of a transition. Thus, in this work, we want to combine two sources of information: the boundary-specific error distribution and a boundary acoustic observation distribution (for now, we simply use MFCC parameters for that purpose). For the reason explained below, we use a dynamic programming method to obtain a final segmentation. In the final extension, the duration distributions are also incorporated into our method.

The rest of this paper is organized as follows: Section 2 introduces the baseline boundary correction algorithms, Sec. 3 introduces the dynamic programming method; in Sec. 4 the experimental results are presented. The paper is concluded in Sec. 5.

2. Boundary-specific correction with regression trees

The boundary-specific statistical correction method (BSC) was introduced in [4]. The average error of the automatic segmentation results compared to manual reference annotation is computed for each type of the boundaries. For the test set, each individual boundary b_i is corrected by shifting the transition point by its boundary-specific mean deviation:

$$\hat{b_i} \;=\; b_i - \mu_{C(b_i)}.$$

In [4], Czech phonemes were divided into 10 clusters, reflecting their phonetic and acoustic features. As each boundary is described by its left and right-hand context, this resulted in a total of 100 types of boundaries.

As an extension of BSC, a regression tree based method (CART) was proposed in [1]. A single binary decision tree is constructed by asking questions on phonetic properties of the left and right-hand phones separated by a particular transition. The tree is applied to the testing data by moving the boundaries by the time found in a specific leaf of the tree.

In our paper, the CART was implemented for Polish, using ca. 40 contextual questions. The tree was trained with a minimum of 70 supporting units in each leaf.

3. Dynamic boundary correction

As it was explained above, the main motivation was to combine two sources of information in one refinement method: the boundary-specific error estimates and boundary acoustic observation distributions.

3.1. Correction ranges

First, it should be noted that while state-chain based systems perform the segmentation "globally", the presented algorithm will perform the optimization considering acoustic data near the fine-tuned boundary only. Thus, in the last stage, to preserve the

boundaries to unrestrictedly migrate along the utterance (ignoring the HMM stage results), the transition points can only be shifted along the adjacent segments (left and right) of the particular boundary. This is equivalent to the assumption that the state-based segmentation stage does not introduce gross errors (a gross error occurs when an automatically obtained boundary passes beyond one of the adjacent segments of a reference manual transcription [3]; this kind of error is considered to be the most harmful one in case of unit-selection synthesis). Obviously, gross errors happen during the state-chain segmentation; however, this assumption was still consistent with the semi-automatic segmentation paradigm we mention below. It should be also noted that the assumption does not mean that the number of gross errors cannot change during the fine-tuning stage.

3.2. Time discretization (tuning precision)

The presented methods are mainly oriented for the use within a unit-selection synthesis. As the segment transition time-points obtained from one of these methods can become concatenation points during the synthesis, we may try to avoid *some* of the negative effects (e.g. cracks) of linking two clips with unmatched soundwave levels by placing the potential transition points on rising zero crossings.

In the ideal case, we would expect an algorithm to render results with high precision, even to the level of one sample. However, such a measurement is unreachable because of asynchronicity of speech. Thus, we decided to spread potential boundary points in 1 to 5 ms intervals.

First, we search all rising zero-crossings throughout the wavefile. To assure that at least one potential boundary point appears within each 5 ms, we also allow a falling zero crossing to be used as a boundary, where necessary. If still no zero crossings are found within 5 ms or more, as many as needed potential transition points are spread regularly along such a period. Finally, wherever three zero-crossing points appear within 1 ms, the middle one is discarded.

The drawback of this approach is that it requires a calculation of MFCC vectors at a few hundred or more potential boundary points in each utterance. As the experiments show, the time required for the calculation exceeds the actual dynamic optimization time. We are going to work on a more effective approach in the future.

3.3. Information sources

3.3.1. Boundary-specific error distribution

Boundary-specific segmentation error information used in this work is, in contrast to the BSC/CART method, not limited to the mean error. The individual errors are collected from the training set (a signed difference between state-chain stage result and a

reference transition point) and clustered with a maximum likelihood criterion, yielding a statistical model of the boundary-specific segmentation error:

$$\ln P_{\mathrm{BSE}}(b_i - t \mid C(b_i)).$$

For modeling the error distribution we use single Gaussian pdf. The example values are demonstrated in Fig. 1.

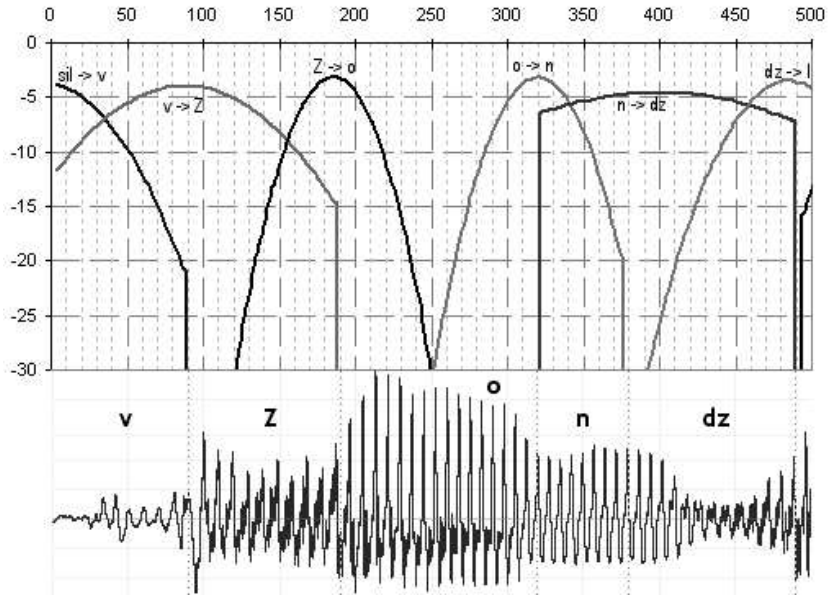

Fig. 1. Log probability of placing subsequent boundaries at a particular moment in time inside a 500 ms fragment of an utterance sample, considering the state-based segmentation results and a boundary-specific error distribution.

3.3.2. Boundary acoustic information

To introduce the acoustic information into the refinement stage, we collect the boundary observations from the training set, as demonstrated in Fig. 2. The MFCC vectors are calculated for two intersecting Hamming windows (plus 4 windows at each side required for the derivative calculation); the window length and frame distance are the same as in the baseline segment model stage, i.e. 25 ms and 10 ms, respectively.

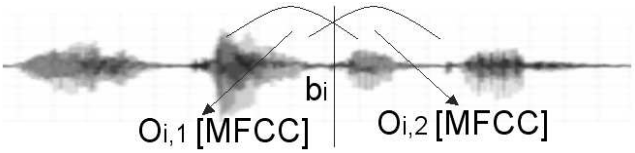

Fig. 2. Boundary acoustic information acquisition.

The above data are clustered according to the context (separate clustering for each of the sides) to form decision trees. The left-side and right-side observation likelihood is combined using the formula:

$$\ln P_{\text{MFCC}}(O_t \mid b_i) = \ln P(O_{t,1} \mid C(b_{i,1})) + \ln P(O_{t,2} \mid C(b_{i,2})).$$

The sample values are demonstrated in Fig. 3.

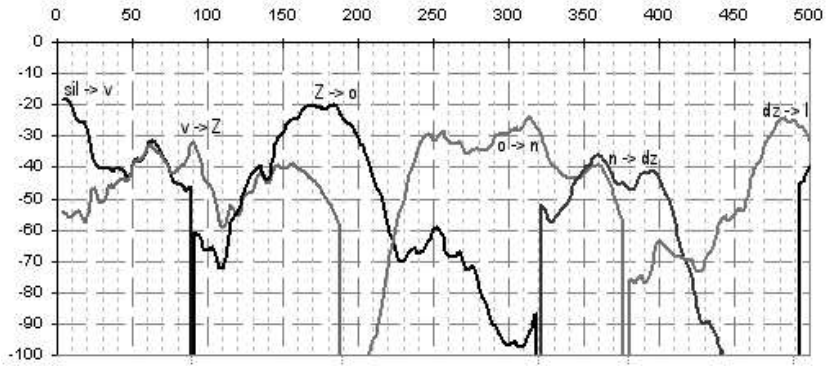

Fig. 3. Log probability of placing boundaries at a particular point inside a fragment of the same utterance sample, based on the acoustic boundary likelihood (limited to the correction range).

3.4. Dynamic programming formulae

The cost of placing a boundary b_i at the potential transition point occurring at time t depends on the sum of two elements: $\ln P_{\text{BSE}}(b_i - t \,|\, C(b_i))$ and $\ln P_{\text{MFCC}}(O_t \,|\, b_i)$. Let us note, however, that a simple, local maximization of such a sum could theoretically lead to "swapping" two consecutive boundary points, which would implicate negative duration segments. For this reason, we propose a dynamic programming algorithm, in which $\phi_{i,t}$ will denote a likelihood of putting a boundary b_i at time t:

$$\begin{aligned}
\phi_{1,t} &= \ln P_{\text{BSE}}(b_i - t \mid C(b_i)) + \ln P_{\text{MFCC}}(O_t \mid b_i), \\
\phi_{i,t} &= \ln P_{\text{BSE}}(b_i - t \mid C(b_i)) + \ln P_{\text{MFCC}}(O_t \mid b_i) \\
&\qquad + \max_{0 \le u < t} (\phi_{i-1,u}).
\end{aligned}$$

The optimal solution is obtained in a backward pass of the dynamic algorithm.

3.5. Duration model version

Finally, we extended the above method by incorporating the duration likelihood of a left-side segment of a examined boundary into the optimization formula:

$$\begin{aligned}
\phi_{i,t} &= \ln P_{\text{BSE}}(b_i - t \,|\, C(b_i)) + \ln P_{\text{MFCC}}(O_t \,|\, b_i) \\
&\qquad + \max_{0 \le u < t} (\phi_{i-1,u} + \ln P_{\text{DUR}}(t - u \mid b_i, 1)).
\end{aligned}$$

We assume that a phrase starts and ends with a silence segments (possibly of zero duration). As silence segments were not subject to the duration modeling, the presented extension does not change the formula for $\phi_{1,t}$.

It may be noted that the dynamic programming approach is required in a duration model version, while in the basic version (Sec. 3.4) it was used only to assure positive duration of subsequent segments.

4. Experimental results

In the experiments we use a part of the Polish Corpora [2] database. It consists of a total of 5 hours of speech, inside 28 folders of 365 separate sentences each, coming from 24 different speakers. Hence, we deal with a speaker-independent segmentation.[(1)] The baseline HMM models were trained for the MFCC target rate of 10 milliseconds, while the manual segmentation was done with a 5 ms precision.

The tests were performed in 7–fold cross-validation, repeated 3 times. The development of the segmentation methods is demonstrated in Table 1. For each of them we calculated the number of gross error and the Root Mean Square Error (RMS). In the last three cases, BSC and DBC methods were applied as a refinement stage to the segmentation result obtained from the segment-model approach (2nd line).

Table 1. Developement of the segmentation method.

Method	% gross errors	RMS error [ms]
Viterbi dec.	0.051	17.75
Segment model	0.032	17.15
BSC/CART	0.031	14.81
DBC	0.031	14.62
DBC+Duration	0.021	13.91

The results show that the largest accuracy boost is obtained by introduction of the CART technique. Dynamic programming extension constitutes a smaller improvement, although incorporating the duration modeling seems to reduce the number of gross errors.

5. Conclusions

We have presented an approach for fine-tuning the transition points obtained from a state-chained based algorithm. This final segmentation stage successfully combines the error distribution information and boundary acoustic observation models in a dynamic

[(1)] It should be noted that the database was specifically designed to contain as much different diphones as possible. Since some sentences were not very common, this might have influenced the statistical models used in this work.

programming algorithm. We have also shown that incorporation of phoneme duration information reduces the number of gross errors, both in the case of baseline and fine-tuning stage. The main drawback of the approach is the time required for the boundary observation acquisition; we think, however, that the calculation time is not crucial in case of the segmentation task.

As far as a fine-tuning methods are concerned, further works include:

- incorporating other feature parameters into the boundary observation models;
- developing a more effective method for calculation of the boundary acoustic likelihood at potential boundary points.

Also, in [7] we presented a semi-automatic approach to the segmentation task, in which we had a human expert manually annotating a small part of a corpus *prior* to the automatic segmentation of the rest of a speech database. In the second stage, the segmentation algorithm considers the enforced expert boundary points and does not change them. For that purpose we also plan to design a confidence measure, that would decide which boundaries should be inserted manually prior to the automatic stages, in order to reduce the number of gross errors. Having the number of gross errors very close to zero allows to make the assumption described in Sec. 3.1.

Acknowledgment

Work supported by the grant KBN No 11C 03827.

References

[1] ADELL J., BONAFONTE A., *Towards phone segmentation for concatenation speech synthesis*, Proc. 5th Speech Synthesis Workshop, pp. 139–144, Pittsburgh 2004.

[2] GROCHOLEWSKI S., *CORPORA – Speech database for Polish diphones*, Proc. Eurospeech'97, pp. 1735–1738, 1997.

[3] KVALE K., *Segmentation and labelling of speech*, Ph.D. Thesis, Inst. for Teleteknikk, Trondheim 1993.

[4] MATOUSEK J., TIHELKA D., PSUTKA J., *Automatic segmentation for Czech concatenative speech synthesis using statistical approach with boundary-specific correction*, Proc. Eurospeech 2003, pp. 301–304, Geneva 2003.

[5] OSTENDORF M., DIGALAKIS V. V., KIMBALL O. A., *From HMM's to segment models: a unified view of stochastic modeling for speech recognition*, IEEE Trans. on Speech and Audio Proc., **4,** 5, September 1996.

[6] SZYMAŃSKI M., GROCHOLEWSKI S., *Implementation of speech segmentation algorithm with statistical duration models* [in Polish], Tuning the Model Parameters, RB–004/05, Poznań Univ. of Technology, Inst. of Computing Science 2005.

[7] SZYMAŃSKI M., GROCHOLEWSKI S., *Semi-automatic segmentation of speech: manual segmentation strategy*, Problem Space Analysis, Proc. CORES'05, Wrocław 2005.

[8] TAYLOR P. A., ISARD S. D., *Automatic phone segmentation*, Proc. Eurospeech, pp. 709–711, Genova 1991.