# Segmentation of Monologues in Audio Books for Building Synthetic Voices

Kishore Prahallad and Alan W. Black

***Abstract*—One of the issues in using audio books for building a synthetic voice is the segmentation of large speech files. The use of the Viterbi algorithm to obtain phone boundaries on large audio files fails primarily because of huge memory requirements. Earlier works have attempted to resolve this problem by using large vocabulary speech recognition system employing restricted dictionary and language model. In this paper, we propose suitable modifications to the Viterbi algorithm and demonstrate its usefulness for segmentation of large speech files in audio books. The utterances obtained from large speech files in audio books are used to build synthetic voices. We show that synthetic voices built from audio books in the public domain have Mel-cepstral distortion scores in the range of 4–7, which is similar to voices built from studio quality recordings such as CMU ARCTIC.**

***Index Terms*—Audio books, forced-alignment, large speech files, text-to-speech (TTS).**

## I. Introduction

Current text-to-speech (TTS) systems use speech databases such as CMU ARCTIC [1]. These speech databases consist of isolated utterances which are short sentences or phrases such as "*He did not rush in.*" and "*It was edged with ice.*." These utterances are selected to optimize the coverage of phones. Such utterances are not semantically related to each other, and possess only one type of intonation, i.e., declarative. Other variants of intonation corresponding to paragraphs and utterances such as wh-questions (*what time is it?*), unfinished statements (*I wanted to…*), yes/no questions (*Are they ready to go?*) and surprise (*What! The plane left already!?*), are typically not captured.

A prosodically rich speech database includes intonation variations; pitch accents which make words perceptually prominent, as in, *I didn't shoot AT him*, *I shot PAST him*; and phrasing patterns which divide an utterance into meaningful chunks for comprehension and naturalness. Development of a prosodicially rich speech database requires a large amount of effort and time. An alternative is to exploit story style monologues in audio books which already encapsulate rich prosody including varied intonation contours, pitch accents, and phrasing patterns. However, there exist several research issues in using audio books for building synthetic voices. A few of them are as follows.

*Segmentation of monologues*: Monologues in audio books are long speech files. The issue in segmentation of large speech files is to align a speech signal (as large as 10 hours or more) with the corresponding text to break the speech signal into utterances corresponding to sentences in text and/or provide phone-level time stamps.

*Detection of mispronunciations*: During the recordings, a speaker might delete or insert at syllable, word, or sentence level and thus the speech signal may not match with the transcription. It is important to detect these mispronunciations using acoustic confidence measures so that the specific regions or the entire utterances can be ignored while building voices.

*Features representing prosody*: Another issue is the identification, extraction and evaluation of representations that characterize the prosodic variations at sub-word, word, sentence, and paragraph level. These include prosodic phrase breaks and emphasis or prominence of specific words during the discourse of a story.

In this paper, we deal with the problem of segmentation of monologues. Typically, segmentation can be accomplished by force-aligning an entire utterance with its text using the Viterbi algorithm. However, such solution fails for utterances longer than a few minutes, since memory requirements of the Viterbi algorithm increase with the length of utterances. Hence, earlier works break long speech files into smaller segments using silence regions as breaking points [2]. These smaller segments are given to an automatic speech recognition (ASR) system to produce hypothesized transcriptions. As the original text of utterances is also available, the search space of ASR is constrained using $n$-grams or finite state transducers based language model [3], [4]. In spite of search space being constrained, the hypothesized transcriptions are not always error-free; especially at the border of small segments where the constraints represented by language models are weak [4], [5]. Apart from practical difficulty in implementing this approach in the context of a TTS system, it strongly implies that a speech recognition system should be readily available before building synthetic voices.

In this paper, we propose an approach based on modifications to the Viterbi algorithm to process long speech files in parts. Our approach differs significantly from the works in [2]–[4], as we do not need a large vocabulary ASR, or employ language models using $n$-grams or finite state transducers to constrain the search space. Since the proposed approach is based on modifications to the Viterbi algorithm, it is suitable for languages (especially for low resource languages), where ASR systems are not readily available. Other applications include highlighting text being read in an audio book.

## II. Viterbi Algorithm (FA-0)

Let $Y = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(T)\}$ be a sequence of observed feature vectors[1] extracted from an utterance of $T$ frames. Let $S = \{1, \ldots, j, \ldots, N\}$ be a state sequence corresponding to the sequence of words in text of the utterance. A forced-alignment technique aligns the feature vectors with the given sequence of words using a set of existing acoustic models.[2] The result is a sequence of states $\{x(1), x(2), \ldots, x(T)\}$ unobserved and hidden so far, corresponding to the observation sequence $Y$. The steps involved in obtaining this unobserved hidden state sequence are as follows.

Let $p(\boldsymbol{y}(t)|x(t) = j)$ denote the emission probability of state $j$ for a feature vector observed at time $t$ and $1 \leq j \leq N$, where $N$ is the total number of states. Let us define $\alpha_t(j) = p(x(t) = j, \boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(t))$. This is the joint probability of being in state $j$ at time $t$, having observed all the acoustic features up to and including time $t$. This joint probability could be computed frame-by-frame using the recursive equation

$$\alpha_t(j) = \sum_i \alpha_{t-1}(i) a_{i,j} p(\boldsymbol{y}(t)|x(t) = j) \qquad (1)$$

Manuscript received March 18, 2010; revised July 06, 2010; accepted September 04, 2010. Date of publication September 30, 2010; date of current version May 13, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gary Geybbae Lee.

K. Prahallad is with the International Institute of Information Technology, Hyderabad 500032, India, and also with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: kishore@iiit.ac.in).

A. W. Black is with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: awb@cs.cmu.edu).

Digital Object Identifier 10.1109/TASL.2010.2081980

[1] Speech signal is divided into frames of 10 ms using a frame shift of 5 ms. Each frame of speech data is passed through a set of Mel-frequency filters to obtain 13 cepstral coefficients.

[2] The acoustic models used to perform segmentation of large audio files are built using about four hours of speech data collected from four CMU ARCTIC speakers (*RMS, BDL, SLT, and CLB*).

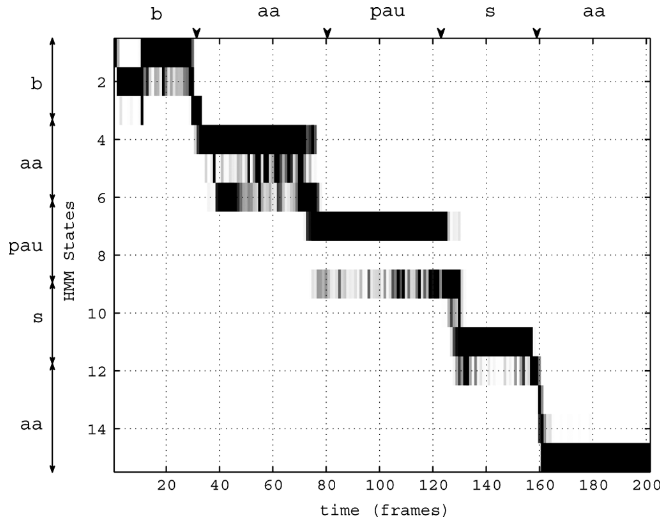
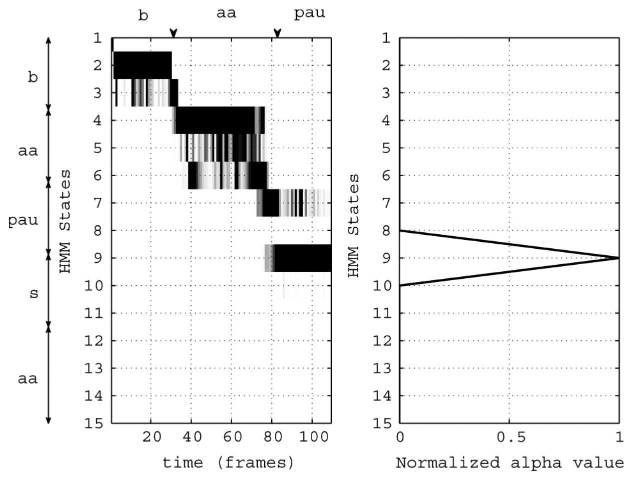


Fig. 1. Alpha matrix obtained for the alignment of feature vectors corresponding to utterance of "$ba\langle pau\rangle sa$" with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/, and /aa/. The markers on the time axis indicate manually labeled phone boundaries.



(a) (b)

Fig. 2. (a) Alpha matrix obtained for the alignment of feature vectors corresponding to utterance of "$ba\langle pau\rangle$" with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/, and /aa/. (b) Alpha values of all states at the last frame ($T = 109$).

where $a_{i,j} = p(x(t) = j|x(t-1) = i)$. Note that (1) indicates sum of paths, and it transforms to the Viterbi algorithm if the summation is replaced with a max operation, as shown as

$$\alpha_t(j) = \max_i\{\alpha_{t-1}(i)a_{i,j}\}p(\boldsymbol{y}(t)|x(t) = j). \quad (2)$$

The values of $a_{i,j}$ and $p(\boldsymbol{y}(t)|x(t) = j)$ are significantly less than 1. For large values of $t$, $\alpha_t(.)$ tends to zero exponentially, and its computation exceeds the precision range of a machine. Hence, $\alpha_t(.)$ values are scaled with term $1/\max_i\{\alpha_t(i)\}$, at every time instant $t$. This normalization ensures that values of $\alpha_t(.)$ are between 0 and 1 at time $t$.

Given $\alpha(.)$ values, a backtracking algorithm is used to find the best alignment path. In order to backtrack, an additional variable $\phi$ is used to store the path as follows:

$$\phi_t(j) = \arg\max_i\{\alpha_{t-1}(i)a_{i,j}\} \quad (3)$$

where $\phi_t(j)$ denotes a state at time $(t-1)$ which provides an optimal path to reach state $j$ at time $t$. Given $\phi(.)$ values, a typical backtracking for forced-alignment is as follows:

$$x(T) = N \quad (4)$$
$$x(t) = \phi_{t+1}(x(t+1)),\ t = T-1, T-2, \ldots, 1. \quad (5)$$

It should be noted that we have assigned $x(T) = N$. This is a constraint in the standard implementation of forced-alignment which aligns the last frame $\boldsymbol{y}(t)$ to the final state $N$. An implied assumption in this constraint is that the value of $\alpha_T(N)$ is likely to be maximum among the values $\{\alpha_T(j)\}$ for $1 \le j \le N$ at time $T$. The forced-alignment algorithm implemented using (4) and (5) is henceforth referred to as FA-0. In order to provide a visualization of the usefulness of (4), let us consider the following example. A sequence of two syllables separated by a short pause, as in "$ba\langle pau\rangle sa$," is uttered and feature vectors are extracted from the speech signal. This sequence of feature vectors is forced-aligned with a sequence of HMM states corresponding to phones /b/, /aa/, /pau/, /s/, and /aa/. Fig. 1 displays the values in alpha matrix (HMM states against time measured in frames) obtained using (2). The dark band in Fig. 1 is referred to as beam and it shows how the pattern of values of $\alpha$ closer to 1 is diagonally spread across the matrix. From Fig. 1, we observe that at the last frame ($T = 201$), the last HMM state ($N = 15$) has highest value of $\alpha$ thus justifying the use of (4) in standard backtracking.

## III. Modifications to the Viterbi Algorithm

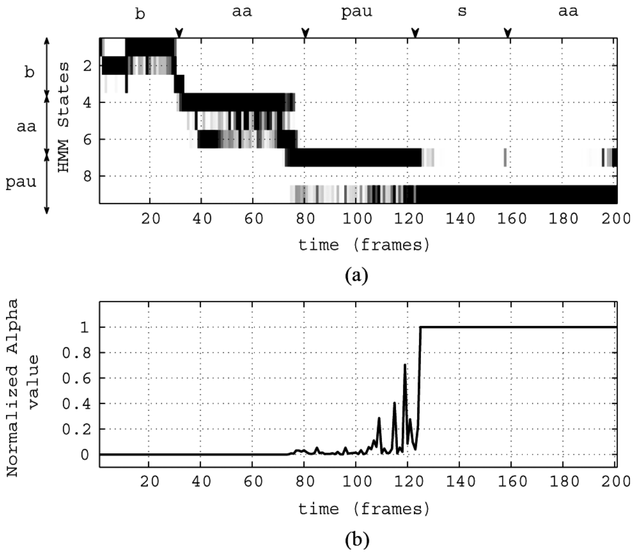
The constraint of forcing $x(T) = N$ is useful when we have the prior knowledge that the sequence of feature vectors $Y$ is an emission of the state sequence $S$. However, such constraints need to be modified when the state sequence $S$ emits $Y' \subset Y$, where $Y' = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(T')\}$, $T' < T$, and when the state sequence $S' \subset S$ emits $Y$, where $S' = \{1, \ldots, j, \ldots, N'\}$, $N' < N$. Such situations arise in processing large speech files in parts (see Section IV for more details). The following subsections describe the proposed modifications to the Viterbi algorithm to handle situations of $S' \subset S$ emitting $Y$ and $S$ emitting $Y' \subset Y$.

### A. Emission by a Shorter State Sequence (FA-1)

Given that $Y$ is an emission sequence for a corresponding sequence of states $S' \subset S$, the backtracking part of forced-alignment can be modified as follows:

$$x(T) = \arg\max_{1 \le j \le N}\{\alpha_T(j)\} \quad (6)$$
$$x(t) = \phi_{t+1}(x(t+1)),\ t = T-1, T-2, \ldots, 1. \quad (7)$$

Equation (6) poses the modified constraint that the last frame $\boldsymbol{y}(T)$ could be aligned to a state which has the maximum value of $\alpha$ at time $T$. This modified constraint allows the backtracking process to pick a state sequence which is shorter than $S$. The forced-alignment algorithm implemented using (6) and (7) is henceforth referred to as FA-1. In order to examine the suitability of (6), the feature vectors corresponding to utterance of "$ba\langle pau\rangle$" are force-aligned with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/, and /aa/. Fig. 2 (a) displays the alpha matrix of this alignment. It should be noted that the dark band in Fig. 2(a)—the beam of alpha matrix—is not diagonal. Moreover, at the last frame ($T = 109$), the last state ($N = 15$) does not have highest value of $\alpha$. Thus, (4) will fail to obtain a state sequence appropriate to the aligned speech signal. From Fig. 2(b), we can observe that the HMM state 9 has highest alpha value at the last frame, and (6) can be used to pick HMM state 9 automatically as the starting state of backtracking. Thus, the use of (6) and (7) provides a state sequence,



(a)

(b)

Fig. 3. (a) Alpha matrix obtained for the alignment of feature vectors corresponding to utterance of "$ba\langle pau \rangle sa$" with the HMM state sequence corresponding to phones /b/, /aa/, and /pau/. (b) Alpha values of the last state ($N = 9$) for all frames.

which is shorter than the originally aligned state sequence, but has an appropriate match with the aligned speech signal.

### B. Emission of a Shorter Observation Sequence (FA-2)

When a given state sequence $S$ emits a sequence $Y' \subset Y$, the backtracking part of forced-alignment can be modified as follows. Let $T' < T$ be the length of $Y'$. To obtain the value of $T'$, the key is to observe the values of $\alpha_t(N)$ for all $t$. If $1 \leq t \ll T'$ then $\alpha_{T'}(N) < 1$, and as $t \longrightarrow T'$ then $\alpha_t(N) \longrightarrow 1$.[3] This property of state $N$ could be exploited to determine the value of $T'$. Equation (8) formally states the property of state $N$, and could be used to determine the value of $T'$.

$$\alpha_t(N) = \begin{cases} < 1 & 1 \leq t < T' \\ = 1 & t \geq T'. \end{cases} \tag{8}$$

Given $T' < T$, the backtracking algorithm is modified as follows:

$$x(T') = N \tag{9}$$

$$x(t) = \phi_{t+1}(x(t+1)), \ t = T' - 1, \ldots, 1. \tag{10}$$

Equation (9) poses the modified constraint that the last state $N$ could be aligned to a feature vector at time $T' < T$, where $T'$ denotes the length of $Y'$ by satisfying (8). The modified constraint in (9) allows the backtracking process to pick an observation sequence which is shorter than $Y$. The forced-alignment algorithm implemented using (9) and (10) is henceforth referred to as FA-2. In order to examine the suitability of (9), the feature vectors corresponding to utterance of "$ba\langle pau \rangle sa$" are force-aligned with the HMM state sequence corresponding to phones /b/, /aa/, and /pau/. Fig. 3(a) displays the alpha matrix of this alignment. From Fig. 3(b), it is observed that the time instant at which the alpha value for the last state ($N = 9$) reaches 1 also denotes the length of shorter observation sequence ("$ba\langle pau \rangle$"). Thus, (9) and (10) can be used to pick a shorter observation sequence corresponding to the state sequence used for alignment.

## IV. Segmentation of a Large Audio File

So far, we have discussed the modifications to the Viterbi algorithm to handle cases of $S' \subset S$ emitting $Y$, and $S$ emitting $Y' \subset Y$. In this section, these modifications are shown to be useful in processing large speech files. Two different methods to process large speech files are described below.

### A. Segmentation Using FA-1 (SFA-1)

In this method, the large speech file is sliced into chunks of $d_b$ seconds. To begin with, the first chunk of speech is force-aligned with a sequence of words from the beginning of text. The unknown variable here is the length of this sequence. To resolve this issue, we overestimate the length based on average speaking rate of three words per second. Thus, a longer sequence of words is force-aligned with the chunk of speech. This leads to the case of $S' \subset S$ emitting $Y$, which could be handled by FA-1. The result of using FA-1 is the correct length of word sequence corresponding to the first chunk of speech. This process is repeated until the end of large speech file. This method of segmenting a large audio file using FA-1 is henceforth referred to as SFA-1. The formal description of SFA-1 is as follows.

Let $\Phi$ denote a large audio book and $\{w(1), \ldots, w(m), \ldots, w(M)\}$ denote the sequence of words present in $\Phi$. Let $\{\boldsymbol{y}(1), \ldots, \boldsymbol{y}(t), \ldots, \boldsymbol{y}(T)\}$ be the sequence of $T$ feature vectors extracted from $\Phi$. Let $n_f$ denote the number of frames in a chunk of $d_b$ seconds of speech. The value of $d_b$ is 30 seconds in this experiment, and the choice of this value is not critical. Let $n_w$ denote the number of words in $d_b$ seconds, estimated as $n_w = \eta * d_b * \lambda$, where $\eta = 3$ indicates a speaking rate of three words per second. The value of $\lambda = 1.5$ is chosen such that the estimate of $n_w$ is *higher* than the actual number of words in $d_b$ seconds of speech. The sequence of steps involved in SFA-1 is as follows.

1) $m = 1, t = 1$.
2) Let $F = \{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n_f)\}$ and let $W = \{w(m), w(m+1), \ldots, w(m+n_w)\}$. A sentence HMM representing $W$ is constructed such that it introduces an *optional* silence model between every word. This optional silence HMM model helps to capture any pauses inserted by the speaker between two adjacent words.
3) Force-align $F$ with a sentence HMM of $W$ using FA-1. Let $x(t), x(t+1), \ldots, x(t+n_f)$ be the state sequence obtained as a result of forced-alignment between $F$ and $W$. In FA-1, the observation vector $\boldsymbol{y}(t+n_f)$ is aligned to a state $x(t+n_f)$, which has the maximum alpha value at time $(t+n_f)$.
4) Note that the speech block $F$ is an *ad hoc* block considered without any knowledge of pause/word/sentence boundary. Thus, the state $x(t+n_f)$ need not be an ending state of a word HMM and hence only an initial portion of state sequence is considered. Let $\delta$ be the minimum non-negative integer value ($\delta \geq 0$) such that $x(t+n_f-\delta)$ is an ending state of a word HMM in the vicinity of $x(t+n_f)$. Considering the state sequence $\{x(t), x(t+1), \ldots, x(t+n_f-\delta)\}$, the corresponding sequence of words $W'\{w(m), w(m+1), \ldots, w(m+n'_w)\}$ is obtained, where $n'_w \leq n_w$. Starting from $w(m)$, a word $w(m+n''_w)$ is located such that there exists a pause ($\geq$ 150 ms) after the word $w(m+n''_w)$, where $n''_w < n'_w$. Let $n''_f$ be the number of frames aligned with the word sequence $\{w(m), w(m+1), \ldots, w(m+n''_w)\}$.
5) $t = t + n''_f, m = m + n''_w$.
6) Repeat the steps 2–6 until the end of $\Phi$.

### B. Segmentation Using FA-2 (SFA-2)

In this method, the text of large speech file is divided into paragraphs. In an audio book, the text is naturally arranged in paragraphs. Each

[3] From (2), it is trivial to observe that a state $j$ achieves an alpha value of 1 at time $t$, only if it is highly likely to be observed at $t$. This is dictated by the terms $\max_i\{\alpha_{t-1}(i)a_{i,j}\}$ and $p(\boldsymbol{y}(t)|x(t) = j)$. The alpha value of state $N$ being 1 at time $T'$ implies that the state $N$ is highly likely to be observed at $T'$, and thus the length of observation sequence $Y'$ is $T'$.

paragraph consists of one or more sentences, and usually deals with a single thought or topic or quotes a character's continuous words. Let $\Phi$ consist of a sequence of $K$ paragraphs $\{u(1), \ldots, u(k), \ldots, u(K)\}$. The words in first paragraph $u(1)$ are force-aligned with first $d_u$ seconds of speech data. As $d_u$ is not known *a priori*, we overestimate its value. Thus, a longer speech chunk is force-aligned with the words in $u(1)$. This leads to the case of $S$ emitting $Y' \subset Y$, which could be handled by FA-2. The result of FA-2 is the correct length of speech chunk corresponding to words in first paragraph $u(1)$. This process is repeated for the remaining paragraphs until the end of text. The method of segmenting a large audio file using FA-2 is henceforth referred to as SFA-2. The sequence of steps involved in SFA-2 is as follows.

1) $k = 1, t = 1$.
2) Let $U = [u(k), u(k+1)]$.
3) A heuristic estimate of duration of $U$ is defined as $d_u = n_p * d_p$, where $n_p$ is the number of phones in utterance $U$ and $d_p$ denotes the duration of a phone. The value of $d_p$ is chosen as 0.13 seconds such that the estimated value of $d_u$ is *higher* than the actual duration of the utterance $U$. Let $n_f$ denote the number of frames in $d_u$ seconds and let $F = \{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n_f)\}$ denote the sequence of feature vectors.
4) Force-align $F$ with the sentence HMM representing $U$ using FA-2. As a result of this forced-alignment, the shorter observation sequence $\{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n'_f)\}$ emitted by $U$ is obtained, where $n'_f < n_f$.
5) Given that $U$ is force-aligned with a longer observation sequence, the ending portion of alignment may not be robust—for example, the silence HMM model at the end of $U$ might observe a few observation vectors of next utterance $u(k+2)$, especially if $u(k+2)$ begins with a fricative sound. Hence, the observation sequence $\{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n''_f)\}$ corresponding to utterance $u(k)$ alone is considered, where $n''_f < n'_f$.
6) $t = t + n''_f, k = k + 1$.
7) Repeat steps 2–6 until $k < K$.
8) In order to obtain phone boundaries for the last utterance $u(K)$ perform forced-alignment of $u(K)$ with $\{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(T)\}$ using FA-0.

## V. Evaluation

To evaluate SFA-1 and SFA-2, we have used the speech databases of *RMS* from the *CMU ARCTIC* [1] and *EMMA* from Librivox [6]. The *RMS* speech database consists of 1132 utterances corresponding to 1132 paragraphs in text. Here, each paragraph contains only one sentence. For our experiments, 1132 utterances were concatenated to create an artificial large speech file, henceforth referred to as $\Phi_r$. The duration of $\Phi_r$ is 66 minutes. It could be argued that an artificial long speech file may not represent a distribution of pauses in an authentic long speech file. To enable such comparison, around 45 minutes of speech data from a story—first three chapters of *EMMA* in Librivox—was hand labeled with beginning and ending of sentences. This database is referred to as $\Phi_m$. Segmentation of long speech files can be evaluated in the following ways.

- *Boundaries of utterances:* The utterance boundaries obtained automatically from long speech files can be compared with hand labeled utterance boundaries in *EMMA* ($\Phi_m$) or known utterance boundaries in RMS ($\Phi_r$). This evaluation methodology is referred to as $E1$.
- *Phone boundaries:* The phone boundaries obtained automatically from long speech files can be compared with phone boundaries obtained from FA-0 (forced-alignment of utterances with their paragraph-length text). This evaluation methodology is referred to as $E2$.

TABLE I
Evaluation of SFA-2 on *RMS* ($\Phi_r$) and *EMMA* ($\Phi_m$) Using $E1$ and $E2$. Mean ($\mu$) and Standard Deviation ($\sigma$) are in Milliseconds

| | $E1$<br>$(\mu, \sigma)$ ms | $E2$<br>$(\mu, \sigma)$ ms |
|---|---|---|
| $\Phi_r$ | (35, 21) | (35, 22) |
| $\Phi_m$ | (138, 88) | (20, 52) |

It is important to note that SFA-2 segments long speech files into utterances corresponding to paragraphs in text, and hence it could be evaluated for $E1$ and $E2$. SFA-1 segments long speech files into chunks of 1–30 seconds long. These chunks need not correspond to paragraph length utterances and hence SFA-1 does not allow its chunks for $E1$ and $E2$ evaluations. However, evaluation of SFA-1 can be done using Mel-cepstral distortion (see Section VI for details).

Table I shows $E1$ and $E2$-based evaluation of SFA-2 on *RMS* and *EMMA*. The results of $E1$ for *EMMA* indicate that the mean (138 ms) and standard deviation (88 ms) of difference in boundary locations is higher for naturally long speech files than that of *RMS*. This is due to pauses at the beginning and ending of utterances in *RMS* are short. They are also not representative of a true distribution of pauses, as occurring in naturally long speech files. However, the implication of this higher difference is less significant as observed from the results of $E2$, which measures the difference between phone boundaries of SFA-2 and FA-0. The results of $E2$ for *EMMA* ($\mu = 20$ ms, $\sigma = 52$ ms) indicate a reasonable agreement of SFA-2 with FA-0 on phone boundaries. This suggests that the difference in boundary locations of SFA-2 and hand labeled data as measured by $E1$ is in the pause regions at the beginning and ending of utterances, and has lesser effect on phone boundaries. From the results of $E1$ and $E2$ on *RMS* ($\Phi_r$) and *EMMA* ($\Phi_m$), it can be argued that segmentation of long speech files can be obtained using modifications to the Viterbi algorithm.

## VI. Building Voices From Audio Books

The methods of segmenting long speech files using SFA-1 and SFA-2 are implemented as a package named INTERSLICE. This is integrated in FestVox (www.festvox.org), which is an open source toolkit for building synthetic voices. The process of building synthetic voices using audio books, INTERSLICE and FestVox is as follows.

- Use INTERSLICE for segmentation of long speech files in audio books.
- The output of SFA methods in INTERSLICE is beginning and ending boundaries of utterances as well as phone boundaries in these utterances. The acoustic models used in INTERSLICE to obtain these phone boundaries are speaker-independent. It is known that speaker-dependent acoustic models are better than speaker-independent to obtain phone boundaries [7]. Also, optimizing phone boundaries is important for statistical speech synthesis such as CLUSTERGEN [8]. Hence, the output of INTERSLICE pertaining to only utterance boundaries is used.
- Given the utterances and the corresponding text, CLUSTERGEN engine in FestVox builds a synthetic voice. An important step in building a CLUSTERGEN voice is to obtain phone and HMM state level boundaries in the utterances [9]. This is accomplished by using flat-start initialization in Baum–Welch training of speaker-dependent HMMs.

As an initial experiment, we collected 17.35 hours of recordings of *EMMA*, henceforth referred to as $\Phi_e$. We downloaded the associated text from the Project Gutenberg, and added text at the beginning and end of each chapter to match the introductions and closings made by the speaker. The text was arranged into 2693 paragraphs. Both SFA-1 and SFA-2 were applied on $\Phi_e$, and CLUSTERGEN voices were built

TABLE II
MCD Scores Obtained on TTS Voices of *Emma* ($V_e^1$, $V_e^2$), *Pride and Prejudice* ($V_p^2$), *Walden* ($V_w^2$) and *Sense and Sensibility* ($V_s^2$). Here the Superscripts $^1$ and $^2$ Indicate the use of SFA-1 and SFA-2, Respectively

| | Gender | MCD | # utts. (train) | # utts. (held-out) |
|---|---|---|---|---|
| $V_e^1$ | F | 5.09 | 13757 (15.57 hrs) | 1528 (1.74 hrs) |
| $V_e^2$ | F | 5.04 | 2424 (15.67 hrs) | 269 (1.67 hrs) |
| $V_p^2$ | F | 6.02 | 2218 (11.99 hrs) | 246 (1.41 hrs) |
| $V_w^2$ | M | 4.96 | 1134 (12.84 hrs) | 126 (1.46 hrs) |
| $V_s^2$ | M | 5.12 | 2087 (12.17 hrs) | 232 (1.30 hrs) |

TABLE III
DND Listening Tests on $V_e^1$ and $V_e^2$

| | diff | no-diff |
|---|---|---|
| $V_e^1$ vs $V_e^2$ | 17/50 | 33/50 |

[9]. Let $V_e^1$, $V_e^2$ denote the TTS voices built from $\Phi_e$ using SFA-1 and SFA-2, respectively.

Table II shows the Mel-cepstral distortion (MCD) scores obtained on TTS voices of $V_e^1$ and $V_e^2$. MCD is a widely used objective measure in speech coding, voice conversion, and statistical parametric speech synthesis [10]–[12]. This measure can be viewed as an approximate log spectral distance between the synthetic speech and natural speech. As the MCD decreases, the corresponding voice quality is found to be perceptually better [12]. It is empirically observed that studio quality recordings such as CMU ARCTIC have MCD scores in the range of 4–7 [9], [10]. Thus the MCD scores 5.09 and 5.04 obtained on $V_e^1$ and $V_e^2$, respectively, indicate that the output of methods SFA-1 and SFA-2 is useful for building synthetic voices. Table III shows the results of listening tests conducted on $V_e^1$ and $V_e^2$. A set of five speakers (henceforth referred to as subjects) participated in the listening test. A set of ten utterances were synthesized from these two voices. Each subject was asked to listen to an utterance synthesized by $V_e^2$ and compare it against the utterance of same text synthesized by $V_e^1$. The subject was asked whether there is a difference or no-difference in the pair of utterances. We henceforth refer to this listening test as DND (difference-no-difference) test. The results indicate that in 33 out of 50 utterances, the subjects did not perceive any difference between the voices $V_e^1$ and $V_e^2$. The subjects perceived a difference between $V_e^1$ and $V_e^2$ in 17 out of 50 utterances. A subset of these 17 utterances were manually analyzed, and a difference of 500 ms was found in the duration of these utterances. Each of these utterances is approximately 30 s long. CLUSTERGEN predicts duration of phones based on contextual features, and this prediction is learnt based on duration of phone boundaries observed in the training set. Given that the voices being examined here have different duration models, it is difficult to pinpoint reasons for variations in predicted durations of phones. Other than these minor variations in durations, we did not perceive any difference in the spectral quality of voices.

To demonstrate that the algorithms in INTERSLICE are directly applicable to different audio books, we have selected three more audio books from Librivox. Let $\Phi_p$, $\Phi_w$, $\Phi_s$ denote the audio books of *Pride and Prejudice*, *Walden*, *Sense and Sensibility*. On these audio books, INTERSLICE was used to segment large speech files using SFA-2 method. Let $V_p$, $V_w$, $V_s$ denote the CLUSTERGEN voices built from $\Phi_p$, $\Phi_w$ and $\Phi_s$, respectively. Table II shows that the MCD values obtained for $V_p$, $V_w$, and $V_s$ are in the acceptable range of 5–7. This experiment demonstrates that the algorithms used in INTERSLICE are directly applicable to several audio books without any modifications.

TABLE IV
Example Utterances From SFA-1 and SFA-2 on *Emma*

| **Utterances obtained from SFA-1** |
|---|
| 1. I do not know what your, opinion may be. Mrs Weston, said Mr Knightley,<br>2. of this great intimacy, between Emma and Harriet Smith,<br>3. but I think it a bad thing,<br>4. A bad thing. Do -<br>5. you really think it a bad thing,<br>6. why so. I think they will neither of them, do the other any good. |
| **Utterances obtained from SFA-2** |
| 1. "I do not know what your opinion may be, Mrs. Weston," said Mr. Knightley, "of this great intimacy between Emma and Harriet Smith, but I think it a bad thing."<br>2. "A bad thing! Do you really think it a bad thing?–why so?"<br>3. "I think they will neither of them do the other any good." |

### A. SFA-1 (Speech-Driven) Versus SFA-2 (Text-Driven)

While both SFA-1 and SFA-2 perform segmentation of long speech files, there are differences in the output of these methods. SFA-2 segments the long speech files into utterances corresponding to paragraphs in text. As discussed earlier, a paragraph could be one or more sentences expressing a single thought or character's continuous words. The definition of a paragraph is not critical here, but it is important to understand that utterances obtain from SFA-2 corresponds to boundaries of grammatical units (sentences) and logical units (thoughts, character's turns, etc.) as shown in Table IV. Such segmentation is useful for modeling prosody at sentence and paragraph level, especially in text-to-speech. On the contrary, SFA-1 segments the long speech file into chunks of 1–30 seconds as shown in Table IV. These chunks may or may not be complete sentences. Hence, they may provide inaccurate representation of sentence boundaries and the corresponding prosodic boundaries. The input to TTS systems is sentences or paragraphs. Hence, modeling prosody at sentence or paragraph level helps to incorporate the same during synthesis. Thus, it is preferred to use the output of SFA-2 for prosody modeling in text-to-speech, as it provides paragraph length utterances.

## VII. Conclusion

In this paper, we have proposed modifications to the Viterbi algorithm and shown that the proposed methods (speech-driven and text-driven) could be employed to segment large speech files. These methods are applicable when there is a good match between the text and speech, with almost no disfluencies and pronunciation mistakes; and are helpful especially for languages where a speech recognition system is not readily available. However, we assume the existence of speech databases (of 1-2 hours long similar to CMU ARCTIC) to pre-train a set of context-independent acoustic models. This requirement is easier to fulfill than that of a large-vocabulary ASR.

The algorithms presented in this paper are implemented as a package named INTERSLICE in FestVox—an open-source toolkit for building synthetic voices. We have also shown that INTERSLICE can be used for segmentation of monologues in audio books, and thus synthetic voices could be built from these audio books. The synthetic voices built from the audio books in public domain were found to have Mel-cepstral distortion scores in the range of 4–7, which is similar to voices built from studio-quality recordings such as CMU ARCTIC.

## Acknowledgment

The authors would like to thank Dr. M. Ravishankar for useful discussions and comments on this work.

## REFERENCES

[1] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *Proc. 5th ISCA Speech Synth. Workshop (SSW5)*, Pittsburgh, PA, 2004, pp. 223–224.

[2] P. J. Moreno, C. Joerg, J. M. van Thong, and O. Glickman, "A recursive algorithm for the forced-alignment of very long audio segments," in *Proc. Int. Conf. Spoken Lang. Process.*, Sydney, Australia, 1998.

[3] I. Trancoso, C. Duarte, A. Serralheiro, D. Caseiro, L. Carrico, and C. Viana, "Spoken language technologies applied to digital talking books," in *Proc. Interspeech*, Pittsburgh, PA, 2006, pp. 1990–93.

[4] P. J. Moreno and C. Alberti, "A factor automaton approach for the forced alignment of long speech recordings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Taipei, Taiwan, 2009, pp. 4869–4872.

[5] A. Stolcke and E. Shriberg, "Automatic linguistic segmentation of conversational speech," in *Proc. Int. Conf. Spoken Lang. Process.*, Philadelphia, PA, 1996, vol. 2, pp. 1005–1008.

[6] "LibriVox, Acoustic Liberation of Books in the Public Domain," Jul. 2010 [Online]. Available: http://www.librivox.org

[7] B. Angelini, C. Baralo, D. Falavigna, M. Omologo, and S. Sandri, "Automatic diphone extraction for an italian text-to-speech synthesis system," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 581–584.

[8] A. W. Black and J. Kominek, "Optimizing segment label boundaries for statistical speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Taipei, Taiwan, 2009, pp. 3785–88.

[9] A. W. Black, "CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling," in *Proc. Interspeech*, Pittsburgh, PA, 2006, pp. 1762–65.

[10] J. Kominek, T. Schultz, and A. W. Black, "Synthesizer voice quality on new languages calibrated with mean Mel-Cepstral distorion," in *Proc. Inte. Workshop Spoken Lang. Technol. for Under-Resourced Lang. (SLTU)*, Hanoi, Vietnam, 2008.

[11] J. Yamagishi, B. Usabaev, S. King, O. Watts, J. Dines, J. Tian, R. Hu, Y. Guan, K. Oura, K. Tokuda, R. Karhila, and M. Kurimo, "Thousands of voices for HMM-Based speech synthesis—analysis and application of TTS systems built on various ASR corpora," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 18, no. 5, pp. 984–1004, Jul. 2010.

[12] T. Toda, A. W. Black, and K. Tokuda, "Mapping from articulatory movements to vocal tract spectrum with Gaussian mixture model for articulatory speech synthesis," in *Proc. 5th ISCA Speech Synth. Workshop (SSW5)*, Pittsburgh, PA, 2004, pp. 31–36.