ISCA Archive
http://www.isca-speech.org/archive

7th ISCA Workshop on
Speech Synthesis (SSW-7)
Kyoto, Japan
September 22-24, 2010

Handling Large Audio Files in Audio Books for Building Synthetic Voices

Kishore Prahallad[1,2], *Alan W Black*[2]

[1]International Institute of Information Technology, Hyderabad, India.
[2]Language Technologies Institute, Carnegie Mellon University, USA.
kishore@iiit.ac.in, awb@cs.cmu.edu

Abstract

One of the issues in using audio books for building a synthetic voice is the segmentation of large audio files. The use of standard forced-alignment 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 work, we propose suitable modifications to the standard forced-alignment algorithm and demonstrate its usefulness for segmentation of large audio files. Experimental results are provided on audio files including an artificially created large audio file and on EMMA speech corpus of 17.5 hours. Synthetic voices are also built using these large audio files.

Index Terms: Large audio file, audio books, forced-alignment, text-to-speech

1. Introduction

Development of a unit selection voice or a statistical parametric voice requires a speech corpus, i.e., a database of audio files and text transcriptions. The text transcriptions are sentences (often incomplete) with five to ten words in each sentence. These sentences are typically recorded by a professional native speaker in a noise-free environment or in a recording studio. The total duration of audio files varies from two to ten hours. The selection of sentences used for recording is based on optimal coverage of phones in a language and hence there may not be a semantic relationship between any two successive sentences. In other words, a speech corpus built specifically for the purpose of text-to-speech does not capture prosody beyond an isolated sentence and it takes considerable amount of effort and time to build such corpus. Thus it is important to investigate methods where the amount of effort involved in obtaining a speech corpus for text-to-speech should be lesser than that of conventional method of having a professional speaker record in a studio. At the same time, the obtained audio files should have the following properties: 1) The recordings of audio files are done in a noise-free environment by a single speaker and 2) the aspects of prosody beyond an isolated sentence are encapsulated in speech files.

In this context, it should be noted that the advent of audio and video sharing networks such as You-Tube and Podcasts, has increased the availability of audio and video data at exponential rates. In this type of data, audio books are of particular interest to us as they provide text and the speech data recorded in a noise free environment. For example, Librivox (librivox.org) and Loudlit (loudlit.org) are two portals where volunteers record books from Gutenberg project and make the audio available for public usage. These freely available audio books (recorded by a single speaker in a noise-free environment) act as excellent candidates for building synthetic voices. Moreover, the text in these books being a story, is arranged in paragraphs. Hence the utterances in audio books encapsulate prosody different from that of an isolated sentence.

However, there do exist a few issues in using audio books for building synthetic voices. One of the issues is the segmentation of large audio files. The audio books have large audio files whose durations vary from 5 to 30 minutes. The use of standard forced-alignment technique to obtain phone level labels on these audio files fails primarily because of huge memory requirements [1] [2]. For an audio file of 30 minutes, a standard implementation of forced-alignment technique typically requires an allocation of 2-D array of 36 GB size. Earlier works have attempted to resolve this problem by segmenting the speech at prosodic phrase breaks or by using speech recognition. In [3], prosodic phrase break locations are first estimated in the speech signal, and then words are placed around breaks based on mean word length and likelihoods of breaks occurring after each word. In [4], a speech recognizer based on finite state transducers is employed to segment the large audio files. In [5], the alignment of large audio files is attempted as a recursive speech recognition problem with a restrictive dictionary and language model. While the work in [3] relies on additional steps to estimate phrase breaks and mean word length, the works in [4], [5] and [2] rely on availability of large vocabulary speech recognizer to segment the large audio files. In this work, we demonstrate that the segmentation of large audio files could be done with simple modifications to standard implementation of forced-alignment algorithm. Thus our work reported in this paper makes use of acoustic models built at phone level as done in a standard implementation of forced-alignment algorithm, but differs significantly from the work reported in [4], [5] and [2], as we do not rely on availability of large vocabulary speech recognizer or employ restricted dictionary and language model to constrain the search space. The fact that our approach is a simple modification to forced-alignment makes it suitable for languages (especially less resource languages) where there is no availability of large vocabulary speech recognition.

The remainder of this paper is organized as follows: Section II and III discusses the extraction of features from a speech signal and the set of acoustic models used in this work. Section IV describes a standard implementation of forced-alignment algorithm. Section V discusses two suitable modifications to forced-alignment which could be applied to segment a large audio file. Section VI proposes two different methods of segmenting a large audio file. In Section VII, the proposed methods for segmenting a large audio file are evaluated on an artificially created large audio file. In Section VIII, experimental results are provided on EMMA speech corpus.

2. Extraction of features from a speech signal

To extract the feature vectors from a speech signal, the characteristics of the speech signal are assumed to be stationary over a short duration of time (between 10-30 ms). The speech signal is preemphasized using a difference operator and is divided into frames of 10 ms using a frame shift of 5 ms. Each frame of speech data is passed through a Hamming window and then through a set of Mel-Frequency filters to obtain 13 cepstral coefficients. Thus each frame of speech data is represented by a vector of 13 coefficients [6].

3. Acoustic models

In this work, 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). These acoustic models are context-independent (CI) HMM models where each phone has three emitting states and two null states. The states in a phone HMM are connected in left-to-right fashion with out any skip arcs. The exception is *pau*, a silence HMM, where a skip arc is provided to optionally omit the middle emitting state. Each state is modeled by a two component Gaussian mixture model. Each Gaussian component is modeled by a 13-dimensional mean vector and a diagonal covariance matrix. The HMM models were initialized using a flat start and were trained using Baum-Welsh reestimation algorithm.

4. A Standard Implementation of Forced-Alignment Algorithm

Let $Y = \{y(1), y(2), \ldots, y(T)\}$ be a sequence of observed feature vectors (see Section 2) extracted from a speech signal of T frames. A forced-alignment technique aligns the feature vectors extracted from a speech signal with a given transcription using a set of existing acoustic models (see Section 3). Let $S = \{1, \ldots, j, \ldots, N\}$ be the state sequence corresponding to a sequence of words used to force-align the feature vectors Y, and let $x(1), x(2), \ldots, x(T)$ be the unobserved sequence of hidden states for Y. Let $p(y(t)|x(t) = j)$ denote the emission probability of state j for the feature observed at time t and $1 \leq j \leq N$, where N is the total number of states.

Let us define $\alpha_t(j) = p(x(t) = j, y(1), y(2), \ldots, y(t))$. This is a joint probability of being in state j at time t and of 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

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

where $a_{i,j} = p(x(t) = j|x(t-1) = i)$. Note that the Eq. (1) indicates sum of paths and it transforms to Viterbi if the summation is replaced with a max operation. Thus Eq. (1) transforms to Viterbi as shown in Eq. (2).

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

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

$$\phi_t(j) = \operatorname*{argmax}_i \{\alpha_{t-1}(i) a_{i,j}\}, \qquad (3)$$

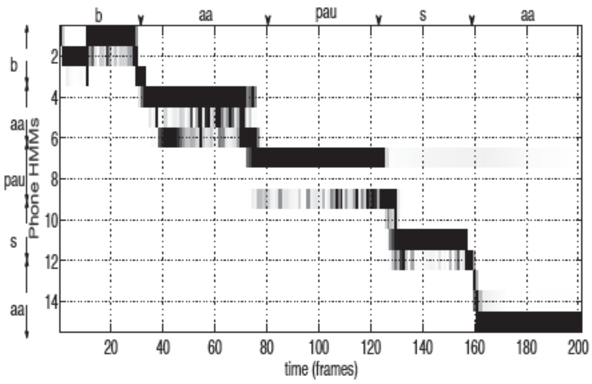


Figure 1: *An alpha matrix obtained for the alignment of feature vectors corresponding to utterance of "ba pau sa" with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/ and /aa/. The markers indicate manually labeled phone boundaries.*

where $\phi_t(j)$ denotes a state at time $(t-1)$ which provides an optimal path to reach state j at time t.

4.1. A standard backtracking (FA-0)

Given the $\phi(.)$ values, a typical backtracking for forced-alignment is as follows:

$$x(T) = N \qquad (4)$$

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

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 $y(t)$ to the final state N and 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 $1 \leq j \leq N$ at time T. The forced-alignment algorithm implemented using Eq. (4) and Eq. (5) is henceforth referred to as FA-0.

In order to provide a visualization of the usefulness of Eq. (4), let us consider the following example. A sequence of two syllables *"ba pau sa"*, separated by a short pause 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). These values are obtained using Eq. (2) and are normalized between 0 and 1 at each time frame. The dark band in Fig. 1 is referred to as beam and it shows how the pattern of values of α 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 α thus justifying the use of Eq. (4) in standard backtracking.

5. Modifications to Forced-Alignment

The constraint of forcing $x(T) = N$ is useful when we have the prior knowledge that the sequence of feature vectors Y are emissions of the state sequence S. However, such constraints need to be modified when the state sequence S emits Y', where $Y' = \{y(1), y(2), \ldots, y(T')\}$ and $T' < T$ or when the state sequence $S' = \{1, \ldots, j, \ldots, N'\}$ emits Y, where $N' < N$.

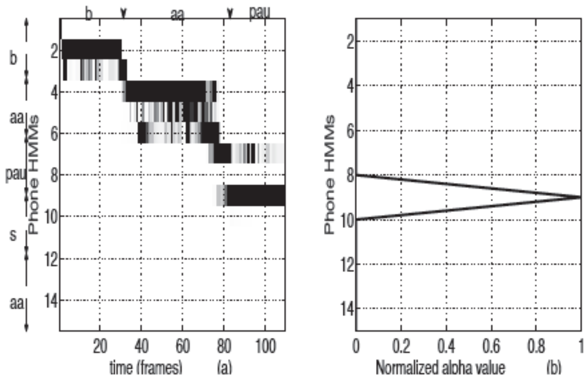


Figure 2: *(a) An alpha matrix obtained for the alignment of feature vectors corresponding to utterance of "ba pau" with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/ and /aa/. (b) Alpha values of all states at the last frame* ($T = 109$). *The markers indicate manually labeled phone boundaries.*

In other words, the constraint $x(T) = N$ needs to be modified for situations when 1) the sequence of feature vectors Y is an emission of a sequence of states S', where S' is shorter than the given sequence S, and 2) the state sequence S emits a sequence of feature vectors Y' whose duration is shorter than Y.

5.1. Emission by a shorter state sequence (FA-1)

When the given sequence of feature vectors Y is an emission of a sequence of states S' which is shorter than the given sequence S, then the backtracking part of forced-alignment is modified as in Eq. (6).

$$x(T) = \underset{1 \leq j \leq N}{\operatorname{argmax}} \{\alpha_T(j)\} \tag{6}$$

$$x(t) = \phi_{t+1}(x(t+1)),\ t = T-1, T-2, \ldots, 1. \tag{7}$$

Equation (6) poses a modified constrained that the last frame $y(T)$ could be aligned to a state which has maximum value of α at time T. This modified constraint allows the backtracking to pick a state sequence which is shorter than S. The forced-alignment algorithm implemented using Eq. 6 and Eq. 7 is henceforth referred to as FA-1.

In order to examine the suitability of Eq. (6) the feature vectors corresponding to utterance of *"ba pau"* 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 could be noted that the beam of the alpha matrix is not diagonal and moreover at the last frame ($T = 109$), the last state ($N = 15$) does not have highest value of α. Thus the use of Eq. (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 Eq. (6) could be used to pick HMM state 9 automatically as the starting state of backtracking. Thus the use of Eq. (6) and Eq. (7) provides a state sequence, which is shorter than the originally aligned state sequence, but has an appropriate match with the aligned speech signal.

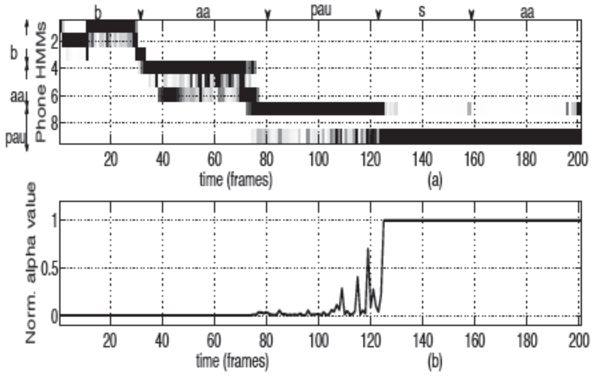


Figure 3: *(a) An alpha matrix obtained for the alignment of feature vectors corresponding to utterance of "ba pau sa" with the HMM state sequence corresponding to phones /b/, /aa/ and /pau/. (b) Alpha values of the last state* ($N = 9$) *for all frames. The markers indicate manually labeled phone boundaries.*

5.2. Emission of a shorter observation sequence (FA-2)

When the given state sequence S emits a sequence of feature vectors Y' whose duration is shorter than Y, then the backtracking part of forced-alignment is modified as follows. The key idea of obtaining a sequence of feature vectors Y' aligned to a state sequence S lies in observing at the values of α at state N. The alpha values at state N are typically zeros until the beam of alpha matrix (as dictated by the observation sequence and state transition probabilities) reaches the state N. When the beam reaches the state N, then the alpha value for state N tends to 1.

Let T' be the time instant at which the state N attains an alpha value of 1 and $\alpha_t(N) < 1$ for all $1 \leq t < T'$. Given T', the backtracking algorithm is modified as follows.

$$x(T') = N,\ T' < T \tag{8}$$

$$x(t) = \phi_{t+1}(x(t+1)),\ t = T'-1, \ldots, 1. \tag{9}$$

Equation (8) poses a modified constraint that the last state N could be aligned to a feature vector at time T' where $T' < T$. If the state sequence S is emitting an observation sequence Y' which is shorter in length than Y, then the time instant at which the beam in the trellis has reached state N denotes the length of Y'. This modified constraint allows the backtracking to pick an observation sequence which is shorter than Y. The forced-alignment algorithm implemented using Eq. 8 and Eq. 9 is henceforth referred to as FA-2.

In order to examine the suitability of Eq. (8) the feature vectors corresponding to utterance of *"ba pau 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 could be 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 pau"*) corresponding to state sequence representing /b/, /aa/ and /pau/. Thus Eq. (8) and Eq. (9) could be used to pick a shorter observation sequence corresponding to the state sequence used for alignment.

6. Segmentation of a Large Audio File

Let Φ denote a large audio book containing a sequence of K utterances $u(1), \ldots, u(k), \ldots, u(K)$, where each utterance is a sentence / paragraph. The objective here is to obtain phone level boundaries for each of the utterances. A standard method is to force-align the text transcription of utterance $u(k)$ $(1 \leq k \leq K)$ with the corresponding speech signal. However, in the case of large audio book Φ, the beginning and the ending of the utterances are not known. Hence we apply force-alignment technique using FA-1 or FA-2 for segmentation of a large audio file.

6.1. Segmentation using FA-1 (SFA-1)

In FA-1, we assume that an observation sequence is being force-aligned with a state sequence longer than the one which has emitted the observation and hence the objective is to find this shorter state sequence. Thus in order to use FA-1, we process the large audio file Φ in blocks of $d_b = 30$ s duration and force-align each block with a sequence of words whose length is longer than the actual sequence of words. The method of segmenting a large audio file using FA-1 is henceforth referred to as SFA-1.

Let $w(1), \ldots, w(m), \ldots, w(M)$ be the sequence of words present in audio file Φ. Let $\boldsymbol{y}(1), \ldots, \boldsymbol{y}(t), \ldots, \boldsymbol{y}(T)$ be the sequence of T feature vectors extracted from Φ. Let n_f denote the number of frames in a block of d_b s of speech. Let n_w denote the number of words in d_b s, heuristically 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 larger than the actual number of words in a 30 s of speech data. The sequence of steps involved in using FA-1 for segmentation of large audio file 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 *optional* silence model between every word. This optional silence model helps to capture any pause regions inserted by the speaker between any two adjacent words.
3. Force-align F with sentence HMM of W using the method 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 maximum alpha value at time $(t+n_f)$. Note that the speech block F is an ad hoc block considered without any knowledge of pause or word/sentence boundary and hence the state $x(t+n_f)$ need not be an ending state of a word HMM.
4. Let δ 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 $n'_w \leq n_w$.
5. To obtain a more robust alignment, only the initial portion of W' is considered. Starting from $w(m)$, a word $w(m+n''_w)$ is located such that there exists a pause of at least $150-200$ 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 $w(m), w(m+1), \ldots, w(m+n''_w)$.
6. $t = t + n''_f$, $m = m + n''_w$.
7. Repeat the steps 2-6 until the end of Φ.

6.2. Segmentation using FA-2 (SFA-2)

In FA-2, we assume that a given state sequence is being force-aligned with a larger observation sequence and hence the objective is to find this shorter state sequence. Thus in order to use FA-2, we process the large audio file Φ in terms of utterances $u(1), \ldots, u(k), \ldots, u(K)$ where each utterance $u(k)$ is a sentence/paragraph. The idea is to force-align an utterance with an observation sequence which is longer than the actual observation sequence. The method of segmenting a large audio file using FA-2 is henceforth referred to as SFA-2. The steps involved in SFA-2 are 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 $d_p = 0.1$ s is chosen 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 s and let F = $\{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n_f)\}$ denote the sequence of feature vectors extracted from Φ.
4. Force-align F with the sentence HMM representing U. This forced-alignment is performed 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, it was observed that the ending portion of such 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.

6.3. SFA-1 Vs SFA-2

While both SFA-1 and SFA-2 performs 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. 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 obtained from SFA-2 correspond to boundaries of grammatical units (sentences) and logical units (thoughts, character's turns etc.) as shown in Table 1. Such segmentation is useful for modeling prosody at sentence and paragraph level, especially in text-to-speech. In contrast, as shown in Table 1, SFA-1 segments the long speech file into chunks of 1-30 seconds. These chunks need not be complete sentences, hence many provide

Table 1: Example utterances obtained from SFA-1 and SFA-2

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

inaccurate representation of sentence boundaries and the corresponding prosodic boundaries. Thus it is preferred to use SFA-2 for text-to-speech, as it provides paragraph length utterances.

7. Evaluation on an Artificially Created Large Audio File

To evaluate the effectiveness of SFA-1 and SFA-2 for segmentation of large audio files, we have made use of RMS voice from CMU ARCTIC database. The RMS voice consists of 1132 utterances in US accented English. Let this original database of RMS (i.e., 1132 utterances and the corresponding speech wave files) be referred to as Θ_r. For our purposes, all the wave files of 1132 utterances were concatenated to create an artificial large audio file, henceforth referred to as Φ_r. The duration of Φ_r is 66 minutes. To compare and evaluate FA-0, SFA-1 and SFA-2, we conducted the following experiments on Θ_r and Φ_r.

- Apply FA-0 on each utterance in Θ_r to obtain phone boundaries in each utterance.
- Apply SFA-1 on Φ_r as described in Section 6.1. This process results in segmentation of Φ_r into short utterances of approximately $5 - 30$ s long and also provides phone boundaries in each of these utterances.
- Apply SFA-2 on Φ_r as described in Section 6.2. This process results in segmentation of Φ_r into 1132 utterances (the same number of utterances as in Θ_r) and also provides phone boundaries in each of these utterances.

The criteria to evaluate the performance of FA-0, SFA-1 and SFA-2 is as follows:

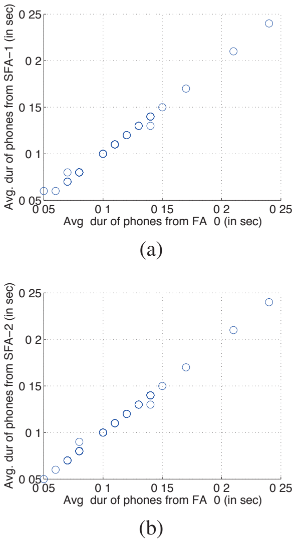
7.1. Duration of utterances:

The number of utterances obtained by SFA-2 are equal to number of utterances in the original recordings. Thus the duration of utterances obtained from SFA-2 is compared with the duration of utterances in original recordings.

Let d_k be the duration of an utterance k obtained from SFA-2 and let $\widehat{d_k}$ be the duration of the original recording of k. Let $\epsilon_k = \widehat{d_k} - d_k$. The mean and standard deviation of ϵ obtained for all 1132 utterances was found to be -0.023 s and 0.028 s respectively. These values indicates that SFA-2 could detect the beginning and ending of each of these utterances with an average difference of 23 milliseconds and a standard deviation of 28 milliseconds. In informal listening tests we found that the beginning and ending of utterances obtained SFA-2 were indistinguishable from the original recordings.

7.2. Mean duration of phones:

The second criteria is to compare the mean duration of phones obtained using FA-0, SFA-1 and SFA-2. Here we assume that the mean duration of phones obtained using FA-0 acts as benchmark to evaluate the performance of SFA-1 and SFA-2. Fig. 4 shows the scatter plot of mean duration of phones from FA-0 and SFA-1 or SFA-2 for RMS voice. The linear trend of scatter plots in Fig. 4 show that the segment boundaries obtained from SFA-1 and SFA-2 are nearly as good as that of FA-0.



(a)

(b)

Figure 4: (a) Scatter plot of mean duration of phones from FA-0 and SFA-1 for RMS voice. (b) Scatter plot of mean duration of phones from FA-0 and SFA-2 for RMS voice.

7.3. Text to Speech based evaluation

The third criteria is to build a TTS with the phone segments obtained from FA-0, SFA-1 and SFA-2. The process to build a TTS voice follows the framework of CLUSTERGEN [7]. CLUSTERGEN is a statistical parametric synthesis engine which learns the spectral, a 25-D Mel-Cepstral (MCEP) vector computed for every frame, and duration parameters from the speech data using classification and regression trees. During synthesis time, spectral (MCEP) and duration parameters are predicted from the input text. The predicted MCEPs are passed through Mel-Log Scale Spectral Approximator (MLSA) and are excited with white noise or pulse train to generate the speech signal. To evaluate the quality of a TTS voice, one could perform a subjective evaluation and an objective evaluation on a held-out test set of utterances. Let V_r^0, V_r^1 and V_r^2 denote the TTS voices built from Θ_r using FA-0, Φ_r using SFA-1 and Φ_r using SFA-2 respectively.

7.3.1. Objective evaluation of a TTS voice

A TTS voice is built using a set of utterances from train set and the utterances from held-out set are synthesized using this TTS

Table 2: MCD scores obtained on TTS voices of RMS (V_r^0, V_r^1, V_r^2) and EMMA (V_e^1, V_e^2).

	MCD	# utts. (train)	# utts. (held-out)
V_r^0	5.27	1019	113
V_r^1	5.27	1049	116
V_r^2	5.29	1019	113
V_e^1	5.09	13757	1528
V_e^2	5.04	2424	269

Table 3: DND listening tests on TTS voices of RMS (V_r^0, V_r^1, V_r^2) and EMMA (V_e^1, V_e^2).

	diff	no-diff
V_r^0 vs V_r^1	15/50	35/50
V_r^0 vs V_r^2	12/50	38/50
V_e^1 vs V_e^2	17/50	33/50

voice. Mel Cepstral Distortion (MCD) is an objective measure for evaluating the quality of synthesized utterances [7]. The synthesized wave file is aligned with original wave file using dynamic programming and Mel-Ceptral Distortion (MCD) is computed between the synthesized and original wave file. MCD is computed as given in equation below.

$$MCD = 10/\ln(10) * \sqrt{2 * \sum_{l=1}^{25} (c_l^s - c_l^o)^2} \qquad (10)$$

where c_l^s and c_l^o denotes the l^{th} coefficient of the synthesized and the original wave files, respectively.

Table 2 shows the MCD scores obtained on TTS voices V_r^0, V_r^2 and V_r^2. From Table 2, we can observe that the MCD scores obtained on these three different voices are similar indicating that the voices built from large audio file Φ_r perform as good as that of Θ_r.

7.3.2. Subjective evaluation of a TTS voice

In order to evaluate the TTS voices V_r^0, V_r^1 and V_r^2 we have also conducted a perceptual listening test. A set of five speakers (henceforth referred to as subjects) participated in this listening test. A set of 10 utterances were synthesized from these three voices. Each subject was asked to listen to an utterance synthesized by V_r^1/V_r^2 and compare it against the same utterance synthesized by V_r^0. The subject was asked whether there was a difference or no-difference in the pair of utterances. We henceforth refer to this listening test as DND (difference-no-difference) test. Table 3 summarizes the results obtained on 50 utterances (five subjects x 10 utterances). From Table 3, it could be observed that in a majority of utterances the subjects did not find any difference between the voices V_r^1/V_r^2 and V_r^0.

8. Evaluation on EMMA Speech Corpus from Librivox

LibriVox (http://librivox.org) is an on-line resource which provides public domain recordings of a range of fiction and non-fiction works in numerous languages and provides information on where to download the associated text. For our experiments we have collected the recordings of EMMA by Jane Austen. These recordings are done by a female speaker . All the recordings were concatenated to form a large audio file, henceforth referred to as Φ_e, whose duration is 17.35 hours. We downloaded the associated text from Project Gutenberg (http://www.gutenberg.org), 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 utterances, where each utterance is of sentence/paragraph length.

Both SFA-1 and SFA-2 were applied on Φ_e, and CLUSTERGEN voices were built. Let V_e^1, V_e^2 denote the TTS voices built from Φ_e using SFA-1 and SFA-2 respectively. Table 2 shows the MCD scores obtained on TTS voices V_e^1 and V_e^2. The lower MCD scores of V_e^1 and V_e^2 in comparison with MCD scores of V_r^0/V_r^1/V_r^2 could be attributed to the large amount of speech data available in Φ_e as compared to one hour of RMS voice. The fact that a decent MCD score was obtained on V_e^1/V_e^2 indicates that the methods SFA-1 and SFA-2 could be applied for segmentation of large audio files such as EMMA corpus. Table 3 shows the DND listening tests conducted on V_e^1 and V_e^2. The results indicate that in a majority of utterances the subjects did not perceive any difference between the voices V_e^1 and V_e^2.

9. Conclusion

In this paper, we have proposed modifications to the standard forced-alignment technique and showed the proposed modifications could be employed to develop two different methods (SFA-1 and SFA-2) to segment a large audio file. Thus it alleviates the need of a large vocabulary speech recognition system (and the corresponding algorithms) for segmenting a large audio file. More importantly, the methods SFA-1 and SFA-2 enable the forced-alignment algorithm to be used in less resource language where a large vocabulary speech recognition system is not readily available.

10. References

[1] K. Prahallad, A. R. Toth, and A. W. Black, “Automatic building of synthetic voices from large multiparagraph speech databases,” in *Proceedings of INTERSPEECH*, Antwerp, Belgium, 2007.

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

[3] A. Toth, “Forced alignment for speech synthesis databases using duration and prosodic phrase breaks,” in *Proceedings of 5th ISCA Speech Synthesis Workshop*, Pittsburgh, USA, 2004.

[4] I. Trancoso, C. Duarte, A. Serralheiro, D. Caseiro, L. Carrico, and C. Viana, “Spoken language technologies applied to digital talking books,” in *Proceedings of INTERSPEECH*, Pittsburgh, USA, 2006.

[5] 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 *Proceedings of Int. Conf. Spoken Language Processing*, Sydney, Australia, 1998.

[6] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition.* Prentice-Hall, 1993.

[7] A. W. Black, “CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling,” in *Proceedings of INTERSPEECH*, Pittsburgh, USA, 2006.