UDC 004.421.6, DOI: 10.2298/CSIS1001075T

A Dynamic Alignment Algorithm for Imperfect Speech and Transcript

Ye Tao[12], Xueqing Li[1], and Bian Wu[2]

[1]Shandong University, Department of Computer Science and Technology,
Ji Nan, 250101, P. R. China
[2]Shanghai Qitai Internet Technology Co. Ltd.,Shanghai, 201203, P. R. China

Abstract. This paper presents a novel alignment approach for imperfect speech and the corresponding transcription. The algorithm gets started with multi-stage sentence boundary detection in audio, followed by a dynamic programming based search, to find the optimal alignment and detect the mismatches at sentence level. Experiments show promising performance, compared to the traditional forced alignment approach. The proposed algorithm has already been applied in preparing multimedia content for an online English training platform.

Keywords: Text-Audio Alignment; Dynamic Programming

1. Introduction

The motivation of this research comes from a content producing module of an English tutor platform [1], which tests and evaluates learners' spoken English level as a foreign language. To perform pronunciation analysis, it is necessary to have the time-aligned word/phoneme transcriptions with audio data. In fact, time-aligned labels can be used not only in the area of language training [11] [3], they are also required by audio/video indexing techniques applied in search engines [15] [4] [5]. Moreover, as a fundamental task in speech processing, it could be useful in model training for Automatic Speech Recognition (ASR) systems [21]. Though aligning speech with its corresponding text might seem a solved problem [8], situation could be difficult if the transcription does not match the media, as the decoder is forced to accept the input transcription. Unfortunately, audio files and their transcriptions are not fully-matched in many cases. Alignment of audio and text with imperfections has applications in subtitling, spoken books and etc. For example, a dialogue script often includes speaker names which are not uttered in audio to make it more readable, and broadcast/TV news scripts sometimes skip speech from interviewees. Manually scanning the discrepancies is a tedious and time-consuming work that requires skill.

Recently, alternate approaches have been investigated to align the speech with its approximate text. Moreno et al [10] developed a recursive method to progressively reduce the forced alignment process with a gradually restricting dictionary and language model. A similar approach has been presented in [7] for human generated transcriptions of audio files. These techniques are based on the methodology of comparing ASR result with the approximate transcript. However, the recognition quality highly depends on the acoustic model, and thus could be severely degraded without speaker adaption and appropriate model training. In contrast, most Viterbi-based forced alignment algorithms give satisfactory result, even the acoustic environment of the input speech is quite different from the one used for model training. Another limitation of Moreno's approach is that the anchor selector is difficult to handle repeated words, phrases and sentences, which are widely used in language tutor. In this case, some recognized anchors can become ambiguous in the original text, and cause alignment errors. Experiment in section 4 indicates that the performance of this approach is relatively low on teaching materials and online courses, even for perfect matched audio and text.

Other authors have included HMM garbage models to allow for text/speech skips and substitutions [12], which works well in discovering and correcting low level (word/phrase) errors, where most parts are matched. The approach presented in this paper focuses more on detecting the insertion and deletion errors at sentence/paragraph level. We convert the alignment problem into a series of overlapping sub-problems, which are solved recursively by a dynamic programming algorithm. The algorithm has been implemented in a content producing system, processing speeches from varied sources, such as news, online courses, lectures and etc.

The rest of the paper is organized as follows. Section 2 introduces the methodology of sentence boundary detection. Section 3 focuses on the dynamic alignment algorithm and the pruning policy. Section 4 gives some preliminary experiment results and section 5 concludes the paper.

2. Pre-processing

Input audio and text are required to be segmented into small (e.g. sentence) unit. Transcript can be segmented by using the maximum entropy approach [2], which is one of the state-of-the-art natural language processing techniques. Speech sentence boundary detection is much more challenging, since typical cues in text (e.g. headers, paragraphs, punctuation and etc.) are absent in utterances [17]. Quite a few jobs have been done in automatic detection of prosodic boundaries in speech [20] [16]. We use a multi-stage pre-processing approach to find the approximate sentential boundaries, as shown in Fig. 1.

1. The source audio file provided by user often contains non-speech parts, e.g. lectures with prelude and epilogue music are very common in multimedia courses. Firstly, these non-speech clips are separated and removed. Different strategies have been investigated for speech/non-speech detection during the last decade [9]. And our previous research [19]

also proposed a fuzzy logic based approach that combines different features to label the boundaries of voice segments.

2. Pauses detection on speech can typically be done with a high accuracy off-line VAD algorithm. Numerous solutions have been reported to achieve precise detection results [13]. The method we used in this system is based on the Order Statistics Filtering Sub-Band Spectral Entropy [6], which measures the sub-band spectrum divergence between speech and background noise. Long-term speech features [14] can also be considered as contextual information to estimate the threshold more precisely and benefits for detecting speech presence in noisy environments.
3. Finally, boundaries detected by the VAD algorithm need to be filtered to get the prosodic boundaries. It has been studied in previous research that a sentence boundary is often marked by some combination of a long pause, a preceding final low boundary tone, and a pitch range reset [18]. Therefore, pause information can be important cues to eliminate inner-sentence boundaries.

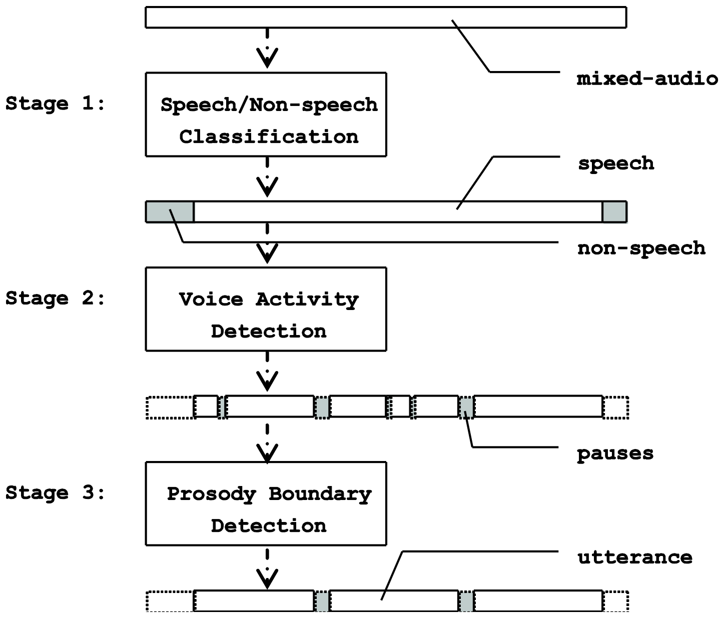


Fig. 1. 3-stage sentence boundary detection

Sentence segmentation is a very difficult problem in spontaneous speech such as lectures, and thus addressed by many works recently. It is easily possible that sentential boundary detection based on prosodic cues can produce errors causing split or concatenated sentences within this pre-processing stage. An algorithm designed for correcting these errors (including false alarms and missed alarms) is presented in the following section.

3. Dynamic Alignment

Let *s*(*i*,*j*) and *t*(*m*,*n*) denote the utterance and transcript whose boundaries are positioned at *i*, *j* and *m*, *n*, where *i* ≤ *j* and *m* ≤ *n*. We are going to find the most feasible sentential matches between the text and audio, i.e.

$$\underset{1\le i\le j\le N,1\le m\le n\le M}{\arg\max}\sum s\left(i,j\right)\leftrightarrow t\left(m,n\right), \qquad (\mathbf{1})$$

where *M* and *N* are the number of segments in speech and text, *s*(*i*,*j*)↔*t*(*m*,*n*) measures the likelihood of the alignment.

3.1. Dynamic programming

If the audio and text are presented in order, the problem can be broken up into stages with an alignment required at each stage. Let the ordered pair (*h*,*k*) denotes a hypothetic alignment for *s*(*0*,*h*)↔*t*(*0*,*k*). To find the best solution at stage (*h*,*k*), it is necessary to go through all the possible matches in previous stages, and see how to make an alignment for the remainder. Denoted by *F*(*h*,*k*) as the maximum similarity accumulation at stage (*h*,*k*), we have the following induction:

$$\begin{aligned}&F\left(h,k\right)=\max_{t_h,t_k}F\left(h-t_h,k-t_k\right)+P\left(h-t_h,h,k-t_k,k\right),\\&\left(0<h<N,0<k<M,t_h\in\left[0,\varepsilon\right],t_k\in\left[0,\varepsilon\right],t_h+t_k\neq 0\right)\end{aligned}\qquad (\mathbf{2})$$

where *ε* is the width of search beam. *P*(*i*,*j*,*m*,*n*) computes the acoustic likelihood of the alignment for *s*(*i*,*j*)↔*t*(*m*,*n*), which indicates the strength of belief that how much they are matched.

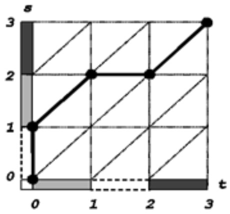
3.2. Alignment function

Fig. 2 presents the alignment result of an utterance and its corresponding transcript, where the value of *P* is computed as follows:

$$\begin{aligned}&P\left(i,j,m,n\right)=\left\{\sum_{\left[f_i,f_{bs}\right]\cup\left[f_{be},f_j\right]}\omega_b\times Acc+\sum_{\left[f_{bs},f_{be}\right]}\omega_c\times Acc\right\}/\left(f_j-f_i\right),\\&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\left(i<j,m<n\right)\end{aligned}\qquad (\mathbf{3})$$

where f_i and f_j are the start/end frame indices of the utterance *s*(*i*,*j*), f_{bs} and f_{be} define the boundaries frame indices, ω_b and ω_c specify the weights of the boundaries and internal parts respectively, and *Acc* is the normalized acoustic score.

Acoustic score value indicates the likelihood that a speech segment represents a particular symbol according to the statistical models. However, the value depends on the length of the segment, and thus needs to be normalized for a particular segment, which simply entails dividing the score by the number of frames contained in the segment. *Acc* thus represents the average log likelihood per frame for the given segment, and can be used to compare speech segments of different lengths (typically different phones) to determine which segments fit better. In addition, *P* needs to be distinguishable between the fully-matched (e.g. *s*(1,2)↔*t*(0,1)) and partially-matched pairs (e.g. *s*(0,2)↔*t*(0,1)), and thus weights are at boundaries to guide the solver towards the global optimal solution.



P	***t*(0,1)**	***t*(0,2)**	***t*(0,3)**	***t*(1,2)**	***t*(1,3)**	***t*(2,3)**
***s*(0,1)**	0	0	0	0	0	0
***s*(0,2)**	516	0	0	507	0	508
***s*(0,3)**	159	171	0	143	161	152
***s*(1,2)**	**640**	0	0	0	0	0
***s*(1,3)**	280	293	0	266	289	279
***s*(2,3)**	0	0	0	0	0	**667**

Fig. 2. An example of dynamic alignment. *s*(0,1) is unuttered and *t*(1,2) is untranscribed

3.3. Sentential Boundary Correction

As discussed earlier, speeches (e.s.p. unprepared speech or conversational speech) often contains pauses due to speech errors, false starts, train-of-thought gaps and etc, which causes false and missed alarms in sentential boundaries detection in pre-processing stage. However, it is possible to correct most of these errors by scanning and comparing the forced alignment results.

False alarm

We compare the alignment results of successive speeches upon the same text ($w_1,\ldots,w_N$), *i.e.* from *P*(*i*, *i*+1, *m*, *n*) to *P*(*i*, *i*+*L*, *m*, *n*), where *L* is a pre-defined value to control the size of the search window, as shown in Fig. 3(a). When a monotonically increasing is detected, i.e.

$$P(i,i+1,m,n) < \cdots < P(i,i+L,m,n) \tag{4}$$

which indicates that alignments are stably improved by extending the speech, we then remove the last L_t inner boundaries, where $0 \leq L_t < L$ is a threshold.

Missed alarm

Fig. 3(b) shows an example of the missed alarm detection. Our forced alignment engine uses a generic speech model (garbage model) to absorb the out-of-vocabulary words, thus the notable detected silence at the end of the utterance indicates the existence of a text-skip, and in this case, a new sentence boundary should be inserted at the end of word w_N.

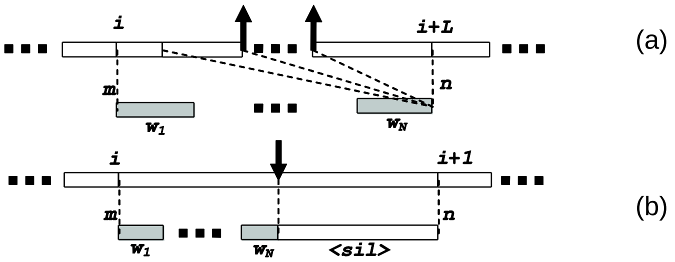


Fig. 3. Sentential boundary correction. (a) false alarm. (b) missed alarm

3.4. Pruning strategy

Pruning strategies are applied to eliminate most unlikely search hypotheses. A letter-to-sound algorithm can be used to predict the length of a sentence, by simply counting the number of words and phones. This could be helpful to avoid attempts on most impossible alignments. If continuous confidence score descents are detected on two utterances upon the same text, e.g. P(1,2,0,1) and P(1,3,0,1) in Fig. 2, we then eliminate any successor, by setting P to 0. Due to the antithesis of this problem, another pruning rule can be applied in text domain, as stated below, where γ* are pre-defined thresholds.

$$\begin{gathered} P(i,j,m,n)-P(i,j-1,m,n)\le\gamma_1\to P(i,x,m,n)=0,\\ P(i,j,m,n)-P(i,j,m,n-1)\le\gamma_2\to P(i,j,m,y)=0,\\ \left(0<i<j<N,0<m<n<M,x\in\left[j+1,M\right],y\in\left[n+1,N\right]\right). \end{gathered}\qquad(5)$$

Moreover, to accelerate the solving speed, we save those *P* and *F* we have already computed. If we need to solve the same problem later, we then retrieve and reuse our already-computed values.

4. Experiment and Discussion

Alignment performance was evaluated on a data set collected from lectures (15%, speeches and interviews), multimedia courses (65%, English teaching materials for K-12 students), broadcasting/television news (20%, live news and BBC/VOA special programs). Table. 1 summarizes the experimental data set, where each type of data is a composition of clips that are fully-matched (CP), and clips that contains mismatched parts (CI), to test the robustness and compatibility of the proposed algorithm.

Table 1. Data set and comparison exepriment results.

types	clips	sentences P/I	length (min.)	F.A	A.A	D.A	
						$\mathcal{E} = 1$	$\mathcal{E} = 2$
lectures	P	37/-	7	37(100%)	30(81%)	33(90%)	34(92%)
	I	38/14	9	33(63%)	39(75%)	45(86%)	48(92%)
news	P	32/-	10	32(100%)	28(88%)	26(81%)	28(88%)
	I	39/16	11	28(50%)	39(71%)	47(74%)	47(74%)
courses	P	80/-	29	80(100%)	61(76%)	69(86%)	73(91%)
	I	49/44	37	54(58%)	69(74%)	73(78%)	77(83%)

Fig. 4 shows the sentence boundary detection results, where false alerts were the major source of errors for the pure VAD algorithm (stage 1). Richness of such errors is related to the corpus we chose, as many of the speeches are designed for teaching and thus contains long inner-sentence pauses. The existence of missed boundary error often related to the variability in the user speaking state, e.s.p. when the user tends to speed up the speech at the end of a sentence. Pause (stage 2) and pitch (stage 3) information are helpful to reduce the false alarm rate. And the VAD correction algorithm (Section 3.3) also provides a sustained improvement in both sentence boundary hit rate and false alarm rate over the 3-stage sentence boundary detection algorithm.

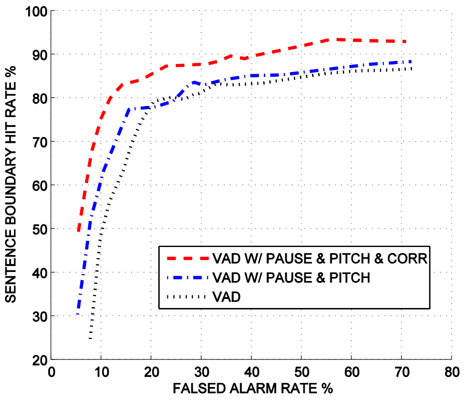


Fig. 4. Results of sentence boundary detection

The acoustic score in *P* is achieved with a basic speaker independent recognizer tuned to run in the forced alignment mode. An alignment of a speech-text pair *s*(*i*,*j*)↔*t*(*m*,*n*) is defined as successful, when the normalized confidence score of exceeds a threshold, i.e. *P*(*i*,*j*,*m*,*n*)>*T*, where *T* is a pre-defined value to guarantee that there is no alignment flaws.

As shown in Table. 1, we found that the anchor-based algorithm (A.A) may fail to give the expected result for fully-matched test cases, when the news and lectures are record in a noisy environment. In particular, the performance of A.A. degrades seriously on those clips designed for teaching and learning, due to a large amount of repeat words and phrases. For speech and text that are not well-matched, our approach significantly increases the ratio of successful alignment, compared to the traditional forced alignment.

An examination of the results shows that most failures are caused by the consecutive mismatches. Performance highly depends on the quality of the sources, e.s.p. the ratio of the mismatched parts, and *ε* a trade-off between accuracy and speed. Better results can be achieved by setting a wider search beam, e.g. changing *ε* from 1 to 2 will increase the correct alignment radio, it will however direct the algorithm to try more possibilities and slow down the alignment process. In general, on each stage, the number of search paths *C* and width of search beam follows:

$$C = \varepsilon^2 + 2\varepsilon \qquad (\mathbf{6})$$

Pruning policies limits the range of alignments and removes most infeasible searches, and it can reduce the amount of computation, as shown in Fig 5.

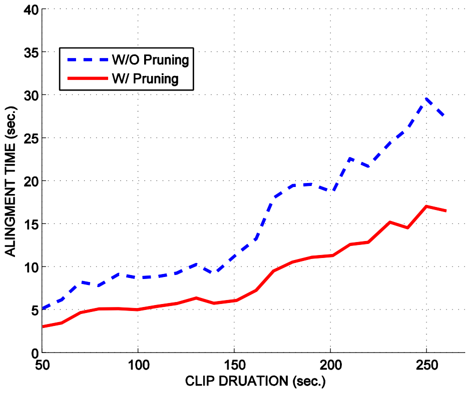


Fig. 5. Results of pruning. ε = 1, evaluated on a P4 2.4GHz computer with 2G RAM installed

5. Conclusion

We introduced an approach for the temporal alignment of speech with imperfect transcripts, based on the acoustic likelihood marked chunks of speech signals, that are associated with partitioned audio segments, and a word level symbol sequence given in the erroneous transcription. In this paper, speech and text are first segmented into units, which are then aligned with a dynamic alignment algorithm. The proposed algorithm has been implemented and validated by an easy-to-use content producing tool for preparing multimedia content for English training. The experiment result shows an increase of the correct matching ratio, in particular for those clips whose speech and transcription are not well-matched, compared to the traditional forced alignment approach.

Though most speech and text are presented in order, a limitation of the algorithm is it is not efficient for the re-ordering of phrases or sentences in transcription. Future investigations also include launching ASR on only the mismatched parts, using the idea described in [10] in conjunction to correct the errors and accelerate the detection speed.

6. References

1. http://www.mytalkpal.com.
2. Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. Computational Linguistics, 22(1), 39-71. (1996)
3. Jared Bernstein, Michael Cohen, Hy Murveit, Dimitry Rtischev, and MitchelWeintraub. Automatic evaluation and training in english pronunciation. In Proc. of ICSLP 90, 1185-1188. (1990)
4. Satya Dharanipragada, Martin Franz, and Salim Roukos. Audio-indexing for broadcast news. In Proc. of TREC6, 115-119. (1997)
5. Jonathan Foote. An overview of audio information retrieval. ACM Multimedia Systems, 7, 2-10. (1999)
6. Zhang Ya-Xin Lv Yue Guo Li-Hui, He Xin. An order statistics filtering-based real-time voice activity detection algorithm. ACTA AUTOMATICA SINICA, 34(4), 419-425. (2008)
7. Timothy J. Hazen. Automatic alignment and error correction of human generated transcripts for long speech recordings. In Proc. of ICSLP 06, 1606-1609. (2006)
8. Picone J., Goudie-Marshall K., Doddington G., and Fisher W. Automatic text alignment for speech system evaluation. IEEE Transactions on Acoustics, Speech and Signal Processing, 34(4), 780-784. (1986)
9. Lie Lu, Hong-Jiang Zhang, and Hao Jiang. Content analysis for audio classification and segmentation. IEEE Transactions on Speech and Audio Processing, 10(7), 504-516. (2002)
10. P. Moreno, C. Joerg, J.-M. Van Thong, and O. Glickman. A recursive algorithm for the forced alignment of very long audio segments. In Proc. of ICSLP 98, 2711-2714. (1998)
11. Leonardo Neumeyer, Horacio Franco, Mitchel Weintraub, and Patti Price. Automatic text-independent pronunciation scoring of foreign language student speech. In Proc. of ICSLP 96, 1457-1460. (1996)
12. Long Nguyen and Bing Xiang. Light supervision in acoustic model training. In Proc. of ICASSP 04, 1, 185-188. (2004)
13. J. Ramirez, J. M. Gorriz, and J. C. Segura. Robust Speech Recognition and Understanding, Chapter 1: Voice Activity Detection. Fundamentals and Speech

Recognition System Robustness, 460-481. I-Tech Education and Publishing. (2007)

14. J. Ramirez, J. C. Segura, Carmen Benitez, Angel de la Torre, and Antonio Rubio. Efficient voice activity detection algorithms using long-term speech information. Speech Communication, 42(3-4), 271-287. (2004)
15. Deb Roy and Carl Malamud. Speaker identification based text to audio alignment for an audio retrieval system. In Proc. of ICASSP 97, 1099-1102. (1997)
16. Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tur, and Gukhan Tur. Prosody-based automatic segmentation of speech into sentences and topics. Speech Communication, 32(1-2), 127-154. (2000)
17. Mark Stevenson and Robert Gaizauskas. Experiments on sentence boundary detection. In Proceedings of the sixth conference on Applied natural language processing, 84-89. (2000)
18. Marc Swerts and Mari Ostendorfc. Prosodic and lexical indications of discourse structure in human-machine interactions. Speech Communication, 22(1), 25-41. (1997)
19. Ye Tao, Daren Zu, Xueqing Li, and Ping Du. A fuzzy logic based speech extraction approach for e-learning content production. In Proc. of ICALIP 2008, 1, 298-302. (2008)
20. Arthur R. Toth. Forced alignment for speech synthesis databases using duration and prosodic phrase breaks, Fifth ISCA Workshop on Speech Synthesis, (2004)
21. Steve Young, Gunnar Evermann, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. The HTK Book (for HTK Version 3.2.1). Cambridge University Engineering Department, (2002)

Ye Tao is working for his PhD in Department of Computer Science and Technology, Shandong University, P.R.China. He obtained M. Tech. Degree in software engineering from Shandong University. His areas of interests are Speech Processing and Computer Graphics.

Xueqing Li is a Professor in Department of Computer Science and Technology, Shandong University, P.R.China. He obtained BSc, MSc, and PhD all from Shandong University. His research interests include Human Computer Interaction & Virtual Reality, Computer Graphics, Image Processing and Computer Geometry.

Bian Wu is the CTO of Shanghai Qitai Internet Technology Co. Ltd., P.R.China. He obtained PhD from Shanghai Jiaotong University. His research interest is Speech Processing.

Received: July 25, 2009; Accepted: October 28, 2009.