WORD LEVEL LYRICS-AUDIO SYNCHRONIZATION USING SEPARATED VOCALS

Sang Won Lee

University of Michigan
Computer Science and Engineering
Ann Arbor, MI 48109
snaglee@umich.edu

Jeffrey Scott

Gracenote, Inc.
Emeryville, CA 94608
jscott@gracenote.com

ABSTRACT

The massive amount of digital music data available necessitates automated methods for processing, classifying and organizing large volumes of songs. As music discovery and interactive music applications become commonplace, the ability to synchronize lyric text information with an audio recording has gained interest. This paper presents an approach for lyric-audio alignment by comparing synthesized speech with a vocal track removed from an instrument mixture using source separation. We take a hierarchical approach to solve the problem, assuming a set of paragraph-music segment pairs is given and focus on within-segment lyric alignment at the word level. A synthesized speech signal is generated to reflect the properties of the music signal by controlling the speech rate and gender. Dynamic time warping finds the shortest path between the synthesized speech and separated vocal. The resulting path is used to calculate the timestamps of words in the original signal. The system results in approximately half a second of misalignment error on average. Finally, we discuss the challenges and suggest improvements to the method.

Index Terms— Lyric Music Alignment, Vocal Separation, Synthesized speech.

1. INTRODUCTION

Automatically scrolling lyrics are useful in music listening applications such as streaming services and karaoke, however the process of aligning lyrics to music is a challenging task to automate. Alignment can be done on a phrase, word or phoneme level with each particular problem providing unique challenges. Overall, lyrics-audio alignment is a complex problem that requires an integrated approach involving various techniques in music information retrieval: vocal separation, music segmentation, speech modeling, pitch detection and rhythmic analysis.

Assuming we are given a text document of lyrics and an associated audio file, our task is to assign timestamps in the audio stream with the starting point of each lyric in the text document. Current applications that would benefit from automating this process include scrolling lyrics displays for streaming music, audio thumbnailing and karaoke.

One common approach to lyric alignment is to apply a speech recognition model and use the model output to align the audio signal to the lyric text [1]. A caveat of this approach is that the acoustic model used in the alignment is trained on speech signals and requires a large amount of manually annotated data to train the model on singing voices. Previous works use a model trained on speech data and apply various techniques that can improve the overall accuracy of the model on a singing voice. One of the common improvements made to this approach is to incorporate domain knowledge of music (temporal and structural) [2, 3, 4, 5]. In addition, the speech model can be adapted to singing voice with manually annotated labels [1, 6, 7]. Other approaches leverage multi-modal inputs available online that have additional temporal information regarding the lyrics (e.g. guitar tabs, lyric synced MIDI file) [8, 9]. However, it can be difficult to find this additional information for all songs. Other researchers investigated translating the lyrics into a common language to utilize the language-specific acoustic models [6, 7].

A number of lyrics-audio alignment systems take a hierarchical approach to first match a music segment (or line) with a block of lyric text and then run a more fine level alignment (line, word or syllable) with the given result [2, 3, 4, 10]. While there is variance in each work, in general, the standard approach is to use dynamic programming search (or forced alignment) with a hidden Markov model framework.

The arguably most advanced algorithm in lyrics-audio alignment incorporates various techniques: vocal separation/segmentation, harmonic structure analysis, unvoiced consonants detection, modified hidden Markov model, adaptation of a phoneme model for singing voice, and Viterbi alignment, resulting in 85.2% accuracy [11].

Our work follows a similar framework to [5] where we assume that we have large scale alignment similar to the results presented in [4]. From there we attempt a finer grain word level alignment. Since we wish our approach to be language-agnostic, we prefer to avoid training our own language-specific model for the sung vocals. We would like

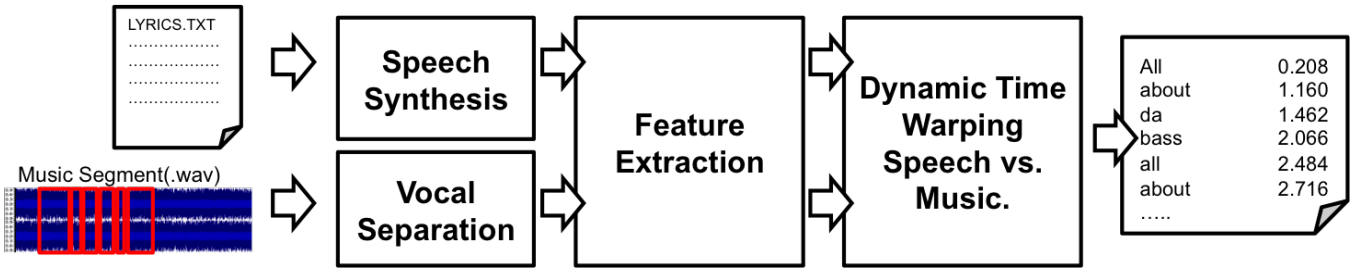

Fig. 1. System diagram for lyrics synchronization. Given lyrics-audio segment pairs, the system returns the timestamps for words in the lyrics.

to just as easily match English lyrics with sung English vocals as we can match Spanish lyrics with the same underlying model. As presented later, we do use language specific *speech generation* models, but they are readily available for use.

Our approach to the problem focuses on developing a system that is practical and scalable for large databases common in commercial companies that deal with digital media. We focus on unsupervised methods since to our knowledge no large database of word level aligned lyrics and audio exists.

2. METHOD

As stated in Section 1, this work approaches the lyrics-audio alignment task under the assumption that for a given audio file and associated lyric file, there exists a paragraph/segment pairing. That is, each paragraph in the lyrics is associated with the proper segment in the audio that contains the words for the given paragraph. We are focusing on automatically producing timestamps where each word in the paragraph occurs in the audio segment.

Figure 1 depicts the steps to generate word level alignment from a given music/text segment pair. The music segment is processed to remove as much of the background accompaniment as possible. This is accomplished by using a speech/music classifier as well as vocal source separation techniques. The text segment is processed using a Text-To-Speech (TTS) algorithm with the word rate and gender adjusted to account for the length of the audio segment and pitch range, respectively. Once we have the processed music segment and the synthesized speech audio, we extract a set of features from each signal and use Dynamic Time Warping (DTW) to align the features from the synthesized speech to the features from the processed musical segment. Since the timestamps of each word are known in the synthesized speech, we map them to locations in the audio segment.

One key component of the system is to process the musical segment to make it resemble the synthesized speech segment. Similarly, the synthesized speech segment is processed to increase its similarity with the processed music segment.

2.1. Music Segment Processing

The preprocessing of the music segment prior to feature extraction and dynamic time warping are designed to increase the similarity of the music segment to the synthesized speech segment. This is a two step process consisting of vocal separation and vocal/non-vocal detection.

2.1.1. Vocal Separation

To separate the vocals we utilize the multi-pass median filtering process described in [12]. This version of the popular Harmonic Percussive Source Separation (HPSS) algorithm comprises a first pass percussive separation using a very high frequency resolution and then a second stage HPSS filter using a lower frequency resolution. Since vocal signals have harmonic energy that is spread across more frequency bins compared to instrumental signals, a significant portion of the vocal energy will survive the first stage of horizontal filtering. The percussive output, P contains percussive and vocal sources. The output of the percussive filter is then put though a harmonic (horizontal) filter using a lower frequency resolution to remove the background percussive elements present from the previous step. For more details see [12]. The end result is a signal that is mostly vocal with some background elements still audible.

The first pass vertical filter uses a Discrete Fourier Transform (DFT) size of 16,384, hop size of 512 and filter width of 17. The low frequency resolution filter has a DFT size 1024, hop size of 512 and filter width 17.

Now that our music signal resembles the speech signal more due to the removal of significant amounts of background noise (instrumental accompaniment), we now must account for some of the fundamental differences between spoken language and vocal melody. The next step is designed to handle the difference in time between our speech and music signals. The speech is synthesized using the Google WebSpeech API and is generated using a constant word rate that is designed to model spoken language pauses and cadence[1]. The rhythm of sung vocals is significantly different, incorporating longer pauses and extended vowel sounds over numerous beats.

[1] http://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html

Initial experiments revealed that a significant source of error was due to the alignment process being 'confused' by large pauses between lines or phrases in the sung vocal lines. To alleviate this issue, we employ a vocal/non-vocal detection scheme to use as a feature in DTW.

2.1.2. *Vocal/non-vocal Estimation*

The vocal/non-vocal detection is based on work in [13] and uses a Gaussian Mixture Model (GMM) as a vocal classifier and a second GMM for the non-vocal classifier. The vocal model outputs a single probability $P_{vocal}(k)$ of whether the frame, k, is vocal. The non-vocal classifier outputs $P_{non-vocal}(k)$, the probability that the frame contains no vocals. Using these two models a soft decision about the frame is given as

$$V_{soft} = \frac{P_{vocal}(k)}{P_{vocal}(k) + P_{non-vocal}(k)}. \quad (1)$$

We utilize the soft weight in Equation 1 as one of the features in our time alignment. To train the GMMs we used 108 songs (7 hours 17 minutes total) from the MedleyDB database [14]. The songs are labeled with activation times for each instrument.

Vocal model training is performed on the separated vocal tracks computed as described in Section 2.1. This is done to best approximate the signal characteristics of the data we want labeled by the vocal/non-vocal classifier. To train the vocal model, we compute MFCCs (13 dimensions) using a DFT size of 2048 and hop of 512. We then compute the first and second derivatives of the MFCCs and concatenate the MFCCs, Δ MFCC and Δ^2 MFCC together to form a 39-dimension vector as our input to the GMM. We use k = 64 mixture components and initialize each distribution using k-means, k being chosen from [1]. Lastly, we smoothed the V_{soft} vector with a moving average filter of length 23, which was chosen from validation.

2.1.3. *Feature Selection*

Features used in addition to the vocal estimation and MFCCs (Δ and Δ^2) are Moving Average Crossing Rate and Moving Average Subtracted Variance [15]. Stacking these features yields a 42-dimensional vector. We employ sequential feature selection to reduce the amount of features used.

2.2. Synthesized Speech Processing

The speech synthesis is performed in a manner that the synthesized speech approximates the extracted vocal signal as close as possible. The speaking rate is adjusted so that the overall time of the synthesized speech is the same as the target music segment. The gender of the speaker is *manually* selected for each example. After the speech signal is generated we compute the same features as in Section 2.1.3 above.

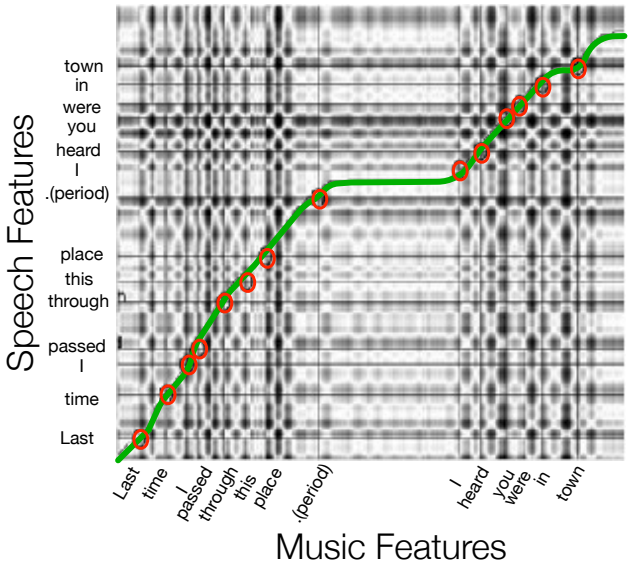

Fig. 2. Similarity matrix and word alignment for the song *Word Gets Around* by St. Vitus. The speech signal is on the vertical axis and the music segment on the horizontal axis. The green line is the shortest path found by DTW and the circles depict the ground truth location for each word.

This manual selection of gender is recognized by the authors as a significant impairment to a fully automated system. One method of automating the gender selection would be to use the predominant pitch range [13, 16].

3. RESULTS

We use a small dataset of labeled vocal onsets that we have to tune the parameters of our model. This is a subset of the multi-track dataset in [14]. The particular songs were chosen based on availability of lyrics and genre diversity and the lyric onsets were hand labeled by the authors.

Figure 2 depicts a least cost path through a distance matrix. The red circles indicate the ground truth point for each lyric in both the separated vocal and synthesized speech. Ideally, our least cost path would pass through each point exactly. The green line in the figure is the actual path computed using DTW and shows how the system is able to wait during an instrumental break between phrases before continuing tracking the lyric.

For the test set, we choose a small set of popular music. In particular, we attempt to add songs that we believe will be challenging for various reasons: polyphonic vocal segments (*ABBA*), non-English lyrics (*2NE1*), unique vocalist (*Daft Punk, Michael Jackson*) and genre variety (*Metallica*, *Notorious B.I.G*). For each song, we choose a short segment (typically from the first verse to the end of the first chorus), the length of which are ranged from 29 to 79 seconds. The

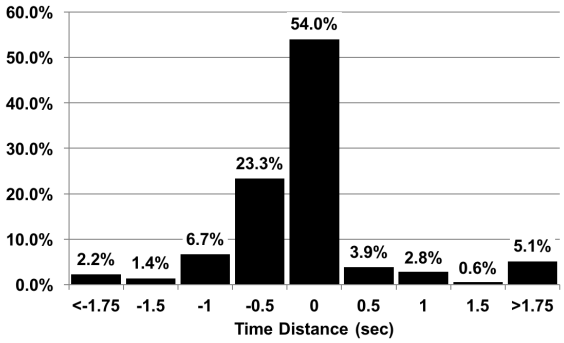

Fig. 3. Histogram of distances between the ground truth and predicted time stamps for each word in the test set.

Artist	Song Name	(seconds) Avg	Std Dev	Max
2NE1	I'm The Best (Korean)	0.632	0.662	2.136
ABBA	Dancing Queen	0.652	0.812	3.255
Alanis Morrisette	One Hand In My Pocket.	0.297	0.484	2.314
Daft Punk	Harder Better Faster	0.346	0.679	3.102
Metallica	Enter Sandman	2.622	1.691	5.034
Michael Jackson	The Way You Make Me Feel	0.111	0.123	0.556
Notorious B.I.G.	Big Poppa	0.308	0.414	1.862
Smashing Pumpkins	1979	0.371	0.717	3.573
Taylor Swift	Shake It Off	0.29	0.272	0.984
Total Error (word-level)		0.469	0.844	5.034
Total Error (line-level)		0.525	0.849	4.516

Table 1. Mean, Standard Deviation and Maximum of absolute error (in seconds) for each song in the test dataset.

segmentation was done manually so that both the test set and the validation set removes any error that can potentially propagate from the lyric-segmentation paring problem. The word-level alignment results synchronized with the all songs can be found in an interactive demo here[2].

To validate the results, we use the average absolute difference between the ground truth and predicted timestamp for each word as a performance measure. Statistics of lyrics-audio alignment for each song in validation set are shown in Table 1. The mean absolute error was 0.469 seconds with the standard deviation of 0.844 seconds. The median absolute error is 0.142, which indicates that, nearly half of the times, the estimated timestamps are within 150 msec window of the true timestamps. The distribution of errors is shown in Figure 3, the width of each bin being 500ms. 87.2% of all words are within +/- 1 second error range.

3.1. Alignment Errors

Typically, we observed that the method performs better in the songs in which the vocal segments and non-vocal segments are evenly distributed and non-vocal segments are short. For example, the best result was shown in the song *The Way You Make Me Feel* by Michael Jackson, where non-vocal segments between lines are short and regular over the selected snippet file. Additionally the vocal is very prominent and clear. The performance was poor for Metallica's *Enter Sandman* as the time interval between each lyrical line is long and also varies within a verse. This is compounded by the usual problem of heavy distorted guitar associated with Metal music. The selected segment contains verse, pre-chorus and chorus, and they differ in their temporal vocal patterns. In the first part of *Harder Better Faster* by Daft Punk, timestamps of words after instrumental segments are estimated earlier than the ground truth. In the second half of the segment, the predicted timestamps are much more accurate as the instrumental segments are very short. This particular error is due to the speech segment matching with an instrumental segment of music. The instrumental segment is too long for DTW to find a path without entering the speech segment. Additional improvements may be gained from emphasizing the relative importance of the vocal/non-vocal classification in the DTW distance function.

[2] http://sangatgracenote.github.io/lyric.html

4. DISCUSSION

In this work we present a system for automatically aligning a lyrics text file to an associated audio stream. We work under the assumption that a rough segment alignment exists between the text and audio. Though our test set is small, this preliminary work shows promise for lyrics-audio alignment across a large corpus. We cannot equivalently compare this result with most of the previous works as most prior works perform alignment on the whole song while this runs within a segment. We found this to be a common theme throughout publications on the topic of lyrics-audio alignment. Also, there is no canonical dataset used for this task as there is in other, more popular, problems. We found a few works that the algorithm was performed in a similar condition (alignment within manually matched segment-lyrics pair). For example, in [17], the algorithm accomplished 1.40 seconds of mean absolute error on line-level alignment. In [7], the system accomplished the mean absolute error of 1.98 seconds on the phoneme-level alignment in Turkish traditional music (a more difficult task than word alignment). Although we cannot make concrete judgment on the performance compared to previous works for various reasons (within-segment, different test sets, different performance metrics, etc.), we do recognize several benefits to the method presented.

The system is simple and blind so that we can attain results without the need for a large, manually annotated training set. While a sufficient dataset is needed for testing, generally the size requirement is much lower for testing than training. Additionally, the algorithm is language-neutral as long as there exists a sufficient speech synthesis engine available for a language.

5. REFERENCES

[1] Hiromasa Fujihara, Masataka Goto, Jun Ogata, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno, "Automatic synchronization between lyrics and music CD recordings based on Viterbi alignment of segregated vocal signals," in *8th IEEE Int. Symp. on Multimedia*. IEEE, 2006, pp. 257–264.

[2] Ye Wang, Min-Yen Kan, Tin Lay Nwe, Arun Shenoy, and Jun Yin, "LyricAlly: automatic synchronization of acoustic musical signals and textual lyrics," in *Proc. 12th Ann. ACM Int. Conf. Multimedia*. ACM, 2004, pp. 212–219.

[3] Denny Iskandar, Ye Wang, Min-Yen Kan, and Haizhou Li, "Syllabic level automatic synchronization of music signals and text lyrics," in *Proc. 14th Ann. ACM Int. Conf. Multimedia*. ACM, 2006, pp. 659–662.

[4] Kyogu Lee and Markus Cremer, "Segmentation-based lyrics-audio alignment using dynamic programming.," in *Proc. 9th Int. Conf. Music Inform. Retrieval (ISMIR)*, 2008, pp. 395–400.

[5] Namunu C. Maddage, Khe Chai Sim, and Haizhou Li, "Word level automatic alignment of music and lyrics using vocal synthesis," *ACM Trans. Multimedia Computing, Commun., and Appl. (TOMM)*, vol. 6, no. 3, pp. 19, 2010.

[6] Kai Chen, Sheng Gao, Yongwei Zhu, and Qibin Sun, "Popular song and lyrics synchronization and its application to music information retrieval," in *Electronic Imaging*. International Society for Optics and Photonics, 2006, pp. 607105–607105.

[7] Georgi Dzhambazov, Sertan Sentürk, and Xavier Serra, "Automatic lyrics-to-audio alignment in classical turkish music," in *The 4th International Workshop on Folk Music Analysis*, 2014, pp. 61–64.

[8] Meinard Muller, Frank Kurth, David Damm, Christian Fremerey, and Michael Clausen, *Lyrics-Based Audio Retrieval and Multimodal Navigation in Music Collections*, pp. 112–123, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[9] Matthias Mauch, Hiromasa Fujihara, and Masataka Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 200–210, 2012.

[10] Gijs Geleijnse, Dragan Sekulovski, Jan Korst, Steffen Pauws, Bram Kater, and Fabio Vignoli, "Enriching music with synchronized lyrics, images and colored lights," in *Proc. 1st Int. Conf. Ambient Media Syst.* ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 1.

[11] Hiromasa Fujihara, Misako Goto, Jun Ogata, and Hiroshi G Okuno, "LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics," *IEEE Journal Select. Topics Signal Process.*, vol. 5, no. 6, pp. 1252–1261, 2011.

[12] Derry FitzGerald and Mikel Gainza, "Single channel vocal separation using median filtering and factorisation techniques," *ISAST Trans. Electron. Signal Process.*, vol. 4, no. 1, pp. 62–73, 2010.

[13] Hiromasa Fujihara, Tetsuro Kitahara, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno, "F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and Viterbi search," in *IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*. IEEE, 2006, vol. 5, pp. V–V.

[14] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research.," in *15th Int. Soc. Music Inform. Retrieval*, 2014, pp. 155–160.

[15] Chao-Ling Hsu, Jyh-Shing Roger Jang, and Te-Lu Tsai, "Separation of singing voice from music accompaniment with unvoiced sounds reconstruction for monaural recordings," in *125th Audio Engineering Soc. Convention*, Oct 2008.

[16] Alain De Cheveigné and Hideki Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoustical Soc. America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[17] Annamaria Mesaros and Tuomas Virtanen, "Automatic alignment of music audio and lyrics," in *Proc. 11th Int. Conf. Digital Audio Effects (DAFx-08)*, 2008.