MIREX 2019 - AUDIO-TO-LYRICS ALIGNMENT CHALLENGE 2019

Emir Demirel
e.demirel@qmul.ac.uk

ABSTRACT

We participate in this year's MIREX (2019) Audio-to-Lyrics Challenge using a traditional forced-alignment technique which adapts a triphone GMM-HMM model trained using context-dependent Singer-Adaptive features on a large open-source Karaoke corpus that has sentence-level annotations provided. Based on this pretrained system, we apply forced alignment to align given lyrics in text format with the music signal. This is an initial system we have proposed for the audio-to-lyrics alignment task. This extended abstract gives the details of our system and concludes with the possible solutions and ideas for achieving a better alignment system.

1. INTRODUCTION

Text alignment to its corresponding verbal content in audio recordings is a crucial step for achieving robust performances in most of the state-of-the-art Automatic Speech Recognition (ASR) systems. In musical domain, the task of text-to-audio alignment is mostly referred as 'Lyrics-to-Audio Alignment', where most of the prior research applied similar techniques to that of speech domain [7] [4] [3].

In our work, we also take on the standard approach of aligning text (or the lyrics) to audio, which employs a context-aware GMM-HMM acoustic phone model that are trained on singer adaptive features [9]. For training, we have used a recently published open-source dataset released by Smule [1] , the Sing 300x30x3 corpus, which consists of solo-singing karaoke recordings of 300 unique arrangements collected from 30 different countries [1]. In order to use word-level data for training phone states, we used the CMU English Pronunciation Dictionary [13] to represent words in sequences of phonemes. A language model trained on lyrics from the Sing 300x30x3 corpus is used for building n-gram word models. For building the speech (lyrics) recognizer, we use Weighted Finite State Transducers (WFST) [8] combining the lexicon, the acoustic and the language models. Finally, we tune some of the hyperparameters that show better performance for our task.

[1] http://www.smule.com

This is an extended abstract for the system submitted for "MIREX 2019 Audio-to-Lyrics Alignment" challenge and is structured as follows: First, we introduce the training dataset used to build the acoustic and the language model. Then we explain the details of the overall pipeline of our alignment system. We, then, conclude with final remarks with a mention of the possible future directions our research would take in order to achieve an improved alignment performance in music recordings with lyrics.

2. DATASET

The DAMP Sing! 300x30x2 corpus [1] is the latest-by-the-date dataset made publicly available for research by Smule. This dataset is a better suited one compared to earlier versions of the DAMP releases due to several factors. First of all, Sing! 300x30x2 provides over 18,676 arrangements of 5,690 popular songs performed by 13,154 performers which has equal number of recordings per gender and country of origin of the performer. This property of this dataset makes it a good balanced set for training. Secondly, the recordings in the dataset are chosen according to the votes cast by the users of Smule app which provides us the assumption of recordings being at least moderately good quality. Moreover, the dataset has prompt-level time annotations for the utterances to be sung. For utilization we have used the preprocessed version of the text data [2].

3. METHOD

The alignment of the lyrics to music recordings is performed using a forced-alignment strategy which requires a pretrained speech recognizer. In our system, we use Kaldi ASR Toolkit [10] for building the automatic speech recognizer, which is an open-source toolkit for speech recognition which has numerous features and functions that are used in many state-of-the-art ASR systems.

3.1 Pretrained Speech Recognizer

We trained an automatic speech recognizer following the traditional approach of building context aware phone acoustic models based on a GMM-HMM architecture. The fundamental idea behind an automatic speech recognition system is to create a model that successfully predicts the sequence of words given an audio recording of an utterance. If the sequence of words are defined as,

$$\mathbf{W} = W_1 W_2 ... W_N$$

and the feature vectors,

$$\mathbf{X}= X_1 X_2 ... X_N$$

then the probability of predicted sequence of words can be defined as;

$$\hat{\mathbf{W}} = \underset{w}{\operatorname{argmax}} P(\mathbf{W}|\mathbf{X}) = \underset{w}{\operatorname{argmax}} \frac{P(\mathbf{X}|\mathbf{W})}{P(\mathbf{X})} \quad (1)$$

$$= \underset{w}{\operatorname{argmax}} P(\mathbf{W}) P(\mathbf{X}|\mathbf{W}) \quad (2)$$

According to the probabilistic model in Equation 1, $P(\mathbf{W})$ is learned as a language model trained on text and $P(\mathbf{X}|\mathbf{W})$ is the acoustic model which can be learned via supervised learning using labeled audio data.

The speech recognition system we inherit uses Weighted Finite State Transducers (WFST) [8] for decoding state transition probabilities to sequences of phoneme / word symbols. WFST uses the composition of the language, lexicon and acoustic components when building the decoding graph.

3.1.1 Acoustic Model

We begin with training a monophone GMM-HMM acoustic model using 13 MFCC features using a 25 milliseconds of window size and 10 milliseconds of hop size, then apply global cepstral mean and variance normalization (CMVN). To include context dependency on the acoustic models, we retrain the model using the delta and delta-delta features, which models the phones as 'triphones'. Dimensionality reduction is then applied to feature vectors using Linear Discriminant Analysis (LDA). We then apply 'feature-space Maximum Linear Likelihood Regression' (fMLLR) [9] transformation to the input features, adapting the GMM parameters for obtaining singer-independent representation of the feature space. Until this stage, the training data used was a small portion of the entire dataset, which consists of the recordings that are obtained from only native-English speaking countries in the dataset (Great Britain, the United States and Australia). We have used such strategy for saving training time. After building context aware triphone GMM-HMM models, we obtain alignments for the entire datasets, which are required for GMM-HMM training, then retrained the acoustic model on the entire dataset (including English recordings from 30 countries). In our experiments, we have seen this way of training has similar, even slightly better performance than training the system on the entire training set from the beginning.

3.1.2 Language Model

The language model is built on the corpus that consists of the lyrics of the songs that are in the DSing! dataset. We have followed similar data clean-up strategies with that of [2]. Some of these strategies include removal of non-ASCII characters and non-lyrics words (such as 'verse' or 'chorus', etc.). We corrected the spelling of certain words in the raw lyrics data that involve repeated vowels indicating a sustain on the corresponding syllable (e.g. 'YEEEEAAAAAH' to 'YEAH'). Numbers are discarded if they are represented as digits. As a result, we obtain 1,747,287 lyrics lines and 91,654 unique words as the text data. We have built a 3-gram maximum entropy Language Model (MaxEnt LM) [6] for our speech recognizer. In general, 4-gram models outperform 3-gram models, yet we have observed the opposite in our triphone GMM-HMM model word error rate (WER) results. We have built the maximum entropy LM using SRILM toolkit [12].

3.1.3 Lexicon

In our alignment system, we use a pretrained triphone acoustic model to predict the sequence of frame-level words and consequently get time alignments. Since the end goal is to perform alignment on the word-level, instead of phoneme-level, a linguistically informed mapping of words to their phonemic representation is required. Such mapping is commonly referred as the (pronunciation) lexicon. In our system, we used the CMU Sphinx English Pronunciation Dictionary [13] as the lexicon for decomposing words into phonemes.

3.2 Alignment

The alignment of phonemes to the audio signal is performed via a 'forced alignment' method. Forced alignment is the procedure of finding the best path from a sequences of target events that minimizes the overall cost. In our system, the phoneme transition probabilities are estimated after training the HMM acoustic states in the speech recognizer described above. We, then, use the Viterbi decoding algorithm for obtaining the most probable chain of phonemes by matching the transition probabilities with the acoustic features extracted from the audio frames. The Viterbi algorithm uses the beam search algorithm for finding the best path, where low probability phone occurrences are pruned to avoid accumulated alignment errors and for memory efficiency.

Beam search method is usually applied in shorter sequences for the ASR task. Typically, a common value for the number of beams varies between 8 to 20 depending on the task and the data. We had to modify this value for own task, due to the average length of audio files in the MIREX evaluation sets. The audio recordings in the dataset are mentioned to have lengths of 4-5 minutes on average. Using the smaller values for the beam search, our speech recognizer fails to align given text to the entire audio recording. For this reason, we have tried to use larger beam lengths. For the initial search, we use a beam length of 500, and retry beam length of 1000 in case the former value fails. Even though using larger beam lengths slows down the alignment procedure, it can achieve alignments on the entire audio recordings.

In the speech recognizer, silence segments '<SIL>' are included in the class set representing a 'silence' phone. Kaldi ASR Toolkit involves an attribute ('boost-silence') that parametrizes the weight assigned to the silence phone. In our experiments, we have observed that reducing the this parameter (from 1 to 0.17) helped aligning different musical structures of the songs (verses, choruses) separately, resulting in less accumulated alignment errors.

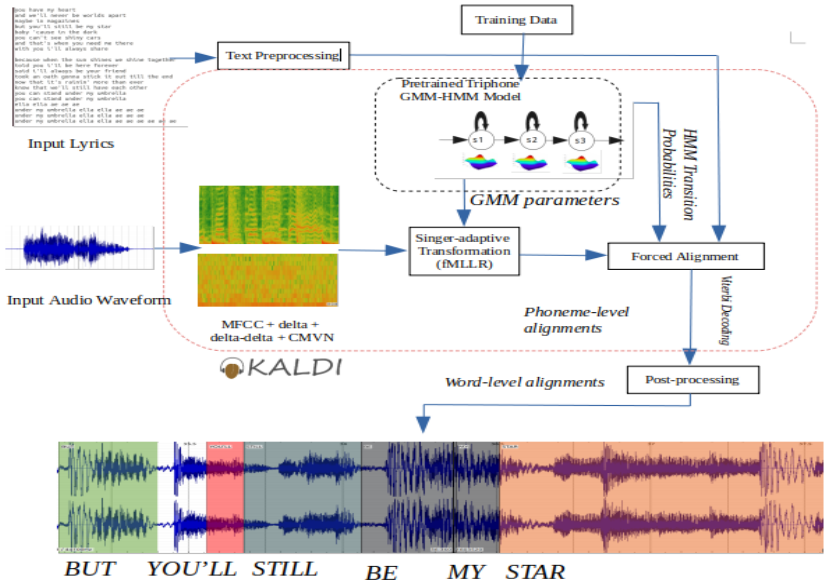

Figure 1. The overall lyrics-to-audio alignment system. The excerpt is taken from the Song *'Umbrella - Rihanna'*

The CMU Pronunciation Dictionary decomposes words in the sequences of phonemes where each phoneme label contains a position suffix which indicates the position of the phoneme with respect to the corresponding word. According to this representation, if a phoneme label has a ' `_B` ' suffix, it means that the phoneme is the beginning of the word, whereas ' `_I` ' represents an intermediate phoneme and ' `_E` ', the ending phoneme of the word and ' `_S` ' a standalone phoneme. When converting phonetic alignments into word-level alignments we have exploited these suffixes to determine the borders of the words.

In Figure 1, the overall system is depicted as a block diagram and the output alignments are shown on the corresponding audio segment * (footnote for sonic visualizer). Note that, even though the training of acoustic models is done on monophonic singing voice recordings, the alignment below is obtained on polyphonic music.

4. FUTURE WORK & DISCUSSIONS

The lyrics alignment output seen in Figure 1 shows that our GMM-HMM based system does not perform horrible even in the case of long polyphonic audio files. However, it is also clear that not all the word alignments are perfect. Moreover, there are accumulated misalignments when the results on the entire song is investigated. In this section, we scrutinize the cause of these misalignments and propose possible solutions for an improved system.

There are errors in regions when there are no vocals present but only the accompaniment. Since the GMM-HMM model is trained on monophonic recordings, the system models non-vocal regions as silence regions which is not the case in polyphonic music recordings. Furthermore, the alignment performance gets lower as the length of the audio exceeds few minutes. To overcome both of these issues, there are few strategies that can be applied to improve the alignments. Forced alignment performs well for shorter utterances, hence automatically segmenting the audio into shorter segments would help achieving an improved performance. The segmentation can be done using music structural segmentation and then matching the corresponding lyrics for each part in the song (verse, chorus, bridge, etc.) It is also possible to segment the audio excerpts based on vocal presence. For that 'Vocal Activity Detection (VAD)' methods can be exploited. 'Source Separation (SS)' could be applied and then segments would be obtained based on silence. Overall, the alignment performance can boost on source separated vocal signals.

Another possibility for an improved alignment system is to build a better automatic speech recognizer. The phoneme transition probabilities can be learned using Deep Neural Networks (DNN) instead of GMMs in building the acoustic model, which forms the basis of most of the state-of-the-art ASR systems. Additionally, a more comprehensive Language Model can be learned using a bigger corpus. For instance, crawling web to retrieve all the lyrics (in English) available online might be used as the training text corpus. In our experiments, we have seen that not all the words in lyrics of popular songs exist in our lexicon for training the LM and the pronunciation dictionary. For this reason, a strategy needs to be developed to include those words that do not exist in the pronunciation dictionary. Last but not least, the pronunciation dictionary needs to be modified in order to take different pronunciations of words in singing into account. This is also a necessary step for a better phoneme duration modeling in singing. In [5], the authors a strategy to the lexicon where the vowels are replicated in pronunciation considering that vowels and voiced phonemes are pronounced longer in singing [11].

In conclusion, we have presented our automatic lyrics-to-audio alignment system that we have submitted for the MIREX 2019: Audio-to-Lyrics Alignment Challenge. We

have given the details of training and alignment procedures. Finally, we mention the possible strategies we plane to take on for boosting the performance of our system.

Acknowledgments

The author E.D received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowsa-Curie grant agreement No. 765068.

5. REFERENCES

[1] Smule sing! 300x30x2 dataset. "https://ccrma.stanford.edu/damp/". Accessed in 2019-08-30.

[2] Gerardo Roa Dabike and Jon Barker. Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system. *Proc. Interspeech 2019*, pages 579–583, 2019.

[3] Georgi Bogomilov Dzhambazov and Xavier Serra. Modeling of phoneme durations for alignment between polyphonic audio and lyrics. In *Timoney J, Lysaght T, editors. 12th Sound and Music Computing Conference*, 2015.

[4] Hiromasa Fujihara and Masataka Goto. Lyrics-to-audio alignment and its application. In *Dagstuhl Follow-Ups*, volume 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[5] Chitralekha Gupta, Haizhou Li, and Ye Wang. Automatic pronunciation evaluation of singing. In *Interspeech*, pages 1507–1511, 2018.

[6] Raymond Lau, Ronald Rosenfeld, and Salim Roukos. Trigger-based language models: A maximum entropy approach. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 45–48. IEEE, 1993.

[7] Annamaria Mesaros and Tuomas Virtanen. Automatic alignment of music audio and lyrics. In *Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, 2008.

[8] Mehryar Mohri, Fernando Pereira, and Michael Riley. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*, pages 559–584. Springer, 2008.

[9] Sree Hari Krishnan Parthasarathi, Bjorn Hoffmeister, Spyros Matsoukas, Arindam Mandal, Nikko Strom, and Sri Garimella. fmllr based feature-space speaker adaptation of dnn acoustic models. In *Sixteenth annual conference of the international speech communication association*, 2015.

[10] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[11] Polina Proutskova. *Investigating singing voice: quantitative and qualitative approaches to studying cross-cultural vocal production*. PhD thesis, Queen Mary, University of London, 2018.

[12] Andreas Stolcke. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*, 2002.

[13] Robert L Weide. The cmu pronouncing dictionary. *URL: http://www. speech. cs. cmu. edu/cgibin/cmudict*, 1998.