# IMPROVING LYRICS ALIGNMENT THROUGH JOINT PITCH DETECTION

*Jiawen Huang* [*1], *Emmanouil Benetos*[1], *Sebastian Ewert*[2]

[1] Centre for Digital Music, Queen Mary University of London, UK
[2] Spotify

## ABSTRACT

In recent years, the accuracy of automatic lyrics alignment methods has increased considerably. Yet, many current approaches employ frameworks designed for automatic speech recognition (ASR) and do not exploit properties specific to music. Pitch is one important musical attribute of singing voice but it is often ignored by current systems as the lyrics content is considered independent of the pitch. In practice, however, there is a temporal correlation between the two as note starts often correlate with phoneme starts. At the same time the pitch is usually annotated with high temporal accuracy in ground truth data while the timing of lyrics is often only available at the line (or word) level. In this paper, we propose a multi-task learning approach for lyrics alignment that incorporates pitch and thus can make use of a new source of highly accurate temporal information. Our results show that the accuracy of the alignment result is indeed improved by our approach. As an additional contribution, we show that integrating boundary detection in the forced-alignment algorithm reduces cross-line errors, which improves the accuracy even further.

***Index Terms***— Lyrics alignment, multi-task learning, pitch detection, music information retrieval

## 1. INTRODUCTION

Given the lyrics to a song, audio-to-lyrics alignment aims at identifying for each lyric the corresponding position in a recording of the song, at a line, word or phoneme level [1]. Applications include the generation of karaoke-style lyrics for music players and subtitles for music videos. Moreover, lyrics alignment can serve as a basic block for singing voice analysis and can benefit other tasks such as cover song identification and music structure analysis.

Since the release of large datasets providing audio with lyrics annotations, such as DALI [2] and DAMP [3, 4], there has been significant progress in reducing the alignment error. Many previous methods are adapted from automatic speech recognition (ASR) [4, 5, 6, 7]. However, singing voice tends to be more complex than speech signals. Singing voice has a wider dynamical range, the pronunciation of words varies a lot more, while singing techniques allow the artist to control the sound in various ways [1, 8, 9, 10]. Moreover, the background music is often highly correlated with the singing voice, making it more challenging to analyze the vocals with high accuracy. Gupta et al. [5] explored music-related features and adapted the acoustic model on polyphonic data. In their follow-up work [11], they trained a genre-informed acoustic model and found improvements in both lyrics alignment and transcription. These approaches indicate that domain knowledge could help train the acoustic model.

One challenge of training the acoustic model is the lack of fine-grained annotation. A frame-level annotation of lyrics is hard to obtain. Some previous works trained a Gaussian mixture hidden Markov model (GMM-HMM) to predict the frame-level annotation and take it as the ground truth for training [12, 5]. Others adopted the connectionist temporal classification (CTC) loss [13] and trained the acoustic model in an end-to-end way [6, 14]. However, the CTC loss is a weaker form of supervision as it only enforces that a symbol is observed but does not specify when [15].

In this paper, we explore multi-task learning [16] for lyrics alignment, where the auxiliary task is pitch detection. At first this seems to be an unusual combination as in traditional speech synthesis, pitch and lyrics are modeled as two independent attributes and thus it might not be possible to share many feature representations between the two tasks - an aspect that typically drives which tasks in multitask learning are combined. However, using this unusual combination we can integrate pitch as an additional source of highly accurate temporal information. In particular, the pitch output of our method has to be temporally precise to achieve high accuracy and the idea is to evaluate whether this might act as an inductive bias that encourages the lyrics output of the network to be more precise as well. In other words, by adding a frame-level loss function through multi-task learning, it implicitly applies a stronger restriction about the timing as many of the phonemes and pitches share onsets. Additionally, previous studies have shown that joint learning of unrelated tasks might lead to sparser and more informative representations [17], which could lead to improved accuracy as well. Hung et al. [18] proposed a multi-task learning method for instrument detection which models pitch and instrument. Inspired by their work, we propose a model that outputs a representation for both pitch [1] and phoneme, where the task-wise loss can be computed by applying pooling along the other axis. We favor this method over a multi-head architecture commonly used in multi-task learning as the pitch information is kept close to the phoneme information all the way.

In real-world applications such as generating karaoke-style scrolling lyrics, cross-line misalignment is less tolerable than in-line errors, and should therefore be avoided. Just as lyrics can be segmented line by line, the corresponding cut points in audio can be regarded as line-level boundaries, which can be estimated by a network. Previous works in audio-to-score alignment show that onset information can enhance the temporal precision by adding an onset term to the cost function [19]. Similarly, we propose to incorporate the boundary probability of audio frames into the alignment algorithm to further improve the alignment performance.

To the best of the authors' knowledge, this is the first attempt to apply multi-task learning to lyrics alignment. Although multi-

---

---

[1]In this study, pitch detection refers to the quantized pitch (as typically specified as the pitch of a note in MIDI) as opposed to the fundamental frequency.
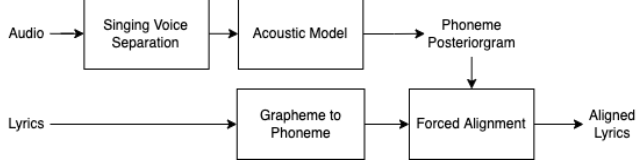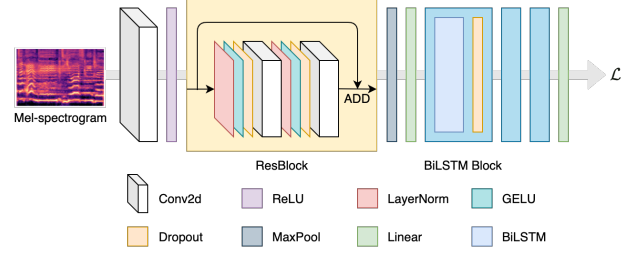
**Fig. 1**. System overview.



**Fig. 2**. The network architecture for the baseline, boundary detection, and the MTL models. The only differences among them are the dimensions of the BiLSTMs and the last fully-connected layer.

task learning has been extensively explored in ASR [20, 21, 22], we choose a musically meaningful auxiliary task to take advantage of the correlation between lyrics and pitch [23, 24]. This is also the first time an audio-to-lyrics alignment system incorporates boundary information into the forced-alignment.

The paper is structured as follows: The proposed architecture and loss computations are discussed in Sec. 2. The proposed method to incorporate boundary information is explained in Sec. 3. The dataset, baseline and experimental settings are described in Sec. 4. The results are presented and discussed in Sec. 5. Finally, we conclude and look into future directions in Sec. 6.

## 2. JOINT PHONEME RECOGNITION AND PITCH DETECTION

A general overview of the various modules involved in our system is given in Fig. 1 – our proposed extensions and their details will be discussed below. It consists of a singing voice separation module, an acoustic model, a grapheme to phoneme (g2p) module, and the alignment algorithm. The lyrics are first converted to a phoneme sequence via g2p (Sec. 4.1). The singing voice separation model extracts the vocals to remove the effect of the background music (Sec. 4.1). The acoustic model takes the Mel-spectrogram of the separated vocals as input, and produces the phoneme posteriorgram (Sec. 2.1). Then the forced-alignment algorithm is applied on the posteriorgram and the corresponding lyrics (Sec. 3).

### 2.1. Acoustic Model

The acoustic model takes a 128-bin Mel-spectrogram with a fixed duration as input. The sampling rate is 22050 Hz, the fft size is 512 and the hop size is 256. To fix notation, let $X = \{x_1, x_2, ...x_m\}$ be the frames of the input Mel-spectrogram, and $L = \{l_1, l_2, ...l_n\}$ be the phoneme sequence for the corresponding lyrics, where $m$ and $n$ are the lengths of the Mel-spectrogram and the phoneme sequence. The size of the phoneme set $\mathcal{S}_{\text{phone}}$ is $N_{\text{phone}}$, and the size of the pitch set $\mathcal{S}_{\text{pitch}}$ is $N_{\text{pitch}}$.

Fig. 2 shows the network architecture. It consists of a convolutional layer, a residual convolutional block, a fully-connected layer, 3 bidirectional LSTM (Long Short-Term Memory) layers, a final fully-connected layer, and non-linearities in between. The kernel size of the convolutional layers is $3 \times 3$, with stride and padding equal to one. The number of filters for the three convolutional layers is 32. The dimensions of the bidirectional LSTMs (BiLSTM) are 256. Layer normalization is applied on the feature dimension, with a mini-batch size of 128. All dropout rates are set to 0.1. The last fully-connected layer is time-distributed (applied to each frame), with a target size of $N_{\text{phone}} \times N_{\text{pitch}}$. After that, the output is reshaped to an order-3 tensor $\mathcal{D}$ of size $N_{\text{time}} \times N_{\text{phone}} \times N_{\text{pitch}}$, where $N_{\text{time}}$ is the output frame number. This representation $\mathcal{D}$ can be considered as the joint probability distribution before a `softmax` operation. This model is later referred to as **MTL** (**M**ulti-**T**ask **L**earning).

### 2.2. Loss Function

Fig. 3 describes the steps to compute the loss functions. Average pooling is applied on $\mathcal{D}$ along the pitch axis to get the (log-) posteriorgram of phoneme $P_{\text{phone}}$ of size $N_{\text{time}} \times N_{\text{phone}}$, where the CTC loss [13] $\mathcal{L}_{\text{phone}}$ is computed. The same is applied along the phoneme axis to get the pitch posteriorgram $P_{\text{pitch}}$ of size $N_{\text{time}} \times N_{\text{pitch}}$ and compute the frame-level cross entropy loss $\mathcal{L}_{\text{pitch}}$. The final loss is a weighted sum of these two:

$$\mathcal{L} = \mathcal{L}_{\text{phone}} + \lambda \cdot \mathcal{L}_{\text{pitch}} \qquad (1)$$

where $\lambda$ is an adjustable parameter. We tested different values in $\{0.5, 0.8, 1.0, 1.2, 1.5\}$ and found $\lambda = 0.5$ to yield the best results and thus we used this setting in our experiments below.

The baseline model shares the same architecture except for the dimension of the last fully-connected layer, but focuses on phoneme recognition only. The output is a $N_{\text{time}} \times N_{\text{phone}}$ posteriorgram, where the CTC loss is computed.

## 3. ALIGNMENT

### 3.1. Viterbi Forced Alignment

In the calculation of the CTC loss [13], the probabilities of all possible alignment paths are accumulated. These paths are generated by inserting blank ($\epsilon$) and repeated labels to the original sequence. By applying the same rules to the lyrics to be aligned at inference time, the best path can be decoded via Viterbi forced alignment [25].

To be more specific, the phoneme sequence is expanded by inserting $\epsilon$ symbols in between the phonemes to $L^* = \{\epsilon, l_1, \epsilon, l_2, ...\epsilon, l_n, \epsilon\}$. Let $P_{\text{phone}}(t, p)$ be the log-probability of the input frame $x_t$ being a phoneme $p \in \mathcal{S}_{\text{phone}}$. A path $\{(t_q, l_q^*)\}$ is a sequence of index pairs indicating $x_{t_q}$ is aligned to $l_q^*$. The target is to find the path that maximizes the score:

$$\underset{path}{\operatorname{argmax}} \sum_{(t,l^*) \in path} P_{\text{phone}}(t, l^*) \qquad (2)$$

The solution can be computed efficiently via dynamic programming.

### 3.2. Incorporating Boundary Information

A boundary detection model is trained independently to predict the line-level boundary probability. The input and the network architecture are the same as the baseline except that it has a smaller BiLSTM dimension 32 and output size is 1 per time step. The target label is a boundary activation curve (range: $0 \sim 1$). The start time of a line of lyrics is considered a boundary event. Each event is converted to a
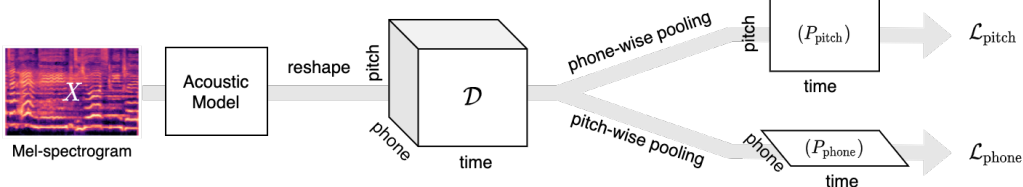
**Fig. 3**. Multi-task loss computation.

| Dataset | SS | FW | Jamendo | | Mauch | |
|---|---|---|---|---|---|---|
| Metric | - | - | AAE | PCO | AAE | PCO |
| SDE2 [6] | Y | E2E | 0.39 | 0.87 | 0.26 | 0.87 |
| GC [11] | N | STD | 0.22 | 0.94 | 0.19 | 0.91 |
| VHM [14] | Y | E2E | 0.37 | 0.92 | 0.22 | 0.91 |
| Baseline | Y | E2E | 0.31 | 0.94 | 0.20 | 0.89 |

**Table 1**. Comparison with the state-of-the-art systems for lyrics alignment. The **SS** column indicates if the input has been source-separated (Yes or No). The **FW** column indicates if the method takes an end-to-end (E2E) or standard (STD) ASR framework.

Gaussian window centered at the event time, at a size of 0.7 sec. Let $P_{bdr}(t)$ be the boundary log-probability of $x_t$ predicted by the model.

We propose to add the boundary probability to the Viterbi score as a bonus at the line beginning of the lyrics, which is a 'boundary' in the text. This is to encourage aligning a phoneme at the beginning of a line to a boundary-like audio frame. Equation (2) is updated as follows:

$$\underset{path}{\operatorname{argmax}} \sum_{(t,l^\star) \in path} P_{phone}(t, l^\star) + \alpha \sum_{\substack{(t,l^\star) \in path \\ l^\star \text{ is a line start}}} P_{bdr}(t) \quad (3)$$

where $\alpha$ is an adjustable weighting parameter. We test different values in $\{0.5, 0.8, 1.0, 1.2, 1.5\}$ and found $\alpha = 0.8$ to yield the best results and thus we used this setting in our experiments below. This alignment method is referred to as **BDR** (**B**oun**Da**R**y**) later in text.

## 4. EXPERIMENTS

### 4.1. Dataset

All models are trained on the DALI v2 dataset [2]. There are 7756 songs in total with word-level lyrics annotations and note-level pitch annotations. We only use the English subset according to the language label. The training set contains 4224 songs and the validation set contains 1056 songs for the baseline and the multi-task model [2]. The samples are generated by applying a 5.6 sec sliding window with a hop size of 2.8 sec. The target lyrics for an audio segment are the words fully covered within the window.

For the boundary detection model, a subset of the above validation set is left out for evaluation. The subset contains 45 songs.

For lyrics alignment, we run evaluations on the Jamendo [26, 6] and Mauch [27] datasets. Each of them has 20 Western pop songs and word-level timestamps and boundary annotations. They are also used in the MIREX [3] lyrics alignment challenges.

| Metric | COnPOff | COnP | COn |
|---|---|---|---|
| pYIN [31] | 5.9% | 12.2% | 43.5% |
| MTL | 4.9% | 16.3% | 32.6% |

**Table 2**. Pitch detection results.

| Metric | Precision | Recall | F-score | AUC |
|---|---|---|---|---|
| BDR | 79.3% | 55.9% | 64.3% | 89.9% |

**Table 3**. Boundary detection results.

For pitch detection, we evaluate the MTL model using the RWC Music Database - Popular Music [28]. It contains 94 popular songs (74 Japanese and 20 English) [4].

Vocals are extracted by an implementation of [29] for all songs mentioned above before feeding to the network. The open-source g2p tool [5] is used to convert the lyrics to phoneme sequences. The phoneme set follows the convention of the CMU pronouncing dictionary [6] and has a size of 39. In practice, we added the space ␣ and the epsilon $\epsilon$ (for the CTC loss) to the phoneme set so that the number of classes $N_{phone}$ is 41. The target pitch range is D2-C6, therefore $N_{pitch}$ is 47 (with one additional class for silence).

### 4.2. Training

The models are trained using a learning rate of $1e^{-4}$ and the ADAM optimizer [30]. Early stopping is adopted when the validation loss does not decrease for 20 epochs after the $20^{th}$ epoch. We employed this strategy as we found some early checkpoints to produce a relatively low loss value but which turned out to be underfitting. The boundary detection model is trained with binary cross entropy loss, and we do not wait for 20 epochs before early-stopping.

### 4.3. Evaluation

For lyrics alignment at the word level, we report the Average Absolute Error (AAE) [32] and Percentage of correct onsets with a tolerance window of 0.3 seconds (PCO) [27]. These metrics are averaged over all the songs in the dataset. In addition to the word-level evaluation, we also report the line-level results to demonstrate the benefit from boundary information.

For boundary detection, we compute the precision, recall, and F-measure. A predicted boundary is considered a hit if there is a reference boundary within $\pm0.5$ seconds [33]. We also report the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) between the boundary activation curve and the ground truth.

---

[2] The data splits and the code can be accessed through the link: `https://github.com/jhuang448/LyricsAlignment-MTL`

[3] `https://www.music-ir.org/mirex/`

[4] The dataset contains 100 songs, but 6 of them (No. 3, 5, 8, 10, 23, and 66) are removed because they have multiple singers.

[5] `https://github.com/Kyubyong/g2p`

[6] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

| Level | Word | | | | Line | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | Jamendo | | Mauch | | Jamendo | | Mauch | |
| Metric | AAE | PCO | AAE | PCO | AAE | PCO | AAE | PCO |
| Baseline | 0.31 | 0.94 | 0.20 | 0.89 | 0.30 | 0.93 | 0.37 | 0.87 |
| Baseline + BDR | 0.29 | 0.94 | 0.20 | 0.90 | 0.25 | 0.95 | 0.33 | 0.88 |
| MTL | 0.23 | 0.94 | 0.21 | 0.90 | 0.27 | 0.93 | 0.43 | 0.86 |
| MTL + BDR | 0.23 | 0.94 | 0.20 | 0.91 | 0.25 | 0.94 | 0.42 | 0.89 |

**Table 4**. Lyrics alignment results. Significant improvement in AAE on Jamendo for the MTL model can be observed for word-level metrics. All line-level metrics increase after introducing boundary information.

For pitch detection, we compute 3 F-scores (COn, COnP, and COnPOff) originally proposed in [34]. COn only evaluates the onset times, COnP evaluates both onsets and the pitches, and COnPOff evaluates onsets, pitches, and offsets. The onset tolerance is 50ms, while the offset tolerance is max(50ms, 0.2*note duration). Since the ground truth of RWC Pop contains many octave errors, instead of evaluating the absolute pitch, we compute the metrics on a octave-wrapped pitch with a tolerance of 50 cents.

## 5. RESULTS AND DISCUSSION

### 5.1. State of the Art Comparison

The comparison of our baseline with state-of-the-art lyrics alignment systems is listed in Tab. 1. Both GC [11] and VHM [14] are trained on DALI v1, while VHM takes an end-to-end approach on separated vocals and GC uses a standard ASR framework trained on polyphonic audio. SDE2 [6] is a wave-U-net model trained on an internal dataset consisting 44,232 songs. It is also trained on separated vocals.

The results show that our baseline performs similarly to the current state-of-the-art on both Jamendo and Mauch. Among the E2E models trained on separated vocals, our baseline outperforms SDE2 and VHM on all metrics except PCO on Mauch. Besides the network architecture, the key difference is the alignment unit. In our baseline, the CTC loss and the alignment are computed on phoneme sequences, while SDE2 and VHM are computed on characters. This further verifies that using phonemes over characters for alignment has better performance [14].

### 5.2. Boundary Detection and Pitch Detection

The boundary detection results are listed in Tab. 3. An AUC of 89.9% indicates it is a decent boundary detector. However, it is hard to provide comparison with existing systems because most of them detect structural boundaries (segmentation).

The pitch detection results of the MTL model are listed in Tab. 2. For comparison, we also list the pYIN [31] (note mode) results. Our model is trained on DALI, which is mostly Western music, while 80% of the songs in the testing set are in Japanese. We argue that there is a large difference in data distribution between the training and testing sets, leading to a limited performance.

The predicted pitch results and the ground truth on a short clip from Jamendo are visualized in Fig. 4. It can be observed that our pitch tracker generally follows the singing voice, but produces some short notes that should be merged.

### 5.3. MTL Evaluations

Results of different combinations of Baseline / MTL with BDR are listed in Tab. 4. Comparing with the baseline, the improvement is
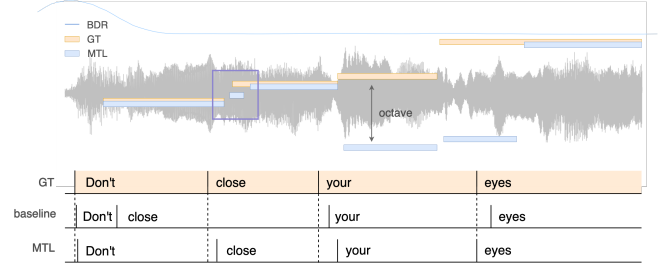


**Fig. 4**. Lyrics alignment, pitch detection, and boundary detection results on a clip of `Avercage_-_Embers.mp3` from Jamendo. The GT (ground truth) notes are slightly shifted for visualization purposes. Two notes have the same pitch if there is an overlap.

significant in AAE on Jamendo for the MTL model, but limited in the other metrics. As can be observed in Tab. 1, results on Jamendo are worse than those on Mauch. This is probably due to more slurred pronunciation in Jamendo [6]. We argue that our MTL model alleviates this problem by estimating the pitch at the same time. Since pitch and phoneme often share onsets, a change in pitch can be a good indicator of a change in phoneme as well. The lyrics alignment results on the same clip are presented in Fig. 4. Comparing to the baseline, MTL is able to place the word "close" at the right time with the help of changes detected in pitch (within the purple frame).

Though the benefit from adding boundary information is marginal in word-level metrics, it is clear in line-level. Both metrics on Jamendo and Mauch are better with BDR.

## 6. CONCLUSION

In this work, we propose a multi-task learning approach for lyrics-to-audio alignment by learning a joint representation for pitch and phoneme, and add boundary information to enhance the alignment. The proposed approach is built upon a phoneme-based end-to-end acoustic model as the baseline, which outperforms state-of-the-art end-to-end systems for lyrics alignment. The proposed model also outperforms the baseline in both word-level and line-level metrics.

Meanwhile, we recognize some limitations of our system. By introducing the additional boundary model, the performance is slightly improved at the cost of efficiency ($\times 1.3$ computation time). Besides, the pitch concept might not apply well to speech and rap in music. One possible solution is to add one pitch class for such voice. For future work, we plan to evaluate the acoustic model on lyrics transcription as well. To further extend the application scenario, we plan to align the pitched notes to the lyrics with the learned representation $\mathcal{D}$. In this way, we can achieve melody and lyrics transcription of singing voice in one pass.

## 7. REFERENCES

[1] Hiromasa Fujihara and Masataka Goto, "Lyrics-to-audio alignment and its application," in *Multimodal Music Processing*, vol. 3 of *Dagstuhl Follow-Ups*, pp. 23–36. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2012.

[2] Gabriel Meseguer-Brocal, Alice Cohen-Hadria, and Geoffroy Peeters, "Creating DALI, a large dataset of synchronized audio, lyrics, and notes," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 55–67, 2020.

[3] Smule Sing!, "Smule.digital archive mobile performances (damp)," https://ccrma.stanford.edu/damp/, Last accessed September, 2021.

[4] Chitralekha Gupta, Rong Tong, Haizhou Li, and Ye Wang, "Semi-supervised lyrics and solo-singing alignment," in *Proc. ISMIR*, 2018, pp. 600–607.

[5] Chitralekha Gupta, Emre Yılmaz, and Haizhou Li, "Acoustic modeling for automatic lyrics-to-audio alignment," in *Proc. INTERSPEECH*. 2019, pp. 2040–2044, ISCA.

[6] Daniel Stoller, Simon Durand, and Sebastian Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *Proc. ICASSP*. 2019, pp. 181–185, IEEE.

[7] Bidisha Sharma, Chitralekha Gupta, Haizhou Li, and Ye Wang, "Automatic lyrics-to-audio alignment on polyphonic music using singing-adapted acoustic models," in *Proc. ICASSP*. 2019, pp. 396–400, IEEE.

[8] Anna M. Kruspe, "Keyword spotting in a-capella singing," in *Proc. ISMIR*, 2014, pp. 271–276.

[9] Anna M. Kruspe, "Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing," in *Proc. ISMIR*, 2016, pp. 358–364.

[10] Alex Loscos, Pedro Cano, and Jordi Bonada, "Low-delay singing voice alignment to text," in *Proc. ICMC*. 1999, vol. 11, pp. 27–61, Michigan Publishing.

[11] Chitralekha Gupta, Emre Yılmaz, and Haizhou Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?," in *Proc. ICASSP*. 2020, pp. 496–500, IEEE.

[12] Emir Demirel, Sven Ahlbäck, and Simon Dixon, "Automatic lyrics transcription using dilated convolutional neural networks with self-attention," in *Proc. IJCNN*. 2020, pp. 1–8, IEEE.

[13] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*. 2006, vol. 148, pp. 369–376, ACM.

[14] Andrea Vaglio, Romain Hennequin, Manuel Moussallam, Gaël Richard, and Florence d'Alché-Buc, "Multilingual lyrics-to-audio alignment," in *Proc. ISMIR*, 2020, pp. 512–519.

[15] Hu Liu, Sheng Jin, and Changshui Zhang, "Connectionist temporal classification with maximum entropy regularization," in *Advances in Neural Information Processing Systems*, 2018, pp. 839–849.

[16] Rich Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[17] Bernardino Romera-Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil, "Exploiting unrelated tasks in multi-task learning," in *Proc. AISTATS*. 2012, vol. 22 of *JMLR Proceedings*, pp. 951–959, JMLR.org.

[18] Yun-Ning Hung, Yi-An Chen, and Yi-Hsuan Yang, "Multi-task learning for frame-level instrument recognition," in *Proc. ICASSP*. 2019, pp. 381–385, IEEE.

[19] Sebastian Ewert, Meinard Müller, and Peter Grosche, "High resolution audio synchronization using chroma onset features," in *Proc. ICASSP*. 2009, pp. 1869–1872, IEEE.

[20] Xuesong Yang, Kartik Audhkhasi, Andrew Rosenberg, Samuel Thomas, Bhuvana Ramabhadran, and Mark Hasegawa-Johnson, "Joint modeling of accents and acoustics for multi-accent speech recognition," in *Proc. ICASSP*. 2018, pp. 5989–5993, IEEE.

[21] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*. 2017, pp. 4835–4839, IEEE.

[22] Zhangyu Xiao, Zhijian Ou, Wei Chu, and Hui Lin, "Hybrid ctc-attention based end-to-end speech recognition using subword units," in *Proc. ISCSLP*. 2018, pp. 146–150, IEEE.

[23] Eric Nichols, Dan Morris, Sumit Basu, and Christopher Raphael, "Relationships between lyrics and melody in popular music," in *Proc. ISMIR*, 2009, pp. 471–476.

[24] Rong Gong, Philippe Cuvillier, Nicolas Obin, and Arshia Cont, "Real-time audio-to-score alignment of singing voice based on melody and lyric information," in *Proc. INTERSPEECH*. 2015, pp. 3312–3316, ISCA.

[25] G David Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[26] Mathieu Ramona, Gaël Richard, and Bertrand David, "Vocal detection in music with support vector machines," in *Proc. ICASSP*. 2008, pp. 1885–1888, IEEE.

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

[28] Masataka Goto, "AIST annotation for the RWC music database," in *Proc. ISMIR*, 2006, pp. 359–360.

[29] Andreas Jansson, Eric J. Humphrey, Nicola Montecchio, Rachel M. Bittner, Aparna Kumar, and Tillman Weyde, "Singing voice separation with deep u-net convolutional networks," in *Proc. ISMIR*, 2017, pp. 745–751.

[30] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[31] Matthias Mauch and Simon Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. ICASSP*. 2014, pp. 659–663, IEEE.

[32] Annamaria Mesaros and Tuomas Virtanen, "Automatic alignment of music audio and lyrics," in *Proc. DAFx*, 2008, pp. 321–324.

[33] Douglas Turnbull, Gert R. G. Lanckriet, Elias Pampalk, and Masataka Goto, "A supervised approach for detecting boundaries in music using difference features and boosting," in *Proc. ISMIR*. 2007, pp. 51–54, Austrian Computer Society.

[34] Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, and Isabel Barbancho, "Evaluation framework for automatic singing transcription," in *Proc. ISMIR*, 2014, pp. 567–572.