IMPROVING LYRICS-TO-AUDIO ALIGNMENT USING FRAME-WISE PHONEME LABELS WITH MASKED CROSS ENTROPY LOSS

Tian Cheng, Tomayasu Nakano, and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan
tian.cheng@aist.go.jp t.nakano@aist.go.jp m.goto@aist.go.jp

ABSTRACT

This paper addresses the task of lyrics-to-audio alignment, which involves synchronizing textual lyrics with corresponding music audio. Most publicly available datasets for this task provide annotations only at the line or word level. This poses a challenge for training lyrics-to-audio models due to the lack of frame-wise phoneme labels. However, we find that phoneme labels can be partially derived from word-level annotations: for single-phoneme words, all frames corresponding to the word can be labeled with the same phoneme; for multi-phoneme words, phoneme labels can be assigned at the first and last frames of the word. To leverage this partial information, we construct a mask for those frames and propose a masked frame-wise cross-entropy (CE) loss that considers only frames with known phoneme labels. As a baseline model, we adopt an autoencoder trained with a Connectionist Temporal Classification (CTC) loss and a reconstruction loss. We then enhance the training process by incorporating the proposed frame-wise masked CE loss. Experimental results show that incorporating the frame-wise masked CE loss improves alignment performance. In comparison to other state-of-the art models, our model provides a comparable Mean Absolute Error (MAE) of 0.216 seconds and a top Median Absolute Error (MedAE) of 0.041 seconds on the testing Jamendo dataset.

1. INTRODUCTION

Lyrics-to-audio alignment is the task of aligning lyrics with the corresponding music audio, with the aim of automatically providing time-synchronized lyrics. This task is particularly important for real-world applications that require accurate large-scale lyric alignment, such as karaoke systems, lyrics-based music retrieval [1], and lyrics animation for music videos [2].

Although lyrics-to-audio alignment is related to speech-to-text alignment, it presents unique challenges due to the distinct characteristics of singing. First, the singing voice is generally more expressive than spoken voice [1, 3, 4]. Lyrics are sung with varying dynamics, pitches, and durations, often incorporating vocal techniques such as vibrato. This introduces considerable variability in pronunciation. Second, the musical accompaniment is not simply background noise but is highly correlated with the vocal, which reduces the effective signal-to-noise ratio for processing the vocal. To address these challenges, some models utilize advanced source separation methods to extract vocals before alignment [5, 6, 7, 8, 9]. Alternatively, other approaches exploit temporal correlations between lyrics and musical elements

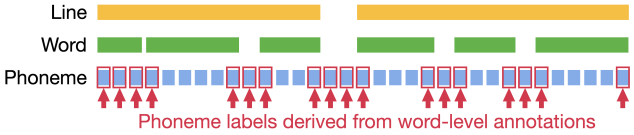

Figure 1: An illustration of alignment annotations at the line, word, and phoneme levels. The proposed method uses phoneme labels derived from word-level annotations, with labeled frames indicated by red arrows. In this example, all frames of the first (single-phoneme) word are labeled, while only the first and last frames are labeled for multi-phoneme words. Silent frames between words are labeled as silence.

such as melody, chords, and accompaniment, learning lyrics-to-audio alignment jointly with related tasks such as chord estimation [10], singing pitch detection [8], singing voice separation [11], and singing voice synthesis [12].

Another practical issue in developing a high-quality lyrics-to-audio alignment system is the scale and quality of available datasets, which are crucial for modern data-driven approaches. With the release of the DALI dataset [13, 14], lyrics-to-audio alignment performance has improved substantially [15, 8, 16, 17]. However, most of these datasets provide only line-level and word-level annotations, lacking frame-wise annotations necessary for supervised training. To train models using such weak annotations, methods based on conventional automatic speech recognition have relied on Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) frameworks to predict frame-wise phoneme labels, which are then treated as ground truth for training [18]. Alternatively, Stoller et al. [19] proposed an end-to-end model based on the Wave-U-Net architecture and trained it using the Connectionist Temporal Classification (CTC) loss [20]. The use of the CTC loss simplifies model training by allowing learning from weak, line-level annotations without explicit alignment. Due to its simplicity, the CTC loss has attracted increasing attention in recent work [6, 21, 8, 9]. However, the CTC loss is designed for transcription tasks and is not sensitive to the precise timing of individual elements in a sequence, which may limit alignment accuracy. To address this limitation, Teytaut et al. [21] added a spectral reconstruction loss to the CTC loss to improve temporal coherence in the model's predictions. Huang et al. [8] proposed a joint learning approach that integrates pitch estimation to enhance alignment accuracy.

While CTC-based models are commonly used, other methods leverage cross-modal embeddings from audio and text domains for lyrics-to-audio alignment. To process cross-modal content, contrastive learning has been employed to map audio and text embeddings into a shared latent space [22, 17]. Kang et al. [16] computed a cross-correlation matrix between latent representations of vocal

and lyric encoders, and trained the model to predict line and word onsets using cross-entropy (CE) loss. These cross-modal methods have achieved state-of-the-art results by utilizing large in-house training datasets.

For reasons of computational cost or performance, many alignment methods adopt a hierarchical alignment strategy, performing alignment at the line level followed by word-level refinement [8, 16, 22]. Demirel et. al. [7] used a similar two-step method, first segmenting the audio based on detected anchor words and then performing alignment within those segments.

In this paper, we address lyrics-to-audio alignment using an autoencoder model trained with the CTC loss. To mitigate the timing imprecision inherent in the CTC loss, we propose a more efficient use of available alignment annotations. We observe that even when only word-level annotations are available, they still contain useful information that has not yet been fully exploited for training. As illustrated in Figures 1 and 4, we can derive phoneme labels for three types of frames: (1) the first (onset) and last (offset) frames of words that contain two or more phonemes (we assign the first and last phoneme labels to those frames, respectively); (2) frames within the duration of words that contain only a single phoneme (we assign the same phoneme label to all such frames); (3) silent frames, identified as frames that do not belong to any word segment (we assign a silence label to such frames). Based on these observations, we assign frame-wise phoneme labels to the three types of frames and construct a mask that includes only those frames with phoneme labels, excluding all others. We then apply a masked CE loss, computed over the masked frames using frame-wise CE loss, to assist the training process.

In our experiments, we adopt an autoencoder model based on Convolutional Recurrent Neural Network (CRNN) [21]. We combine the masked CE loss with the CTC and reconstruction losses to train the model using the DALI dataset. Evaluation on the testing Jamendo dataset demonstrates that incorporating the masked CE loss improves alignment performance, achieving a comparable Mean Absolute Error and a top Median Absolute Error in the comparison experiment (see Section 3.5).

The contributions of this paper are summarized as follows:

- We propose an effective method for leveraging alignment annotations by deriving frame-wise phoneme labels—at word onsets, offsets, and silent frames—from word-level annotations. Since annotation creation is labor-intensive and costly, it is crucial to make the most of the available data.
- We demonstrate that combining the masked CE loss with the CTC and the reconstruction losses improves alignment performance and yields an alignment method with state-of-the-art results.
- Owing to the network simplicity of the CRNN-based model, our method enables song-level alignment using the entire song as input, without the need for separate line-level processing. This allows the model to capture long-range temporal context while also simplifying inference.

The rest of this paper is organized as follows. Section 2 introduces the baseline model and the proposed masked CE loss. Section 3 presents the experimental setup and results. Section 4 concludes the paper.

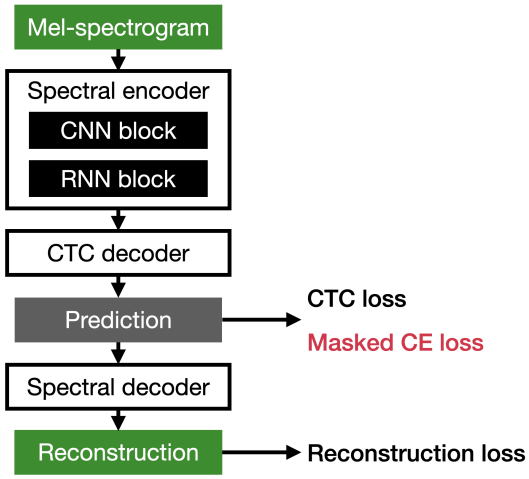

Figure 2: Model architecture. The baseline model is trained using the Connectionist Temporal Classification (CTC) loss and the reconstruction loss [21]. We propose a frame-wise masked cross-entropy (CE) loss (highlighted in red) to enhance training.

2. METHOD

We employ an autoencoder model for phoneme prediction, illustrated in Figure 2. The baseline model is trained using a CTC loss and a reconstruction loss. To improve alignment accuracy, we propose applying a masked CE loss to constrain model predictions at word onsets, offsets, and silent frames.

2.1. Audio and Lyrics Preprocessing

We first apply an advanced source separation method, HT Demucs [23], to the audio signals (stereo, sampled at 44.1 kHz) to extract the singing vocals. The separated vocals are then converted to mono and downsampled to 16 kHz. We compute log-scaled magnitude mel-spectrograms with 128 mel bins using a window size of 1024 and a hop size of 256. The mel-spectrogram of each song is normalized to a range between 0 and 1.

During training and validation, the vocal audio is divided into 10-second segments with a 5-second hop size. Only words that are completely included within each segment are used. In contrast, during testing, the full vocal audio for each song is used without segmentation, and its mel-spectrogram is fed into the model to obtain predictions.

For the lyrics corresponding to the audio, we follow the same preprocessing procedure as in [8]. We convert the lyrics into phoneme sequences using an open-source grapheme-to-phoneme (G2P) tool[1], and adopt a set of 39 phoneme tokens, following the conventions of the CMU Pronouncing Dictionary[2]. In addition to the 39 phoneme tokens, we add a token `0` to represent the blank symbol ϵ (staying on the same phoneme) used in the CTC loss, and a token `40` for silence or space, resulting in 41 distinct tokens. An example of lyrics with the corresponding phonemes and token representations is shown in Figure 3.

[1] `https://github.com/Kyubyong/g2p`
[2] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

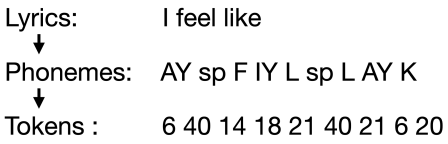

Figure 3: Example of lyrics preprocessing. "sp" stands for the silence or space between words.

2.2. The Baseline Autoencoder Model

The baseline is an autoencoder model trained with both CTC loss and reconstruction loss.

2.2.1. Network Architecture

We adopt the network architecture used in [21] as the basis for our model (see Figure 2). The mel-spectrogram is first fed into a spectral encoder and a CTC decoder to produce frame-wise phoneme probabilities as model predictions. These probabilities are then passed to a spectral decoder for spectral reconstruction.

The spectral encoder consists of a CNN block followed by an RNN block. The CNN block contains two 2D convolutional layers with a kernel size of 3×3 and ReLU activations. The number of filters is 16 and 32 for the first and second layers, respectively. Each convolutional layer is followed by batch normalization, a pooling operation that halves the feature dimension, and a 25% dropout.

The RNN block in the spectral encoder, the CTC decoder, and the spectral decoder all share the same architecture: two bidirectional LSTM layers with 512 units. In both the CTC decoder and the spectral decoder, a linear layer follows the LSTM layers, with output dimensions corresponding to the number of tokens (41) and the number of mel-spectral bins (128), respectively.

The above CRNN-based model used in this paper is a simplified version of the model in [21], with the attention mechanism removed, as we found that adding attention degraded alignment performance in our preliminary experiments. There are also differences in the model input: we use only the log-scaled mel-spectrogram, normalized to the range [0, 1], whereas Teytaut et al. [21] used the log-scaled mel-spectrogram in the range [-1, 1] along with its temporal spectral derivatives. As a result, we apply a `sigmoid` function for spectral reconstruction instead of the `tanh` function. While Teytaut et al. [21] excluded the blank from the spectral decoder input, we include it, as this yielded better results in our preliminary experiments.

2.2.2. The Connectionist Temporal Classification (CTC) Loss

We predict frame-wise phoneme probabilities $\mathbb{P}(\hat{\mathbf{y}}|\mathbf{X})$ from the input mel-spectrogram $\mathbf{X} \in \mathbb{R}^{128\times T}$, where the phoneme sequence $\hat{\mathbf{y}} = \{\hat{y}(t)\}, t \in [0, \cdots, T-1]$, and T is the total number of frames in the mel-spectrogram. However, frame-wise annotations are not available. Instead, we can use the phoneme sequence of lyrics $\mathbf{y} = \{y(m)\},\ m \in [0, \ldots, M-1]$, where $y(m) \in \mathcal{A}$, M is the length of the phoneme sequence, and $\mathcal{A}$ is the set of all possible phonemes.

Using the CTC loss [20], we train the learnable model parameters Θ to maximize the probability of generating an output sequence $\hat{\mathbf{y}}$ with the CTC constrains of $\hat{\mathbf{y}} \in \mathcal{A} \cap \{\epsilon\}$, and $\mathcal{B}(\hat{\mathbf{y}}) = \mathbf{y}$,

$$\mathbb{P}(\mathbf{y}|\mathbf{X};\Theta) = \sum_{\hat{\mathbf{y}},\mathcal{B}(\hat{\mathbf{y}})=\mathbf{y}} \prod_{t=0}^{T-1} \mathbb{P}(\hat{y}_t|\mathbf{X};\Theta), \tag{1}$$

where $\mathcal{B}$ is a mapping function that collapses repeated labels and removes blank symbols ϵ. For example, $\mathcal{B}(\epsilon aaa\epsilon\epsilon bb\epsilon\epsilon) = ab$. The CTC loss is defined as the negative log-likelihood of the above probability:

$$\mathcal{L}_{\text{CTC}}(\Theta) = -\log \mathbb{P}(\mathbf{y}|\mathbf{X};\Theta). \tag{2}$$

We apply a `softmax` function to the output of the CTC decoder to obtain the frame-wise phoneme probabilities and convert the probabilities into log scale for efficient computation of the CTC loss.

2.2.3. The Reconstruction Loss

Although frame-wise phoneme probabilities can be predicted using the CTC loss, it is computed over all possible alignments with the target phoneme sequence and is not sensitive to the exact temporal location of each phoneme. To improve the temporal precision of phoneme alignment, the spectral decoder is used to reconstruct the input spectrogram from the model's phoneme predictions, following [21]. The model is then trained to minimize the reconstruction loss defined as the L2 distance between the input and reconstructed mel-spectrograms, thereby enforcing temporal coherence in its predictions:

$$\mathcal{L}_{\text{REC}} = ||\hat{\mathbf{X}} - \mathbf{X}||_2, \tag{3}$$

where $\mathbf{X}$ is the input spectrogram and $\hat{\mathbf{X}}$ is the reconstructed mel-spectrogram obtained by applying a `sigmoid` function to the output of the spectral decoder. In our implementation, we use the Mean Squared Error (MSE) loss as a normalized squared L2 loss.

2.3. The Masked Cross-Entropy (CE) Loss

We observed that although frame-wise phoneme labels are not available for all frames, labels for specific frames can be derived from word-level annotations—such as the onset and offset frames of words. For example, consider the clip shown in Figure 4: the annotations include the lyrics "I feel like" along with the onset and offset times for the three words. From such annotations, we can derive phoneme labels for three types of frames:

1. For words containing two or more phonemes (e.g., "feel" and "like"), the first phoneme is assigned to the onset frame and the last phoneme to the offset frame.
2. For words consisting of a single phoneme (e.g., "I"), all frames within the word duration are labeled with that phoneme.
3. For frames outside any word segments, a silence/space label (token `40`) is assigned.

These labeled frames play an important role in determining word onsets and offsets, which is essential for word-level alignment.

We construct a binary mask to compute the CE loss between the model predictions and the frame-wise phoneme labels y_t at frames with known phoneme labels (masked frames). We first create an all-zero matrix as the mask matrix $\mathbf{B}$ with the same size as the model predictions: $\mathbf{B} = \mathbf{0}^{41\times T}$. For the masked frames, we set the corresponding elements $B_{i,t}$ in $\mathbf{B}$ to 1, excluding the blank

Word	Phonemes	Phoneme labels	Onset	Offset
I	AY	6	0.087 seconds	0.184 seconds
feel	F IY L	14 18 21	0.281 seconds	0.377 seconds
like	L AY K	21 6 20	0.474 seconds	0.571 seconds

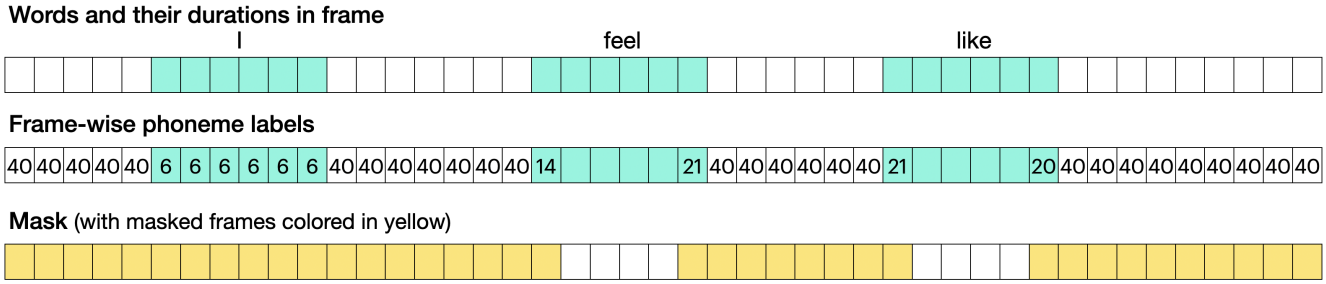

Figure 4: Frame-wise phoneme labels (tokens) derived from word-level annotations and the corresponding mask.

(token 0): $B_{i,t} = 1$ for token index $i \in [1, \cdots, 40]$ and frame index $t \in$ the masked frames. This ensures that the presence and absence of each phoneme label at the masked frames are properly accounted for during the loss computation. Then, we enhance the onsets by including the blank at the onset frames: $B_{0,t} = 1$ for $t \in$ the onset frames. We set $B_{0,t}$ to 1 at the onset frames to ensure that the absence of the blank token ($i = 0$) is explicitly enforced in the computation. This is because at the onset frames, the phoneme label must never correspond to the blank symbol ϵ, which represents staying on the same phoneme, and this mask setting ensures that its absence is properly accounted for during the loss computation.

By using this mask $\mathbf{B}$, the masked CE loss is computed as follows:

$$\mathcal{L}_{\text{maskedCE}} = -\frac{1}{N_{\text{mask}}} \sum_i \sum_t B_{i,t} H(L_{i,t}, D_{i,t}), \tag{4}$$

where $H(a, b) = a \log(b)$. $\mathbf{L} \in \mathbb{R}^{41 \times T}$ is the one-hot encoding of the frame-wise phoneme labels (i.e., tokens derived from the word-level annotations), with elements denoted by $L_{i,t}$. $L_{i,t} = 0$ for all $i \in [0, \cdots, 40]$ if frame t is not a word onset, offset, or silent frame (i.e., when no phoneme label can be derived). $\mathbf{D} \in \mathbb{R}^{41 \times T}$ represents the model predictions from the CTC decoder, with elements denoted by $D_{i,t}$, where $D_{i,t}$ is the predicted probability for token i at frame t. N_{mask} is the total number of masked frames.

In the experiments, the proposed model is trained to minimize the following combined loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{CTC}} + \lambda_1 \mathcal{L}_{\text{REC}} + \lambda_2 \mathcal{L}_{\text{maskedCE}}, \tag{5}$$

where λ_1 and λ_2 are weights for the reconstruction loss and the masked CE loss, respectively.

2.4. Postprocessing

Since our method performs song-level alignment, we remove silent segments at the beginning and end of each song prior to forced alignment to improve alignment performance and robustness.

2.4.1. Removing Silent Segments at the Beginning and End

To remove the silent segments at the beginning and end, we compute the sum of the mel-spectrogram along the frequency axis, resulting in a vector that represents the frame-wise energy of the song. We normalize this energy vector so that its maximum value is 1, and apply a threshold of 0.05 to identify non-silent frames. Among the frames with energy above the threshold, we take the first frame as the starting point and the last frame as the ending point. We then extend both points by one second (63 frames) to ensure that all singing parts are included. Only the audio segment between these two points is used for alignment.

2.4.2. Forced Alignment

We align the phoneme sequence to the corresponding audio using the CTC-segmentation, a force alignment method adapted for CTC-based model predictions [24].

First, we compute the trellis matrix $\mathbf{k}$, with an element $k_{t,m}$ of $\mathbf{k}$ representing the maximum joint probability of aligning the phoneme sequence up to index m with the audio up to frame t. For each time frame, this maximum probability is computed by considering the more likely of two possible transitions: staying on the same phoneme as in the previous frame ($\mathbb{P}_{\text{stay}}$), or transitioning to the next phoneme ($\mathbb{P}_{\text{transit}}$). The trellis matrix is computed according to the following recurrence:

$$k_{t,m} = \begin{cases} \max(\mathbb{P}_{\text{stay}}, \mathbb{P}_{\text{transit}}) & \text{if} \quad t > 0 \text{ and } m > 0 \\ 0 & \text{if} \quad t = 0 \text{ and } m > 0 \\ 1 & \text{if} \quad m = 0 \end{cases} \tag{6}$$

where

$$\begin{aligned} \mathbb{P}_{\text{stay}} &= k_{t-1,m} \mathbb{P}(\epsilon|t) \\ \mathbb{P}_{\text{transit}} &= k_{t-1,m-1} \mathbb{P}(y_m|t). \end{aligned} \tag{7}$$

In Equation 7, $\mathbb{P}(y_m|t)$ denotes the probability of the m^{th} phoneme label y_m at frame t, and $\mathbb{P}(\epsilon|t)$ denotes the probability of staying on the same phoneme (blank symbol ϵ) at frame t.

Once the trellis matrix is constructed, we perform backtracking to determine the most probable sequence of phoneme labels. These are then merged into words, and the word onsets are extracted for word-level alignment. We use the PyTorch implementation[3] for forced alignment and word onset detection.

[3] https://pytorch.org/audio/stable/tutorials/forced_alignment_tutorial.html

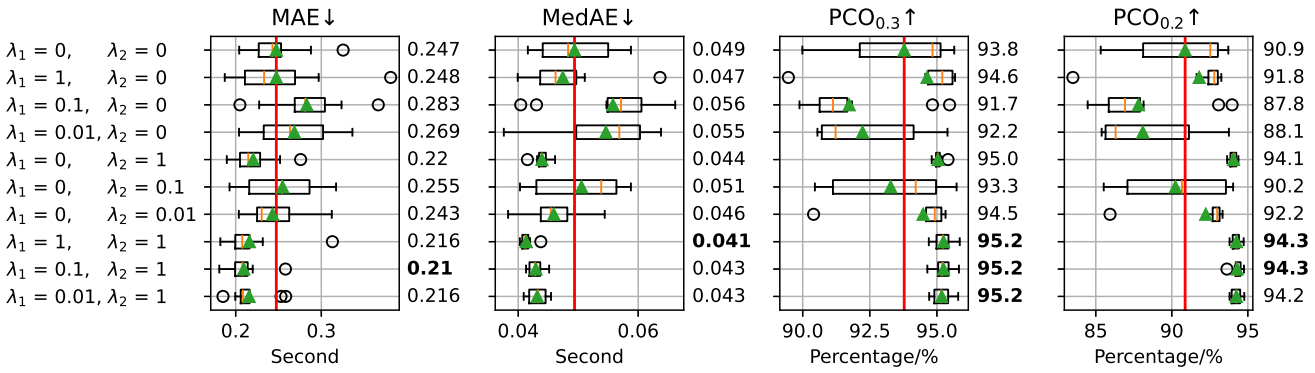

Figure 5: Word-level alignment results on the Jamendo dataset (test set). λ_1 and λ_2 are the weights for the reconstruction loss and the masked CE loss, respectively. For each weight setting, the results of 10 trained models (see Section 3.2) are shown, with the average value indicated by the green triangle and shown on the right. The baseline results ($\lambda_1 = 0$, $\lambda_2 = 0$) are marked by red vertical lines for comparison. Bold numbers indicate the best results.

3. EXPERIMENTS AND RESULTS

In this section, we describe the datasets, training settings, and evaluation metrics used in our experiments, followed by the results and discussion.

3.1. Datasets

We use the DALI dataset [13] for training and validation. DALI contains 5,358 songs annotated at four hierarchical levels: note, word, line, and paragraph. For our experiments, we use 3,352 audio-available songs from the English subset of DALI. The songs are randomly split into training and validation sets with an 80/20 ratio, resulting in 2,681 training songs and 671 validation songs.

To evaluate our method, we use the publicly available Jamendo dataset [19], which is widely used for benchmarking lyrics-to-audio alignment. This allows for direct comparison with other methods that report results on the same dataset. The Jamendo dataset contains 20 English songs with both line-level and word-level annotations.

3.2. Training Settings

We train the model with a batch size of 32 for up to 20 epochs using the RMSprop optimizer with a learning rate of 10^{-4}. To mitigate variability due to random initialization and dropout, we train each model 10 times with different random seeds and report both the results of the 10 trained models and their average.

3.3. Evaluation Metrics

We evaluate word-level alignment performance by comparing the predicted word onsets with the ground-truth word onsets. Four evaluation metrics are used: Mean Absolute Error (MAE), Median Absolute Error (MedAE), and Percentage of Correct Onsets (PCO) with two tolerance windows, following [10]. The metrics are defined as follows:

$$\text{MAE} = \frac{1}{W}\sum_{w=1}^{W}|t^w_{pred} - t^w_{ref}|, \tag{8}$$

$$\text{MedAE} = \underset{1\le w\le W}{\text{median}}(|t^w_{pred} - t^w_{ref}|), \tag{9}$$

$$\text{PCO}_\tau = \frac{1}{W}\sum_{w=1}^{W} 1_{|t^w_{pred}-t^w_{ref}|<\tau} \times 100\%, \tag{10}$$

where W denotes the total number of words in a song, and w denotes the word index. t^w_{pred} and t^w_{ref} represent the predicted and reference onset times of the w^{th} word, respectively. We compute PCO with two tolerance window sizes: $\text{PCO}_{0.3}$ and $\text{PCO}_{0.2}$, representing the percentage of correctly predicted word onsets within 0.3 seconds and 0.2 seconds, respectively. All results are averaged over songs.

Among these metrics, MAE, MedAE, and $\text{PCO}_{0.3}$ are commonly used for evaluating lyrics-to-audio alignment performance. $\text{PCO}_{0.2}$ is additionally reported for comparison with the results in [16].

3.4. Results

Figure 5 shows the results of the proposed method trained with different combinations of the CTC loss, reconstruction loss, and masked CE loss, using various weighting values. Compared to the baseline model (CTC loss only, with $\lambda_1 = 0$ and $\lambda_2 = 0$), a clear improvement in MAE was observed—from 0.247 seconds to 0.22 seconds—by adding the masked CE loss ($\lambda_1 = 0$ and $\lambda_2 = 1$). The MAE was further improved by using both the reconstruction loss and the masked CE loss (with $\lambda_1 = 1$, 0.1, or 0.01, and $\lambda_2 = 1$). Similar trends were observed for the MedAE and PCO results, confirming the effectiveness of the proposed masked CE loss.

3.5. Comparison to State-of-the-Art Results

Considering all the above results, we selected the proposed model with weights $\lambda_1 = 1$ and $\lambda_2 = 1$ for comparison with other state-of-the-art models, with the results summarized in Table 1. Our model achieved an MAE of 0.216 seconds, which was comparable to other models trained on the DALI dataset. Lower MAEs were reported by methods trained on large-scale in-house datasets. DSE [22] achieved an MAE of 0.15 seconds using an in-house dataset of 87,785 songs with professional-quality recordings in English, Spanish, German, and French. HX-IH [16] achieved an MAE of 0.16 seconds using an in-house dataset consisting of approximately 67,000 Korean and English songs.

Table 1: Comparison with state-of-the-art methods. Bold numbers indicate the best results across all methods, while underlined numbers indicate the best results among methods trained using the DALI dataset.

	MAE ↓	MedAE ↓	$PCO_{0.3}$ ↑	$PCO_{0.2}$ ↑	Training Dataset
GC [15]	0.22	0.05	94%		DALI
HBE [8]	0.23		94%		DALI
KGLW [17]	<u>0.20</u>		94%		DALI
HX-D [16]	0.42	0.043	89%	87%	DALI
HX-IH [16]	0.16	0.043	93%	91%	In-house 67k
DSE [22]	**0.15**		92%		In-house 88k
Proposed ($\lambda_1 = 1, \lambda_2 = 1$)	0.216	**<u>0.041</u>**	**<u>95.2</u>%**	**<u>94.3</u>%**	DALI

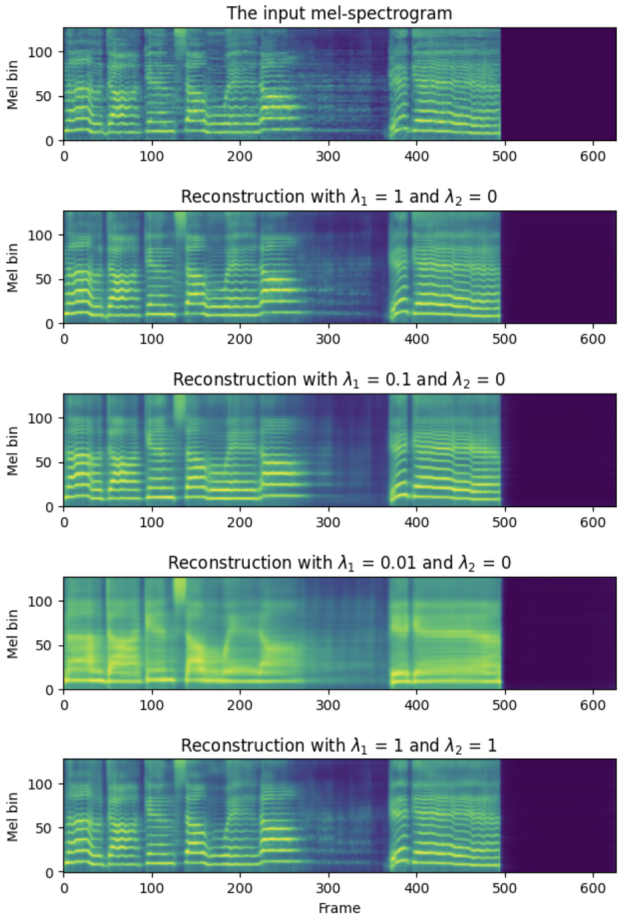

Figure 6: The input mel-spectrogram and reconstructed mel-spectrograms obtained with different reconstruction loss weights ($\lambda_1 = 1$, 0.1, and 0.01).

With respect to the MedAE, $PCO_{0.3}$, and $PCO_{0.2}$ metrics, our method outperformed all other methods, including those trained on in-house datasets. We inspected our alignment results and found that they contained a large proportion of small absolute errors (as reflected by the good MedAE and PCO results), but also several outliers with large absolute errors that contributed to the relatively high MAE. We expect that incorporating a line-level alignment stage could help reduce these outliers.

3.6. Discussion

We observed an interesting phenomenon: the reconstructed mel-spectrograms vary depending on the weight of the reconstruction loss, as shown in Figure 6. With larger reconstruction weights (e.g., $\lambda_1 = 1$ and $\lambda_2 = 0$), the reconstruction closely resembles the input mel-spectrogram. In contrast, as λ_1 decreases, the reconstructed mel-spectrograms become smoother, with pitch information increasingly suppressed (e.g., $\lambda_1 = 0.01$ and $\lambda_2 = 0$). Since the reconstructed mel-spectrogram is generated from the model's predictions of frame-wise phoneme probabilities, it tends to focus more on phonetic content and less on pitch information as the reconstruction weight λ_1 decreases. Note that the masked CE loss has little impact on the reconstruction (see the result with $\lambda_1 = 1$ and $\lambda_2 = 1$).

Although obtaining smoothed reconstructed mel-spectrograms with suppressed pitch information (e.g., $\lambda_1 = 0.01$) is not the goal of this study, such smoothed mel-spectrograms could be useful for other singing voice research tasks, such as singer identification [25] and singing voice synthesis [26]. Exploring their potential usefulness would be an interesting direction for future work.

4. CONCLUSIONS

In this paper, we proposed a masked CE loss to constrain model predictions at word onsets, offsets, and silent frames for training a lyrics-to-audio alignment model. Through experiments, we demonstrated its effectiveness by showing that combining the masked CE loss with the CTC loss and reconstruction loss improves alignment performance. The resulting model achieved state-of-the-art word-level alignment accuracy. The proposed method offers several advantages. First, it provides a more efficient way to leverage limited and valuable annotation data. Second, frames with derived phoneme labels—such as onsets, offsets, and silent segments—are especially important for word-level alignment, and incorporating them via the masked CE loss enhances performance. Third, the model is based on a simple CRNN architecture and operates at the song level, taking the entire song as input. This enables the model to capture long-range temporal dependencies while simplifying inference.

As for future work, we are interested in investigating whether additional frame-wise phoneme labels can be derived for the intermediate frames within words, beyond just the onset and offset frames. Identifying such labels could help further improve alignment performance by providing the model with more detailed guidance within word segments.

5. ACKNOWLEDGMENT

This work was supported in part by JST CREST Grant Number JPMJCR20D4, Japan. We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use."

6. REFERENCES

[1] Hiromasa Fujihara and Masataka Goto, "Lyrics-to-Audio Alignment and its Application," in *Multimodal Music Processing*, Meinard Müller, Masataka Goto, and Markus Schedl, Eds., vol. 3 of *Dagstuhl Follow-Ups*, pp. 23–36. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2012.

[2] Jun Kato, Tomoyasu Nakano, and Masataka Goto, "TextAlive: Integrated Design Environment for Kinetic Typography," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (ACM CHI 2015)*, 2015, pp. 3403–3412.

[3] Anna M. Kruspe, "Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, 2016, pp. 358–364.

[4] Eric J. Humphrey, Sravana Reddy, Prem Seetharaman, Aparna Kumar, Rachel M. Bittner, Andrew Demetriou, Sankalp Gulati, Andreas Jansson, Tristan Jehan, Bernhard Lehner, Anna Krupse, and Luwei Yang, "An introduction to signal processing for singing-voice analysis: High notes in the effort to automate the understanding of vocals in music," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 82–94, 2019.

[5] Bidisha Sharma, Chitralekha Gupta, Haizhou Li, and Ye Wang, "Automatic lyrics-to-audio alignment on polyphonic music using singing-adapted acoustic models," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 396–400.

[6] Andrea Vaglio, Romain Hennequin, Manuel Moussallam, Gaël Richard, and Florence d'Alché Buc, "Multilingual lyrics-to-audio alignment," in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 512–519.

[7] Emir Demirel, Sven Ahlbäck, and Simon Dixon, "Low resource audio-to-lyrics alignment from polyphonic music recordings," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2021)*, 2021, pp. 586–590.

[8] Jiawen Huang, Emmanouil Benetos, and Sebastian Ewert, "Improving lyrics alignment through joint pitch detection," in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2022)*, 2022, pp. 451–455.

[9] Jun-You Wang, Chon-In Leong, Yu-Chen Lin, Li Su, and Jyh-Shing Roger Jang, "Adapting pretrained speech model for mandarin lyrics transcription and alignment," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2023)*, 2023.

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

[11] Kilian Schulze-Forster, Clement S. J. Doire, Gaël Richard, and Roland Badeau, "Phoneme level lyrics alignment and text-informed singing voice separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2382–2395, 2021.

[12] Yi Ren, Xu Tan, Tao Qin, Jian Luan, Zhou Zhao, and Tie-Yan Liu, "DeepSinger: Singing voice synthesis with data mined from the web," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2020)*, 2020, pp. 1979–1989.

[13] Gabriel Meseguer-Brocal, Alice Cohen-Hadria, and Geoffroy Peeters, "DALI: a large Dataset of synchronized Audio, LyrIcs and notes, automatically created using teacher-student machine learning paradigm," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018, pp. 431–437.

[14] 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 (TISMIR)*, vol. 3, no. 1, pp. 55–67, 2020.

[15] Chitralekha Gupta, Emre Yılmaz, and Haizhou Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2020)*, 2020, pp. 496–500.

[16] Minsung Kang, Soochul Park, and Keunwoo Choi, "HCLAS-X: Hierarchical and cascaded lyrics alignment system using multimodal cross-correlation," arXiv 2307.04377, 2023.

[17] Timon Kick, Florian Grötschla, Luca A Lanzendörfer, and Roger Wattenhofer, "Contrastive lyrics alignment with a timestamp-informed loss," in *Proceedings of the NeurIPS Workshop on AI-Driven Speech, Music, and Sound Generation*, 2024.

[18] Chitralekha Gupta, Emre Yılmaz, and Haizhou Li, "Acoustic modeling for automatic lyrics-to-audio alignment," in *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, 2019, pp. 2040–2044.

[19] Daniel Stoller, Simon Durand, and Sebastian Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 181–185.

[20] Alex Graves, Santiago Fernández, Faustino John Gomez, and Jürgen A Schmidhuber, "Connectionist Temporal Classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, 2006, pp. 369–376.

[21] Yann Teytaut and Axel Roebel, "Phoneme-to-audio alignment with recurrent neural networks for speaking and singing voice," in *Proceedings of the 22th Annual Conference of the International Speech Communication Association (Interspeech 2021)*, 2021, pp. 61–65.

[22] Simon Durand, Daniel Stoller, and Sebastian Ewert, "Contrastive learning-based audio to lyrics alignment for multiple languages," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2023)*, 2023.

[23] Simon Rouard, Francisco Massa, and Alexandre Défossez, "Hybrid transformers for music source separation," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2023)*, 2023.

[24] Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll, "CTC-segmentation of large corpora for german end-to-end speech recognition," in *Speech and Computer*, Alexey Karpov and Rodmonga Potapova, Eds., 2020, pp. 267–278.

[25] Dorian Desblancs, Gabriel Meseguer-Brocal, Romain Hennequin, and Manuel Moussallam, "From real to cloned singer identification," in *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR 2024)*, 2024, pp. 327–334.

[26] Chitralekha Gupta, Haizhou Li, and Masataka Goto, "Deep learning approaches in topics of singing information processing," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2422–2451, 2022.