GENRE-CONDITIONED ACOUSTIC MODELS FOR AUTOMATIC LYRICS TRANSCRIPTION OF POLYPHONIC MUSIC

Xiaoxue Gao[1], *Chitralekha Gupta*[2] *and Haizhou Li*[3]

[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[2]Department of Communications and New Media, National University of Singapore, Singapore
[3]The Chinese University of Hong Kong, Shenzhen, China

ABSTRACT

Lyrics transcription of polyphonic music is challenging not only because the singing vocals are corrupted by the background music, but also because the background music and the singing style vary across music genres, such as pop, metal, and hip hop, which affects lyrics intelligibility of the song in different ways. In this work, we propose to transcribe the lyrics of polyphonic music using a novel genre-conditioned network. The proposed network adopts pre-trained model parameters, and incorporates the genre adapters between layers to capture different genre peculiarities for lyrics-genre pairs, thereby only requiring lightweight genre-specific parameters for training. Our experiments show that the proposed genre-conditioned network outperforms the existing lyrics transcription systems.

Index Terms— Lyrics transcription of polyphonic music, singing voice separation, music information retrieval

1. INTRODUCTION

Despite much progress in automatic speech recognition (ASR) [1, 2] and deep learning [3–5], there are fewer studies in lyrics transcription of polyphonic music. The aim of automatic lyrics transcription is to transcribe lyrics from a song that contains singing vocals mixed with background music. Lyrics transcription has attracted a lot of interest to aid applications such as automatic generation of karaoke lyrical content, music video subtitling and query-by-singing [6].

Background music often correlates with the singing vocals, making the task of lyrics transcription challenging. Past studies have broadly taken two approaches to deal with the background music: 1) an *extraction-transcription* approach, in which singing vocals are extracted from the polyphonic music [7–9] as a pre-processing step, and lyrics are transcribed from these extracted vocals; and 2) a *music-aware* approach, whereby the background music knowledge is used to help the transcription model [10, 11].

In the *extraction-transcription* approach, several singing vocal separation techniques have been studied to suppress the background music and extract the singing vocals from the polyphonic music that are used for acoustic model training [7–9, 12]. However, due to imperfections in music removal and the distortions associated with the inversion of a magnitude spectral representation, the extracted time-domain singing vocal signals often contain artifacts. Acoustic model trained on such extracted vocals are far from perfection [7, 11].

Another way is to use acoustic model trained on clean singing vocals, and at the time of inference, apply source separation technique to extract singing vocal from the input polyphonic song to transcribe lyrics [13–15]. However, this two-step approach not only needs a large amount of data for training the vocal separation model as well as the acoustic model [16], but also suffers from mismatch between the acoustic features between training and testing, thereby causing degradation of the performance of acoustic modeling in lyrics recognition.

Gupta et al. [17] found that an acoustic model trained on a large amount of solo singing data, when adapted with a small amount of in-domain polyphonic data outperforms the lyrics transcription performance of solo-singing acoustic models adapted with extracted singing vocals. This suggests that the polyphonic data, i.e. singing vocals+background music, helps in learning the spectro-temporal variations of the background music more than the extracted vocals.

In the *music-aware* approach, instead of removing the background music, the systems [10, 11] make use of music information by directly training with the polyphonic music input for lyrics recognition. For example, Stoller et al. [10] adopted an end-to-end wave-U-net model to predict character probabilities from the polyphonic audio, while Gupta et al. [11] designed a music genre-based kaldi-based acoustic model, that outperforms all previous techniques. These studies show that the task of lyrics transcription in polyphonic music can benefit from the knowledge of background music.

Different genres exhibit significantly different levels of lyric intelligibility in polyphonic music [18], since the genres vary in their musical characteristics such as instrumental accompaniment, singing vocal loudness, syllable rate, reverberation, and singing style [11, 18]. In [11], the music genre tag was embedded in the pronunciation dictionary at the time of training, while at the time of inference, genre information was not provided. In contrast, in this work, we believe that the predictable genre-class information would help an automated lyrics transcription system with lyrics intelligibility. We propose a music genre-conditioned training strategy to adapt an end-to-end lyrics transcription system according to the music genre. Inspired by the success of adaptive fine-tuning with pre-trained models in natural language processing [19] and speech translation [20–22], we propose to incorporate genre-specific adapters to a pre-trained transformer-based polyphonic lyrics transcription model [23].

2. GENRE-CONDITIONED AUTOMATIC LYRICS TRANSCRIPTION

Music genre are categorical labels created by human experts to characterize and structure pieces of music [24, 25]. Musical genres differ from each other in their instrumental accompaniment, rhythmic structure, vocal harmonization, reverberation and pitch content of the music [24,25]. Lyrics intelligibility in polyphonic music is found to be influenced by these genre specific characteristics [11, 18]. Factors such as instrumental accompaniment, vocal harmonization, and reverberation are expected to interfere with lyric intelligibility, while predictable rhyme schemes and semantic context might improve intelligibility [18]. For example, as observed in [18], in *metal* songs,

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

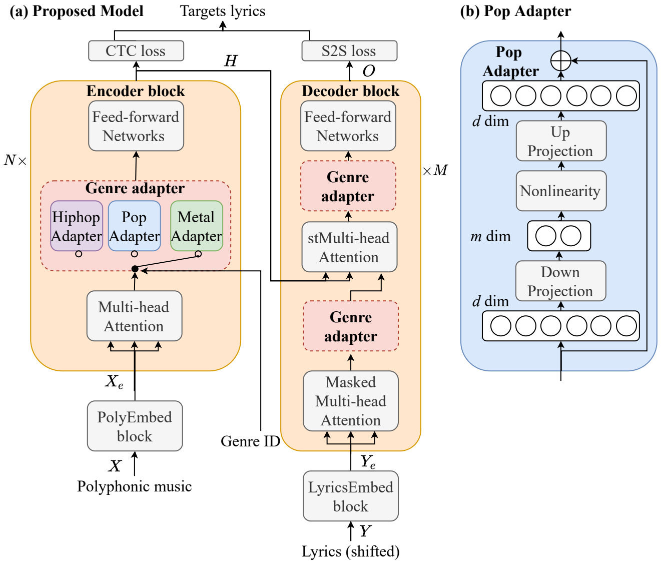

Fig. 1. An overview architectures of (a) the proposed genre-conditioned lyric transcriber with a Transformer architecture. For simplicity, layer normalization and residual connection are omitted, and genre ID is applied for all genre adapters; and (b) the pop adapter, which has the same design for metal and hiphop adapters.

the accompaniment is loud and interferes with the vocals, while is relatively softer in *jazz*, *country*, and *pop* songs. As a result "Death Metal" songs shows lyrics intelligibility scores of zero, while "Pop" songs achieve scores close to 100% [18]. Another difference is the syllable rate between genres. In [18], it was observed that *rap* songs, that have a higher syllable rate, show lower lyric intelligibility than other genres. We expect that these factors are important for building an automatic lyrics transcription framework.

2.1. Genre-conditioned Acoustic Model

The proposed lyric transcriber consists of an encoder block, and a decoder block. The encoder block is designed to produce an acoustic latent representation from the input polyphonic music, while the decoder block combines the encoded acoustic representation and the previously predicted lyrical information to transcribe the lyrics. The focus of this study is a simple yet effective genre-specific adapter and its integration with a pre-trained model.

Our framework is based on transformer [26] with joint encoder-decoder and connectionist temporal classification (CTC) architecture [27]. We incorporate a genre-conditioned encoder that converts the input polyphonic acoustic features to intermediate genre-specific representations, and a genre-conditioned decoder that predicts lyrical tokens, i.e. sub-words in this paper, one at a time given the intermediate genre-specific representations and the previously predicted lyrical tokens in an auto-regressive manner, as illustrated in Fig 1 (a). The core module of the genre-conditioned training framework is a genre adapter, that employs a down-projection and an up-projection with non-linear activation function to make use of different genre temporal context of the polyphonic music input sequence.

2.1.1. Genre Broadclasses

Music has been divided into different genres in many different and overlapping ways, based on a shared set of characteristics [25]. The main difference between genres that affects lyrics intelligibility are the relative singing vocal volume compared with background music and syllable rate [11, 18]. Based on the shared characteristics between genres that affect lyrics intelligibility, Gupta et al. [11] mapped all genres into three broad genre classes namely *pop*, *metal* and *hiphop*. The song containing some rap along with electronic music were mapped to *hiphop* broadclass, that includes genres such as Rap, Hip Hop, and Rhythms & Blues. Songs with loud and dense background music were categorized as *metal*, that includes genres such as Metal and Hard Rock. And songs with clear and louder vocals under genres Pop, Country, Jazz, Reggae etc. were categorized as *pop* broadclass. In this work, we follow the same categorization and broadclasses.

2.1.2. Genre-specific Adapters

We incorporate a genre-adapter as a new block of layers between the transformer layers of a pre-trained base model [23] (Fig. 1 (a)). A genre-specific adapter first projects the original *d*-dimensional input feature into a smaller bottleneck dimension, *m*, applies a nonlinearity, and then projects back to *d* dimension. We expect that this bottleneck lower-dimensional feature, *m*, captures genre-specific information. The genre-specific adapter itself has a skip-connection internally as shown in Fig. 1 (b). With the skip-connection, if the parameters of the projection layers are initialized to near-zero, the module is initialized to an approximate identity function.

The idea is that the holistic polyphonic representation, as captured by the pre-trained base model (which is trained on polyphonic, i.e. singing+background music data), provides information that is common across the different genres, while the genre adapters capture the genre-specific characteristics, thereby tuning the base model as per genre identity of the song. The genre broadclass ID, for example pop, is given to the genre adapter to choose the corresponding adapter pathway, i.e. pop adapter, to characterize the corresponding genre attributes as shown in Fig. 1 (b). Most of the weights of the pre-trained base model are untouched, except the layer normalization [28] and the source-target multi-head attention (stMulti-head Attention in Fig. 1 (a)).

In this work, we explore inserting the genre-adapter at multiple locations in the pre-trained base model architecture: in the encoder, and at two places in the decoder. The genre-conditioned encoder would adapt the network according to the genre-specific acoustic characteristics, while the genre-conditioned decoder would also capture the lyrical characteristics specific to genre. In the following subsections, we describe the genre-conditioned modules individually.

2.1.3. Genre-conditioned Encoder

The genre-conditioned encoder consists of an embedding block (PolyEmbed) and N identical encoder blocks where each encoder block contains a multi-head attention (MHA) [26], a genre adapter, and a position-wise feed-forward network (FFN). The input sequence $\mathbf{X}$ is first encoded into $\mathbf{X}_e$ by PolyEmbed block using subsampling and positional encoding (PE) [26]. The genre-conditioned encoder then transform $\mathbf{X}_e$ into a hidden representation $\mathbf{H}$ via the MHA, the genre adapter and the FFN. Residual connection [29] and layer normalization [28] are employed inside each of the encoder blocks for MHA and FFN as in [26].

$$\begin{aligned} \mathbf{X}_e &= \text{PolyEmbed}(\mathbf{X}), \\ \mathbf{H} &= \text{GenreEncoder}(\mathbf{X}_e) \end{aligned} \tag{1}$$

2.1.4. Genre-conditioned Decoder

The genre-conditioned decoder consists of a textual embedding block (LyricsEmbed) and M identical decoder blocks, where each decoder block has a masked MHA, a MHA, a FFN and genre adapters. During training, $\mathbf{Y}$ represents the lyrical token history that is offset right by one position, however, at run-time inference, it represents the previous predicted token history. $\mathbf{Y}$ is first converted to

Table 1. A description of polyphonic music dataset that consists of DALI and NUS collections.

		# songs	# lines	duration
Poly-train	DALI-train	3,913	180,034	208.6 hours
	NUS	517	264,62	27.0 hours
Poly-dev	DALI-dev	100	5,356	3.9 hours
	NUS	70	2,220	3.5 hours
Poly-test	Hansen	10	212	0.5 hour
	Jamendo	20	374	0.9 hour
	Mauch	20	442	1.0 hour

lyrics token embedding $\mathbf{Y}_e$ via a LyricsEmbed block, that consists of an embedding layer, and a positional encoding (PE) operation.

$$\begin{aligned} \mathbf{Y}_e &= \text{LyricsEmbed}(\mathbf{Y}), \\ \mathbf{O} &= \text{GenreDecoder}(\mathbf{H}, \mathbf{Y}_e) \end{aligned} \tag{2}$$

The lyrics embedding $\mathbf{Y}_e$ is fed into the masked MHA that ensures causality, i.e. the predictions for current position only depends on the past positions. The output of the masked MHA is fed into the genre-adapter, whose outputs and the acoustic encoding $\mathbf{H}$ are then fed to the source-target MHA for capturing the relationship between acoustic information $\mathbf{H}$ and the genre-informed textual information. The output of the MHA is then passing through the another genre adapter and a FFN to generate the decoder output $\mathbf{O}$. The residual connection [29] and layer normalization [28] are also employed in each of the decoder blocks for masked MHA, stMHA and FFN [26].

2.1.5. Multi-task Learning Objective

We employ both CTC [27] and sequence-to-sequence (S2S) objective functions for model training. The M decoder blocks are followed by a linear projection and softmax layers, that converts the decoder output $\mathbf{O}$ into a posterior probability distribution of the predicted lyrical token sequence $\mathbf{G}_{s2s}$. The S2S loss, $\mathcal{L}^{\text{S2S}}$, is the cross-entropy of the ground-truth lyrical token $\mathbf{R}$ and $\mathbf{G}_{s2s}$. Also, a linear transform is applied on $\mathbf{H}$ to obtain the token posterior distribution $\mathbf{G}_{ctc}$. CTC loss is computed between $\mathbf{G}_{ctc}$ and $\mathbf{R}$. The network is trained to minimize both S2S and CTC losses jointly with an objective function $\mathcal{L}_{\text{Genre-con}}$,

$$\begin{aligned} \mathcal{L}_{\text{Genre-con}} &= \alpha\mathcal{L}^{\text{CTC}} + (1-\alpha)\mathcal{L}^{\text{S2S}}, \\ \mathcal{L}^{\text{CTC}} &= \text{Loss}_{\text{CTC}}(\mathbf{G}_{ctc}, \mathbf{R}), \\ \mathcal{L}^{\text{S2S}} &= \text{Loss}_{\text{S2S}}(\mathbf{G}_{s2s}, \mathbf{R}) \end{aligned} \tag{3}$$

where $\alpha \in [0, 1]$, $\mathbf{R}$ is the ground-truth lyrical token sequence. During run-time inference, the genre-conditioned model converts input polyphonic acoustic features to output lyrical token sequence.

3. EXPERIMENTS

3.1. Datasets

As shown in Table 1, the polyphonic music training dataset, Poly-train, consists of the DALI-train [30] dataset and a NUS proprietary collection. The DALI-train dataset consists of 3,913 English polyphonic song tracks [1]. The dataset is processed into 180,034 lyrics-transcribed audio lines with a total duration of 208.6 hours. The NUS collection dataset consists of 517 popular English songs. We obtain its line-level lyrics boundaries using the state-of-the-art audio-to-lyrics alignment system [31], leading to 26,462 lyrics-transcribed audio lines with a total duration of 27.0 hours.

Table 2. The genre distribution for polyphonic music Dataset.

Statistics	**Metal**	**Pop**	**Hiphop**
Percentage in Poly-train	35%	59%	6%
Percentage in Poly-dev	48%	49%	3%
Percentage in Poly-test	34%	56%	10%

The Poly-dev dataset consists of the DALI-dev dataset of 100 songs from DALI dataset [31], and 70 songs from a NUS proprietary collection. We adopt three widely used test sets – Hansen [32], Jamendo [10], and Mauch [33] to form the Poly-test as shown in Table 1. The test datasets are English polyphonic songs, that are manually segmented into line-level segments of average duration 8 seconds. We transcribe the lyrics line-by-line, instead of the whole song, to avoid possible accumulated errors in the Viterbi decoding that occur in long duration audio clips [34, 35].

Genre tags for all of the NUS collection dataset and most of the songs in DALI-train and DALI-dev sets are provided in their metadata, except for 840 songs in DALI. For these remaining songs, we apply an automatic genre recognition implementation [36], that has 80% accuracy, to obtain their genre IDs. For the test sets, we scan the web to find the genre tags of each song. For each song in train, dev, and test sets, the genre tags are mapped to one of the three genre broadclasses, pop, metal, or hiphop, according to the mapping described in [11]. The statistics of music genre distribution for the polyphonic music datasets are provided in Table 2.

3.2. Experimental Setup

We use ESPnet [37] with pytorch backend to build all the models (including the pre-trained base model [23] and the proposed genre-conditioned models). We extract 83-dimensional filterbank features along with pitch from the audio files with a window of 25 ms, and a frame-shift of 10 ms. We use sub-words as the model token units for the task of lyrics transcription, and 5,000 sub-words are generated using byte-pair encoding (BPE). All models are trained with the Adam optimizer with a Noam learning rate decay, 25,000 warmup steps, 5,000,000 batch-bin, and 100 epochs as in [26]. The PolyEmbed block contains two CNN blocks with a kernel size of 3 and a stride size of 2.

Other parameters of all the models follow the default settings in the published LibriSpeech model (LS) [2], where attention dim is 512, the number of heads is 8 in MHA, FFN laryer dim is 2048, the interpolation factor α between CTC loss and S2S loss (Eq. 3) is set to 0.3, and there are 12 encoder blocks and 6 decoder blocks ($N = 12$ and $M = 6$). We follow the default setting in ESPnet [37] to average the best 5 validated model checkpoints on the development set Poly-dev (Table 1) to obtain the final model. We follow the common joint decoding approach [27, 38], which takes CTC prediction into account during decoding where we set the same decoding parameters (penalty, beam width and CTC decoding weight are set to 0.0, 10 and 0.3, respectively).

The pre-trained base model [23] is trained using Poly-train, and Poly-dev as the development set and Poly-test to evaluate, and the base model is pre-trained by solo-singing model using the English solo-singing dataset *Sing! 300 $\times$ 30 $\times$ 2* [3], whose weights are initialized by the published LS model. In the genre-conditioned models, we insert the genre adapters in the pre-trained base model, as shown in Figure 1 (a). In each genre-adapter, the down-projection and up-projection layers are linear layers with $d = 512$ and $m = 256$, along with a ReLU non-linearity function.

[1] There are a total of 5,358 audio tracks in DALI, but we only have access to 3,913 English audio links.

[2] Pretrained LS model "pytorch large Transformer with specaug (4 GPUs) + Large LSTM LM" from the ESPNET github `https://github.com/espnet/espnet/blob/master/egs/librispeech/asr1/RESULTS.md`.

[3] The audio files can be accessed from https://ccrma.stanford.edu/damp/

4. RESULTS AND DISCUSSION

We study the effects of genre-conditioned approach and the places to plug-in the genre adapters. We also compare the performance of the proposed models with the SOTA systems for lyrics transcription of polyphonic music and conducts an ablation study. We report the performance in terms of word error rate (WER), which is the ratio of the total number of insertions, substitutions, and deletions with respect to the total number of words per song.

4.1. Genre-Conditioned Training

We study the effectiveness of genre-conditioned training approach compared to the pre-trained base model. The lyrics transcription performance of the framework with and without the genre-conditioned training is presented in Table 3. In Genre MHA+MaskMHA model, the genre adapter is inserted in the encoder and the two places in the decoder, as shown in Fig. 1. In Genre MHA model, we insert the genre-adapter in the encoder and only at one place in the decoder, i.e. the one after stMHA.

The two genre-conditioned models (Genre MHA and Genre MHA+MaskMHA) outperform the base model for pop and metal songs, and comparable with the base model for the hiphop songs (only 5 hiphop songs in Poly-test). This suggests that the genre-specific adapters are able to capture the differences across music genres in a rich polyphonic environment. One should note that the number of pop and metal songs in the train and dev datasets are considerably more than that of hiphop songs (only 6% in Poly-train and 3% in Poly-dev are hiphop songs), as illustrated in Table 2. The lack of data for training the hiphop adapter results in sub-par performance of the system on the songs of this genre.

We further investigate the effect of different places of inserting the genre adapters. The genre-adapter in the encoder captures the genre-specific information from the acoustic features. The first adapter in the decoder intends to capture genre-related features from the previously predicted lyrics, while the second adapter in the decoder captures the genre-information from the combination of acoustic and lyrical features. We find that the Genre MHA model (that does not have the first decoder adapter which only takes lyrical features as input), performs better than the Genre MHA+MaskMHA model (that has all the three adapters) for metal and pop genres. This observation aligns with the fact that the characteristics that define these music genres depend more on the acoustic features (such as loudness, musical instrument types), than on textual features. For hiphop, however, the textual features can indicate the high syllable rate of rap songs, thereby, the first adapter of the decoder might help, resulting in a slightly better performance of Genre MHA+MaskMHA for hiphop genre. However, as mentioned earlier, the amount of training, dev, and test data for hiphop genre was small. Therefore, hiphop genre needs further investigation in the future.

For ablation study purpose, the Genre MHA Ablation experiment is conducted with one common adapter with pop, hiphop and metal parameters shared. The proposed Genre MHA outperforms the Genre MHA Ablation for pop and hiphop songs. The genre-specific adapter is especially beneficial to the hiphop songs compared with the common adapter, which suggests that the different properties of different genres should be considered separately.

4.2. Comparison with Prior Studies

We compare the proposed models with the existing approaches [10, 14, 23, 31, 39, 40] for lyrics transcription of polyphonic music in Table 3. Stoller et al.'s [10] system is based on E2E Wave-U-Net framework, while the remaining systems [14, 31, 39, 40] are based on the Kaldi. A subset of these systems [14, 39, 40] were submitted to the lyrics transcription task in MIREX 2020, where the GGL1 system [40] outperformed other submissions[4].

Table 3. Comparison between the proposed genre-adapter solutions and other existing competitive solutions to lyrics transcription (WER%) of polyphonic music.

Whole songs test	**Hansen**	**Jamendo**	**Mauch**
DS [10]	-	77.80	70.90
RB1 [14]	83.43	86.70	84.98
DDA2 [39]	74.81	72.15	75.39
DDA3 [39]	77.36	73.09	80.66
CG [31]	-	59.60	44.00
GGL2 [40]	48.11	61.22	45.35
GGL1 [40]	45.87	56.76	43.76
Line-level test	**Metal**	**Pop**	**Hiphop**
GGL1 [40]	59.70	37.07	57.08
Base model [23]	50.04	36.52	**51.19**
Genre MHA	**48.17**	**33.34**	52.32
Genre MHA Ablation	48.05	33.41	55.42
Genre MHA+MaskMHA	48.22	33.86	51.55

We first report the lyrics transcription performance of all existing systems on the same test sets for whole songs evaluation. As can be seen, GGL1 [40] performs the best among all and considered as the published state-of-the-art system. Therefore, we use GGL1 [5] as our point of comparison for the genre-based analysis next. To avoid accumulated errors while decoding long duration songs, we segment the songs in the test sets into lines, as described in Section 3.1. We further report the lyrics transcription results for these segmented test-sets across different genres. We observe that the base model performs better than GGL1 for all the genres, implying that the end-to-end transformer based framework works better than the kaldi-based conventional framework. Our proposed models outperform the base model for pop and metal songs, which indicates the superiority of genre-conditioned training over the base model. The genre models and the base model also outperform GGL1 across all the test data with relative 8%-19% improvements. This indicates the superiority of genre-conditioned training and the end-to-end transformer-based model over the conventional multi-step ASR pipeline.

5. CONCLUSION

We propose a novel genre-conditioned lyrics transcription network architecture that captures genre-specific information from the acoustics and the lyrics, and adapts the network as per the music genre. The proposed approach injects genre-specific adapter into the backbone transformer pre-trained model to interpret different genre attributes in a single network. We have presented a study of adapters for lyrics-genre pairs in polyphonic music, and shown the genre adapters can provide genre-related knowledge to help with music interference problem. Integrating genre-adapters with existing pre-trained model also shows the flexibility of using these adapters to explore different kinds of music data for the development of lyrics transcription system for polyphonic music.

Acknowledgement This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project No. A18A2b0046). This work is supported by A*STAR under its RIE2020 Advanced Manufacturing and Engineering Domain (AME) Programmatic Grant (Grant No. A1687b0033, Project Title: Spiking Neural Networks).

[4] `https://www.music-ir.org/mirex/wiki/2020:Lyrics_Transcription_Results`

[5] The results of GGL1 and GGL2 are obtained by standard Kaldi recipe with scoring, which are slightly different from those at MIREX2020 website.

6. REFERENCES

[1] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, “Purely sequence-trained neural networks for ASR based on lattice-free MMI,” in *INTERSPEECH*, 2016, pp. 2751–2755.

[2] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, “Deep convolutional neural networks for lvcsr,” in *IEEE ICASSP*, 2013, pp. 8614–8618.

[3] X. Gao, X. Tian, R. K. Das, Y. Zhou, and H. Li, “Speaker-independent spectral mapping for speech-to-singing conversion,” in *IEEE APSIPA ASC*, 2019, pp. 159–164.

[4] X. Gao, X. Tian, Y. Zhou, R. K. Das, and H. Li, “Personalized singing voice generation using wavernn,” in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 252–258.

[5] X. Gao, B. Sisman, R. K. Das, and K. Vijayan, “Nus-hlt spoken lyrics and singing (sls) corpus,” in *IEEE International Conference on Orange Technologies*, 2018, pp. 1–6.

[6] A. Mesaros, “Singing voice identification and lyrics transcription for music information retrieval invited paper,” in *IEEE Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 2013, pp. 1–10.

[7] C. Gupta, B. Sharma, H. Li, and Y. Wang, “Automatic lyrics-to-audio alignment on polyphonic music using singing-adapted acoustic models,” in *IEEE ICASSP*, 2019, pp. 396–400.

[8] A. Mesaros and T. Virtanen, “Automatic recognition of lyrics in singing,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, no. 1, p. 546047, 2010.

[9] G. B. Dzhambazov and X. Serra, “Modeling of phoneme durations for alignment between polyphonic audio and lyrics,” in *12th Sound and Music Computing Conference*, 2015, pp. 281–286.

[10] D. Stoller, S. Durand, and S. Ewert, “End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model,” in *IEEE ICASSP*, 2019, pp. 181–185.

[11] C. Gupta, E. Yılmaz, and H. Li, “Automatic lyrics transcription in polyphonic music: Does background music help?” *Proc. IEEE ICASSP*, pp. 496–500, 2020.

[12] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, “Lyricsynchronizer: Automatic synchronization system between musical audio signals and lyrics,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252–1261, 2011.

[13] C. Gupta, E. Yılmaz, and H. Li, “Lyrics-to-audio alignment with music-aware acoustic models.”

[14] G. R. Dabike and J. Barker, “The sheffield university system for the mirex 2020: Lyrics transcription task,” MIREX 2021.

[15] E. Demirel, S. Ahlbäck, and S. Dixon, “Low resource audio-to-lyrics alignment from polyphonic music recordings,” *arXiv preprint arXiv:2102.09202*, 2021.

[16] E. Demirel, S. AhlbÄck, and S. Dixon, “Automatic lyrics transcription using dilated convolutional neural networks with self-attention,” in *IEEE IJCNN*, 2020, pp. 1–8.

[17] C. Gupta, E. Yılmaz, and H. Li, “Acoustic modeling for automatic lyrics-to-audio alignment,” in *INTERSPEECH*, 2019.

[18] N. Condit-Schultz and D. Huron, “Catching the lyrics: intelligibility in twelve song genres,” *Music Perception: An Interdisciplinary Journal*, vol. 32, no. 5, pp. 470–483, 2015.

[19] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for nlp,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[20] H. Le, J. Pino, C. Wang, J. Gu, D. Schwab, and L. Besacier, “Lightweight adapter tuning for multilingual speech translation,” *arXiv preprint arXiv:2106.01463*, 2021.

[21] X. Li, C. Wang, Y. Tang, C. Tran, Y. Tang, J. Pino, A. Baevski, A. Conneau, and M. Auli, “Multilingual speech translation with efficient finetuning of pretrained models,” *arXiv preprint arXiv:2010.12829*, 2020.

[22] C. Xu, B. Hu, Y. Li, Y. Zhang, Q. Ju, T. Xiao, J. Zhu *et al.*, “Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders,” *arXiv preprint arXiv:2105.05752*, 2021.

[23] X. Gao, C. Gupta, and H. Li, “Automatic lyrics transcription of polyphonic music with lyrics-chords multi-task learning,” *under review in IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[24] M. Genussov and I. Cohen, “Musical genre classification of audio signals using geometric methods,” in *18th European Signal Processing Conference*. IEEE, 2010, pp. 497–501.

[25] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Neural Information Processing Systems*, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[27] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, “Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration,” in *INTERSPEECH*, 2019, pp. 1408–1412. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-1938

[28] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm,” in *ISMIR*, 2018, pp. 431–437.

[31] C. Gupta, E. Yılmaz, and H. Li, “Automatic lyrics alignment and transcription in polyphonic music: Does background music help?” in *IEEE ICASSP*, 2020, pp. 496–500.

[32] J. K. Hansen, “Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients,” in *9th Sound and Music Computing Conference (SMC)*, 2012, pp. 494–499.

[33] M. Mauch, H. Fujihara, and M. Goto, “Lyrics-to-audio alignment and phrase-level segmentation using incomplete internet-style chord annotations,” in *the 7th Sound and Music Computing Conference (SMC)*, 2010, pp. 9–16.

[34] P. J. Moreno, C. F. Joerg, J.-M. Van Thong, and O. Glickman, “A recursive algorithm for the forced alignment of very long audio segments.” in *ICSLP*, vol. 98, 1998, pp. 2711–2714.

[35] C. Gupta, R. Tong, H. Li, and Y. Wang, “Semi-supervised lyrics and solo-singing alignment.” in *ISMIR*, 2018, pp. 600–607.

[36] “Musical genre recognition using a cnn,” in *https://github.com/f90/Wave-U-Net*, accessed online 5 July 2010.

[37] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “Espnet: End-to-end speech processing toolkit,” in *INTERSPEECH*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[38] T. Hori, J. Cho, and S. Watanabe, “End-to-end speech recognition with word-based rnn language models,” in *IEEE SLT*, 2018, pp. 389–396.

[39] E. Demirel, S. Ahlback, and S. Dixon, “A recursive search method for lyrics alignment,” MIREX 2020.

[40] X. Gao, C. Gupta, and H. Li, “Lyrics transcription and lyrics-to-audio alignment with music-informed acoustic models,” MIREX 2021.