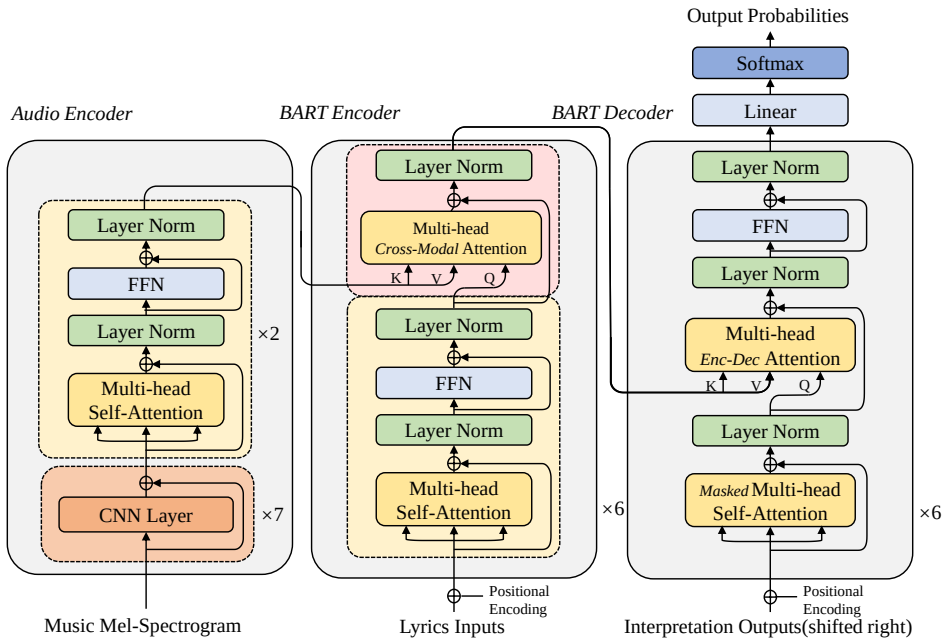

Figure 2. An overview of our proposed model. The model is divided into three parts from left to right: the audio encoder, the BART encoder, and the BART decoder. The fusion of the semantic representation of music audio and lyric text occurs in the upper part of the middle module (pink background). Only at the last two layers of the BART encoder, the music audio representation and the lyric text representation are semantically fused: the music audio representation and the lyric text representation are fed into the cross-modal attention module, and the result is added as an additional embedding to the original lyric text representation. The fused representation is fed into the BART decoder to generate an interpretation.

design an audio encoder to extract a representation following Won *et al.* [16]. The audio encoder uses a stack of CNN layers as a filter to extract local features, followed by a self-attention module to capture the global and temporal features of the audio.

The audio encoder receives an audio clip X_m and transforms it to a mel spectrogram H_m^0 as input. We compute the feature map of the i-th layer of the encoder as follows:

$$\widetilde{H}_m^i = \text{BN}(\text{CNN}_2(\text{ReLU}(\text{CNN}_1(H_m^{i-1})))) \quad (4)$$

$$H_m^i = \widetilde{H}_m^i + \text{BN}(\text{CNN}_3(H_m^{i-1})), \quad (5)$$

where BN is the Batch Normalization operation, and the CNNs are convolutional modules with different parameters. We add a residual connection to each CNN layer.

We then add two Transformer layers, identical to those in the BART encoder, to extract the final representation of music audio. Finally, we get the music audio representation $Z_m \in \mathbb{R}^{L_m \times d_m}$, where L_m and d_m denote the shape of the music audio feature map at the last CNN layer.

2.3 Representation Fusion

As shown in Figure 2, we insert a representation fusion module into the BART encoder to incorporate musical information. Inspired by Tsai *et al.* [17] and Yu *et al.* [14], we apply cross-modal attention to enable the music audio representation to be transferred to the lyric text domain. Jawahar *et al.* [23] have shown that BART encoders tend to extract semantic information in the last few layers, so we only fuse semantic representations at the final two Transformer layers.

For a specific Transformer layer i, we have the lyric text representation $H_t^i \in \mathbb{R}^{L_t \times d_t}$ and the music audio representation $Z_m \in \mathbb{R}^{L_m \times d_m}$, which is the same for each layer. We calculate the domain-adapted music audio representation with a multi-head Cross-Modal Attention (CMA) module:

$$H_{m\to t}^i = \text{CMA}(H_t^i W_Q', Z_m W_K', Z_m W_V') W_a', \quad (6)$$

where $H_{m\to t}^i \in \mathbb{R}^{L_t \times d_t}$ and the symbol $m \to t$ denotes cross-modal attention from music audio to the lyric domain. $d_{a'}$ is the dimensionality of the attention module, and similar to Eq. 3, $W_Q' \in \mathbb{R}^{d_t \times d_{a'}}$, $W_K' \in \mathbb{R}^{d_m \times d_{a'}}$ and $W_V' \in \mathbb{R}^{d_m \times d_{a'}}$ denote linear transformation matrices which map the representations to a common space. $W_a' \in \mathbb{R}^{d_{a'} \times d_t}$ linearly projects the attention value back to the lyric text dimension.

We add the cross-domain representation as an additional embedding [24] to the lyric text representation to get the final representation for the last two layers of the BART encoder:

$$H_{\text{fusion}}^i = H_t^i + H_{m\to t}^i. \quad (7)$$

3. SONG INTERPRETATION DATASET

The lack of suitable datasets has prevented deep learning models from learning to describe songs in natural lan-

guage. Related studies [25, 26] refer to some datasets, but they share two common problems: 1) the amount of data is small and 2) the datasets are not open-sourced. We propose a new dataset, the Song Interpretation Dataset, for the lyric interpretation task. [2]

The Song Interpretation Dataset combines data from two sources: (1) music and metadata from the Music4All Dataset [27], and (2) lyrics and user interpretations from SongMeanings.com [3] . We design a music metadata-based matching algorithm that aligns matching items in the two datasets with each other. In the end, we successfully match 25.47% of the tracks in the Music4All Dataset.

The dataset contains audio excerpts from 27,834 songs (30 seconds each, recorded at 44.1 kHz), the corresponding music metadata, about 490,000 user interpretations of the lyric text, and the number of votes given for each of these user interpretations. The average length of the interpretations is 97 words. Music in the dataset covers various genres, of which the top 5 are: Rock (11,626), Pop (6,071), Metal (2,516), Electronic (2,213) and Folk (1,760).

To the best of our knowledge, this is the first large-scale open-source dataset for lyric interpretation. A comparison with similar datasets is shown in Table 1.

Dataset	Music	Interpretation	Public
Choi *et al.* [25]	800	2000	×
Manco *et al.* [26]	17,354	17,354	×
Ours	**27,384**	**490,000**	✓

Table 1. A comparison of our dataset with previous music description datasets.

We observe three main issues with interpretations written by real users: (1) some interpretations are very short or very long; (2) interpretations can contain content unrelated to the lyrics themselves, and (3) some interpretations are of low quality. We therefore preprocess the dataset using two techniques:

1. We **remove overly short interpretations** with length less than 256 characters to improve data representativeness, since we find that sentences below this length are often meaningless interpretations. For interpretations longer than 2048 characters, we keep only the first 2048 characters, but ensure that the last word is complete.
2. We use a **voting-based filtering mechanism** to improve data quality. Every interpretation on SongMeanings.com has a voting result attached, indicating how much the community approves of it, so an interpretation with a higher vote is more likely to be a high-quality interpretation. We therefore create two subsets, keeping only interpretations with positive votes and interpretations with non-negative votes.

[2] The Song Interpretation Dataset is anonymously open for downloading: `https://doi.org/10.5281/zenodo.7019124`.

[3] `https://songmeanings.com/`

To enable the model to be comparable across datasets, we manually select 800 interpretations and use them as a test dataset after excluding them from the original dataset. We have specifically removed all songs that appeared in the test set from the training set to avoid data leakage issues. After the above preprocessing, the dataset has 3 different subsets with the information shown in Table 2.

Dataset Name	Train	Valid.	Test
Raw dataset	440,000	50,000	800
Dataset Full	279,283	31,032	800
Dataset w/vote ≥ 0	265,360	29,484	800
Dataset w/vote > 0	49,736	5,526	800

Table 2. A comparison of dataset sizes (in number of interpretations) with different filtering methods.

4. EXPERIMENTAL SETTINGS

4.1 Implementation Details

We pre-process the lyric text input data by truncating or padding text to 2048 tokens. All the audio signals are downsampled to 16,000 Hz sample rate and converted to short-time Fourier transform representations with a 512-point FFT and 50%-overlapping Hann window. Finally, we convert these to log mel spectrograms with 128 bins.

We use `BART-base` [7] as the pre-trained language model to construct BART-fusion, which has a 6-layer encoder and decoder. For the audio encoder (CNNSA), we use 3×3 kernels for all layers with [128, 128, 256, 256, 256, 256, 256] channels and [(2, 2), (2, 2), (2, 2), (2, 1), (2, 1), (2, 1), (2, 1)] strides. The cross-modal attention module has 1 head and 1 layer, with $d_a = 768$.

We use AdaFactor [28] as the optimizer. We set the learning rate to 6×10^{-4} and reduce it to 6×10^{-5} from the 11th epoch. For all experiments, we use a batch size of 8. We train all models for 20 epochs with early stopping of 3 epochs using the ROUGE-1 score on the validation set.

4.2 Evaluation Metrics

Since there are no existing metrics for the lyric interpretation task, we borrow several complementary metrics from similar tasks to evaluate the performance level of the model in a comprehensive manner. We use ROUGE, METEOR and BERT-Score to evaluate the generated interpretations. ROUGE is considered as the main metric for evaluation, because our task is closest to the text summarisation task.

4.2.1 ROUGE-{1, 2, L}

ROUGE [29] is a common metric for evaluating abstractive text summaries. It calculates the overlap of 1-gram phrases (R-1), 2-gram phrases (R-2) and their weighted results (R-L). In the context of lyrics, a higher ROUGE score indicates that the generated explanatory text leaves out less necessary information, which is better.

Training dataset	Method	Data size	R-1	R-2	R-L	METEOR	BERT-Score
Dataset w/random	BART	56,470	40.0	12.5	21.7	21.1	**83.7**
Dataset w/random	BART-fusion	56,470	**42.1***	**13.6***	**23.4***	**22.0***	83.3
Dataset w/voting > 0	BART	56,470	41.2$_{+}$	13.0$_{+}$	22.8$_{+}$	22.0$_{+}$	**83.6**
Dataset w/voting > 0	BART-fusion	56,470	**44.3**$^{*}_{+}$	**14.6**$^{*}_{+}$	**24.7**$^{*}_{+}$	**22.6**$^{*}_{+}$	83.3
Dataset Full	BART	316,478	44.1	14.0	24.5	22.5$_{+}$	**83.5**
Dataset Full	BART-fusion	316,478	**46.1***	**15.0***	**25.1***	**23.0***	**83.5**
Dataset w/voting ≥ 0	BART	300,712	44.8$_{+}$	14.9$_{+}$	24.7	22.7	83.9
Dataset w/voting ≥ 0	BART-fusion	300,712	**46.7**$^{*}_{+}$	**15.6**$^{*}_{+}$	**25.5**$^{*}_{+}$	**23.4***	**84.1**

Table 3. Evaluation results of BART-fusion and BART (baseline) on the Song Interpretation Dataset with different settings. $*$: BART-fusion outperforms BART with $p < 0.05$; $+$: the filtered dataset outperforms the unfiltered dataset with $p < 0.05$.

4.2.2 METEOR

METEOR [30] takes into account similar semantic information such as synonyms through WordNet and calculates the similarity score based on F-measure. It complements ROUGE by jointly measuring how semantically similar the model-generated lyric interpretation is to the reference text.

4.2.3 BERT-Score

BERT-Score [31] is mainly used to evaluate the naturalness and fluency of the generated text. We expect that incorporating the audio modality should not sacrifice the generative performance of the language model.

We use `rouge` [4], `nltk` [5] and `bert-score` [6] respectively to compute these metrics.

5. RESULTS AND ANALYSIS

5.1 Main Results

We train BART-fusion and the corresponding original BART on several different dataset settings. Results are shown in Table 3, where all values are the means of multiple independent repeated experiments. We perform a *paired Student's t-test* on the results of BART-fusion and BART, and the results on the filtered datasets and unfiltered datasets respectively, where p-value is 0.05.

Our first finding is that applying a voting-based filtering mechanism to the dataset significantly improves the performance of the models (both BART-fusion and baseline) on this task. For fairness, we take a random subset of *Dataset Full*, which we call *Dataset w/random*, to match the size of *Dataset w/voting* > 0. (We still use *Dataset Full* to compare *Dataset w/voting* ≥ 0.) The experimental results show that the performances of both BART-fusion and the baseline are significantly improved on the filtered datasets.

The experimental results also show that BART-fusion significantly outperforms the baseline, and generates more precise interpretion text for lyrics. For all dataset settings, our BART-fusion models show better performance on the ROUGE and METEOR scores.

Finally, we find that BART-fusion preserves the generative performance of the pre-trained language model while improving the generation accuracy. The performance of BART-fusion is essentially equal to that of baseline on the BERT-Score metric, which is the metric of text quality. It means that the introduction of audio modal information affects the model mainly semantically and does not affect the naturalness or fluency.

5.2 Case Study and Error Analysis

We observe that adding music modality information usually brings the benefits of accurate understanding of the theme, selecting highlighted lyric lines, and emotive sentences from the generated samples. In the case study, we select a representative example showing the generation results from different models with the same lyric input, as shown in Table 4. In the first lines, BART-fusion explains the theme of the lyrics more accurately than BART. Then, BART-fusion selects the highlighted lyric lines and gives further detailed interpretation, while the text generated by BART lacks a clear explanation of them. We find that BART-fusion talks about the mood of the song at the end, which is not present in the original BART example. [7]

We have noticed that when the lyrics are about complex topics, such as religious and philosophical topics, BART-fusion and BART sometimes fail to understand the meaning correctly and generate text that is only superficially correct, which we interpret as a lack of common sense often observed in this class of deep learning models.

5.3 Cross-Modal Retrieval Analysis

In addition, we design a cross-modal retrieval experiment to test the retrieval capability of our models. We randomly extract a sentence from the real interpretation in the test set as a query string, use this to search over a database of lyric text and musical audio on a semantic level, and return a ranked list of all possible results. We expect the correct result corresponding to the query to be as close to the top of this ranking as possible.

To accomplish this task, the model generates interpretations for all song lyrics in the database, calculates their semantic features using Sentence-BERT [32], and stores

[4] `https://github.com/pltrdy/rouge`
[5] `https://github.com/nltk/nltk`
[6] `https://github.com/Tiiiger/bert_score`

[7] We provide a demo page for more examples: `https://sites.google.com/view/bart-fusion-ismir2022`.

Lyrics: I cannot find a way to describe it / It's there inside, all I do is hide / I wish that it would just go away / What would you do, you do, if you knew? / What would you do? / / All the pain I thought I knew / All the thoughts lead back to you / Back to what was never said / Back and forth inside my head / I can't handle this confusion / I'm unable, come and take me away / / I feel like I am all alone / All by myself, I need to get around this / My words are cold, I don't want them to hurt you / If I show you, I don't think you'd understand / 'Cause no one understands / / All the pain I thought I knew / All the thoughts lead back to you / Back to what was never said / Back and forth inside my head / I can't handle this confusion / I'm unable, come and take me away / ...
Reference: I think this song is about a crush that the singer is afraid to tell the person they are in love with. She wonder if maybe the other person feels the same, but is also too afriad to tell her. Thinking "what if". Since, she won't admit it she'll never know and this not knowing ends up eating her up inside. Each day she wonders if she could be with this person who she really loves, but she just gets frustrated and eventually wishes that she never even fell in love with the person since it would make her life so much easier. It builds up so much that she just wishes someone would "come and take me away". That could mean two things. It could be death/suicide because she decides it's not worth it or she could just wish for someone else to date so she can just say she's taken and not have to worry about this crush.
BART: I think this song is about a girl who is in love with a guy, but she doesn't know how to tell him because she's afraid he'll hurt her. She's trying to convince herself that he loves her, but he doesn't understand how she feels about him. She wants him to come and take her away from all the pain she's caused and make her feel better about herself. This is what I think of when I listen to this song. I think it's about someone who is trying to tell someone they love them, but they don't want to tell them because they're afraid that they'll hurt them. I love this song, it's one of my favorites on the album.
BART-fusion: I think this song is about a girl who is in love with a guy but she is afraid to tell him because she doesn't want to ruin the friendship. In the first verse she is talking about how she can 't find a way to describe how she feels and how she wishes it would just go away. The second verse is about how the guy is telling her how he feels but she isn't sure if he really feels the same way. The chorus is her saying how she is confused about what to say and how to tell the guy. "I'm unable, come and take me away" means that she is unable to tell her feelings and she is trying to get the guy out of her head. The last verse is her telling the guy that she feels like she is going nowhere in her life, and she just wants him to take her away from all of this. That's just my take on it. I can totally relate to this song. It's a very powerful song.

Table 4. A generation example for Avril Lavigne - *Take Me Away*.

features in advance. At query time, we also use Sentence-BERT to compute the semantic feature for the query string. We return the song whose feature is most similar to the query feature by computing the cosine similarities between the query feature and features in the database.

We use Mean Reciprocal Rank (MRR) [33], a common metric for information retrieval tasks, to measure the performance of the models on this task. Formally, for a set of query strings $Q = \{q_1, q_2, ..., q_m\}$ and the corresponding database $S = \{s_1, s_2, ..., s_n\}$, the model outputs a list of songs sorted by probability for the i-th query, where the correct song is ranked k_i-th, and the model is scored as:

$$\text{score}_i = \frac{1}{k_i}, \text{where } k_i \leq n. \tag{8}$$

The final MRR is calculated by averaging the scores:

$$\text{MRR} = \frac{1}{m}\sum_{i=1}^{m}\text{score}_i = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{k_i}. \tag{9}$$

From Table 5, we find that BART-fusion outperforms the original BART on all 4 dataset settings, i.e., the interpretations generated by BART-fusion can help users find music more accurately. If we return a random ranking result, the MRR value is about 0.9%, which is much lower than the model performance.

Dataset	Method	MRR (%)
Dataset w/random	BART	25.3
Dataset w/random	BART-fusion	**27.5***
Dataset w/voting > 0	BART	26.1_+
Dataset w/voting > 0	BART-fusion	**28.2**$^*_+$
Dataset Full	BART	26.2
Dataset Full	BART-fusion	**30.5***
Dataset w/voting ≥ 0	BART	26.4
Dataset w/voting ≥ 0	BART-fusion	**32.5**$^*_+$

Table 5. Evaluation results for the music retrieval task. $*$: BART-fusion outperforms BART with $p < 0.05$; $+$: the filtered dataset outperforms the unfiltered dataset with $p < 0.05$.

6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel model to generate interpretations from song lyrics and musical audio. Our modelis based on a pre-trained language model and generates better lyric interpretations by fusing text and music semantic representations. We also have proposed a new dataset and explored the process of dataset creation, and have investigated how different treatments of the dataset can affect the performance of the model. We have designed an additional experiment that shows that our model outperforms BART on cross-modal music retrieval tasks. Our work has a range of potential applications, such as helping people better understand English lyrics (especially for non-native English speakers) and natural language based music discovery.

The current model still has shortcomings, such as difficulty in understanding complex topics and lack of general common sense. In future work, we will try to improve the model by adding metadata and knowledge base inputs. We also plan to extend the model to describe the musical content of a song in natural language. In addition, large-scale language models may be biased, reinforce stereotypes and introduce harms, which deserves our attention in the future.

7. ACKNOWLEDGEMENT

We want to thank Mark Levy for his great contribution to this work. We also thank Yin-Jyun Luo and Liming Kuang for their enthusiastic help during the writing process of the paper. Yixiao Zhang is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by the China Scholarship Council, Queen Mary University of London and Apple Inc.

8. REFERENCES

[1] K. Watanabe and M. Goto, "Lyrics information processing: Analysis, generation, and applications," in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020, pp. 6–12.

[2] M. Fell, Y. Nechaev, E. Cabrio, and F. Gandon, "Lyrics segmentation: Textual macrostructure detection using convolutions," in *COLING*, 2018, pp. 2044–2054.

[3] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, "Music mood detection based on audio and lyrics with deep neural net," *arXiv preprint arXiv:1809.07276*, 2018.

[4] Z. Sheng, K. Song, X. Tan, Y. Ren, W. Ye, S. Zhang, and T. Qin, "Songmass: Automatic song writing with pre-training and alignment constraint," *arXiv preprint arXiv:2012.05168*, 2020.

[5] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," *arXiv preprint arXiv:1707.04678*, 2017.

[6] K. Al-Sabahi, Z. Zuping, and M. Nadher, "A hierarchical structured self-attentive model for extractive document summarization (HSSAS)," *IEEE Access*, vol. 6, pp. 24 205–24 212, 2018.

[7] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[8] J. Son and Y. Shin, "Music lyrics summarization method using textrank algorithm," *Journal of Korea Multimedia Society*, vol. 21, no. 1, pp. 45–50, 2018.

[9] M. Fell, E. Cabrio, F. Gandon, and A. Giboin, "Song lyrics summarization inspired by audio thumbnailing," in *RANLP*, 2019.

[10] W. Li, C. Gao, G. Niu, X. Xiao, H. Liu, J. Liu, H. Wu, and H. Wang, "UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning," *arXiv preprint arXiv:2012.15409*, 2020.

[11] Y. Zhang, Z. Wang, D. Wang, and G. Xia, "BUTTER: A representation learning framework for bi-directional music-sentence retrieval and generation," in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020, pp. 54–58.

[12] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and VQA," in *AAAI*, vol. 34, no. 07, 2020, pp. 13 041–13 049.

[13] J. Chen, H. Guo, K. Yi, B. Li, and M. Elhoseiny, "VisualGPT: Data-efficient adaptation of pretrained language models for image captioning," *arXiv preprint arXiv:2102.10407*, 2021.

[14] T. Yu, W. Dai, Z. Liu, and P. Fung, "Vision guided generative pre-trained language models for multimodal abstractive summarization," *arXiv preprint arXiv:2109.02401*, 2021.

[15] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, "Plug and play language models: A simple approach to controlled text generation," *arXiv preprint arXiv:1912.02164*, 2019.

[16] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," *arXiv preprint arXiv:1906.04972*, 2019.

[17] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *ACL*, vol. 2019. NIH Public Access, 2019, p. 6558.

[18] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MASS: Masked sequence to sequence pre-training for language generation," *arXiv preprint arXiv:1905.02450*, 2019.

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[23] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *ACL*, 2019.

[24] C. Li, X. Gao, Y. Li, B. Peng, X. Li, Y. Zhang, and J. Gao, "OPTIMUS: Organizing sentences via pre-trained modeling of a latent space," *arXiv preprint arXiv:2004.04092*, 2020.

[25] K. Choi, J. H. Lee, X. Hu, and J. S. Downie, "Music subject classification based on lyrics and user interpretations," *AIST*, vol. 53, no. 1, pp. 1–10, 2016.

[26] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Muscaps: Generating captions for music audio," *CoRR*, vol. abs/2104.11984, 2021. [Online]. Available: https://arxiv.org/abs/2104.11984

[27] I. A. P. Santana, F. Pinhelli, J. Donini, L. Catharin, R. B. Mangolin, V. D. Feltrim, M. A. Domingues *et al.*, "Music4all: A new music database and its applications," in *IWSSIP*. IEEE, 2020, pp. 399–404.

[28] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *ICML*. PMLR, 2018, pp. 4596–4604.

[29] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004, pp. 74–81.

[30] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005, pp. 65–72.

[31] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," *arXiv preprint arXiv:1904.09675*, 2019.

[32] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[33] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan, "Expected reciprocal rank for graded relevance," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, pp. 621–630.

TOWARD POSTPROCESSING-FREE NEURAL NETWORKS FOR JOINT BEAT AND DOWNBEAT ESTIMATION

Tsung-Ping Chen and Li Su
Institute of Information Science, Academia Sinica, Taiwan
{tearfulcanon, lisu}@iis.sinica.edu.tw

ABSTRACT

Recent deep learning-based models for estimating beats and downbeats are mainly composed of three successive stages—feature extraction, sequence modeling, and post processing. While such a framework is prevalent in the scenario of sequence labeling tasks and yields promising results in beat and downbeat estimations, it also indicates a shortage of the employed neural networks, given that the post-processing usually provides a notable performance gain over the previous stage. Moreover, the assumption often made for the post-processing is not suitable for many musical pieces. In this work, we attempt to improve the performance of joint beat and downbeat estimation without incorporating the post-processing stage. By inspecting a state-of-the-art approach, we propose reformulations regarding the network architecture and the loss function. We evaluate our model on various music data and show that the proposed methods are capable of improving the baseline approach without the aid of a post-processing stage.

1. INTRODUCTION

Beat and downbeat trackings have been of long-standing interests in the communities of signal processing and music information retrieval (MIR) [1–5]. The two tasks inherently possess the class imbalance issue [6], i.e., the highly imbalanced numbers of beat (downbeat) and non-beat (non-downbeat) frames in music data, and hence remain challenging in many cases even nowadays. To tackle the beat and downbeat tracking problems, early signal processing approaches have developed a three-stage framework which consists of *feature extraction*, *sequence modeling*, and *post-processing* [7–9]. Over the past few years, great progresses on the two tasks have been made through the combination of the three-stage pipeline and deep learning models such as recurrent neural networks (RNNs) [10–12], convolutional-recurrent neural networks (CRNNs) [13], and Transformer-based networks [14]. Among the thriving research, a convolutional approach achieves the state-of-the-art performance on joint estimation of tempo, beat, and downbeat [15], in which convolutional neural networks (CNNs) and temporal convolutional networks (TCNs) [16] are placed in charge of the feature extraction and of the sequence modeling, respectively, while dynamic Bayesian networks (DBNs) [17] are used for the post-processing.

In spite of the promising results obtained by this approach, the employed networks raise three concerns. First, the CNNs for feature extraction convolve an input spectrogram with small kernels at the very beginning, and a max-pooling layer is followed immediately. Therefore, the spectro-temporal patterns might not be well-captured [18]. Second, the dilated convolutions [19] of the TCNs involve a small kernel size and an exponentially increasing dilation rate. Despite the large receptive fields obtained at higher layers, this design leads to extremely sparse samplings which join distantly-separated frames, and consequently the contextual information could be irrelevant [20]. Lastly, the assumptions made for the DBNs that the meter is unchanged or the rhythmic patterns are known [11, 21] are not always applicable to various music genres, even to popular music. For example, the Hainsworth dataset [22], which is often employed for evaluation, includes several clips of pop or rock songs with changing meter. Moreover, the DBNs involve intricate design and assumptions for representing the state and the transition throughout a piece of music, and hence introduce non-trivial works in addition to the training of the preceding neural networks.

In this paper, we cope with the joint beat and downbeat estimation task by addressing the aforementioned issues. To get rid of the post-processing DBNs while maintaining the performance, we consider to improve the network architecture and the loss function. Precisely, we propose to use scaled depthwise separable convolutions [23–25] to aggregate rich contextual information at multiple time scales. To address the class imbalance issue, the focal loss [26] and the Dice loss [27] are employed in place of the commonly used cross entropy loss. Besides, we make our model aware of the periodic structure of beat or downbeat sequences by including a label embedding network [28] during training. We evaluate the proposed architecture on various music data, including both audio and MIDI files, and demonstrate a state-of-the-art performance on the Ballroom dataset. The main contribution of this work is that we thoroughly address the architectural issues for beat and downbeat estimations. Most of the problems are shared by sequence models in general, and therefore the proposed methods could be applied to many other MIR-related tasks.

2. OVERVIEW OF THE STATE-OF-THE-ART

The model of [15] without the DBNs is taken as our baseline upon which we build our new model to address the architectural issues mentioned beforehand. In the following, we briefly introduce the baseline model using the three-stage framework, and present the new model thereafter. We skip the post-processing stage for we aim to expel the DBNs from the system in this work. An overview of the baseline model is illustrated at the top of Figure 1.

2.1 Feature extraction

Given an audio signal represented as a log-magnitude spectrogram of size $T \times J$, where T denotes the number of time steps and J the number of frequency bins, the CNNs of the baseline model extract high-level features using three groups of 2-D convolution and max-pooling layers. The three convolutional layers (with a kernel size of 3×3, 1×12, and 3×3, respectively) [1] capture harmonic content at different frequency scales, while the max-pooling operations (with a kernel size of 1×3) reduce the frequency dimension in three steps. In other words, the CNNs gradually transcribe the harmonic information into high-level features, and output a sequence of size $T \times D$ with D indicating the dimension of high-level features.

2.2 Sequence modeling

The high-level feature sequence from the previous stage is then passed to the TCNs for learning temporal structure. As depicted at the top of Figure 1, each TCN layer performs two 1-D dilated convolutions in parallel with a kernel of size 5 and an exponentially increasing dilation rate 2^l (resp. 2^{l+1}), where l is the layer number. The outputs of the two dilated convolutions are concatenated and reprojected to keep the feature dimensionality constant. As a result, the TCNs efficiently enlarge the receptive field within a few layers with the amount of learnable parameters increasing linearly to the number of layers. Given that the TCNs has 11 layers, the two convolutions at the last layer will expand the kernels across over 4000 (resp. 8000) time frames with each element of the two kernels being spaced 1024 (resp. 2048) frames apart. Therefore, the temporal context modelled by the TCN is more than 80 seconds (assume a frame rate of 100 fps). Afterwards, the output of the TCNs is processed by the DBNs to obtain a sequence of beat or downbeat estimates.

2.3 Potential issues

In the community of computer vision, it has been shown that stacking small convolutional kernels followed by pooling layers is effectual for extracting high-level features of an image [29–31]. Such a architecture is often adopted by audio-based, or more specifically spectrogram-based research [32–34]. Considering the nature of images, it is reasonable to aggregate information hierarchically from spatial associations. However, it is questionable that a spectrogram could be well-represented with the same architecture. The characteristic spectral information of a musical piece does not necessarily reside in the adjacent frequency bins within a small kernel. Moreover, it is also suspicious that the *compressed* frequency dimension after pooling operations is as meaningful as a downsampled image.

On the other hand, it has been observed that dilated convolutions result in the so-called *gridding artifacts* [35, 36], due to that adjacent elements in the output are calculated from completely different sets of elements in the input. Additionally, small convolutional kernels with high dilation rates relate input elements which are highly sparsely distributed and hence might lack of correlations. In the scenario of beat or downbeat estimation, stacked 1-D dilated convolutions could be preferable for they are able to encode periodic structure of the beat (downbeat) through equidistant elements of dilated kernels. However, it is not guaranteed that an inquired musical piece has a steady tempo as well as a constant rhythm. Hence, a network equipped with dilated convolutions for modeling temporal information may have limitations on expressive music.

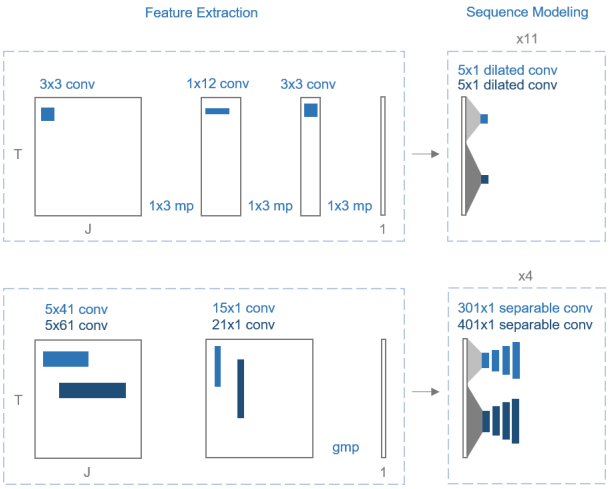

Figure 1: Architecture reformulation with respect to convolution (conv) and max pooling (mp) layers . Top: model of [15] without DBNs. Bottom: proposed architecture. The feature dimension is not shown in the figure.

3. APPROACH

3.1 Task formulation

We treat the joint beat and downbeat estimation task as a sequence labeling problem [37, 38]. Given an input spectrogram $\mathbf{X} \in \mathbb{R}^{T \times J}$, the task involves the assignment of a categorical label $\in \{\texttt{downbeat}, \texttt{xbeat}, \texttt{neither}\}$ [2] to each time step $t \in \mathrm{T}$. We employ a model architecture which mainly consists of a feature extraction network (f_{ex}), a sequence modeling network (f_{seq}), and an estima-

[1] A kernel size $M \times N$ specifies a convolution window across M time frames and N frequency bins.

[2] The label `xbeat` refers to a beat which is not a downbeat.