Learning Music Sequence Representation from Text Supervision

Tianyu Chen[1 3], Yuan Xie[4], Shuai Zhang[1 3], Shaohan Huang[2], Haoyi Zhou[1 3], Jianxin Li[1 3]

[1]BDBC, Beihang University, China [2]Microsoft Research Asia, China [3]SKLSDE, Beihang University

[4]The Institute of Acoustics of the Chinese Academy of Sciences, China

Introduction

- Music data is relatively quantity-small but with sufficient supervision information in their text-form metadata (e.g., lyrics, album, descriptions, lyricist, composer, singer, comments), which are still under-explored.
- We propose a novel text supervision method to learn directly from text-form metadata, called MUSER.
- We design an additional spectrogram encoder that greatly improves data efficiency of the CLIP-style framework.
- We propose a novel tri-modal contrastive pre-training framework and achieve state-of-the-art on music-related benchmarks.

Methodology

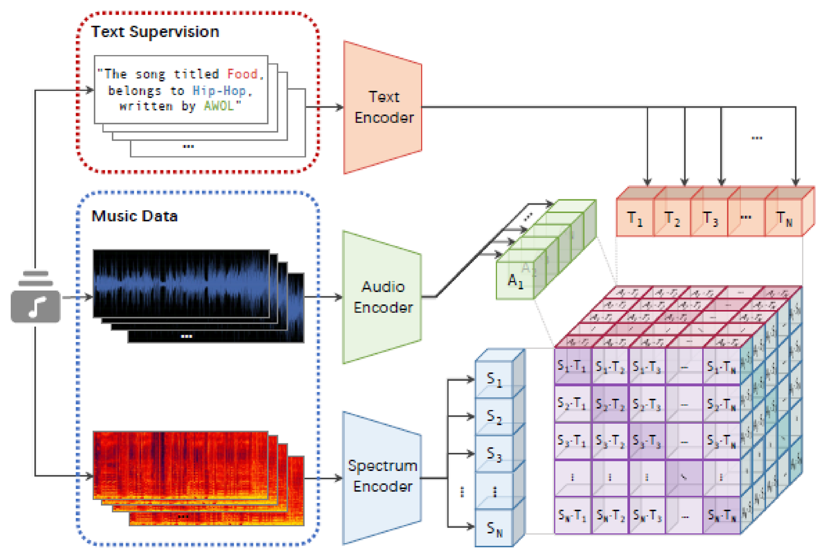

Figure 1 Overview of our training framework. First, we convert different forms of metadata into a unified plain text format. Then, the plain text sequence and the audio sequence of music will be encoded respectively into a shared embedding space. Also, the music spectrum is encoded to learn text-form concepts from multi-view. By using the shared embedding space, we apply CLIP-style contrastive learning, which aims to distinguish music sequence by their corresponding text.

- **A text template example**
  A song of {hip-hop}, belongs to {fast-rhythm music}, whose style is {inspiring}.

- **Tri-modal Embeddings calculation**
  The tri-modal inputs will be encoded respectively into a shared embedding space.

- **Contrastive learning**
  Cosine similarity is used to measure the distance between embeddings. we optimize embeddings together with an asymmetric contrastive loss.

$$L(i,j;\theta_Q,\theta_K) = -\log \frac{\exp(D(E_{Q_i},E_{K_i})/\tau)}{\sum_{j=1,j\neq i}^{N}\exp(D(E_{Q_i},E_{K_j})/\tau)},$$

$$\mathcal{L}(i,j;\theta_A,\theta_T,\theta_S) = \mathcal{L}(i,j;\theta_A,\theta_T) + \mathcal{L}(i,j;\theta_S,\theta_T).$$

Algorithm 1: Contrastive Learning of MUSER.

Data: all labeled training music sequence $\mathcal{A}$, text $\mathcal{T}$, spectrum $\mathcal{S}$ pairs, and label $\mathcal{Y}$.
Input: encoders F_{aud}, F_{txt}, F_{spec}; weights W_a, W_t, W_s; temperature parameter τ; batch size n.

while *not done* **do**
  Sample batches $(A_i, T_i, S_i, Y_i) \sim (\mathcal{A}, \mathcal{T}, \mathcal{S}, \mathcal{Y})$.
  for all (A_i, T_i, S_i, Y_i) **do**
    $E_{A_i}, E_{T_i}, E_{S_i} \leftarrow F_{\text{aud}}(A_i), F_{\text{txt}}(T_i), F_{\text{spec}}(S_i)$.
    $E_{A_i}, E_{T_i}, E_{S_i} \leftarrow W_a E_{A_i}, W_t E_{T_i}, W_s E_{S_i}$.
    Compute logits $\widehat{Y}_{AT}, \widehat{Y}_{TA}, \widehat{Y}_{ST}, \widehat{Y}_{TS}$ as: e.g., $\widehat{Y}_{AT_i} = E_{A_i} E_{T_i} e^{\tau}$. Compute losses $\ell_{AT_i}, \ell_{TA_i}, \ell_{ST_i}, \ell_{TS_i}$ as: e.g., $\ell_{AT_i} = \text{CrossEntropy}(\widehat{Y}_{AT_i}, Y_i, axis = 0)$.
    Compute overall loss $\ell_i = (\ell_{AT_i} + \ell_{TA_i} + \ell_{ST_i} + \ell_{TS_i})/4$.
  end
  Update encoders and weights with loss $\mathcal{L} = \sum_i \ell_i$.
end

Experiment & Results

Datasets

- **Free Music Archive (FMA)** We use the small subset of FMA, a balanced subset containing 8,000 clips. We rely on templates to concatenate the genre, parent genre, and top-level tag of each audio together.
- **GTZAN** We choose GTZAN as the dataset for genre classification task. which contains 1,000 tracks of 30-second length.
- **MagnaTagATune (MTT)** We choose MTT as the benchmark dataset for automatic tagging task. We limit the vocabulary to the top 50 most popular tags.

Baselines

- **VGGish** This baseline is pre-trained on a large-scale video dataset (AudioSet) with a classification task.
- **CLMR** This baseline first introduced the contrastive pre-training techniques, which enable unsupervised music sequence representation learning.
- **CLAM** This baseline first proposed for unconditional speech. It codifies a high-rate continuous audio sequence into low-rate discrete codes. Then a language model is trained on resulting codified audio and optional meta-data to produce high-quality contextual representations.
- **Multi-task** We divide the pre-training into several sub-tasks of a shared encoder according to the annotations.

MUSER with far less pre-training data

Method	Source	Audio / Text
VGGish[21]	YouTube-8M	350000h / 8M
CLMR[22]	Not mentioned	2200h / 260k
CALM[23]	Jukebox	240000h / 1.2M
MUSER	FMA (small subset)	66.7h / 8k
	MTT (train)	127h / 15k
	GTZAN (train)	3.7h / 0.4k
	Total	**195.8h / 23.5k**

Table 1 Datasets for Music Sequence Pre-training. The quantity of pre-training data used is less than 0.1% of other pre-trained models.

MUSER with promising performance

Method	Tags (AUC)	Tags (AP)	Genere (ACC)
VGGish	89.4	42.2	75.2
CLMR	89.4	36.1	68.6
CALM	91.5	41.4	79.7
AE only (MT., PT)	88.7	38.4	59.7
AE only (MT., PT+FT)	88.9	38.9	76.9
State-of-the-art	**91.5**	42.2	82.1
MUSER (AE only)	87.5	36.3	66.6
MUSER (w/o spec)	88.1	39.6	75.2
MUSER (PT)	88.7	41.6	72.6
MUSER (PT+FT)	89.5	**43.0**	**82.5**

Table 2 Performance on music understanding benchmarks. Our MUSER (PT+FT) outperforms SOTA on both tasks. Music spectrum encoder brings a obvious improvement on both tasks.

MUSER with better few-shot ability

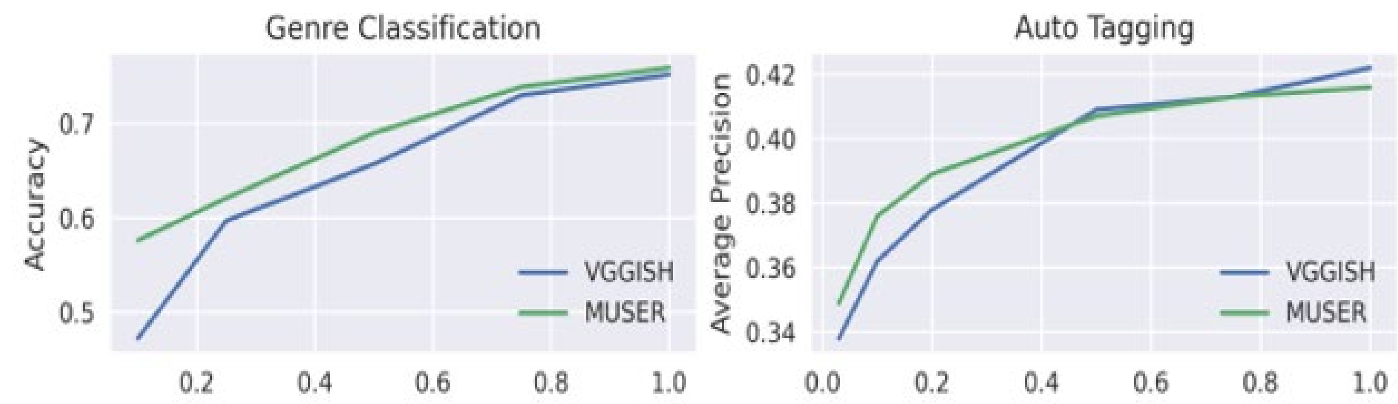

Figure 2 Comparisons of fine-tuning efficiency on % ratio of training samples for different downstream tasks, with only FMA-small dataset for pre-training. Our MUSER uses less pre-training data to have better few-shot performance

Conclusion

- MUSER only requires 0.056% of pre-training data to achieve the state-of-the-art performance, and it has excellent few-shot performance.
- We are the first to introduce text supervision for exploring the fine-grained feature of distributed songs by designing text templates.
- We add a music spectrogram encoder to the CLIP-style framework. It enables the MUSER encoders to learn music from different views.

Microsoft Research 微软亚洲研究院