TRANSCRIPTION-BASED LYRICS EMBEDDINGS: SIMPLE EXTRACTION OF EFFECTIVE LYRICS EMBEDDINGS FROM AUDIO

Jaehun Kim Florian Henkel Camilo Landau Samuel E. Sandberg Andreas F. Ehmann
SiriusXM+Pandora, USA
firstname.lastname@siriusxm.com

ABSTRACT

The majority of Western popular music contains lyrics. Previous studies have shown that lyrics are a rich source of information and are complementary to other information sources, such as audio. One factor that hinders the research and application of lyrics on a large scale is their availability. To mitigate this, we propose the use of *transcription-based lyrics embeddings* (TLE). These estimate 'ground-truth' lyrics embeddings given only audio as input. Central to this approach is the use of transcripts derived from an automatic lyrics transcription (ALT) system instead of human-transcribed, 'ground-truth' lyrics, making them substantially more accessible. We conduct an experiment to assess the effectiveness of TLEs across various music information retrieval (MIR) tasks. Our results indicate that TLEs can improve the performance of audio embeddings alone, especially when combined, closing the gap with cases where ground-truth lyrics information is available.

1. INTRODUCTION

Lyrics play an important role in music consumption [1–3], often providing additional context to the perceived audio, such as lyrical themes and semantic meaning. As such, lyrics also have a wide range of applications in MIR, including mood/sentiment prediction [4–8], recommendation [2, 9], genre [2, 10–12] and music tag prediction [11].

However, the absence of lyrics on a large scale poses a significant challenge. While they are often available for popular music, this might not be the case for the majority of songs in a music catalog, either because they are non-existent, i.e., not yet transcribed by a human, or due to missing copyrights. Automatic lyrics transcription (ALT) systems are an important step towards alleviating this problem by directly transcribing the lyrical content from a piece of audio [13–17]. Still, these systems are not infallible and some efforts have been made to further refine the resulting (potentially faulty) transcriptions, e.g., by using large language models (LLMs) [15].

In this work, we investigate the use of lyrics embeddings on a variety of MIR downstream tasks, ranging from music tagging to recommendation. We focus on a comparison between embeddings stemming from human-transcribed or 'ground-truth' lyrics and their machine-transcribed counterparts, which we refer to as transcription-based lyrics embeddings (TLE) throughout this work. In particular, we are interested in the effectiveness of two TLE variants compared to audio embeddings and 'ground-truth' lyrics embeddings, where we assume the performance of the latter as an upper bound to TLE. To that end, we answer the following research questions:

- **RQ1** Do TLE provide useful additional information compared to audio embeddings alone?
- **RQ2** Can TLE be efficiently refined to close the gap to 'ground-truth' lyrics embeddings?

The remainder of the paper is structured as follows. Section 2 discusses related work on lyrics embeddings and automatic lyrics transcription. In Section 3 we introduce the concept and types of TLEs we evaluate in this work. Section 4 covers our experimental setup including choices of audio/lyrics embeddings as well as datasets and tasks. In Section 5 we investigate and discuss the aforementioned research questions. Finally, we conclude this work in Section 6 and highlight potential future work directions.

2. RELATED WORK

Extracting information from lyrics has long been studied in the MIR community. In particular, representing such information quantitatively, e.g., with feature or latent vectors, has been a strong focus. For instance, linguistic features (e.g., rhyme and stylistic features) are shown to be useful in various tasks [6, 8, 18], as well as approaches using psychologically validated dictionaries [2, 19].

For representation modelling, bag-of-words (BoW) [20] and term frequency inverse document frequency (TF-IDF) have been common and effective choices for lyrics [6, 12], which is further extended to latent document or topic modeling that has been successful in lyrics similarity estimation and exploration [21, 22], as well as genre and mood classification [6, 11, 12]. Another successful method is to employ word2vec [4, 18, 23], where lyrics documents are typically represented as the average of word vectors.

Lately, deep learning (DL) has been a popular choice for lyrics representation learning. In supervised learning,

it typically is accomplished implicitly within hidden layers via end-to-end learning, which proves to be effective on a range of downstream tasks [10, 11, 18]. More recently, LLMs have introduced self-supervised learning based latent text representations, which are shown to be effective on several MIR tasks [24, 25].

Regardless, ALT remains a challenging problem today [15, 16, 26, 27]. Along with efforts in building lyrics-specific transcription systems [16, 17], Automatic Speech Recognition (ASR) applied to the ALT task has also been shown to be effective [27–29].

3. TRANSCRIPTION-BASED LYRICS EMBEDDINGS

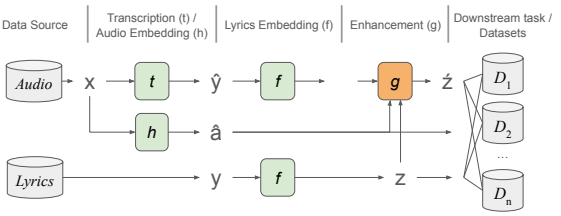

Figure 1. The diagram of proposed lyrics embedding estimation. The models in the green-colored boxes ($\{t, h, f\}$) are assumed to be pre-trained, whereas lyrics enhancement model g is trained employing embeddings obtained from those pre-trained models.

In this work, we propose a system that estimates lyrics embeddings (LE) independent of 'ground-truth' lyrics data $y \in \mathcal{Y}$ by only relying on audio data $x \in \mathcal{X}$, which we generally refer to as transcription-based lyrics embeddings (TLE) in the following. To achieve this, we consider several off-the-shelf pre-trained models, including an ALT model $t : \mathcal{X} \rightarrow \hat{\mathcal{Y}}$, an audio embedding model $h : \mathcal{X} \rightarrow \mathcal{A}$, and finally a lyrics embedding model $f : \mathcal{Y} \rightarrow \mathcal{Z}$.

Given the availability of a pre-trained ALT as well as word or sentence embedding models, it is straight-forward to devise a sequential system that allows one to directly input audio data and obtain high-quality lyrics embeddings that are ready to be used for a variety of downstream music tasks. We propose such an ALT based embedding as the first type of TLE, which is further referred to as $\mathbf{TLE}_T$ and denoted as $\hat{z}$ in Figure 1.

Despite remarkable recent improvements, ALT models are not yet completely error-free, due to the challenging nature of this task [15]. As a result the transcription $\hat{y} \in \hat{\mathcal{Y}}$, and hence an embedding computed from it may contain a certain degree of error when compared to 'ground-truth' lyrics embeddings. We aim to improve the fidelity of TLE_T by introducing an 'enhancement' model which regresses to the ground-truth lyrics embeddings from noisy transcription-based embeddings by using audio embeddings as an additional input. In the following we refer to this approach as $\mathbf{TLE}_R$ (denoted as $\acute{z}$ in Figure 1).

Given pre-configured audio $a \in \mathcal{A} \subset \mathbb{R}^{d_a}$ and lyrics embedding $z \in \mathcal{Z} \subset \mathbb{R}^{d_z}$ spaces, the main goal of 'enhancement' is to find a function $g : \Phi \rightarrow \mathcal{Z}$ which maps the concatenated audio-lyrics embedding $\phi = [a; \hat{z}] \in \Phi \subset \mathbb{R}^{(d_a+d_z)}$ to the lyrics embedding space $z' \in \mathbb{R}^{d_z}$. Specifically, we minimize the sum of squared error between the estimated and ground-truth lyrics embedding as the main learning objective:

$$\min_{\Theta} \sum_{(z,x)\sim\mathcal{D}_{\text{train}}} ||z - g(\phi; \Theta)||^2 + \alpha\mathcal{R}(\Theta) \qquad (1)$$

where Θ are the parameters of the regressor g and $\mathcal{D}_{\text{train}}$ denotes the training dataset where we have access to both the audio x and lyrics y as well as their corresponding embeddings a and z. Finally, $\mathcal{R}$ is the regularizer for the parameters Θ which is controlled by coefficient α.

4. EXPERIMENTAL SETUP

The main hypotheses correspond to each RQ: 1) TLE_T effectively provides lyrics information that is complementary to audio 2) TLE_R improves the effect of TLE_T. Concretely, we design an experiment comparing the performance of three treatments, **LE**, $\mathbf{TLE}_T$, and $\mathbf{TLE}_R$, on relevant downstream tasks, with respect to a range of lyrics and audio embeddings. In the experiment, we define a treatment as a scenario where a single type of (transcription-based) lyrics embeddings is employed to represent text information, both in the training and testing phase of the machine learning (ML) experiment. [1] The rest of this section describes each component of the experimental design.

4.1 Machine Transcription

Similar to [15], we rely on a Whisper-based model [28] to transcribe the lyrics of a song from its audio recording. In contrast to [15], we do not perform a correction step in the form of ChatGPT [2], as this would be too costly on a large scale. Instead we directly create embeddings from the potentially faulty transcriptions ($\mathbf{TLE}_T$) and subsequently try to improve the embeddings using a learned correction function ($\mathbf{TLE}_R$).

Considering that we aim to transcribe a large set of audio recordings (see Table 1), we employ Distil-Whisper for an efficient transcription process without significant performance losses [32, 33]. As suggested in [15] we use "lyrics:" as a prefix prompt.

4.2 Embeddings

In the following, we introduce the different embeddings for each modality used in our experiments. While each embedding is tested separately, we also test *combined cases*, where both audio and lyrics embeddings are provided in the downstream task as a concatenated embedding vector.

[1] We do not consider the scenarios where different treatments are used in training and testing phase to control for a possible data drift [30, 31] and to simplify the experimental design.

[2] https://openai.com/blog/chatgpt

4.2.1 Lyrics Embeddings

We consider three text embeddings, ranging from conventional to more modern transformer-based embeddings to ensure the generality of the study.

Bag-of-Words embeddings (BE): BE embeddings serve as the baseline lyrics embedding approach within the experimental design. Unless the lyrics data is pre-tokenized by words such as the MSD-MusiXmatch (MSD-MXM) dataset [34], we employ the Byte-Pair Encoding (BPE) tokenization [35] instead of actual words. The resulting representation is a sparse song-token count matrix on which we apply TF-IDF [36] and randomized singular value decomposition (rSVD) [37] with a dimensionality of $d = 300$ to subsequently obtain a dense, low-rank vector representation of each lyrics.

Wasserstein embeddings (WE): WE embeddings are learned by applying linear optimal transport which minimizes the Wasserstein distance between distributions of the learned embeddings and given reference vectors [38]. For the reference vectors, we train token embeddings using their co-occurrence matrix. This provides token-to-token transition frequency information on top of the document-token frequency which is the only information source to BE. We choose an embedding dimensionality of $d = 300$.

Sentence BERT embeddings (sBERT): We use a pre-trained sentence BERT model [39], which is fine-tuned using a general language model called MPNet [40, 41]. In particular, the fine-tuning training involved a large scale text corpora to effectively estimate the semantic similarity between paraphrased sentences. Such a property can be crucial for lyrics data, which often is highly abstract and irregular compared to conversational language. The embedding has a dimensionality of $d = 768$.

4.2.2 Audio Embeddings

We employ two open-source and one proprietary music audio embedding models.

OpenL3: is a video-audio multimodal representation trained using self-supervised learning. Specifically, the model encodes video and audio features in respective embeddings, and minimizes the matching error between them, assuming the best matches happen when they are extracted from the same video clip [42,43]. We employ the audio encoding sub-network from the 'music' variant of OpenL3 as the embedding encoder with a dimensionality of $d = 6144$ and 128-band mel spectrograms as input.

MULE: is an open-source music audio embedding model trained in a self-supervised way by using contrastive learning on MusicSet, a large-scale proprietary music audio dataset [44]. We choose this for representing a modern, generic music audio embedding which is effective on wide range of downstream tasks.

MSLE: is the supervised counterpart to MULE where the music labels of MusicSet are used for its supervised learning [44]. We employ it for the proprietary datasets (i.e., InternalLT, InternalRec, see following section). Both embeddings have the same dimensionality $d = 1728$.

4.3 Tasks and Datasets

4.3.1 Automatic Music Tagging (AMT)

AMT has been a popular downstream task in MIR [45]. While there are several datasets [34, 46–48], few of them focus on lyrics specifically. To measure the effect of lyrics more clearly, we devise a subset of the Million Song Dataset (MSD) for tagging [34] that is more relevant for lyrics data, which we refer to as **MSDSnippetLT**.

It is composed as the subset of social tags that MSD provides and that are specifically relevant to the lyrics' subject matter and language. It involves a machine-assisted tag selection process, where we first identify lyric-relevant tags by ranking MSD tags using the correlation with approximately a dozen privately-curated lyric-related tags and language metadata, with songs matched to an annotated proprietary music catalog. Among the top 200 MSD tags per each proprietary lyrics tag, three researchers voted [3] for a final subset based on the following selection rules: 1) the MSD tag has to be clearly related to the targeted proprietary lyrics tag, 2) the MSD tag is not a music genre, 3) the MSD tag is not an artist. After filtering songs that map to the MSD-MXM subset which provides lyrics data for a subset of MSD songs, the resulting dataset contains $74,545$ MSD songs and 87 unique tags in total, where approximately half of them center around the lyrical subject (i.e., "melancholy", "political"), while the other half are related to the lyrics' language (i.e., "british", "Espanol"). We also experiment with a proprietary subset that we refer to as **MSDFullLT** where we have access to the full lyrics. The dataset consists of $35,264$ songs and is a complete subset of MSDSnippetLT. We hypothesize that the dataset can provide useful insights on the effect of incompleteness of the snippet/preview lyrics.

Additionally, we experiment with two popular tagging datasets and one proprietary lyrics subject tagging dataset: **MSDSnippetMT** is a subset of the popular MSD tagging dataset, [34] where we select the commonly used 50 tags [45] to compare to the MSDSnippetLT dataset. As we have to consider the availability of lyrics within MSD, the resulting subset includes a total of $68,363$ songs. We further test with **JamendoMood** dataset which is a subset of the MTG-Jamendo datset [47] specifically focused on music mood. The main purpose of this dataset is to show how effective TLEs are for general music mood tagging when 'ground-truth' lyrics are not available. It contains $17,982$ songs annotated with 56 music mood tags. Finally, **InternalLT** is the subset of a proprietary lyrics-subject dataset providing a set of high-quality lyrics-subject tags as well as full 'ground-truth' lyrics.

We apply 5-fold cross validation for all datasets except JamendoMood, where we use the provided pre-defined split. The model performance is evaluated by the sample-weighted mean average precision (wmAP) averaged across all tags. The main motivation of applying sample weights

[3] A weighted majority voting is conducted where one of three researchers has three times larger weight than the other two, considering the substantial musical experience and training. For further details on the dataset creation, we kindly refer readers to the supplementary material.

Dataset	task	#songs	#tags	text	audio
MSDSnippetMT	music tagging	68,363	50	BoW5k	preview
MSDSnippetLT	lyrics tagging	75,545	87	BoW5k	preview
MSDFullLT	lyrics tagging	35,264	87	full-text	preview
JamendoMood	mood tagging	17,982	56	N/A	full-audio
InternalLT	lyrics tagging	51,240	15	full-text	full-audio
MSDSnippetRec	RecSys	112,769	N/A	BoW5k	preview
InternalRec	RecSys	138,984	N/A	full-text	full-audio

Table 1. Details on the datasets.

is that the majority of the datasets, except JamendoMood, provide the tagging confidence values, which is useful to both training and evaluating the task. The sample weight $w_{i,j} \in [0, 1]$ is defined as the normalized confidence value for the observed annotation of tag j on song i. For all other pairs of i and j (unannotated tags on songs) we set it to 1.

4.3.2 Music Recommendation

We further explore the effectiveness of lyrics embeddings within a music recommendation system (RecSys) problem. To maximize the effect of music content in a RecSys task, we experiment with the 'item cold-start' scenario; a subset of songs lack user interactions (e.g., new releases) hence a content-based recommendation is more effective than collaborative filtering [49, 50].

The dataset consists of triplets of {user, song, listening count} which is translated into a user-song matrix. The interaction data is split into five sets of *train*, *validation* and *test* set by songs in approximately 3:1:1 ratio via 5-fold cross-validation. The user-song interactions within the training (song) set is assumed as 'observed' and thus can be used as the training data, while those within the test (song) set are treated as 'future' interactions which the recommendation system is expected to rank higher. An effective measure commonly used is the binary normalized discounted cumulative gain (nDCG) [51] applied on a truncated list of the top 500 recommended songs. [4]

We employ two datasets for the RecSys task: MSD-Echonest subset [5] is a popular recommendation dataset which contains {user, song, listening count} triplets. We derived a subset by including songs overlapping with the MSD-MXM subset only, which we refer to as **MSDSnippetRec**. We apply 5-core filtering, i.e., we filter out users who interacted with less than or equal to five unique songs, and vice versa. Similarly, we derived a subset of proprietary streaming listening data in the aforementioned format and apply the same pre-processing steps. We refer to this dataset as **InternalRec**.

4.3.3 Pre-processing on Text Representation

The subset of datasets involving MSD-MXM data only provide a pre-tokenized BoW representation, while for the rest we have access to a natural text representation of lyrics, except for JamendoMood where we do not have access to any 'ground-truth' lyrics. We refer to the pre-tokenized BoW representation as *BoW5K* as it specifically is limited to the 5000 most frequent words. This applies to MSDSnippetLT, MSDSnippetMT, MSDSnippetRec.

Furthermore, as the transcription of music previews tend to be substantially shorter, the BoW5K representation of those has less counts compared to the one provided by MSD-MXM, which is extracted from the full-text lyrics. To correct this bias, we apply the following adjustment to the transcription based word count a:

$$\tilde{a}_{i,b} = \tilde{N}_i \big(\gamma p^a_{i,b} + (1-\gamma) p^{\text{prior}}_{i,b}\big) \tag{2}$$

where $p^a_{i,b}$ denotes the normalized frequency of a word b on the ith lyrics based on the transcription, while $p^{\text{prior}}_{i,b}$ represents the global probability of a word b based on the MSD-MXM corpus. $\tilde{N}_i = r_i N_i$ denotes the estimated word count of the full text based on the length ratio r_i between full audio and snippet audio, and the observed word count from the transcription N_i. Based on a preliminary study, we choose the mixing coefficient $\gamma = 0.8$ which yielded the best adjustment quality. [6]

Given that MSDFullLT, InternalLT, InternalRec, and JamendoMood (via transcription) directly provide full text lyrics for the embedding encoding, they do not require any of the aforementioned pre-processing steps. An overview of the datasets can be found in Table 1.

4.4 Experimental Setup Details

4.4.1 Lyrics Embedding Models

Unlike sBERT, for which we use a pre-trained model, we train BE and WE models either with the MSD-MXM dataset or a proprietary lyrics corpus. [7] As discussed in section 4.3.3, downstream task datasets based on MSD-MXM are pre-processed with the BoW5K representation, which lacks the token sequential dependency information. As WE requires the reference embeddings where typically pre-trained token/word embeddings are used, we employ *glove-840B* [52] word embeddings. For training BE and WE embeddings, we only use half of the songs uniformly sampled from MSD-MXM (118,831/237,662) to consider the scenario where lyrics are only available for a subset of songs. For datasets with the full texts available, we employ BE and WE pre-trained on a subset of a proprietary lyrics corpus containing 3 million unique lyrics.

[4] For efficient evaluation, we compute estimates per fold by averaging nDCG over 5 randomly sampled subsets of 3000 users. It is shown that the estimation error is marginal, not impacting the overall conclusion.

[5] http://millionsongdataset.com/tasteprofile/

[6] The BoW5K matrix becomes dense after this adjustment, which still is tractable for computing BE and WE, due to the word truncation at 5000. However, for a large scale dataset, we suggest to set $\gamma = 1$, which disregards the prior but significantly improves computational efficiency.

[7] We use implementations from the vectorizers package.

4.4.2 *Regression Model for TLE$_R$*

For the enhancement model for TLE$_R$, we apply multivariate linear ridge regression where the regularizer $\mathcal{R}(\Theta) = ||\Theta||$ and the optimal α is selected from the range $\{10^p : p = [-6, -5, ...5, 6]\}$ via cross-validation.

4.4.3 *Downstream Task Pre-processing & Models*

We apply standardization followed by Principal Component Analysis (PCA) to embeddings at 99.9% explained variance ratio with whitening. This is especially useful for combined audio-lyrics embeddings in order to balance the contribution of each modality.

For *Tagging* tasks, we apply ridge logistic regression, populated per tag to handle the multi-label classification problem. The regularization coefficient is found by cross-validation, from the same range used for the regressor described in Section 4.4.2.

For the *RecSys* task, we employ item K-Nearest Neighbor (itemKNN) [53]. For each song, it computes and caches the K most similar songs by measuring cosine distances between song embedding vectors, which results in a sparse song-song similarity matrix. Later, it serves songs that are most similar to the users previously listened songs by employing this similarity matrix. Finally, the optimal K is found by cross validation per fold in each feature/dataset combination from the range of $[20, 50, 100, 200, 500, 1000, 2000]$.

5. RESULTS & DISCUSSION

5.1 Are LE in general useful for MIR tasks?

Although our main focus is the effectiveness of TLEs on MIR tasks, we briefly discuss its ideal counterpart, LE. Our main interest is whether LE outperforms the baseline scenario where *only the audio embedding is used*, compared to scenarios where LE is either used alone or in combination with audio embeddings for downstream tasks. As Figure 2 suggests, LE (round points in pink) outperforms the baseline (dashed horizonatal line) particularly when the task is lyrics focused (i.e., MSDSnippetLT, MSDFullLT, InternalLT), or when the combined lyrics and audio embedding is given to downstream task models (i.e., the three "+Audio" columns to the right of each grouping). It is notable that a performance improvement is observed on most of the tagging datasets and one RecSys scenario (i.e., InternalRec using MSLE) when LE is combined with audio embeddings. However, overall we observe a smaller effect for RecSys tasks. We assume that this is due to the smaller relative effect of LE against the baseline in those cases.

Comparing audio baselines, OpenL3 performs worse than MULE and MSLE on most tagging tasks (except MSDFullLT), while performing better in RecSys tasks. Despite these differences, we observe similar trends regarding the performance of LEs compared to those baselines.

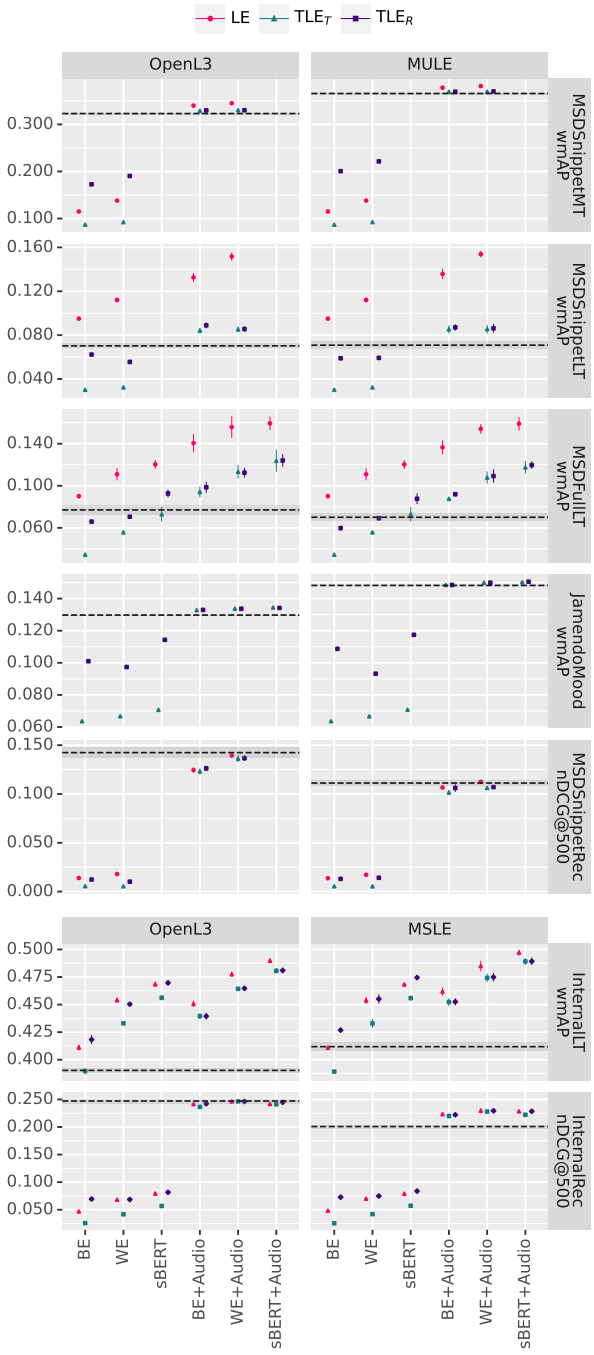

Figure 2. Each sub-figure corresponds to one dataset (row) and audio embedding (column). x and y axes represent embedding combinations and performance measures per task, respectively. Dashed horizontal lines and the shaded gray area in each figure represents the average performance and confidence interval when only the audio embedding is used. Each other point and vertical bar indicates the average performance and confidence interval of an embedding (combination). We set confidence intervals at 95%.

5.2 Does TLE$_T$ provide complementary information to audio embeddings?

In practice, the confirmed effectiveness of LE is unlikely to be helpful due to limited access to 'ground-truth' lyrics. Regardless, our results indicate that TLE$_T$ can also achieve better testing performance compared to audio-only base-

lines, similar to LE.

We observe degradation of performance compared to LE in most of the cases, which is expected due to the transcription error of the ALT process. The error can be seen in Figure 3, where we measure the cosine similarity between corresponding pairs of LE and TLEs. This suggests that the transcription error can be severe such that the cosine similarity of a large number of pairs approaches 0 (i.e., MSDSnippetLT, MSDFullLT).

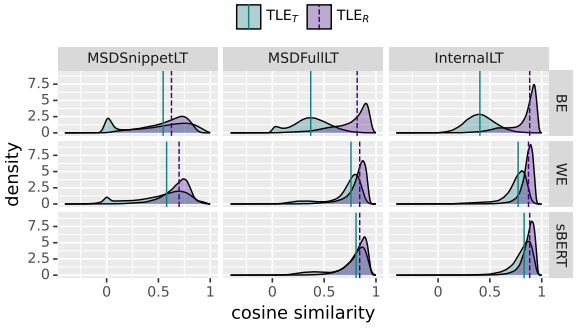

Figure 3. Distribution of cosine similarities measured between pairs of TLE$_T$ – LE and TLE$_R$ – LE, respectively. Each sub-figure represents the result per dataset and lyrics embeddings tested. Vertical bars denote the median.

However, TLE$_T$, even with such transcription errors and hence a loss of fidelity in the resulting embeddings, still provides meaningful performance gains when compared to an audio-embedding-only scenario. This is especially true on lyrics focused tasks or when combining the audio and (noisy) lyrics embeddings. In JamendoMood and MSDSnippetMT, where the lyrics data is not available upfront or the task is not focused on lyrics, we still observe that TLE$_T$ combined with audio features outperforms audio embeddings alone.

This implies that one can build lyrics-based ML systems that can be applied to all the songs in the catalog of interest, with an expectation of a performance gain compared to models solely dependent on audio embeddings.

Comparing the performance of TLE$_T$ between MSDSnippetLT and MSDFullLT, it is notable that the truncation of transcribed lyrics influences the effectiveness of the resulting embeddings. This suggests that providing full-length lyrics transcripts where possible is important.

5.3 Does TLE$_R$ further improve TLE$_T$?

Next, we focus on TLE$_R$ which applies regression on top of TLE$_T$. Ideally, we would expect that the regression improves the fidelity of TLE$_T$, which likely results in an improved downstream task performance. First, we can confirm that the regression indeed improves the fidelity, as suggested by Figure 3. Measured on the testing samples of matching pairs of LE and TLE$_R$, the average cosine similarity is improved in all the cases. The effect is more obvious when the initial TLE$_T$ has lower fidelity (i.e., BE, MSDSnippetLT). This indicates that the regression does increase the fidelity to some degree.

However, on the downstream tasks, the effect is not as consistent as in the cosine similarity (fidelity) result. In the case where the audio embedding is not included as input, the result indicates that TLE$_R$ improves the downstream performance over TLE$_T$, sometimes even outperforming corresponding LEs (i.e., MSDSnippetMT, InternalLT). However, once combined with audio embeddings, the effect is not as distinct. While overall a small positive effect is observed in the RecSys datasets, the effect in the tagging datasets seems to be less clear, with TLE$_R$ generally performing on par with TLE$_T$.

One explanation could be that the concatenation of audio embeddings for downstream tasks would eventually provide the same degree of audio information for TLE$_T$ as already provided for TLE$_R$. The regression of TLE$_R$ is conditioned both by TLE$_T$ and the audio embedding, and thus would likely inherit the audio information. This is a possible explanation for the cases where TLE$_R$ outperforms both LE and TLE$_T$. Similarly, TLE$_T$ combined with the audio embedding explicitly fusing the two modalities via concatenation, shows performance that is on par with TLE$_R$ in most of the cases.

6. CONCLUSION & FUTURE WORK

In this work, we introduce and assess transcription-based lyrics embeddings which tackles the problem of lyrics availability. An experiment is conducted to evaluate the effectiveness of TLEs in popular MIR downstream tasks, assessed against two comparisons, namely 'ground-truth' lyrics embeddings and audio embeddings. The result indicates that TLEs perform generally in between these two contenders, especially when combined with audio embedding and on the lyrics-focused tasks. This implies that TLEs can be an effective approach to be applied in lyrics-relevant MIR tasks where lyrics are often unavailable.

In particular, our results suggest that TLE$_T$ is a simple, yet effective method for various downstream tasks when combined with audio embeddings. It is shown to complement audio embeddings by improving performance when combined with them. These gains can be achieved by using only off-the-shelf pre-trained models, while not requiring any access to 'ground-truth' lyrics whatsoever. Additionally, this approach does not require any subsequent refinement processes as is the case with TLE$_R$.

Furthermore, we identify some areas of exploration by which TLE could be improved: 1) end-to-end learning that directly associates the audio and LE to potentially improve the quality of TLE$_R$ and avoids the transcription process, 2) instead of a simple linear regression model, more advanced methods such as semi-supervised learning [54] could further improve the fidelity of TLE$_R$. Additionally, 3) using context vectors from DL-based ALT models could be a viable alternative TLE, which bypasses the lyrics text embedding models. Finally, 4) while not the main focus due to the prevalence of English language in the data, multi-lingual transcription and embedding models could further improve the results in a more general setup.

7. REFERENCES

[1] A. M. Demetriou, A. Jansson, A. Kumar, and R. M. Bittner, "Vocals in Music Matter: the Relevance of Vocals in the Minds of Listeners," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 514–520.

[2] J. Kim, A. M. Demetriou, S. Manolios, M. S. Tavella, and C. C. Liem, "Butter Lyrics Over Hominy Grit: Comparing Audio and Psychology-Based Text Features in MIR Tasks." in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 861–868.

[3] K. Choi, J. H. Lee, X. Hu, and J. S. Downie, "Music subject classification based on lyrics and user interpretations," in *Proceedings of the 2016 Annual Meeting of the Association for Information Science and Technology*, vol. 53, no. 1, 2016, pp. 1–10.

[4] X. Hu and J. S. Downie, "When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010, pp. 619–624.

[5] S. Naseri, S. Reddy, J. Correia, J. Karlgren, and R. Jones, "The Contribution of Lyrics and Acoustics to Collaborative Understanding of Mood," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 16, 2022, pp. 687–698.

[6] R. Mayer, R. Neumayer, and A. Rauber, "Rhyme and Style Features for Musical Genre Classification by Song Lyrics," in *Proceedings of the 9th International Conference on Music Information Retrieval*, 2008, pp. 337–342.

[7] K. Watanabe and M. Goto, "Query-by-blending: A music exploration system blending latent vector representations of lyric word, song audio, and artist," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 144–151.

[8] A. G. Smith, C. X. S. Zee, and A. L. Uitdenbogerd, "In your eyes: Identifying clichés in song lyrics," in *Proceedings of the Australasian Language Technology Association Workshop*, 2012, pp. 88–96.

[9] P. Knees and M. Schedl, "A Survey of Music Similarity and Recommendation from Music Context Data," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 10, no. 1, pp. 1–21, 2013.

[10] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 694–701.

[11] M. McVicar, B. D. Giorgi, B. Dundar, and M. Mauch, "Lyric document embeddings for music tagging," *arXiv preprint arXiv:2112.11436*, 2022.

[12] C. Laurier, J. Grivolla, and P. Herrera, "Multimodal Music Mood Classification Using Audio and Lyrics," in *Proceedings of the 7th International Conference on Machine Learning and Applications*, 2008, pp. 688–693.

[13] X. Gao, C. Gupta, and H. Li, "Genre-conditioned acoustic models for automatic lyrics transcription of polyphonic music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 791–795.

[14] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 891–899.

[15] L. Zhuo, R. Yuan, J. Pan, Y. Ma, Y. Li, G. Zhang, S. Liu, R. Dannenberg, J. Fu, C. Lin *et al.*, "LyricWhiz: Robust Multilingual Lyrics Transcription by Whispering to ChatGPT," in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023, pp. 343–351.

[16] T. Deng, E. Nakamura, and K. Yoshii, "End-to-end lyrics transcription informed by pitch and onset estimation," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 633–639.

[17] E. Demirel, S. Ahlbäck, and S. Dixon, "Mstre-net: Multistreaming acoustic modeling for automatic lyrics transcription," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 151–158.

[18] K. Watanabe and M. Goto, "A chorus-section detection method for lyrics text," in *Proceedings of the 21th International Society for Music Information Retrieval Conference*, 2020, pp. 351–359.

[19] D. Yang and W. Lee, "Music emotion identification from lyrics," in *Proceedings of the 11th IEEE International Symposium on Multimedia*, 2009, pp. 624–629.

[20] Z. S. Harris, "Distributional structure," *WORD*, vol. 10, no. 2-3, pp. 146–162, 1954.

[21] S. Sasaki, K. Yoshii, T. Nakano, M. Goto, and S. Morishima, "Lyricsradar: A lyrics retrieval system based on latent topics of lyrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 585–590.

[22] B. Logan, A. Kositsky, and P. J. Moreno, "Semantic analysis of song lyrics," in *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo*, 2004, pp. 827–830.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Workshop Track Proceedings of the 1st International Conference on Learning Representations*, 2013.

[24] P. Donnelly and A. Beery, "Evaluating Large-Language Models for Dimensional Music Emotion Prediction from Social Media Discourse," in *Proceedings of the 5th International Conference on Natural Language and Speech Processing*, 2022, pp. 242–250.

[25] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, "MuLan: A Joint Embedding of Music Audio and Natural Language," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 559–566.

[26] 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.

[27] C. Wang, R. Lyu, and Y. Chiang, "An automatic singing transcription system with multilingual singing lyric recognizer and robust melody tracker," in *Proceedings of the 8th European Conference on Speech Communication and Technology*, 2003, pp. 1197–1200.

[28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 28 492–28 518.

[29] O. Cífka, C. Dimitriou, C. Wang, H. Schreiber, L. Miner, and F. Stöter, "Jam-alt: A formatting-aware lyrics transcription benchmark," *arXiv preprint arXiv:2311.13987*, 2023.

[30] A. Mallick, K. Hsieh, B. Arzani, and G. Joshi, "Matchmaker: Data drift mitigation in machine learning for large-scale systems," in *Proceedings of Machine Learning and Systems*, 2022, pp. 77–94.

[31] S. Ackerman, E. Farchi, O. Raz, M. Zalmanovici, and P. Dube, "Detection of data drift and outliers affecting machine learning model performance over time," in *Proceedings of the Joint Statistical Meetings Conference*, 2020, pp. 144–160.

[32] S. Gandhi, P. von Platen, and A. M. Rush, "Distil-Whisper: Robust Knowledge Distillation via Large-Scale Pseudo Labelling," *arXiv preprint arXiv:2311.00430*, 2023.

[33] distill-whisper/distill-large-v2. https://huggingface.co/distil-whisper/distil-large-v2. Accessed: 2024-07-30.

[34] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011, pp. 591–596.

[35] P. Gage, "A new algorithm for data compression," *The C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[36] A. Rajaraman and J. D. Ullman, *Data Mining*. Cambridge University Press, 2011, p. 1–17.

[37] N. Halko, P. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[38] C. Moosmüller and A. Cloninger, "Linear optimal transport embedding: provable Wasserstein classification for certain rigid transformations and perturbations," *Information and Inference: A Journal of the IMA*, vol. 12, no. 1, pp. 363–389, 09 2022.

[39] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 3982–3992.

[40] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, "MPNet: Masked and Permuted Pre-training for Language Understanding," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 16 857–16 867.

[41] sentence-transformers/all-mpnet-base-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2. Accessed: 2024-07-30.

[42] R. Arandjelovic and A. Zisserman, "Look, Listen and Learn," in *IEEE International Conference on Computer Vision*, 2017, pp. 609–617.

[43] J. Cramer, H. Wu, J. Salamon, and J. P. Bello, "Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3852–3856.

[44] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and Unsupervised Learning of Audio Representations for Music Understanding," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 256–263.

[45] K. Choi, G. Fazekas, and M. B. Sandler, "Automatic Tagging Using Deep Convolutional Neural Networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 805–811.

[46] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009, pp. 387–392.

[47] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The MTG-Jamendo Dataset for Automatic Music Tagging," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning*, 2019.

[48] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 316–323.

[49] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems*, vol. 26, 2013, pp. 2643–2651.

[50] S. Oramas, O. Nieto, M. Sordo, and X. Serra, "A Deep Multimodal Approach for Cold-start Music Recommendation," in *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, 2017, pp. 32–37.

[51] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.

[52] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[53] M. Deshpande and G. Karypis, "Item-based top-*N* recommendation algorithms," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143–177, 2004.

[54] G. Kostopoulos, S. Karlos, S. Kotsiantis, and O. Ragos, "Semi-supervised regression: A recent review," *Journal of Intelligent and Fuzzy Systems*, vol. 35, no. 2, pp. 1483–1500, 2018.