Predicting Metadata from Song Lyrics Using NLP

Ashley Ke, Elizabeth Ke, Nitin Kumar, Sadhana Lolla
MIT 6.8610
ashleyke@mit.edu, elizke@mit.edu,
nitink@mit.edu, sadhana@mit.edu

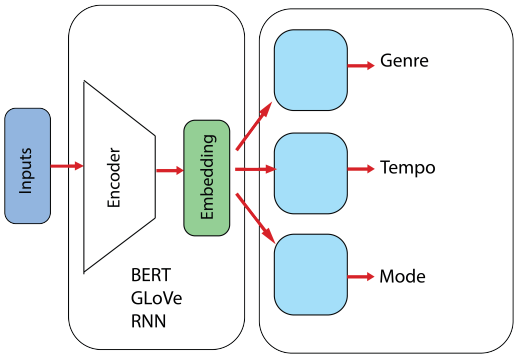

Figure 1: Architecture we use to predict target variables. We vary 3 different encoders and use different prediction heads to perform the downstream tasks.

Abstract

Tracking metadata is crucial for streaming services to accurately recommend songs to listeners and automatically compile playlists. Currently, metrics such as the acousticness, loudness, and other musical attributes are collected by crowdsourcing: users are asked to manually input this information into forms. This can lead to incomplete records of metadata. To standardize this process, we develop an NLP system that automatically predicts musical metadata about a song only from its lyrics. We predict seven different metrics: genre, mode, danceability, energy, valence, tempo, and loudness. We find that we can predict metrics such as danceability and valence with high accuracy, but we cannot predict attributes such as mode and tempo, since these may be independent of the lyrics. We also find that out of all the embedding models we use, the embeddings generated by BERT are the best for these downstream tasks, which we attribute to the necessity of tracking long-term dependencies, which BERT achieves using attention mechanisms.

1 Introduction

Current streaming services, such as Spotify and Apple Music among others, use song metadata to tag tracks with properties such as genre, acoustics, danceability, and more. Spotify metadata is often crowd-sourced and manually inputted, leading to oftentimes incomplete or missing information. In response to this issue, we explore using NLP-based methods to automatically assign song metadata to tracks by analyzing the lyrical content of the song to predict musical attributes. This would enable streaming services to systematically assign these metadata values to their tracks, without relying on possibly inaccurate responses from users.

Currently, NLP-based classification schemes based on song lyrics primarily focus on predicting genre. Previous works approach genre classification by applying various embedding techniques such as Word2Vec with TFIDF, GloVe embeddings, and BERT representations combined with deep learning and RNN approaches.

However, genre classification is unique amongst song metadata in that it is more easily predictable given lyrics than other aspects of a song, which are primarily auditory.

In this paper, we jointly approach genre classification with predicting other attributes about a given song, which is a relatively unexplored field. Using the same song lyric embeddings, we apply several different models to predict the following metadata attributes: genre, mode, danceability, energy, valence, tempo, and loudness. We can uncover trends about the lyrical composition of songs using this method, such as finding lyrics with high association to danceability or musical composition (major or minor key).

The remainder of this paper is structured as follows. Section 2 discusses related works already completed regarding genre prediction given lyrics; section 3 describes construction of our custom dataset; section 4 describes methods and models trained; section 5 discusses experimental results; and section 6 provides concluding remarks and discusses limitations.

2 Related Works

Previous works using song lyrics have largely focused on genre classification.

(Kumar et al., 2018) uses deep learning on Word2Vec song lyric embeddings to categorize songs into one of the 4 genres Christian, Metal, Country, and Rap. They report an accuracy of 65% without TFIDF and 74% with TFIDF when using a 3-layer deep neural network.

(Tsaptsinos, 2017) attempts to classify songs into either 20 or 117 genres by applying recurrent neural networks to GloVe word embeddings of song lyrics. In a 20-genre dataset, their Long Short Term Memory (LSTM) model achieves an accuracy of 49.77%. With a larger 117-genre dataset, a hierarchical attention network (HAN) outperforms the LSTM, with an accuracy of 46.42%.

(Akalp et al., 2021) compares performance of a BI-LSTM model to dense layers applied on top of BERT and DistilBERT in the classification task of labeling songs from 13 genres. They discover that BERT outperforms other models, achieving an accuracy of 77.63% on one-label classification and 71.29% on multi-label classification.

3 Dataset

We construct the dataset in three main steps: (1) gathering song lyrics, (2) finding the corresponding songs on Spotify, (3) getting the metadata for each song, and (4) preprocessing the dataset. Each step presents a set of challenges.

3.1 Gathering song lyrics

We looked into many options and sources for gathering lyrics. Websites that provide lyrics for songs consider the lyrics a product of their work, and protect them under IP laws. Additionally, our datset requires a large volume of songs, which poses challenges because many websites and lyric sites have protections on their websites and API's that limit the rate at which we could pull these lyrics. Ultimately, we decide to use a dataset from Kaggle constructed by Anderson Neisse (Neisse, 2022). The lyrics in the dataset are scraped from the Brazilian website Vagalume (https://www.vagalume.com.br/). The songs are chosen to be the most popular songs of the most popular website, according to the website. In addition to music in English, the website and the dataset have a significant presence of Latin American and Caribbean music in other languages. For the sake of working in a single language with songs we are largely familiar with, we filter the data to consist of only songs in English, with the help of Vagalume's genre tags.

3.2 Finding corresponding Spotify songs

Next, in order to get metadata for each song from Spotify, we get each song's Spotify ID. For each song, we search for the song by name and artist via the Spotify API. This step takes a very long time to run — there are over 150,000 songs to search, and Spotify API empirically seems to allow only about 2 queries per second. Moreover, despite following the backoff-retry strategy suggested by Spotify's API, an API access key gets banned for about 24 hours after about 10,000 requests. We use multiple access keys to somewhat increase this limit. Some songs are not found on Spotify, and sometimes the same ID was found for multiple songs in the Kaggle dataset; we decide that there are too many such cases to review individually, so we simply do not use those songs.

3.3 Gathering metadata

Using Spotify's API, we get the genres for each artist and musical features corresponding to each song, namely:

- mode: whether the song is in major (mode = 1) or minor (mode = 0) key
- danceability: how suitable a track is for dancing
- energy: perceptual measure of intensity and activity
- valence: musical positiveness conveyed by a track
- loudness: overall loudness of the track in decibels
- tempo: pace of a track, given in beats per minute.

These musical attributes are often concentrated around particular values (see Figure 2) and are somewhat correlated with each other (e.g. loudness and energy are positively correlated).

3.4 Preprocessing

We use each song's musical features to find and filter out some tracks that are not really songs in

the dataset (e.g. a blurb by an artist about their album likely has a very high "speechiness" value). We further preprocess the data by dropping remix and karaoke versions of the same song, dropping duplicate lyrics, filtering out extreme values, etc. After all these edits, the remaining final dataset contains 86,242 songs.

4 Methodology

Our approach has two parts: (1) finding effective encodings for the lyrics, and (2) creating a multi-output classifier using these embeddings that classifies them into categories or fits them to the target variables. Our architecture is summarized in Figure 1.

4.1 Generating lyric embeddings

We explore both pretrained and from-scratch embedding generation mechanisms to generate robust representations of song lyrics. We experiment with three different types of embeddings:

1. Pre-trained DistilBERT, which uses a transformer mechanism to embed words. We use this embedding to learn if long-term dependencies between lyrics and the attention mechanism are helpful in encoding the lyrics. We also tried fine-tuning DistilBERT on this dataset, but found that the model did not learn well, possibly due to the large number of parameters.

2. GloVe model, which embeds meaning by taking account of word-word co-occurrence probabilities. We use GloVe pretrained vectors of dimension 300 as one of our embedding types.

3. RNN embeddings. We train these from scratch using an embedding layer, a 2-layer LSTM, and dropout layer. We attempt two methods of from-scratch tuning: training an RNN per head (i.e. optimizing the encoder only for a single objective– whether that is danceability, energy, valence, or any other single category). We also attempt to jointly optimize the RNN for all of these categories at the same time. We do this by feeding the results of the RNN into all of the prediction heads (further explained in 4.2) and summing all of their losses, and then backpropagating through all of the weights. We found that

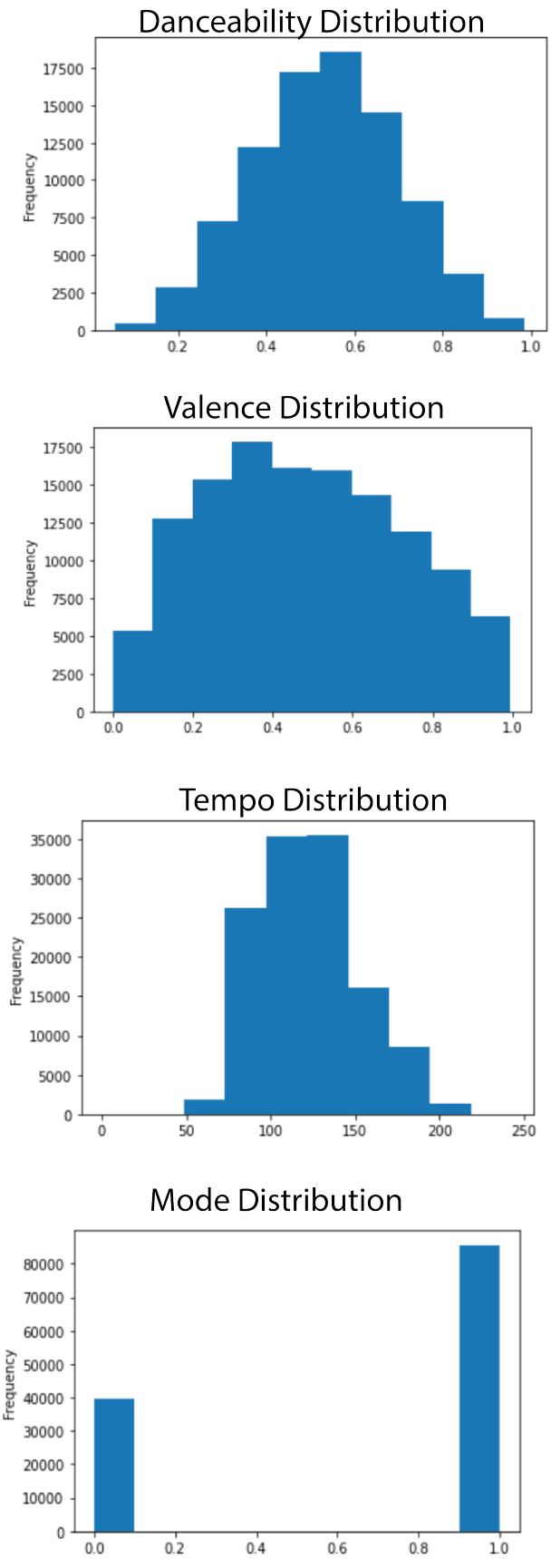

Figure 2: Distribution of values across different metrics in the dataset.

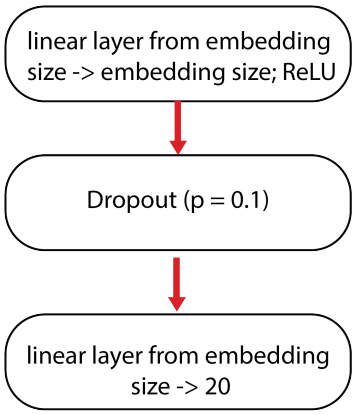

Figure 3: Model Architectures for song classification head

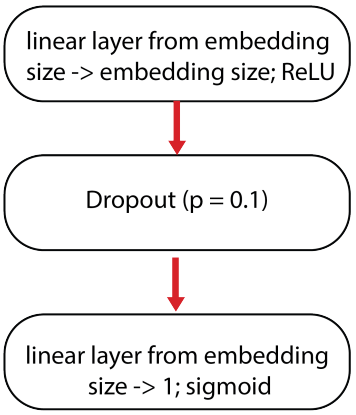

Figure 4: Neural network used for binary classification and regression tasks

the joint optimization didn't perform well, indicating that the feature extraction must be different per output. Therefore, we optimize a different RNN per output.

4.2 Computing downstream tasks

We train different models to find the best approach for computing downstream tasks on the embeddings. Since many our attributes have different formats, we use a separate approach for each format. In all cases, we split the dataset into an 80-20 split for training and testing, respectively, and we train for 20 epochs.

4.2.1 Multi-label classification

First, we attempt to predict genre through multi-label classification. The dataset contains 1047 genres. Classifying into all these genres is an unrealistic classification task due to the sheer number of genres, the similarity and overlap between many of the genres, and the rarity of some genres. To address this, we narrow down our dataset to 20 genres. We pick 20 of the most common genres with low overlap: [rock, dance pop, pop, mellow gold, metal, permanent wave, rap, singer-songwriter, hip hop, punk, indie rock, urban contemporary, neo mellow, r&b, country, soul, adult standards, folk, new romantic, and trap]. Then, we filter out all genres not in this set. For each song, we convert the genres from a string representation to a multi-hot encoding, a 20-dimensional vector where each entry is 1 if the corresponding genre is one of the song's genre labels, and 0 if not.
Then, we train a model to predict a probability for each of the 20 genres. We use the neural network architecture shown in Figure 3.

For training, we use pytorch's BCEWithLogitsLoss() function, which combines a sigmoid layer with binary cross entropy loss. For our optimizer, we use the Adam optimizer. To measure performance of the model, for each song we take the argmax of the model's output and treat that genre as the model's prediction for the song's genre. We consider a prediction correct if it is in fact one of the song's genre labels, and incorrect otherwise. Using this method, we compute and record the accuracy for the test dataset. We also evaluate performance using the AUCROC. To do this, we assign different thresholds to designate a positive and calculate the true positive rate and false positive rate. We plot this and calculate the area under the curve (AUC) as an additional measure of accuracy.

4.2.2 Binary classification

We use binary classification for predicting mode, since mode is always either 0 or 1. We pass the lyric embeddings from 4.1 through the neural network architecture shown in Figure 4. The neural network outputs a value between 0 and 1, which we round to get our prediction. We train the model using binary cross entropy loss and the Adam optimizer. Finally, we measure accuracy by calculating the classification accuracy of our rounded outputs on the test dataset.

4.2.3 Regression

Finally, we use regression to predict the remaining attributes: danceability, energy, valence, tempo,

	BERT	GloVe	RNN
AUC-ROC	0.89	0.85	0.83
Accuracy	54.6%	48.4%	45.8%

Table 1: Genre results.

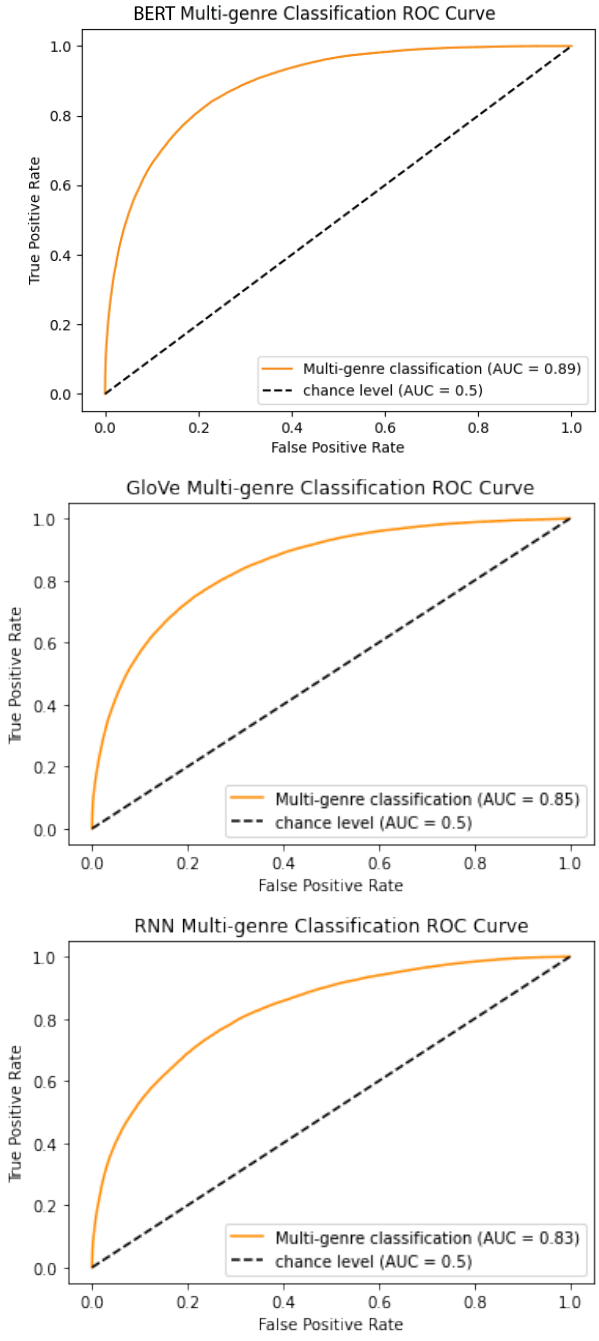

Figure 5: AUC-ROC curves for genre. BERT exhibits better AUC than the other metrics

and loudness, which are all numerical attributes. Danceability, energy, and valence are defined to be numbers between 0 and 1, and we normalize tempo and loudness so that they also lie in this range.
To predict each value, we pass the lyric embeddings through the same neural network architecture as used in binary classification, shown in Figure 4. However, we no longer round the outputs, and we use mean squared error for our loss function instead of binary cross entropy.

5 Results

In this section, we discuss our results in (1) genre prediction, (2) mode prediction, (3) regression prediction of danceability, energy, valence, tempo, and loudness.

5.1 Genre

Our results of classifying songs into 20 genres using multi-label classification are shown in Table 1. We see that BERT performs the best in both metrics, with the highest AUC-ROC of 0.89 and an accuracy of 54.6%. GloVe is next best, and RNNs perform the worst. As expected, the pre-trained results perform better than our RNN from scratch, as they are trained on larger sets of data. BERT, a transformer based approach, performs the best, and shows better performance than GloVe's word-word co-occurrence probabilities. The three AUC-ROC curves are shown in Figure 5.

We can see that BERT primarily achieves more true positives at lower confidence thresholds than GloVe or RNNs, since it is able to assign label unlikely low genres a lower confidence.

We find that our accuracies are comparable to those achieved by (Tsaptsinos, 2017), and that BERT outperforms these models. Since we chose different genres than the papers cited, it is unclear to what degree the genre selection played a part in this. Our results also support the trend of more genres leading to poorer classification results, which is logical due to genres overlapping as the number of genres increases.

	BERT	GloVe	RNN
Mode	69.96%	68.5%	67.8%

Table 2: Accuracy for mode predictions.

	BERT	GloVe	RNN
Danceability	0.0189	0.0209	0.0268
Energy	0.0381	0.0429	0.0563
Valence	0.0477	0.0503	0.0681
Loudness	0.0048	0.0055	0.0074
Tempo	0.0193	0.0197	0.0210

Table 3: Regression MSE values.

5.2 Mode

Our results of predicting mode through binary classification are shown in Table 2. We observe that all three models achieved an accuracy of about 68%, which did not significantly improve over the 20 epochs of training. In all three cases, the model learns to predict 1 for almost all of the lyric examples, leading to a 68% accuracy because 68% of the values in the test dataset have a mode of 1.

These results suggest that the parameter mode is not well associated with song lyrics. All three models were unable to learn any useful association.

5.3 Regression Categories

For the remaining categories of danceability, energy, valence, tempo, and loudness, we predicted values using regression. Table 3 displays the MSE loss values for each model and category. Once again, we see that overall, BERT performs the best, followed by GloVe, and finally RNNs. Table 4 shows the correlation values for each of these models.

We observe that MSE is not a good indicator of model performance, as we see low correlations for tempo despite the small MSE value. Danceability and energy show the strongest correlation, followed by valence and loudness, and finally minimal correlation in tempo. These results suggest that our models are best able to learn associations with danceability and energy, do moderately well with valence and loudness, and finally learn minimal results for tempo.

	BERT	GloVe	RNN
Danceability	0.4524	0.4238	0.3571
Energy	0.5067	0.4357	0.3587
Valence	0.3341	0.3641	0.2780
Loudness	0.4378	0.3506	0.2916
Tempo	0.0970	0.1163	0.0303

Table 4: Regression correlations of actual and prediced values.

We explore scatterplots for the first 1500 points of the test set for two of these categories. Figure 6 shows scatter plots of danceability and tempo for the three models. The left column displays danceability plots and the right column displays tempo plots, in the order BERT, GloVe, RNN from top to bottom. We observe that all the danceability plots show a moderately positive correlation between actuals and predicted, reflecting moderate success in learning an association between song lyrics and danceability. Meanwhile, the tempo scatter plots all appear to have minimal to no correlation. While the MSEs are low, this appears to be primarily due to predictions falling in the range from 0.4 to 0.6.

6 Impact Statement

We hope that in the future, lyrics will play a larger role in the metadata stored about songs. Lyrics are important to songs, and have an important role to play in tasks related to categorization and recommendation engines, and can also be helpful to increasing the accuracy of existing automatically-generated metadata. Our work demonstrates the potential of NLP techniques in this space by showing that lyrics alone can be used as a strong predictor of genre and other musical attributes.

However, before similar techniques are implemented by a large music service, there are many sensitive issues that merit significant consideration. For instance, we noticed that even if musical attributes of songs are similar, artists and songwriters of different walks of life and cultural backgrounds might have different diction. Consequently, even if a human would say two songs are of very similar genres, a lyric-based system may predict starkly different genres. As another example, lyrics often contain slurs or pejoratives that open the door to model bias. While our focus was mostly on exploring techniques to start using lyrics as metadata, we recognize that our model does not address many of these challenges.

References

Hasan Akalp, Enes Furkan Cigdem, Seyma Yilmaz, Necva Bolucu, and Burcu Can. 2021. Language representation models for music genre classification

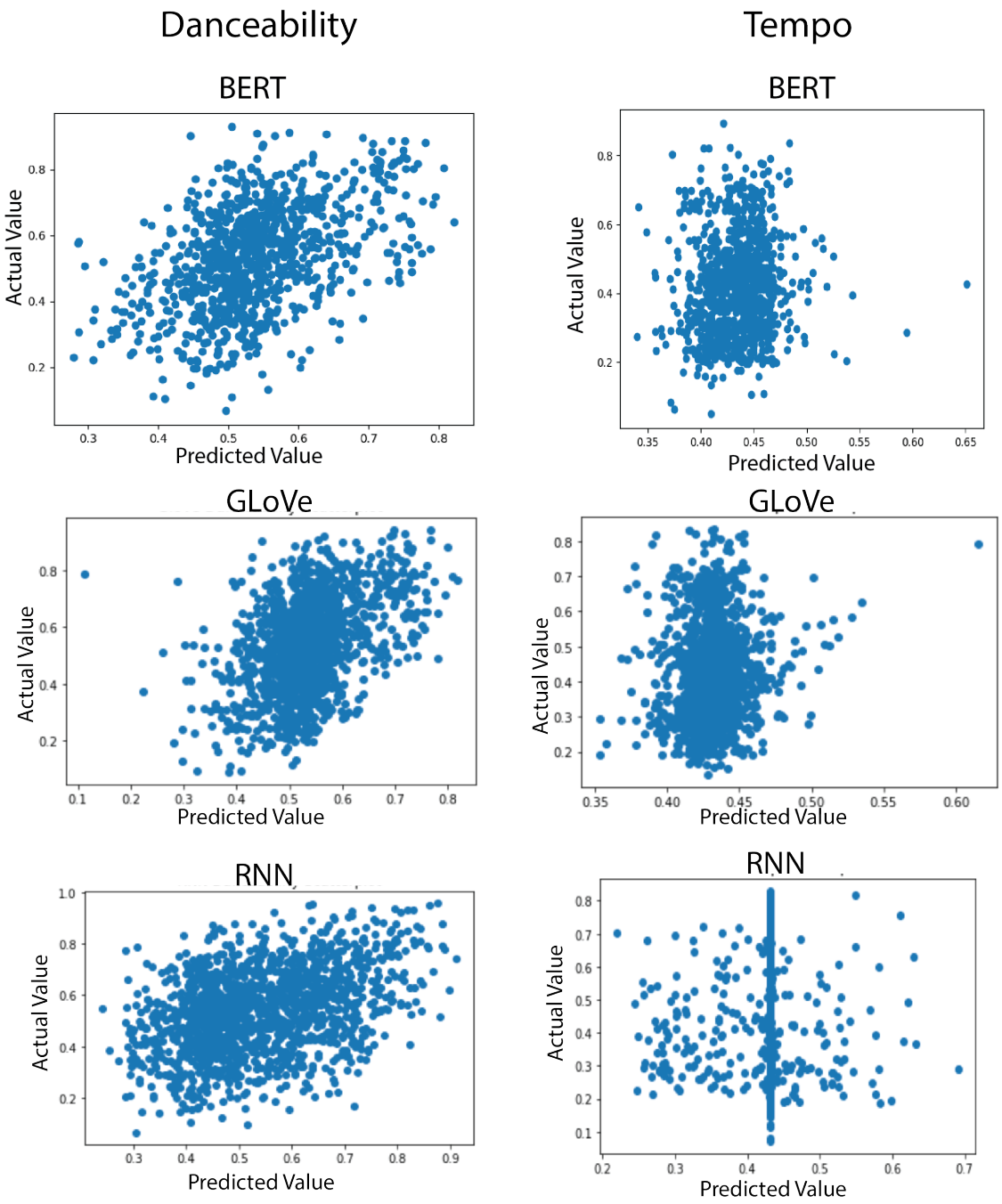

Figure 6: Predicted value vs. actual value for danceability (moderately positive correlation) and tempo (little to no correlation) across embeddings

using lyrics. In *2021 International Symposium on Electrical, Electronics and Information Engineering*, ISEEIE 2021, page 408–414, New York, NY, USA. Association for Computing Machinery.

Akshi Kumar, Arjun Rajpal, and Dushyant Rathore. 2018. Genre classification using word embeddings and deep learning. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2142–2146.

Anderson Neisse. 2022. Song lyrics from 79 musical genres. https://www.kaggle.com/datasets/neisse/scrapped-lyrics-from-6-genres.

Alexandros Tsaptsinos. 2017. Lyrics-based music genre classification using a hierarchical attention network.