Multimodal Music Recommender System: Final Report

Phakawat Wangkriangkri*, Jae Young Kim*, Anirudh Alameluvari*, Xiaoying Zhang*
Jessica D'souza*

*{wangkria, jkim2458, alameluv, xzhang88, jjdsouza}@usc.edu

Abstract

Personalization and recommendation have been an important challenge for the tech companies as it leads to profit. Especially, in the music industry, it is important since new songs are published every day and users have limited information about them. Previously, collaborative filtering models and sequential models were developed and adapted to recommend songs similar to users preferences. However, these models only use play history even though there are various other features of songs that can be leveraged. In this paper, we propose a multimodal recommendation system using some of these datasets to address this limitation.
Our effort can be summarized as follows
1) Data for 4 different modalities were collected: artist biography, lyrics, audio features, and album art.
2) Song embeddings for each of these modalities are generated.
3) Incorporated generated embeddings to the sequential recommendation model, BERT4Rec, by replacing randomly initialized embedding layers.

1 Introduction

With the advent of newer streaming platforms and the increase in size and variety of music libraries, the need for automatic music recommendation algorithms grew tremendously (Schedl et al., 2014). Recommendation systems mainly try to make long-term recommendations using collaborative filtering (Goldberg et al., 1992) and content-based filtering (Balabanović and Shoham, 1997), or a hybrid of both. Another approach could be to explore contextual information such as time, geographical location, weather and use the above methods for short term music recommendation to factor in the fact that user's music interests changes with context (Lee and Downie, 2004). However, most of the traditional music recommender systems tend to utilize a unimodal approach.

One of the earliest works using multimodal approach tried to address the cold start problem using a hybrid recommendation approach (Oramas et al., 2017) using artist biography and audio spectograms. In the artist recommendation task, artist biographies are enriched using Entity Linking (Moro et al., 2014) and then Word2Vec (Mikolov et al., 2013) trained on Google News is then used to get the embeddings. For song embeddings, Convolutional Neural Networks (CNN) are used to learn higher-level features from audio spectrograms. Finally, a late fusion technique is used to combine the feature vectors. Following a similar approach (Oramas et al., 2018) tried to learn and combine multimodal data representations for music genre classification. Album reviews are semantically enriched and classified among 13 genre classes using an SVM classifier (Oramas et al., 2016). Text representations are learned using a feed forward network. Audio representations are learned in a similar pattern as the previous work. Visual representations for the album art are learned using a ResNet (He et al., 2016), initialized with pretrained parameters learned in a general image classification task (Russakovsky et al., 2015), and fine tuned on the classification of music genre labels from the album cover images. Final classification task is done by concatenating chosen combination of these features (normalized) and passing it to a feed forward neural network.

Although these papers achieved good results using the multimodal approach. There are three main drawbacks of these methodologies. Firstly, they utilized architectures like CNN and Word2Vec on Google News to extract the embeddings. While these give good representation of the modailities. Newer state-of-the-art models like SBERT (Reimers and Gurevych, 2019) can be more useful for textual data as it generates embeddings that can show closeness between sentences. Also, Varia-

tional Autoencoders (Kingma and Welling, 2013) can help in dimension reduction for image data without significant loss of information. Secondly, most of the prior works have utilized up to two modalities. There is a lack of experimentation with different combinations of modalities and their overall importance in the final model. We experiment with upto four modalities to verify the efficacy of adding more modalities to improve the music recommendation system. Finally, the fusion technique used is a either a simple feed forward network (Oramas et al., 2017) or GRU4Rec (Oramas et al., 2018), but both of these methods lack capturing attention which can help in longer sessions. We explore the usage of BERT4Rec (Sun et al., 2019a) to improve on the recommendation following the fusion of the modalities.

2 Related Work

Normally, we encode a user's historical interactions into a vector (i.e., representation of user's preference) using a left-to-right sequential model and make recommendations based on this hidden representation. These sequential recommendation systems (Kang and McAuley, 2018a) recommend an object to a user based on interaction history where the user interacted with similar objects. (Sachdeva et al., 2018) explores using Attentive neural networks using tags and prior song history to model short term user preference. This method offered a better performance compared to all the other baseline models in modeling the short term user preference.

Despite their wide usage and effectiveness, such unidirectional models (left to right) are sub-optimal as their architectures restrict the information that can be encoded into the hidden representations. BERT4Rec (Sun et al., 2019b) employs deep bidirectional self-attention to model users' behaviors. However, we cannot just use BERT (Devlin et al., 2018) for recommendation as traditionally, we recommend from left to right and if we directly use it, it might just give out trivial results due to information leakage as it would indirectly allow each item to see the target item. In order to solve this problem, BERT4Rec employed Cloze Task. This is similar to the Masked Language Modeling Task in BERT.

This leads us to our major challenge, unimodal models tend to miss out of representing different aspects of a song that might influence the choice of the next song for any user. How do we improve on BERT4Rec to include multiple modalities? (Lin et al., 2018) also proposes a few efficient encoding schemes for various modalities. A TransR (Lin et al., 2015) for embedding relational information in music data, a Paragraph vector model with a next word prediction task for embedding text data, Variational Autoencoders (Kingma and Welling, 2013)for embedding visual data is suggested.

(Vaswani et al., 2021) suggests another method of generating playlist embeddings from acoustic embeddings and lyrics embeddings of each song in the playlist. Acoustic features are known for their usage in emotion recognition tasks and hence, is very vital in giving enriched music representations. The author highlights the use of Variational Autoencoders (Kingma and Welling, 2013) to generate expanded embeddings of acoustic features as it produces continuous embedding space. Similarly, lyrics also tends to be extremely useful in relaying information about the form of music. SBERT (Reimers and Gurevych, 2019) is then used to extract embeddings for Lyrical features. Finally, the authors note that multiplying attention weight generated from Bi-GRU model similar to (Hidasi et al., 2015) of the music playlist improved the performance of the model.

The novelty in our approach is achieved by combining methods proposed in the above papers, exploring a newer and different architecture for our embeddings, and introducing new modalities to overcome our limited data/computational power.

3 Problem statement

In this paper, we are mainly dealing with music recommendation problem. Music recommendation means that we are predicting songs that the users would like to listen next when users' playlist P, an ordered list of songs, is given. Being able to more accurately predict the songs users want to listen to means being able to more accurately recommend songs to users.

In this work, we propose a Multimodal Music Recommendation System. The main focus points of our project are as follows:

- Collect datasets for Audio Features, Lyrics, Album Art, Artist Biography and Playlist History.
- Extract Embeddings for each of the four modalities using the appropriate model for

the kind of data.

- Create a new Multimodal Approach for Music Recommendation using BERT4Rec as the recommendation system.

- Verify the efficacy of adding multiple modalities to improve upon the current unimodal approaches for music recommendation.

4 Description of solution[1]

We split our model architecture into two logical modules, the pre-training module and the recommendation module. The pre-trained module generates the embeddings for each modality and fuses them to a multimodal embedding space. These embeddings are used to initialize an embedding layer in the recommendation module, which is trained and fine-tuned to learn the weights by predicting the next song in the playlist.

4.1 Pre-training Module

The modalities we utilize are lyrics, audio features, album art, and artist biography. We have a variety of models to generate embeddings for the varied datasets we have. For album art and audio data, we generated embeddings using Variational AutoEncoders. For lyrics and artist biography data, we generated embeddings using sentence transformers (Reimers and Gurevych, 2019) and word embedding techniques. These embeddings, to some extent, represent the features of each modalily that will be used as representations provided to recommendation module.

The overall architecture of the pre-training module is illustrated in Figure 1. We further specify the embedding generation process in more details in Section 4.2.

4.2 Embedding Generation

We explain how we generate each modality's embeddings from their features in the following paragraphs. However, some songs may lack certain modalities' data (See Section 5.1). For songs that didn't have all the modalities, we have multiple possible solutions. Firstly, we could use the average value of all the other embeddings to fill in. Secondly, we could use a linear network to train from the non-missing embeddings, which is the only one with integrity data. Third, we use Expectation Maximization to generate embeddings for missing values. In this work, we use average values for missing data.

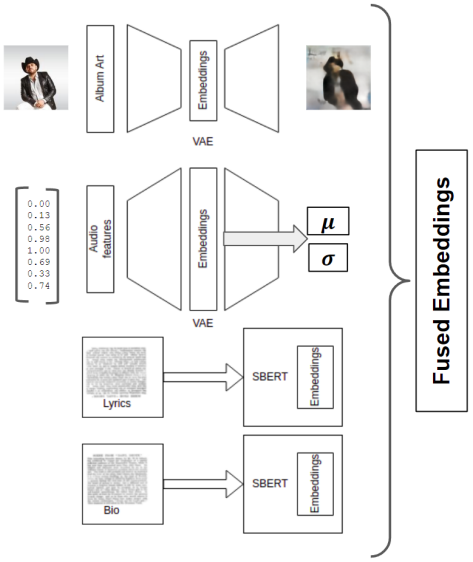


Figure 1: Pre-training Module Architecture

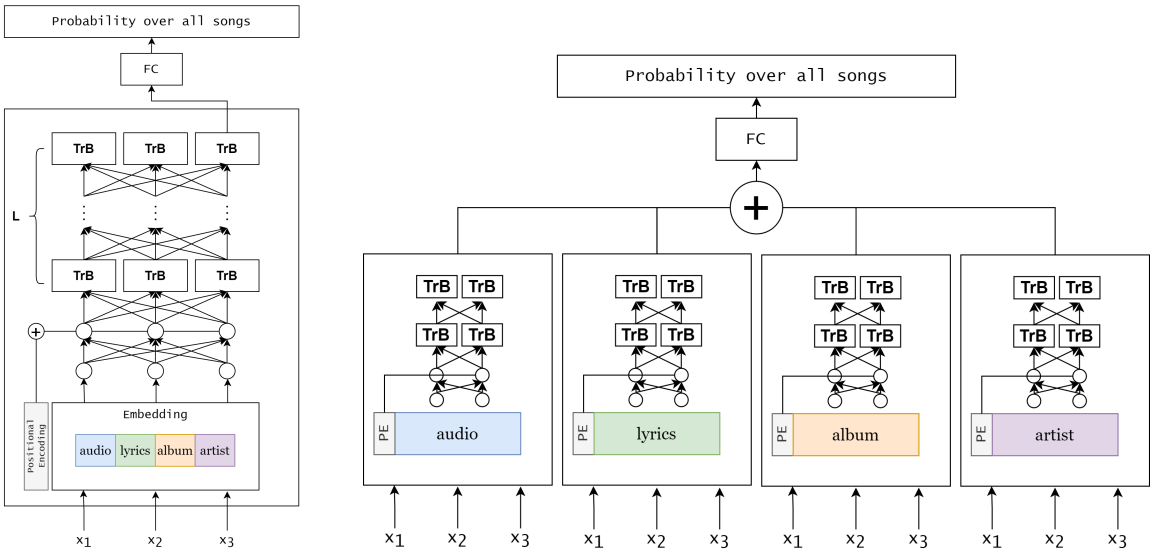
4.2.1 Audio Embeddings

Since the audio features mentioned in Section 5.1.3 were collected as a vector of 11 distinct features rather than a continuous spectrum of audio data, we utilize a Variational Autoencoder (VAE) (Kingma and Welling, 2013) to map these features into a continuous latent space. VAE allows the latent space to capture a meaningful representation of an input. Following a popular procedure (Doersch, 2016), we train a VAE using an unsupervised task by reconstructing an input feature x through an encoder and decoder network. The result from an encoder gives a posterior distribution of a latent z, which we use as an audio embedding of a song. We optimize a VAE with an evidence lower bound (ELBO) objective.

4.2.2 Lyrics Embeddings

Lyrics data are passed through a pre-processing pipeline which first cleared all the artefacts from the Genius API. It was then stripped off all the stop words and low inverse document frequency tokens. A rudimentary spell check was applied on the lyrics dataset to fix some common spelling errors found in lyrics such as missing 'g' at the end of words (Singin' instead of Singing). Embeddings were then generated on this using four different techniques. Two Word embedding tech-

[1] https://github.com/unfortunate-code/Multimodal-Music-Recommendation



(a) Fused modalities

(b) Separated modalities

Figure 2: An illustration of the recommendation module architecture using (a) concatenated embeddings of all modalities and (b) each modality is passed through individual transformers layers

niques, Glove and Word2Vec (300 dimensions) were used to get embeddings of each word which was then averaged to get the embedding for the entire dataset. Two Sentence transformers (Reimers and Gurevych, 2019) techniques, miniLM and mpnet were also used. Lyrics dataset had around 90% of all the MPD songs.

4.2.3 Album Art Embeddings

The first step to generating album art embeddings is pre-processing images. After cropping and resizing images to the same size, all images are converted to an array of $128 \times 128 \times 3$ (three channels for RGB) dimensions, we normalized the array of images. During our experiment, we designed both autoencoder and variational autoencoder (VAE) to generate embeddings. We used MSE loss and visualized reconstructed images to optimize our modal, we extract the latent space as embeddings. We provide three types of embedding to the subsequent recommend experiment, which include the latent space of the autoencoder, the mean value from the latent space of VAE only, and both the mean and variance from the latent space of VAE.

4.2.4 Artist Biography Embeddings

In order to generate the embeddings for textual data (Artist Biography), we use Sentence BERT (Reimers and Gurevych, 2019) rather than BERT. This paper shows that using BERT output layer does not always have the best embeddings to show the similarity between different texts. SBERT is trained to able to distinguish different pieces of text and hence, offers a better word to represent textual data. We extract embeddings using two sentence transformer models: miniLM (size of embedding: 384) and MP-NET (size of embedding: 768). In order to verify the efficacy of each of these approaches, we'll further feed this input into our multimodal recommendation model and utilize the method that proves to give the best results in terms of our different evaluation metrics.

4.3 Recommendation module

Figure 2 illustrates our recommendation module architecture. We used two different architectures for fusing modalities. Figure 2a shows a fused modalities architecture where we concatenated all the modalities before passing to the main recommendation module. Figure 2b shows another approach where each modality was passed through the recommendation module individually and the last layer for each of these is added with weights learned using layer norm.

We used Bert4rec as a sequential recommender system as it showed the best performance and scalability in the baseline experiments. Random initialized embedding layer in the baseline model is replaced with pre-trained embedding from modalities. We train a sequential recommender system while finetuning the embeddings obtained previously. The input to the module is a playlist of

listening history $x_1, x_2, ..., x_n$ where each x_i represents a onehot-encoding of a song. We pass each x to an pre-trained embedding layer to obtain the embedding vector $\hat{z}$. For the fused modalities architecture, $\hat{z}$ is a concatenation of each modality embeddings, while for the separated modalities architecture, $\hat{z}$ only consists of individual modality's embedding vector. Next, we pass the embedding vectors to linear layers to reduce the dimension before adding with a learnable positional encoding vector. Then, the sum vectors are passed to the multi-head transformers to obtain a final representations. Finally, they are fed into a fully connected layer to calculate the probability distribution over all possible songs to recommend to a user.

5 Experiment settings

5.1 Dataset

5.1.1 Playlist Dataset

To run the baseline models with the music playlist of the users, 4 different datasets are used: Million Playlist Dataset Small version (MPD-small), Million Playlist Dataset Frequent song version (MPD-freq), Million Musical Tweets Dataset (MMTD) (Hauger et al., 2013), Million Playlist Dataset Large version (MPD-Large). MPD-small dataset is MPD challenge dataset containing 8000 playlists. The second set, the MPD-freq dataset is a filtered version of MPD-small. Songs appeared less than 3 times and playlists with lengths less than 3 are filtered out. The MMTD is a log data of Twitter users' music listen history. Duplicate songs in the playlists are filtered out. Finally, the MPD-Large dataset is a full version of the MPD dataset. It consists of 1 million playlists and 2 million songs. Due to the sparsity of the dataset, songs that appeared less than 100 times are filtered out. From each playlist in the dataset, the second last song, and last song are assigned to a valid set and test set for each. Table 1 summarizes each dataset.

Data	#user	#songs	Sparsity(%)
MPD-small	8,000	66,243	99.932
MPD-freq	7,543	17,835	99.83
MMTD	46,803	15,984	99.95
MPD-Large	970,877	70,229	99.922

Table 1: Playlist Datasets used in this project

5.1.2 Lyrics Data

Lyrics data was extracted from the genius API to for every song in the MPD dataset. Few different techniques were used to crawl lyrics. Lyrics for over 90% of the MPD songs were obtained this way.

5.1.3 Audio Feature Data

For audio features, since obtaining the raw audio spectrum would require substantial data storage, we opt to use the Spotify API to acquire each song's 11 audio characteristics. These are acousticness, danceability, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, and valence.

5.1.4 Artist Biography Data

Artist biography is extracted using a combination of data from Wikipedia and Million Song Dataset (MSD-A). Firstly, we obtained a list of unique artists (13277) using the playlist dataset that we obtained in section 5.1.1. We then merge MSD-A along with this list to obtain the artist biographies. However, using this method we observed a huge loss in the intersection of the both. Hence, for the missing artist biographies, we scrapped them from Wikipedia using their name. Using this method, we were able to obtain over 94% of the artist biographies.

5.1.5 Album art data

The image URL for album art was obtained by scraping the Genius API which was then used to fetch the image data. Album art for over 90% of the MPD songs were obtained this way.

The summary of data collection for each modalities is shown in Table 2.

	#entries	#collected	(Percentage%)
Audio	70,229	70,229	100
Lyrics	70,229	63,627	90.6
Artist Bio	70,229	65,514	93.2
Album Art	70,229	64,507	91.9

Table 2: Each modality's data collection ratio

5.2 Evaluation Procedure/Metrics

We use two ranking-based metrics: Recall@R and the truncated normalized discounted cumulative gain(NDCG@R). For each user, both metrics compare the predicted rank of the held-out items with their true rank. While Recall@R considers all items

ranked within the first R to be equally important, NDCG@R uses a monotonically increasing discount to emphasize the importance of higher ranks versus lower ones.

Formally, we define $m(r)$ as the item at rank r, and I_u as the held-out items of user feedback for user u.

$$Recall@R = \frac{\sum_{r=1}^{R} \mathbb{1}[(m(r) \in I_u)]}{min(R, |I_u|))}, \quad (1)$$

where $\mathbb{1}[\cdot]$ denotes the indicator function.

$$DCG@R = \sum_{r=1}^{R} \frac{2^{\mathbb{1}[(m(r) \in I_u)]} - 1}{log(r+1)} \quad (2)$$

NDCG@R is the DCG@R linearly normalized to [0,1] after dividing by the best possible DCG@R, where all the held-out items are ranked at the top.

5.3 Baseline models

Four baseline models were used. TransRec (He et al., 2017), SASRec (Kang and McAuley, 2018b), BERT4Rec (Sun et al., 2019b), and EASE (Steck, 2019) were used. First of all, TransRec is a sequential model that embeds items into a 'transition space' where users are modeled as translation vectors operating on item sequences. Secondly, SASRec is a sequential model that allows the model to capture long-term semantics using an attention mechanism. Thirdly, BERT4Rec is a model which employs deep bidirectional self-attention to model user behavior sequences. Finally, EASE is a linear item to item collaborative filtering model which calculates a similarity matrix in a closed-form based on the user to item rating matrix.

5.4 Multimodal Recommender System Model

We used BERT4Rec to develop Multimodal Recommender System Model. In baseline experiment, we used 4 different datasets to do experiment but we only used MPD-Large dataset in the main experiment as we collected multimodal data for MPD-Large dataset. Various parameters and embedding settings were explored to get the best result.

6 Results

6.1 Baseline Experiment Result

Table 3 summarizes the baseline results. Among the baseline unimodal models, generally, BERT4Rec model showed the best performance on all evaluation metrics. EASE model showed the best performance in some cases. However, it could not be used on large datasets due to scalability issues. EASE model approximates the similarity matrix for all users and all songs at once. As data grows, the size of the matrix increases rapidly and this causes memory issues.

Model	MPD-small	MPD-freq	MMTD	MPD-Large
		Recall@20		
TransRec	0.0127	0.0147	0.0373	0.0120
SASRec	0.0100	0.0015	0.0357	0.0810
BERT4Rec	**0.0387**	0.0445	0.0440	**0.1060**
EASE	-	**0.1465**	**0.0860**	-
		Recall@500		
TransRec	0.1454	0.1757	0.2592	0.1348
SASRec	0.0057	0.0327	0.2620	0.1075
BERT4Rec	**0.1557**	0.3263	0.2463	**0.4834**
EASE	-	**0.5166**	0.3196	-
		NDCG@20		
TransRec	0.0048	0.0055	0.0155	0.0049
SASRec	0.0033	0.0005	0.0148	0.0312
BERT4Rec	**0.0148**	0.0163	0.0178	**0.0445**
EASE	-	**0.0643**	**0.0375**	-
		NDCG@500		
TransRec	0.1454	0.1757	0.2592	0.1348
SASRec	0.0057	0.0327	0.2620	**0.1075**
BERT4Rec	**0.1557**	0.3263	0.2463	0.1054
EASE	-	**0.5166**	**0.3196**	-

Table 3: Baselines Experiment Results

At first, Recall@20 and NDCG@20 are used as the evaluation metric. However, the result of the models was too low. Therefore, a higher @k number, 500 is used. NDCG@500 was an evaluation metric for Million Playlist Challenge. With a large @k value, it got much easier to check the result of the models.

6.2 Results of Mulitimodal Recommender System

Table 4 contains the results of our experiments using frozen embeddings[2]. Each section represents our baseline models, single modalities models, and 2 modalities concatenated models respectively. The result shows that unimodal baseline model actually did better than adding any modalities. This means that freezing embedding performs worse than random learnable embedding. Based on the single modality results, we decided to prematurely conclude the experiment and move on to experiments that allow embeddings to be fine-tuned.

Table 5 contains results for the experiments with learnable embeddings. Each section from top to bottom are: baselines, single modalities run under different configurations, two modalities concatenated, three modalities concatenated, and finally

[2]frozen embeddings are fixed and not fine-tuned

Embedding Frozen	Recall@500	NDCG@500	Recall@20	NDCG@20	Embedding dim
BERT4rec	**0.4834**	**0.1054**	**0.106**	**0.0445**	512
BERT4rec	0.4774	0.1006	0.1059	0.0455	128
Audio	0.4339	0.0862	0.0851	0.0348	150
Lyrics	0.4327	0.0864	0.0844	0.0352	128
Album Art	0.4644	0.0951	0.0968	0.0408	256
Artist Bio	0.4136	0.0745	0.0689	0.0323	384
Audio + Lyrics(miniLM)	0.4616	0.0959	0.0991	0.0423	256

Table 4: Experiment results with Frozen Embeddings

Model	Recall@500	NDCG@500	Recall@20	NDCG@20	Embedding dim
BERT4rec	0.4834	0.1054	0.106	0.0445	512
BERT4rec	0.4774	0.1006	0.1059	0.0455	128
Audio	0.4943	0.1075	0.1158	0.0512	128
Audio (128μ)	0.5032	**0.1126**	**0.1236**	**0.0561**	128
Lyrics(miniLM)	0.4944	0.1091	0.1185	0.0532	128
Album Art(VAE) 128μ	0.5003	0.1104	0.1126	0.0494	128
Album Art(VAE) 64μ 64σ	0.4964	0.109	0.1125	0.0495	128
Artist Bio (miniLM-PCA) 384 to 128	0.4923	0.107	0.1153	0.051	128
Artist Bio (mpnet-PCA) 768 to 150	**0.5150**	0.1030	0.0742	0.0456	150
Audio + Lyrics (miniLM)	0.4912	0.1083	0.1172	0.0528	256
Audio + Lyrics (glove)	0.4872	0.1076	0.1166	0.052	256
Audio + Artist Bio	0.4825	0.1055	0.1137	0.0508	256
Audio + Album (VAE)	**0.5102**	**0.1163**	**0.1295**	**0.0596**	256
Audio + Album + Artist	0.412	0.0824	0.0822	0.0337	384
Audio + Album + Artist + Lyrics(glove)	0.4181	0.0807	0.0748	0.0303	512
Audio + Album + Artist + Lyrics (Normalized)	0.4305	0.0836	0.0784	0.032	512
Audio + Album + Artist + Lyrics(glove) + PCA	0.4844	0.1133	0.1158	0.0517	300
Audio + Album + Artist + Lyrics (mpnet) + Linear to 300 dim (with regularization)	0.4929	0.0992	0.1003	0.0412	300
Audio + Album + Artist + Lyrics (mpnet) - Individual BERT + regularization + linear	**0.5004**	**0.1091**	**0.1173**	**0.0522**	300

Table 5: Experiment results with Learned Embeddings

all four modalities concatenated with different variations. The first two results are with all the modalities concatenated similar to the previous entries. The 3rd section is with a PCA applied on the modalities after concatenation. For the fourth and fifth entries, the PCA layer is replaced with a feed forward network to reduce the embedding dimensions. These entries also add some regularization such as layer norm, increasing dropout. In the fifth entry, we replace the architecture from Figure 2a to Figure 2b where each modality is passed through a recommendation module separately and finally added with weights learned with a LayerNorm layer. Fine-tuning embeddings shows an improvement over the baseline model, with the best overall model utilizing two modalities, audio and album art. We discuss a more in-depth analysis of the results in Section 7.

7 Discussion

7.1 Frozen v/s Learned Embeddings

We can make a number of inferences from the results. It is clear that freezing embeddings leads to bad performance with our model performing worse than the baseline. Unfreezing the embeddings leads to significantly better results with our model outperforming the baseline every time. The reason for this would just be that freezing the embeddings would prevent any fine-tuning on the recommendation task.

7.2 Which Single Modality Model performs the best?

We first tried experimenting with single modalities to compare the efficacy of different embedding techniques for each modality. An interesting observation here is that for lyrics, glove based embeddings performed better than one of the sentence transformers and performed almost as well as the other sentence transformer embedding. For audio features, embeddings were generated using VAE. We experimented with a few ways of using the results of this encoding. We tried taking just the mean vector for each song and also tried taking a fixed random representation of the audio vector. It is worth noting that the latter performed better than the former. For album art, we generated embeddings with autoencoder and VAE. As expected, VAE embeddings outperformed autoencoder embeddings in every pre-trained metric. For multimodal embeddings, we concatenated embeddings for each modality and passed this through recommendation module. For 2 modalities cases,

the combination of album art and audio features outperformed other same dimensional embeddings.

We found that models trained on album art or audio features perform better than models trained on any other single modalities with the same dimensions. This could be because users tend to listen to songs of the same genre (correlated to the audio features) or similar albums.

7.3 Is the dimension of the embedding important?

In order to combine the modalities and feed them into the recommender system, we need to reduce and equate the dimension of the combined modalities. Each of the modalities obtained from the pretraining task were reduced to a smaller and equal embedding size with PCA. We experimented with embeddings of sizes 128 and 150. We found that embeddings of size 150 performed significantly better than the ones of size 128. For instance, artist biography of size 150 performed significantly better in term of Recall@500 than any other embeddings of size 128 while artist biography of size 128 was our worst performing modality. Similarly, for the 2 modalities case, an embedding of size 300 (2 150-dimensional embeddings concatenated) performed much better than a 256 dimensional modality. It is also worth noting that, for album art, embeddings of 512 dimensions gave much better reconstruction than embeddings of size 128, however, they performed worse than 128 dimensional embeddings in recommendation tasks after PCA.

7.4 Does adding more modalities cause overfitting?

Another observation is that the models trained on all the 4 modalities combined performed worse than any single modality model or 2-modalities models. We hypothesized that this could be because of overfitting (Wang et al., 2020). We tried several approaches to tackle this issue. We tried reducing the 512 dimensional embedding obtained after concatenation to 128 dimensions or 300 dimensions using PCA. This gave significantly better results than the 512 dimensional embeddings with the 300 dimensional embedding giving better results than 128 dimensional results consistent with our previous findings. We also tried reducing the dimensions using a linear layer to allow the model to learn the right combination of modalities through recommendation rather than a context-blind PCA. In this case, we concatenated the full embeddings for each modality without any PCA and passed it through a feed forward network after the embedding layer. This performed worse than the PCA approach while still performing better than the original 512 dimensional embeddings. Further, we improved on the feed forward model by adding layer norm, initialization and increasing dropout. All this regularization paid off with this model performing better than any of the other fully concatenated embedding models we have. But, it is still unfortunate that this still didn't perform better than the best single or 2-modality models we have.

7.5 Other approaches for multimodal fusion

Inspired by (Vaswani et al., 2021), we tried a different architecture where we passed each of the embeddings separately to the BERT4Rec model and added the final layer of each weighed by a LayerNorm layer. This while outperforming all the other 4-modality models we have, still didn't outperform the best single and 2-modality models. Further research is required to find the right combination of modalities and parameters.

7.6 Future work

Our initial approach to the pretraining tasks involved training embeddings using a supervised genre classification task. This proved to be a non-trivial task. We tried several methods for extracting genres but none of the approaches had a good intersection with the MPD dataset. This led us to pivot to self-supervised learning tasks. We believe this approach is still worth considering and further exploration is needed to get the genres data for each of the songs. For album art, we have only tried VAE and AutoEncoders so far. Other architectures such as ResNet, Visual Transformers could be explored too. Furthermore, as described above, further research is required to tackle the overfitting problem for larger number of modalities. Also, it is worth exploring other modalities such as audio frequencies, song reviews to further enrich the recommendations. Currently, we have preliminary attempts to generate missing embeddings. It shows a slight improvement in results. One more research avenue could be to explore other methods to generate embeddings for missing songs in each modality.

References

Marko Balabanović and Yoav Shoham. 1997. Fab: content-based, collaborative recommendation. *Com-*

munications of the ACM, 40(3):66–72.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70.

David Hauger, Markus Schedl, Andrej Košir, and Marko Tkalcic. 2013. The million musical tweets dataset: What can we learn from microblogs. In *Proc. ISMIR*, pages 189–194.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 161–169.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Wang-Cheng Kang and Julian McAuley. 2018a. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE.

Wang-Cheng Kang and Julian McAuley. 2018b. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Jin Ha Lee and J Stephen Downie. 2004. Survey of music information needs, uses, and seeking behaviours: preliminary findings. In *ISMIR*, volume 2004, page 5th. Citeseer.

Qika Lin, Yaoqiang Niu, Yifan Zhu, Hao Lu, Keith Zvikomborero Mushonga, and Zhendong Niu. 2018. Heterogeneous knowledge-based attentive neural networks for short-term music recommendations. *IEEE Access*, 6:58990–59000.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Sergio Oramas, Francesco Barbieri, Oriol Nieto Caballero, and Xavier Serra. 2018. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*

Sergio Oramas, Luis Espinosa-Anke, Aonghus Lawlor, et al. 2016. Exploring customer reviews for music genre classification and evolutionary studies. In *The 17th International Society for Music Information Retrieval Conference (ISMIR 2016), New York City, United States of America, 7-11 August 2016*.

Sergio Oramas, Oriol Nieto, Mohamed Sordo, and Xavier Serra. 2017. A deep multimodal approach for cold-start music recommendation. In *Proceedings of the 2nd workshop on deep learning for recommender systems*, pages 32–37.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Noveen Sachdeva, Kartik Gupta, and Vikram Pudi. 2018. Attentive neural architecture incorporating song features for music recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 417–421, New York, NY, USA. Association for Computing Machinery.

Markus Schedl, Emilia Gómez Gutiérrez, and Julián Urbano. 2014. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval. 2014 Sept 12; 8 (2-3): 127-261.*

Harald Steck. 2019. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pages 3251–3257.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019a. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019b. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Kunal Vaswani, Yudhik Agrawal, and Vinoo Alluri. 2021. Multimodal fusion based attentive networks for sequential music recommendation. In *2021 IEEE Seventh International Conference on Multimedia Big Data (BigMM)*, pages 25–32. IEEE.

Weiyao Wang, Du Tran, and Matt Feiszli. 2020. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705.

A Appendices

Following Tables 6 and 7 contain the entire list of experiments we've conducted.

Embedding Frozen	Recall@500	NDCG@500	Recall@20	NDCG@20	Embedding dim
BERT4Rec	0.4834	0.1054	0.106	0.0445	512
BERT4Rec	0.4774	0.1006	0.1059	0.0455	128
Audio	0.4339	0.0862	0.0851	0.0348	150
Lyrics(glove)	0.4327	0.0864	0.0844	0.0352	128
Lyrics(miniLM)	0.4327	0.0864	0.0844	0.0352	128
Album Art(AE)	0.4351	0.083	0.0764	0.0304	128
Album Art	0.4644	0.0951	0.0968	0.0408	256
Artist Bio	0.4136	0.0745	0.0689	0.0323	384
Audio + Lyrics(miniLM)	0.4616	0.0959	0.0991	0.0423	256

Table 6: Experiment with Frozen Embeddings

Model	Recall@500	NDCG@500	Recall@20	NDCG@20	embedding$_dim$
BERT4Rec	0.4834	0.1054	0.106	0.0445	512
BERT4Rec	0.4774	0.1006	0.1059	0.0455	128
Audio	0.4943	0.1075	0.1158	0.0512	128
Audio 128μ	0.5032	0.1126	0.1236	0.0561	128
Lyrics(glove)	0.4965	0.1097	0.1195	0.0536	128
Lyrics(miniLM)	0.4944	0.1091	0.1185	0.0532	128
Lyrics(w2v)	0.4901	0.1065	0.1151	0.0501	128
Lyrics(mpnet)	0.3427	0.0611	0.0487	0.0188	128
Lyrics(mpnet)	0.4936	0.1101	0.1207	0.0188	128
Album Art(VAE) 128μ	0.5003	0.1104	0.1126	0.0494	128
Album Art(VAE) 64μ64σ	0.4964	0.109	0.1125	0.0495	128
Album Art(VAE) 128μ	0.4956	0.108	0.1198	0.0536	128
Album Art(AE)	0.4764	0.0945	0.089	0.0361	128
Album Art(PCA) 256 to 128	0.4683	0.0916	0.08	0.0323	128
Album Art(PCA) 512 to 128	0.4596	0.089	0.0715	0.0285	128
Artist Bio (PCA) 768 to 128	0.4748	0.1001	0.1055	0.0453	128
Artist Bio (PCA) 384 to 128	0.4923	0.107	0.1153	0.051	128
Artist Bio (PCA)	0.515	0.103	0.0742	0.0456	150
Audio + Lyrics(miniLM)	0.4912	0.1083	0.1172	0.0528	256
Audio + Lyrics(glove)	0.4872	0.1076	0.1166	0.052	256
Audio + Artist Bio	0.4825	0.1055	0.1137	0.0508	256
Audio + Album(VAE)	0.5102	0.1163	0.1295	0.0596	256
Audio + Album + Artist	0.412	0.0824	0.0822	0.0337	384
Audio + Album + Artist + Lyrics(glove)	0.4181	0.0807	0.0748	0.0303	512
Audio + Album + Artist + Lyrics(glove) (normalize before concat)	0.3604	0.065	0.0512	0.0202	512
Audio + Album + Artist + Lyrics(glove) (normalize after concat)	0.3787	0.069	0.0587	0.023	512
Audio + Album + Artist + Lyrics(glove) (normalized after concat, n$_layer = 1$)	0.4305	0.0836	0.0784	0.032	512
Audio + Album + Artist + Lyrics(glove) + pca to 128dim	0.4777	0.0941	0.0914	0.0371	128
Audio + Album + Artist + Lyrics(glove) + pca to 300 dim	0.4844	0.1133	0.1158	0.0517	300
Audio + Album + Artist + Lyrics(mpnet) + linear to 300 dim	0.466	0.0929	0.0909	0.0377	300
Audio + Album + Artist + Lyrics(mpnet) + linear to 300 dim (with regularization)	0.4929	0.0992	0.1003	0.0412	300
Audio + Album + Artist + Lyrics(mpnet) individual BERT + regulairization + linear	0.5004	0.1091	0.1173	0.0522	300

Table 7: Experiment with Learned Embeddings