Music Recommender System Conditioned on Lyrics and Metadata

Eric Bezzam
eric.bezzam@epfl.ch

Vidit Vidit
vidit.vidit@epfl.ch

Abstract

We investigate the integration of lyrics into a music recommender system. From a learned lyrical embedding space, we retrieve similar songs and are also able to condition on metadata such as genre and various acoustic features. The lyrical embedding space is trained by finetuning a language model on various downstream tasks, e.g. genre classification, and determining clusters within this space. These clusters are then used to look for similar songs. Our code is available at https://github.com/ebezzam/lyrics-mir .

1 Introduction

The majority of music information retrieval (MIR) systems such as Spotify, Deezer, and Shazam rely on acoustic features and/or basic metadata (artist, album, genre) in order to query a database. In this project, we are interested in exploring how lyrics/text can be used as a complement to such tasks. There are a few works that combine melodies and lyrics in order to query a database (Wang et al., 2009; Suzuki et al., 2006) (similar to Shazam (Gikaru, 2020)). Spotify organized a challenge (Chen et al., 2018) for playlist completion based on metadata. This is similar to the MIR task we would like to consider but the metadata provided in their dataset is very sparse and does not contain any text or audio features, but rather metadata on how users created playlists (e.g. playlist title, track titles, artists, duration). To the best of our knowledge, we could not find prior work on recommendations based on lyrics and metadata.

Given a song and a conditioning for the recommendation, e.g. "We are the champions" by Queen in the style of hip-hop, we envision the following workflow for our recommender system:

1. Compute embedding of the lyrics.
2. Identify the closest cluster of similar songs it belongs to. These clusters are determined from our training dataset.
3. Depending on the metadata for conditioning (e.g. genre, release year, more danceable), we subset the songs in the cluster.
4. Return the song(s) in the subset with the smallest embedding distance(s).

The various tasks involved in the above method are described in this paper, which is organized as follows. In Section 2, we describe the dataset, provide some visualizations, and explain how we prepare it for our task. In Section 3, we describe three approaches that we consider for training an encoder specifically for lyrics. This encoder is needed both for computing the embedding of a given song and for determining the clusters from our training dataset. In Section 4, we present results based on clustering metrics that help us decide which training strategy may have led to a suitable encoder. In Section 5, we describe in more detail the proposed recommender system, and we conclude in Section 6.

2 Dataset

There is a lack of publicly available datasets for music lyrics due to copyright issues, therefore we restrict our work on the MusicOSet (Silva et al., 2019)[1]. MusicOSet consists of $20'405$ songs with their lyrics, metadata, and various acoustic features. The dataset is put together from various sources:

1. Genius.com for lyrics through the *LyricsGenius* library.[2]
2. Spotify for metadata (artist(s), genre(s), year, and popularity) and acoustic features through

[1] https://marianaossilva.github.io/DSW2019/

[2] https://github.com/johnwmillr/LyricsGenius

the *Spotipy* library.[3]
3. Wikipedia for additional metadata.[4]

From the twelve acoustic features provided by Spotify, we more closely consider:

- Valence: reflects the positiveness of the song, solely from the audio.
- Danceability: describes how suitable a track is for dancing.
- Energy: represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.
- Tempo: the speed or pace of a given piece.
- Liveness: reflects presence of live audience.

Other more popular, larger datasets such as the Million Song Dataset (MSD) (Bertin-Mahieux et al., 2011) ($1'000'000$ songs and metadata) and Free Music Archive (FMA) (Defferrard et al., 2016) ($106'574$ songs and metadata) do not contain full lyrics (but rather a bag of words) or contain many unknown songs, and not as rich metadata as MusicOSet. Moreoever, MusicOSet is a more manageable size for the scope and timeframe of this project.

2.1 Preprocessing

Since there is little prior work with this new dataset and none on our desired task, some time was spent for data exploration and preprocessing. We remove bad entries and consolidate the different lyrics and metadata files (five in total) into a single *Pandas* dataframe. As shown in Figure 1, the lyrics contain delimiters and annotations which need to be removed for a clean text. Additionally, we reduce the number of genres to 8 (shown in Table 2) from 461 after removing small genres and combining similar ones. Note that combining genres was by no means a trivial task; as there are several subgenres and merging them would mean, all lyrics convey a similar theme. We combine the genres according to our best judgement of what can be close to each other.

After these steps, we are left with $15'863$ songs.

2.2 Data Exploration

One of our goals is to improve the lyric embeddings by training the network on metadata prediction tasks, as described later in Section 3. In order to identify suitable metadata and acoustic features to work with, we visualize their distributions (after preprocessing).[5] A description of each feature can be found on the MusicOSet site, as well as a correlation analysis.[6]

Genre	No. of Songs
Rock	3476
Pop	2774
Country	2412
Soul/Disco	2339
Dance pop	2082
Hip-Hop/Rap	1472
RnB	972
Acoustic/Folk	336

Table 1: Final genres and their song distribution

Several of the metadata are strongly skewed/unbalanced, such as explicit, mode, acousticness, instrumentalness, liveness, and speechiness, making them ill-suited for prediction/classification tasks. Moreover, it intuitively seems unlikely to be able to predict some of the acoustic features, such as loudness and tempo, from solely the lyrics. Nevertheless, we hypothesize that certain features such as valence and danceability could be correlated with the lyrics, as artists tend to write music which reflect the mood of the lyrics (and vice-versa).

Finally, we want to verify that the embeddings of two different kind of songs are "far away" from each other. As a sanity check, we use an MPNet (Song et al., 2020) model[7] to obtain embeddings for the lyrics and use the cosine similarity as a distance metric:

$$\text{cosine similarity} = \frac{\sum_{n=1}^{N} A_n B_n}{\sqrt{\sum_{n=1}^{N} A_n^2}\sqrt{\sum_{n=1}^{N} B_n^2}}, \tag{1}$$

where A and B are length-$N = 768$ embedding vectors for the MPNet model. Cosine *distance*, as we will use later on for clustering, is simple 1 minus the cosine similarity. Figure 2 shows that the cosine similarity of embeddings for similar songs is higher than the score for different kind of songs

[3] https://spotipy.readthedocs.io
[4] https://wikipedia.readthedocs.io
[5] See this notebook: https://github.com/ebezzam/lyrics-mir/blob/main/notebooks/1_dataset_preparation.ipynb
[6] https://marianaossilva.github.io/DSW2019/assets/data/acoustic_features.html#correlation-analysis
[7] HuggingFace model card: https://huggingface.co/sentence-transformers/all-mpnet-base-v2

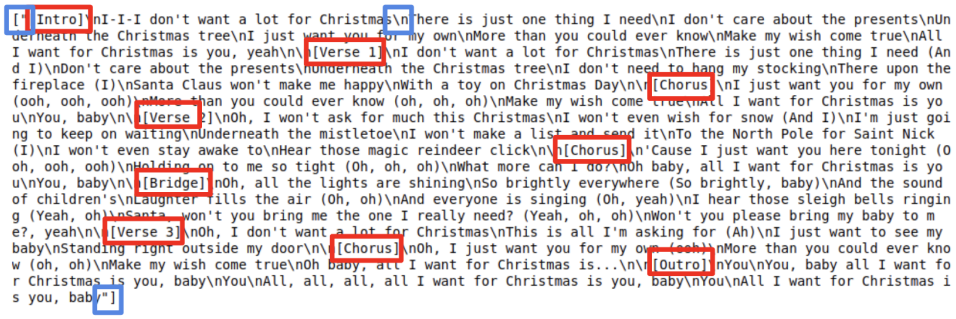


Figure 1: For each lyric, there are different delimiters (blue) and annotations (red) in the text. As a preprocessing step, we remove both of them.

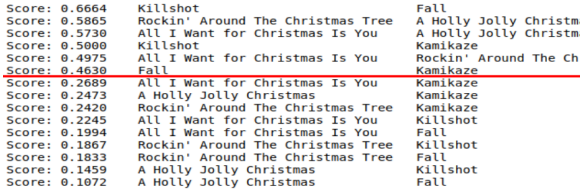


Figure 2: Cosine distance study between Christmas and Eminem songs. The similarity scores are low when we compare embedding of Christmas and Eminem songs (below red line, drop of 0.2), as opposed to when the songs are of similar kind (above the red line).

(Christmas and Eminem in this example). This suggests that there exists a semantic grouping of different lyrics which we can exploit for our goal.

2.3 Train, Validation, Test Split

We split the $15'863$ songs into the following train, validation, and test sets, maintaining the same genre distribution across the different splits.

Split	No. of Songs	Percentage
Train	$11'104$	70%
Validation	$2'379$	15%
Test	$2'380$	15%

Table 2: Train, validation, test split.

3 Computing lyrics embeddings

Our task is to recommend a song from the database which matches the query song in lyrics and metadata criteria. One of the shortcomings of our dataset is that there is no ground-truth available for this specific task. Given the time-frame of the project, it was difficult to curate a test set with a proper recommendation task ground-truth. Hence, we formulate it as a clustering problem. In other words, with our dataset we determine clusters that correspond to lyrically similar songs, by applying an agglomerative clustering method (Day and Edelsbrunner, 1984) in the embedding space. An advantage of this approach is that the computational load of determining similar song(s) is reduced as we only have to search within the cluster (after determining the closest cluster).

One of our objectives for the recommender system is to train an encoder that can produce rich embeddings of song lyrics. In Figure 2, we see how an MPNet model could differentiate Eminem and Christmas songs within its embedding space, even through this model was not trained on song lyrics. This model will serve as our baseline encoder. During our investigation, we also considered a distilled RoBERTa (DistilRoBERTa) model (Liu et al., 2019)[8] and a MiniLM 12-layer model (Wang et al., 2020)[9], as they provided lightweight options for faster iteration. We ultimately stuck with the MPNet model as the clustering metrics of the DistilRoBERTa model did not show much improvement, as will be shown in Section 4. Table 3 offers a comparison of the three models.

Model	Size	Speed
all-mpnet-base-v2	418 MB	2800
all-distilroberta-v1	292 MB	4000
all-MiniLM-L12-v2	118 MB	7500

Table 3: TModel characteristics as reported in https://www.sbert.net/docs/pretrained_models.html. Speed is given by sentence/second on a V100 GPU.

[8] HuggingFace model card: https://huggingface.co/sentence-transformers/all-distilroberta-v1

[9] HuggingFace model card: https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

In the following subsections, we present three approaches in order to finetune our base encoder - all-mpnet-base-v2 - for song lyrics:

1. Finetuning on a single downstream task, e.g. genre classification or valence prediction.
2. Multitask finetuning, e.g. genre classification *and* valence prediction.
3. A self-supervised approach which splits the lyrics in half and tries to bring their embeddings closer.

Another advantage of introducing this clustering step is that we can use quantitative metrics based on inter- and intra-cluster distances in order to quantify the performance of clustering after a particular encoder. These metrics for the finetuned models will be presented in Section 4.

3.1 Finetuning with metadata prediction downstream task

For this approach, we use an MPNet (Song et al., 2020) model[10] as an encoder layer and add a fully connected prediction/classification layer.

Unless mentioned otherwise, we freeze the encoder layers and train with an Adam optimizer with weight decay of 0.01 and a large learning rate, namely $2\mathrm{e}{-3}$, in order to train solely the prediction/classification layer. The goal is to identify which downstream tasks could then be used to finetune the encoder by unfreezing those layers. We select the model at the epoch which produced the largest accuracy on the validation set. All results are reported on the test set.

3.1.1 Genre classification

After training for five epochs with the base encoder frozen, we unfreeze the base encoder layers and train for five more epochs with a smaller learning rate, namely $2\mathrm{e}{-5}$, to avoid over-fitting on the training set. The resulting confusion matrix can be found in Figure 3.

The model has a tough time correctly classifying acoustic/folk, pop, and r&b. This is understandable as acoustic/folk is often confused with rock, and both genres can be quite similar. Moreover, there are very few acoustic/folk examples in the entire dataset (336) compared to rock examples (3476). Pop is a quite generic genre, and can span multiple genres. Whereas r&b is often confused with dance pop and soul/disco with which it shares common traits.

[10]HuggingFace model card: https://huggingface.co/sentence-transformers/all-mpnet-base-v2

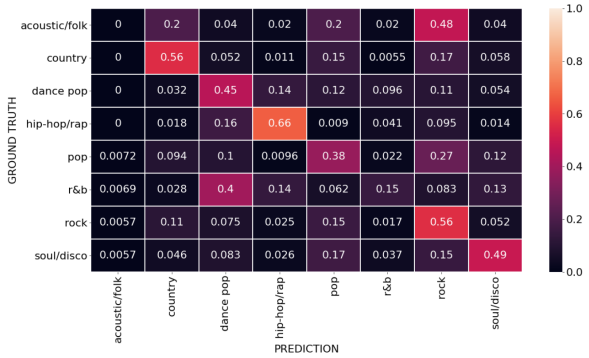


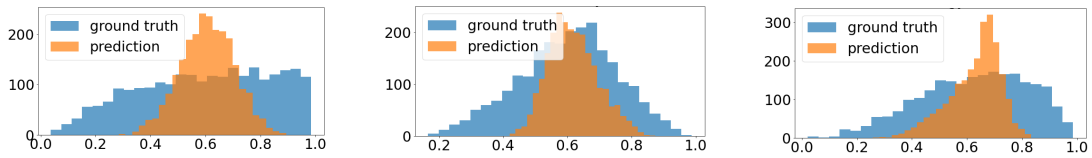
Figure 3: Genre classification confusion matrix after finetuning on an MPNet model.

3.1.2 Acoustic feature prediction

Initially, we framed this task as a regression problem with an l_2-loss function where the network predicts a value in the range of the acoustic feature, e.g. $[0, 1]$ for the valence score. Figure 4 highlights the issue with the l_2 formulation. As the data is not normally distributed, the regressor is not able to fit the training data properly. Therefore, we switch the task to classification by thresholding the values at the median of each feature, so that we also get a balanced dataset. Hence, we predict whether the song belongs to an extremity of the feature or not, e.g. high or low valence. As our objective is to obtain an encoder that creates richer embeddings for lyrics, we can afford losing the granularity in the feature scores.

Figure 5 shows different acoustic feature classification results. We keep the encoders frozen at first to test if the network can predict reliably, as it easy to overfit once the encoder is unfrozen. The features for which the prediction works well are valence, danceability and energy. This can be expected as the lyrics may have a positive message, and therefore be correlated with valence. Similarly, the content of lyrics may be about e.g. partying, which could be why danceability and energy work rather well. Unsurprisingly, we obtain poor performance on features that are very audio-centric: liveness, loudness and tempo.

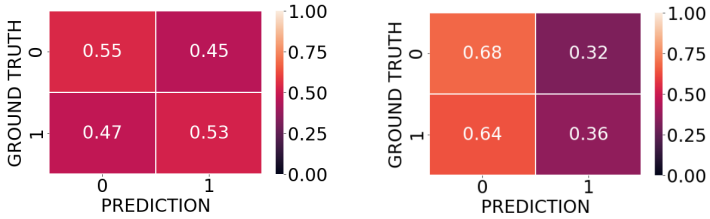
With the acoustic features that can be predicted reasonably well (valence, danceability, and energy), we unfreeze the encoder layers and train for a few epochs with a small learning rate of $2\mathrm{e}{-5}$, to avoid over-fitting on the training set.



(a) Valence (b) Danceability (c) Energy

Figure 4: Mismatch in the prediction vs groundtruth distribution when framed as a regression task.

(a) Valence (b) Danceability (c) Energy



(d) Tempo (e) Liveness

Figure 5: Confusion matrices for binary classification of different acoustic features.

3.2 Multitask finetuning

With the tasks that can be performed decently well (genre, valence, danceability, and energy) we try multitask training. This feature is not supported by HuggingFace's Trainer so we had to build some custom code for this;[11] however this proved to be quite complicated. The training code could run, but we quickly ran out of memory and were not able to save/load checkpoints correctly.

Therefore, we approach multitask by finetuning on multiple tasks sequentially, i.e. in the following order: genre, valence, energy, danceability, and finally genre again.[12] Except for the first genre finetuning where we take the model from Section 3.1.1, we train for five epochs with the encoder frozen with Adam optimization with weight decay of 0.01 and a learning rate of 2e−3 to train the classifier layer, then three epochs with the encoder unfrozen with Adam optimization with weight decay of 0.01 and a learning rate of 2e−6 to finetune the encoder. The order valence, energy, and danceability was chosen in decreasing order of the individual task's clustering scores, as can be seen in Section 4. We finished off with genre finetuning as valence/energy/danceability finetuning tends to promote two clusters (as can be seen in Section 4); this tendency to two classes is probably because the downstream task is binary classification.

3.3 Self Supervised Training

As a final approach for finetuning a model for song lyrics, we employ a SimSiam-based (Chen and He, 2021) training strategy as shown in Figure 6. We split our lyrics for a given song into two halves and try to bring their embeddings closer. In this approach, the network does not learn a trivial solution because *stop-grad* restricts the encoder update with respect to one half of the lyrics. The advantage of this approach compared to other self-supervised methods is that there is no need for negative examples, which are not available in our case. An l_2-loss is used as a similarity function. We report clustering results with this method in the following section.

[11] https://github.com/ebezzam/lyrics-mir/blob/main/notebooks/3_multitask_simulateneous.ipynb

[12] https://github.com/ebezzam/lyrics-mir/blob/main/notebooks/3_multitask-sequential.ipynb

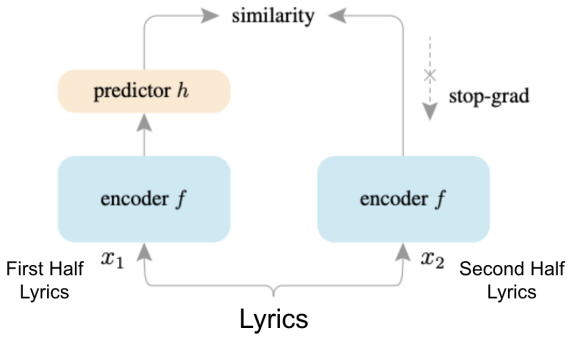


Figure 6: SimSiam-based (Chen and He, 2021) self-supervised training strategy.

4 Results

We use Agglomerative Clustering (Day and Edelsbrunner, 1984) to group embeddings with cosine distance as a metric and maximum linkage. This is a bottom-up approach where initially all the data points are considered as a separate clusters and successively merged together if they fall within a distance threshold. We use three metrics to evaluate the generated clusters:

- **Calinski-Harabaz Index (CH):** measures the ratio of mean inter-clusters dispersion and the intra-cluster dispersion. Higher is better.
- **Davies Bouldin Score (DB):** measure average similarity of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Smaller is better.
- **Silhouette Score (SH):** the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The silhouette coefficient for a sample is $(b - a)/max(a, b)$. The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters.

We compute these metrics for a varying number of clusters, and we are interested in the model and number of clusters which produces the best scores.

In Figure 7 we compare the clustering metrics between a MPNet and a DistilRoBERTa model for a varying number of clusters, and how finetuning on genre classification affects the embeddings clustering. For MPNet, finetuning on genre classification improves all metrics, and we even see a peak for **CH** and **SH** and a small dip for **DB** for 4 clusters. For DistilRoBERTa, there is no clear peak/dip or elbow in the curves and finetuning is actually worse according to the **DB** metric. So we stick with MPNet for the rest of the investigation. For both, the **SH** scores are close to 0 which indicates overlapping cluster, although slightly less so for hte binary classification tasks.

In Figure 8, we compare clustering metrics when finetuning on various downstream tasks, as described in Section 3.1. For valence, danceability, and energy, we notice clearly improved scores for 2 clusters. This is expected as the embeddings have been finetuned on a binary classification task. Once again, the **SH** scores are close to 0 which indicates overlapping clusters.

In Figure 9, we see the effect of multitask finetuning as described in Section 3.2. If we finish on one of the binary classification tasks (green), we end up with a model that performs best with just 2 clusters. Finishing off the fine-tuning with genre classification (red) fixes this. However, there is no improvement from the multitask training (orange still better than red across all metrics).

In Figure 10, we compare metadata finetuning with the SimSiam approach described in Section 3.3. The latter seems to improve the clustering metrics significantly; note that the **CH** and **DB** scores are plotted on a logarithmic scale. 6 clusters seems to be a good tradeoff, as it has a large **CH** score which plateaus and an **SH** score that is still close to 1.

In Figure 11, we can see through a PCA visualization how finetuning on downstream tasks and via SimSiam creates more distinct clusters as compared to the original MPNet model. SimSiam yields more distinct clusters as also reflected by the **SH** scores. However, a closer inspection of the SimSiam clusters reveals something peculiar, namely the scale of the reduced dimension space is very small: $1e{-}7$. Moreover, when we look closer to the embeddings produced by SimSiam they are nearly identical for all songs. Unfortunately, we found this out quite late as the clustering metrics seemed to indicate good performance.

Overall, MPNet model with genre prediction task gives us the better clusters.

5 Proposed Recommender System

We describe our recommendation system in the following pseudocode.

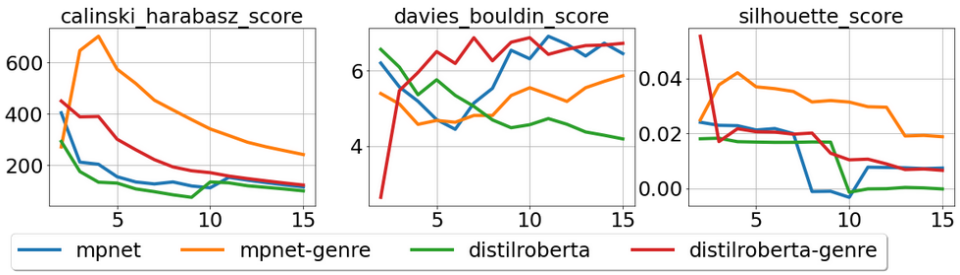


Figure 7: Clustering metric comparison between MPNet and DistilRoBERTa.

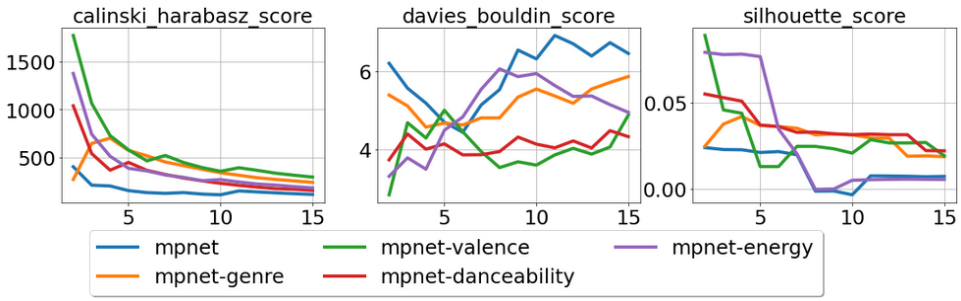


Figure 8: Clustering metric comparison for single task finetuning on MPNet as described in Section 3.1.

```
#D: Dataset of Lyrics
#M: Lyrics Metadata
#E: Trained Encoder
#Q: Query Lyric
#m: Query metadata
def recommendation(L,M,E,Q,m):
    #compute embeddings of lyrics
    L = E(D)
    #get cluster centers and assignments
    c,assgn = AgglomerativeClustering(L)
    #get embeddings for the query lyrics
    qE = E(Q)
    #match the query with centers having
    #max cosine similarity
    mC = argmax(qE.dot(c))
    #set of lyrics their metadata belonging
    #to this cluster
    mL, mM = L[assgn==mC], M[assgn==mC]
    mD = D[assgn==mC]
    #lyrics whose metadata matches
    #with the query metadata
    mL = mL[mM==m]
    mD = mD[mM==m]
    #cosine similarity between query
    #and subset of lyrics
    dist = qE.dot(mL)
    #order highest similarity first
    ind = argsort(-dist)
    #return the ordered lyrics
    reco = mD[ind]
    return reco
```

The finetuned encoders are used to get embeddings of the lyrics in the dataset. These embeddings are then used to generate clusters and corresponding lyrics assignment using Agglomerative clustering with cosine affinity and maximum linkage.[13] In practice, this is precomputed.

Given a new query lyrics/text with a set of metadata, we first find the cluster to which this lyrics belong to and then filter out the lyrics in this cluster which do not have same metadata as query. Finally, we get a list of lyrics which are similar to the query via the cosine similarity metric.

The following notebook can be used to query an arbitrary song (with the LyricsGenius and Spotipy API) and using encodings and clusterings we have trained for this project: https://github.com/ebezzam/lyrics-mir/blob/main/notebooks/5_recommendation_demo.ipynb Figures 12 and 13 show example recommendations with our system.

6 Conclusion

In this work, we propose a simple recommendation system based on lyrics and metadata of songs. We faced different challenges in the process, majority being gathering appropriate data and figuring out an achievable recommendation method given the

[13] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering

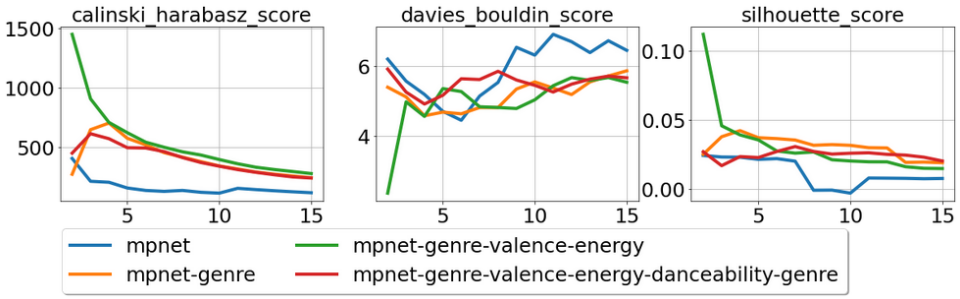


Figure 9: Clustering metric comparison for multitask finetuning on MPNet as described in Section 3.2.

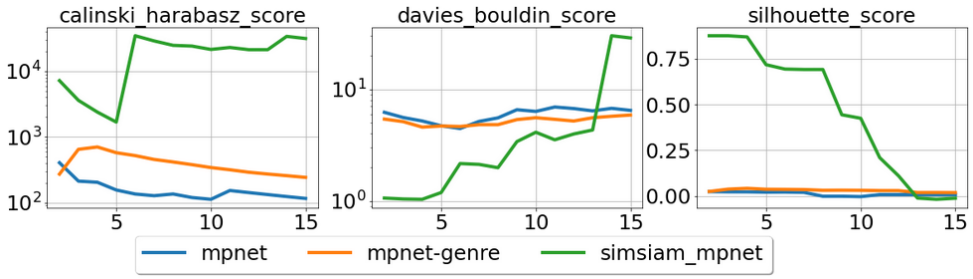
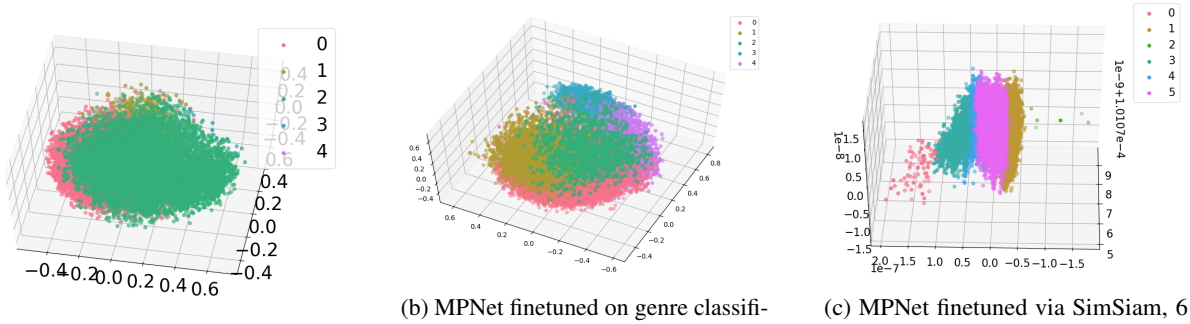


Figure 10: Clustering metric comparison between metadata finetuning and SimSiam as described in Section 3.3. Note the logarithmic scale for the **CH** and **DB** scores.



(a) MPNet original, 5 clusters.

(b) MPNet finetuned on genre classification, 5 clusters.

(c) MPNet finetuned via SimSiam, 6 clusters.

Figure 11: PCA visualization of clusters

time frame of the project. Our best performance is achieved with MPNet on genre classification finetuning task.

There are different directions which we could not fully explore, like self-supervised training. Our SimSiam based method led to inaccurate embeddings for the lyrics, hence formulation of pretraining tasks needs to be explored. In the multitask setup, we faced issue due to low compute resources and could not fully train a model simultaneously on all tasks. We expect multitask setup should be better than single, as we observed rather good accuracy on the individual tasks. Additionally, we should curate a ground truth list of similar songs so that it is better to judge the performance of the method than on proxy clustering metrics.

References

Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset.

Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani. 2018. Recsys challenge 2018: Automatic music playlist continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 527–528.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758.

William HE Day and Herbert Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24.

```
Score: 0.60823315
Genre: hip-hop/rap Artist: Jessie J SongName: Masterpiece
Lyrics: So much pressure, why so loud?. If you don't like my sound you can turn it down. I got a road, and I walk it
alone. Uphill battle, I look good when I climb. I'm ferocious, precocious I get braggadocios. I'm not gonna stop, I
like the view from the top. . You talk that blah blah. That la la, that rah
*****
Score: 0.64838994
Genre: hip-hop/rap Artist: fun. SongName: Some Nights
Lyrics: Some nights, I stay up cashing in my bad luck. Some nights, I call it a draw. Some nights, I wish that my li
ps could build a castle. Some nights, I wish they'd just fall off. But I still wake up, I still see your ghost. Oh,
Lord, I'm still not sure what I stand for, oh. What do I stand for?. What do
*****
Score: 0.6648902
Genre: hip-hop/rap Artist: Nas SongName: I Can
Lyrics: I know I can. Be what I wanna be. If I work hard at it. I'll be where I wanna be. I know I can (I know I ca
n!). Be what I wanna be (be what I wanna be!). If I work hard at it (If I work hard it!). I'll be where I wanna be
(I'll be where I wanna be!). Be, b-boys and girls, listen up. You can be anyth
*****
```

(a) Genre-finetune MPNet.

```
Score: 0.34370732
Genre: hip-hop/rap Artist: Jeezy SongName: Amazing
Lyrics: It's amazing, I'm the reason. Everybody fired up this evening. I'm exhausted, barely breathing. Holding on t
o what I believe in. . No matter what, you'll never take that from me. My reign is as far as your eyes can see, it's
amazing. So amazing, so amazing, so amazing, it's amazing. So amazing, so a
*****
Score: 0.3842497
Genre: hip-hop/rap Artist: Eminem SongName: Survival
Lyrics: This is survival of the fittest. This is do or die. This is the winner takes it all. So, take it all, a-all,
a-all, a-all. . Wasn't ready to be no millionaire, I was ill-prepared. I was prepared to be ill though, the skill wa
s there. From the beginning it wasn't 'bout the ends, it was 'bout. Bustin'
*****
Score: 0.4012431
Genre: hip-hop/rap Artist: Jessie J SongName: Masterpiece
Lyrics: So much pressure, why so loud?. If you don't like my sound you can turn it down. I got a road, and I walk it
alone. Uphill battle, I look good when I climb. I'm ferocious, precocious I get braggadocios. I'm not gonna stop, I
like the view from the top. . You talk that blah blah. That la la, that rah
*****
```

(b) Original MPNet.

Figure 12: Recommendations for "We are the champions" by Queen (song about perseverance) in the style of hip-hop and more upbeat. Both are quite similar; however the genre finetuned model seems to focus more on the aspect of facing various trials, as also expressed in the Queen song ("I've paid my dues... I've done my sentence.. Kicked in my face").

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. 2016. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*.

Mari Gikaru. 2020. https://www.musicgateway.com/blog/how-to/a-complete-guide-to-shazam-music-app.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mariana O Silva, Laıs M Rocha, and Mirella M Moro. 2019. Musicoset: An enhanced open dataset for music data mining. In *XXXII Simpósio Brasileiro de Banco de Dados: Dataset Showcase Workshop, SBBD*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Motoyuki Suzuki, Toru Hosoya, Akinori Ito, and Shozo Makino. 2006. Music information retrieval from a singing voice using lyrics and melody information. *EURASIP Journal on Advances in Signal Processing*, 2007:1–8.

Tao Wang, Dong-Ju Kim, Kwang-Seok Hong, and Jeh-Seon Youn. 2009. Music information retrieval system using lyrics and melody information. In *2009 Asia-Pacific Conference on Information Processing*, volume 2, pages 601–604. IEEE.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

```
Score: 0.53017956
Genre: country Artist: The J. Geils Band SongName: Looking for a Love
Lyrics: Somebody help me. Somebody help me now. Somebody help me now. Somebody help me find my baby. Somebody help m
e find my baby right now. I'm looking for a love. I'm looking for a love. I'm looking here and there. I'm searching
everywhere. I'm looking for a love. To call my own. Well, someone to get up
*****
Score: 0.57674146
Genre: country Artist: Paul Davis SongName: Keep Our Love Alive
Lyrics: You're the sunlight that opened my eyes. When the world was tumbling down. You're the river that gave me a d
rink. When a drop of water couldn't be found. You give me what I want when I want it. And all I'll ever want is you,
ooh, ooh. I was born to be loved by somebody. And I know that somebody is y
*****
Score: 0.58109486
Genre: country Artist: Humble Pie SongName: I Don't Need No Doctor
Lyrics: I don't need no doctor 'cause I know what's ailing me. I don't need no doctor 'cause I know what's ailing m
e. Yes I do, all I need is my baby. You don't know I'm in misery. I don't need no doctor. I don't need no doctor. I
don't need no doctor. My prescription tells me that. I don't need no doctor.
*****
```

(a) Genre-finetune MPNet.

```
Score: 0.25241828
Genre: country Artist: Gene Autry SongName: Here Comes Santa Claus (Right Down Santa Claus Lane)
Lyrics: Here comes Santa Claus. Here comes Santa Claus. Right down Santa Claus Lane. Vixen and Blitzen and all his r
eindeer. Pullin' on the reins. Bells are ringin', children singin'. All is merry and bright. So hang your stockings
and say your prayers. 'Cause Santa Claus comes tonight. Here comes Santa Cla
*****
Score: 0.33566386
Genre: country Artist: Kenny Rogers SongName: The Greatest Gift of All
Lyrics: Dawn is slowly breaking. Our friends have all gone home. You and I are waiting. For Santa Claus to come. The
re's a present by the tree. Stockings on the wall. And knowing you're in love with me. Is the greatest gift of all.
The fire is slowly fading. Chill is in the air. All the gifts are waiting. F
*****
Score: 0.36401868
Genre: country Artist: Merle Haggard SongName: If We Make It Through December
Lyrics: If we make it through December. Everything's going to be all right I know. It's the coldest time of winter.
And I shiver when I see the falling snow. If we make it through December. Got plans to be in a warmer town come summ
er time. Maybe even California. If we make it through December we'll be fine
*****
```

(b) Original MPNet.

Figure 13: Recommendations for "All I want for Christmas is you" by Mariah Carey (song about needing someone for Christmas) in the style of country. The genre-finetuned model focuses more on the theme of needing/wanting someone, while the original model focuses on the Christmas theme.