SI630 Project Final Report

Kailin Wang

Abstract

Music genre is one of the most widely used labels when featuring songs. However, people usually label a song with a specific genre based on the melody or rhythm of songs, which involves complex preprocessing work of audio data when leveraging ML methods for classification. Therefore, we hope to find an easier to classify songs based on text song lyrics.

In this project, I implemented two language representation models for multi-label genre classification based on song lyrics. To further improve the performance of models on highly imbalanced data, three balancing strategies combined with language models are constructed for comparison.

Experimental results show that BERT models overall give better performances than DistilBERT models. The best BERT model gives a micro F1 of 0.82 and overall accuracy of 67.5%. On the other hand,balancing strategies can also help model give better performance, especially when combined with DistilBERT model. The best DistilBERT model with WeightedSampler can give a comparable performance as the best BERT model, while only takes half of time for training compared to BERT models.

1 Introduction

The primary goal of the task is to explore the relationship between song lyrics and the music genres by leveraging language representation models. Music genre is one of the most widely used labels on featuring songs.Particularly, it could be the key feature when building up music recommendation engines, which brings more real-world meanings to this study.

Besides, we could also be interested in the following questions: How accurate can the classifier work when tagging multiple genre labels to songs when the number of tags is not fixed? When the data is highly imbalanced, are there any strategies that can help addressing the problems cause by imbalance? To answer these questions, we can use NLP methods on the basic input data that only includes text lyrics, and produce multi-label classification outputs that represent predicted probabilities of assigning samples to each genre.

To be specific, I employed two models that combine language representation models with deep learning models: BERT(Bidirectional Encoder Representation from Transformers) and DistilBERT for lyrics embedding, together with multiple balancing strategies for addressing imbalance.

Various works have been finished on investigating similar multi-class classification tasks by dto and Akalp et al. (2021). Most of the related works focus on classification on less than 5 music genres, where the number of labels attached to each song is fixed to 1. In this project, I would combine language representation models with deep learning models, and further explore how the models perform on multiple-label genre classification tasks with more than 5 genres, where the number of labels attached to each song is NOT fixed. Besides, most of the papers did not put an emphasis on solving the problems caused by imbalanced data. Here, I would also compare the performances of different balancing strategies.

In terms of result, BERT models overall give better performances than DistilBERT models. The best BERT model gives a micro F1 of 0.82 and overall accuracy of 0.675, which is a competitive performance when compared to similar works conducted by others. Besides, BERT models are alsp more robust than DistilBERT in this task with highly imbalanced data. However, WeightedSampler balancing strategy has been proven to be helpful on improving the performance

of DistilBERT model on this task. Therefore, DistilBERT model with WeightedSampler can give a comparable performance as the best BERT model, while only takes half of time for training compared to BERT models.

2 Data

2.1 Data source and basic information

In this project, the main data resource is song lyrics, along with artist information and song information collected by Lim and Benson (2021) from Genius[1]
The main lyric dataset 'lyrics.jl' contains 37993 rows of 2 columns including unique song names and the transcribed lyrics on Genius.
The joint table of 'lyrics.jl' and 'song_info.json' has 37907 rows. the joint table obtained by inner-joining three datasets has 36720 rows.

2.2 Data preprocessing

2.2.1 Removing of non-English songs

In terms of data preprocessing works, there exist song lyrics that are non-English. To avoid the inconsistency and unnecessary complexity that multilingual task may bring in, I removed all the non-English song by using package 'langdetect[2] to recognize the main language used in the songs. After language filtering, 34872 English songs are kept in the final data set.

2.2.2 Relabeling of music genres

In terms of music genres, a list of the top 15 genre tags,together with frequencies and proportion in the full data set is displayed below. The bar plot representing the distributions are displayed in figure 1. You can match the genre IDs to genre names based on the table below.

[1] https://genius.com/
[2] https://pypi.org/project/langdetect/

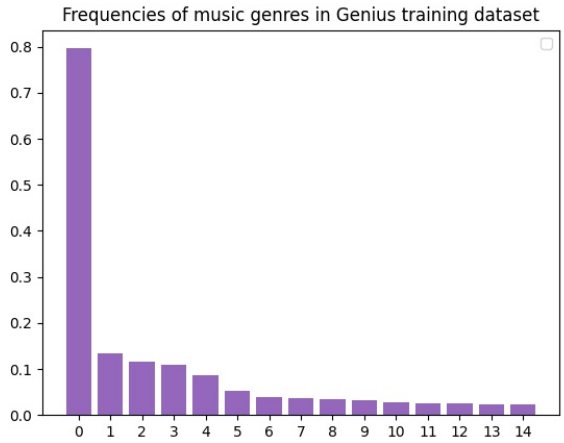

Figure 1: Frequencies of music genres

Genre	Frequency	Genre id
Rap	20471	0
Pop	3458	1
R&B	2977	2
Trap	2780	3
Rock	2232	4
West Coast	1364	5
Alternative	1008	6
UK	936	7
Alternative Rock	886	8
East Coast	843	9
Canada	728	10
France	679	11
French Rap	665	12
Atlanta	608	13
Electronic	595	14
...		

By reading related papers and implementing EDA on the training data, I realized the granularity of genres labels in the data set are not uniform. To solve this problem, I re-partitioned the genre set with 6 main genres based on music styles and the frequency of each genre appeared in the data set. With new genres partition rules, I relabeled each song by capturing key words of the existing labels. The table below displays the frequencies of new labels of the training data set:

New genres	Frequency	Proportion
Rap	27438	0.7868
Pop	5444	0.1561
R&B	4338	0.1243
Trap	3919	0.1123
Rock	3540	0.1015
Alternative	2552	0.07318
Others	252	0.0072

2.2.3 Encoding of genre labels

To encode the genre labels of songs into numerical representations, one-hot encoding was adapted here. For each song, a vector with length K=6 with binary values(1/0) would represent whether the song belongs to each of all 7 genres.

2.2.4 Train-validation-test split

The final data set is split by 60/20/20 to get data sets for training, validation and testing purposes. The split is based on stratified sampling method, which keeps the distribution of music genres in each data set consistent with the original distribution in the full data set.

3 Related Work

Machine learning methods are widely used in related works and research on music genre classification. Ying et al. (2012) uses traditional methods, including SVMs, K-NN, and Naive Bayes classifiers for the categorization of music genres and moods in a song.

Neural network methods are usually implemented in classification task with a combination of audio and text data. Costa et al. (2017) compares the performance of CNNs on spectrograms data with SVMs for genre classification purpose. The result shows that CNN outperforms other classifiers in several scenarios.Tsaptsinos utilised variants of hierarchical attention network and LSTM for genre classification using intact lyrics, while LSTM reaches up to 49.77% prediction accuracy on the 20 Genre dataset.

Since music genre classification based on lyrics is essentially a text classification task, more and more researchers are combining language representation models with deep learning models in music genre classification tasks. Kumar et al. (2018) apply two Word Embedding techniques, Word2Vec and Word2Vec with TFIDF (Term Frequency-Inverse Document Frequency), on the preprocessed data, and implement Support Vector Machine, Random Forest, XGBoost (eXtreme Gradient Boosting) and Deep Neural Networks on the resultant word vectors to predict the genre of 4 classes. A mean accuracy of 61.70% and 71.05% in Simple Word2Vec and Word2Vec with TFIDF respectively with maximum accuracy of 65.0% and 74.0% using a 3 Layer Deep Learning model in case of both the techniques.

Akalp et al. (2021) investigates the usage of BERT and DistilBERT in music genre classification by comparing the results with a traditional deep neural network based on a BILSTM. Experimental results show that BERT outperforms other models on one-label and multi-label classification with accuracy of 77.63% and 71.29% respectively. And BERT runs 4 times faster than DistilBERT.

The majority of papers focuses on the task of one-label classifications or multi-label classification problems with less than 5 music genres, where the number of labels attached to each song is fixed to 1. In this project, I would combine language representation models with deep learning models, and further explore how the models perform on multiple-label genre classification tasks with more than 5 genres, where the number of labels attached to each song is NOT fixed. Besides, most of the papers did not put an emphasis on solving the problems caused by imbalanced data. Here, I would also carry out comparison experiments to better address this problem under a more extreme and complex scenario regarding data imbalance.

4 Methodology

In this study, my primary goal is to combine language representation models with deep learning models on the classification of music genres using lyrics. For this purpose, I plan to compare different bert-based models: BERT[3](Bidirectional Encoder Representation from Transformers) and a variant of DistilBERT model, MiniLM [4]. Comparisons would be carried out in two folds: the performance on this multi-label classification task, and computational costs as a plus. Besides, another focus of this project is how to overcome the difficulty that caused by the highly imbalanced labels of music genres. For this purpose, I carried out comparison experiments with different weighted

[3] https://huggingface.co/bert-base-uncased

[4] https://huggingface.co/microsoft/MiniLM-L12-H384-uncased

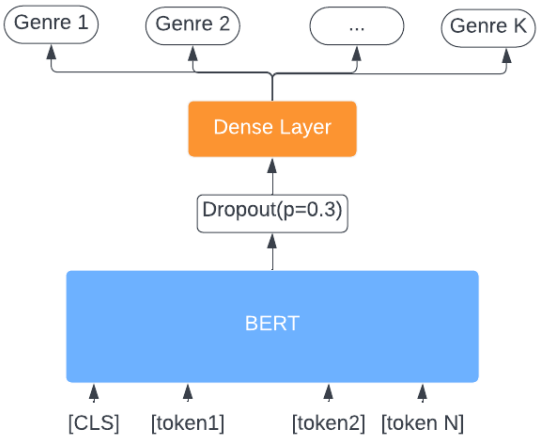

Figure 2: BERT-based models

loss function and sampling strategies that could help to address this problem.

4.1 BERT-based models

The main structure of BERT-based models can is illustrated in figure BERT-based models.

4.1.1 BERT-based models

The BERT model involves 3 main parts: BERT encoder layer [CLS], one dense layer with dropout layer, and sigmoid function for the final output. Here, the Sigmoid function is added by adapting BCELogitsLoss as loss function instead of manually adding in.
The BERT encoder layer receives an input of a sequence of lyric tokens with limited number(parameter). The outputs is the first token of the sequence [CLS] as the representation of the lyrics sequence. For the 'bert-base-uncased' model, the dimension of the pooler_output, i.e. the dimension of [CLS] token, is 768.
Dropout layer is used here to prevent overfitting, while the sigmoid activation function is used on top of the dense layer for multi-label classification purpose.
In terms of the default setup of the BERT model, the vocabulary size of the embedding layer is 30522. It has encoders with 12 transformer block, 13 self-attention heads, and a hidden size of 768. The input of a sequence should contain no more than 512 tokens.

4.1.2 DistilBERT

DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased. In terms of the dimension of [CLS] token,the dimension of the pooler_output is 384, which is half of that in BERT model. The DistilBERT model shares a similar structure as the BERT model: DistilBERT encoder layer [CLS], one dense layer with dropout layer attached before, and sigmoid function for the final output. The main difference, as mentioned before, is that the size of the encoder layer is much smaller in this case.

4.2 Preprocessing of lyrics

To tokenize the lyrics, I leveraged the universal BERT tokenizer to realize the tokenization of all the lyrics. In this case, the lyrics of a song should be considered as a whole entity, thus one [CLS] token is added at the beginning of the lyrics and one [SEP] is attached to the end of it. The maximum embedding length is set as 512.

4.3 How to address the imbalanced data

4.3.1 Loss function

When dealing with classification problem, cross-entropy, or BCEWithLogitsLoss[5] is the most commonly used loss function. However, in this case when the classes are highly imbalanced, the original BCEWithLogitsLoss could be misleading and result in 'one-sided' classification result. That is, the prediction of a specific genre would also be the majority population in this genre.
To solve this problem, one way is to leverage weighted BCELoss function by adding more weights to the positive cases for the minority class in population. Here, I constructed the weight as follows:

$$W_c = \frac{N_c - n_c}{n_c}$$

where n_c is the number of instances in class c that are positive, N_c is the number of all instances in class c. The smaller n_c is, the larger the weight is, then the harsher the punishment on the minclassification would be.
However, how to create good weights could be another difficulty under this scenario of a multi-label

[5] https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

classification problem with the number of labels unfixed.

4.3.2 Resampling

Except for making adjustment on the loss function, another way to address this problem is by leveraging WeightedResampler[6]. Intuitively speaking, we want to remove the imbalance by resampling the minority group with higher probability. Thus, the way how the weights are constructed here are similar to the weight in the previous part. However, weights here correspond to samples in the training data set instead of classes. To achieve this, I used the weights generated from the last part, and used the sum total of all weights from all classes as weight for resampling. It can be expressed as follows:

$$W(X_i) = \sum_{c=1}^{K} W_c * X_{i,c} + \frac{1}{W_c} * (1 - X_{i,c})$$

where K is the number of classes or genres, $X_{i,c}$ represents the c class values(0 if not belongs to this genre, otherwise 1) of the ith sample. W_c is the class weight defined in the previous part.

5 Evaluation and Results

5.1 Evaluation metrics

5.1.1 Loss function

During the training of models, I would use weighted cross-entropy, or BCELoss as the loss function, which can be expressed as:

$$H(p,q) = -\sum_{n=1}^{N}\sum_{c=1}^{K} w_c(y_{n,c}q_{n,c}+(1-y_{n,c})(1-q_{n,c}))$$

where N is the number of training samples, K is the number of genres, $p_{n,c}$ represents the true possibility that the n-th sample belongs to genre k, which usually takes value 0 or 1. $q_{n,c}$ represents the predicted possibility of the n-th sample belongs to genre k. w_c is the weight given to the k class. When no weights are given, $w_c = 1$

5.1.2 F1 score, Accuracy, Precision and Recall

For a multi-label classification problem under this scenario, we can obtain F1 score, accuracy, recision and recall [7] metrics with respect to each binary classification problem. These metrics can be considered in discussion part for further diagnostics.

To measure the overall performance of models, we can look at the micro and macro combination of these metrics.

Another metric that give better interpretation is the overall accuracy, which can be defined as the proportion of samples are correctly classified in terms of every binary music genre classification. It can be expressed as:

$$ACC_{overall} = \frac{N_{All\ True}}{N}$$

where $N_{All\ True}$ is the number of samples are correctly classified in all K binary music genre classification tasks. N is the size of the whole data set.

The primary metrics that I adapted as measurements of the performance of the models are: macro F1-score, micro F1-score.[8],and overall accuracy given that the data is highly imbalanced.

5.2 Baselines

Baseline 1. Random Classification

Suppose the samples are randomly assigned to each class with the probability equals $\frac{1}{2}$.

With this setup, the corresponding evaluation metrics on the test data set are:

	Micro	Macro
Precision	0.23	0.23
Recall	0.5	0.5
F1	0.31	0.26
Overall Acc	0.015	

Baseline 2. Prediction based on population

In this case, we classify every sample to majority class in each genre.

Suppose for class C, the estimated probability of $P_C = W_C$, where W_C is the proportion of samples that belong to class C in the data set, then the cross-entropy is: With this setup, the corresponding evaluation metrics on the test data set are:

	Micro	Macro
Precision	0.78	0.13
Recall	0.57	0.17
F1	0.66	0.15
Overall Acc	0.579	

[6] https://pytorch.org/docs/stable/data.html?highlight=weightedrandomsampler#torch.utils.data.WeightedRandomSampler

[7] https://en.wikipedia.org/wiki/Precision_and_recall

[8] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

5.3 Results of models

The key parameters and performances on the test dataset of BERT, DistilBERT models are displayed in the table below:

5.3.1 DistilBERT

Hyperparameters

Parameter	Values
vocab size	30522
hidden size	384
num hidden layers	12
num attention heads	12
intermediate size	1536
hidden dropout prob	0.1
attention probs	0.1
dropout prob	0.3
max position embeddings	512
epoch	2
Optimizer	Adam[9]

The epoch parameter is tuned based on the model performance on the validation data set. After 2 epochs, the validation error starts to going up, thus the finalized model is the one trained after 2 epochs.

Performances

	BCELoss	WBCE	WSampler
micro F1	0.76	0.48	0.80
macro F1	0.39	0.52	0.65
Overall ACC	0.6324	0.4347	0.6232
Train time(ep)	6m18s	6m18s	6m18s

Different balancing strategies obviously give different degree of improvements on the performance of the DistilBERT model. Among them, DistilBERT model with WeightedSampler gives the best per formance with micro F1=0.80, macro F1 score=0.65 and overall accuracy=62.3%.

5.3.2 BERT

Hyperparameters

Parameter	Values
vocab size	30522
hidden size	768
num hidden layers	12
num attention heads	12
intermediate size	3072
hidden dropout prob	0.1
attention probs	0.1
dropout prob	0.3
max position embeddings	512
type vocab size	16
epoch	5/2(Wsampler)
Optimizer	Adam[10]

Performances

	BCELoss	WSampling
micro F1	0.82	0.81
macro F1	0.65	0.66
Overall ACC	0.6755	0.6437
Train time(epoch)	15m24s	15m24s

Experimental results show that balancing strategies don't bring in further improvements to BERT models in terms of F1 scores and overall accuracy.The best BERT model is the one with the original BCELoss without WeightedSampler, which yields micro F1=0.82, macro F1 score=0.65 and overall accuracy=0.675.

6 Discussion

6.1 Comparison between balancing strategies

6.1.1 DistilBERT model

Based on the result table from the DistilBERT model, it's easy to see that WeightedSampler method gives the best performance in terms of F1 and overall accuracy. The overall accuracy on 6 genres classification problems is about 62.3%. Besides, from figure 3, we can see that WeightedSampler method helps to correct the 'one-sided' prediction problem that usually occurs in the prediction of the genre that appears very infrequently. As a comparison, the Alternative music genre is always predicted as 'No' when adapting BCELoss without resampling. This could help to explain why the WeightedSampler model outperforms the others.

The model trained with the original BCELoss function also gives a good result, but with a much lower macro F1 score. The reason why can be found in the full output table in figure 4. Since macro F1 score is taking unweighted average of all the classificaition f1-scores, a zero f1-score of Al-

Overall Accuracy: MLTC_model_state_resamp.bin 0.6232258064516129

	precision	recall	f1-score	support
Rap	0.97	0.94	0.95	5440
Pop	0.66	0.68	0.67	1134
R&B	0.54	0.54	0.54	882
Trap	0.61	0.55	0.58	777
Rock	0.62	0.76	0.68	728
Alternative	0.40	0.54	0.46	535
micro avg	0.79	0.80	0.80	9496
macro avg	0.63	0.67	0.65	9496
weighted avg	0.81	0.80	0.80	9496
samples avg	0.84	0.85	0.82	9496

Figure 3: DistilBert WeightedSampler

Overall Accuracy: MLTC_model_state.bin 0.6324014336917563

	precision	recall	f1-score	support
Rap	0.97	0.94	0.96	5440
Pop	0.60	0.78	0.68	1134
R&B	0.58	0.35	0.44	882
Trap	0.63	0.41	0.50	777
Rock	0.76	0.50	0.60	728
Alternative	0.00	0.00	0.00	535
micro avg	0.84	0.74	0.79	9496
macro avg	0.59	0.50	0.53	9496
weighted avg	0.79	0.74	0.76	9496
samples avg	0.86	0.81	0.81	9496

Figure 4: DistilBert BCE

ternative genre dramatically drags down the average score. In this sense, WeightedSampler model is more robust.

Despite of that, both of the two models outperform the baseline model.

However, it is surprising that the Weighted BCELoss gives the worst performance in terms of micro F1, which is even lower than the baseline micro F1 score. This could be resulted from the difficulty on determining the best weights that should be assigned to each genres. Here, I adapted a commonly used method to generate the genre weights, but it can't be easily fine-tuned as other parameters in the model. From figure 5, we can see that the current model with weights dramatically improved the recall for the minority genres and precision of the majority genres. This improvement is actually very meaningful and also not observed in the other models. By looking at the precisions, I believe this phenomena may indicate that the weights are too large to well-balance recall and precision. If I have more time, I would try to train the model with smaller weights, which I believe can help the model to give better performances.

6.1.2 BERT model

Since the weighted BCE method didn't give satisfactory result in DistilBERT model, I chose not to conduct this method on the training of BERT

Overall Accuracy: ML_Weight_model_state.bin 0.4346953405017921

	precision	recall	f1-score	support
Rap	1.00	0.65	0.79	5440
Pop	0.50	0.89	0.64	1134
R&B	0.33	0.76	0.46	882
Trap	0.63	0.31	0.41	777
Rock	0.52	0.85	0.64	728
Alternative	0.29	0.71	0.41	535
micro avg	0.62	0.68	0.65	9496
macro avg	0.54	0.69	0.56	9496
weighted avg	0.77	0.68	0.68	9496
samples avg	0.63	0.70	0.63	9496

Figure 5: DistilBert Weighted BCE

model. Here, only to methods are adapted for comparison.

For the BERT model, it is surprising that the original BCELoss without WeightedSampler model outperforms the others, which yields micro F1=0.82, macro F1 score=0.65 and overall accuracy=0.675.

The WeightedSampler method in BERT model also gives a performance with micro F1=0.81, macro F1 score=0.66 and overall accuracy=0.63. Compared to the DistillBERT model, there is an 1% overall improvement on all the metrics.

In terms of f1-score, these two BERT models give similar performances. However, if we can take a closer look at the output table, we can see that the model with WeightedSampler performs better on the recall metrics, but gives lower precisions. This may indicate that the weightedsampler can help improve the recall, which is the pain point in this imbalance data classification problem, but the resampling method may need to be more carefully designed.

In terms of accuracy, BERT model with BCEloss gives a much higher overall accuracy. Besides, the 'one-sided' effect on the minority genre classification disappeared. This may indicate that BERT model naturally perform better than the DistilBert model on imbalanced data like this! This is a very interesting and meaningful discovery.

6.2 Comparison between two BERT-based models

To explain the differences between the performances of two BERT-based models, different pretraining process could be a potential factor. Besides, the larger dimension of the encoder layer is more likely to be the reason. The [CLS] token is of length 768 in BERT while 384 in DistilBERT, thus more parameters exists in the BERT model. However, as a trade-off, the training time of

Overall Accuracy: 0.6755555555555556

	precision	recall	f1-score	support
Rap	0.96	0.96	0.96	5440
Pop	0.71	0.67	0.69	1134
R&B	0.70	0.45	0.55	882
Trap	0.72	0.46	0.56	777
Rock	0.73	0.69	0.71	728
Alternative	0.56	0.35	0.43	535
micro avg	0.87	0.78	0.82	9496
macro avg	0.73	0.60	0.65	9496
weighted avg	0.85	0.78	0.81	9496
samples avg	0.89	0.84	0.84	9496

Figure 6: Bert BCE

Overall Accuracy: 0.6437275985663082

	precision	recall	f1-score	support
Rap	0.97	0.94	0.96	5440
Pop	0.66	0.74	0.70	1134
R&B	0.58	0.57	0.57	882
Trap	0.60	0.58	0.59	777
Rock	0.67	0.75	0.71	728
Alternative	0.56	0.34	0.42	535
micro avg	0.82	0.81	0.81	9496
macro avg	0.67	0.65	0.66	9496
weighted avg	0.82	0.81	0.81	9496
samples avg	0.86	0.86	0.83	9496

Figure 7: Bert WeightedSampler

BERT model is more than 2 times as that of the DistilBERT model.

Given above, DistilBERT models are faster and lighter then BERT, but may encounter difficulties on classification tasks with imbalanced data. As a comparison, BERT models give better performances on this classification task with imbalanced data, but is more computationally expensive. To conclude, when computational expense is more of concern, DistilBERT model with WeightedSampler is preferred. Instead, if the user is pursuing higher accuracy, BERT model is a better choice. Compared to the baselines and similar experiments conducted by Akalp et al. (2021), which gives about 70% accuracy on multi-label classification problem on 6 genres that are less imbalanced, I think the performances of these models in this study are satisfactory and have met my expectation.

7 Conclusion

In this project, I compared two language representation models (BERT and DistilBERT) with different strategies on solve the common problems occurred in multi-label classification tasks on imbalanced data. Multiple data preprocesing works are finished to remove non-English songs, inconsistency between definitions of music genres, and genres that has extremely small population.

In terms of balancing strategies, DistilBERT model requires balancing strategies to give better performances. Among these models, DistilBERT model with WeightedSampler gives the best performance. As a comparison, BERT models are naturally more robust on imbalanced data classification tasks, but balancing strategies can also help BERT model give better trade-off between precision and recall.

For the comparison between two BERT-based model, BERT model with BCEloss give the best performance, with micro F1=0.82, macro F1 score=0.65 and overall accuracy equals 67.5%, thus is perferred when the user is more focused on improving accuracy. However, DistillBERT model with WeightedSampler also gives comparable performance in terms of f1-score, but takes less than half of the time for training.Thus, DistillBERT model is probably preferred when computational cost is of concern.

For future work, I hope to work out better balancing strategies with the weighted BCEloss and WeightedSampler. Since we have validate the strengths of these two strategies, more efforts can be devoted to the work of fine-tuning the weights assigned to each genres' positive and negative samples.

8 Other Things I Tried

At the very beginning, I also tried out the multi-label classification with pretrained AutoModelForSequenceClassification[11] model, hoping to get a satisfactory performance on the single-label classification task. However, the model works poorly and gives zero recall since it just simply classified every song to the overwhelmingly populated genre *Rap*. That's the time when I realized the necessity of adapting different balancing strategies.

Besides, I also tried out multi-label classification with the number of genre tagged to each song fixed based on AutoModelForSequenceClassification[12] model. The logic here is setting up a dynamic threshold for each binary classification

[11] https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#automodelforsequenceclassification

[12] https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#automodelforsequenceclassification

based on the pre-specified number of tags k and predicted probabilities. The top k genres with the highest probabilities would be classified as positive. However, I didn't finish all the coding parts here, so I haven't gotten an output that can be placed in the report. Nevertheless, considering that the current models have already provided some satisfactory results, I believe this model could hardly give better predictions given that it requires pre-specified numbers of tags.

9 What You Would Have Done Differently or Next

Balancing Strategies If I could start again, I would spend more time on developing better weighting strategies that can be applied in WeightedSampler and Weighted BCELoss. With better weights, I believe that there stands a chance of getting models that perform better, especially in terms of regrading the f1-score and overall accuracy.

Correlation between genres In this project, the correlation between genres are ignored and assumed to be somehow independent since all these binary classifiers are almost independent except for sharing the same token layers. However, there could exists positive or negative correlations between genre pairs. If so, we can further incorporate this prior knowledge into the model by setting up restrictions on the output of some genres(e.g. combined two binary classifications into one multi-label classification when negatively correlated, into one binary classification when highly positively correlated).

More BERT-based models In this work, only two BERT-based models, Bert-base-uncased and MiniLM,are adapted for comparison. However, there are a great variety of variants of BERT-based models that are trained with different pretrained tasks and have different structures. If I have more time, I would try out more models on this task, and make further comparisons among them to see the strengths and weakness of each model. Maybe a perfect model would show up and dominate all the other models on this task on highly imbalanced data.

References

????

Hasan Akalp, Enes Furkan Cigdem, Seyma Yilmaz, Necva Bolucu, and Burcu Can. 2021. Language representation models for music genre classification using lyrics. In *2021 International Symposium on Electrical, Electronics and Information Engineering*. pages 408–414.

Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. 2017. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing* 52:28–38.

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)*. IEEE, pages 2142–2146.

Derek Lim and Austin R Benson. 2021. Expertise and Dynamics within Crowdsourced Musical Knowledge Curation: A Case Study of the Genius Platform. In *Proceedings of the International AAAI Conference on Web and Social Media*. volume 15, pages 373–384.

Alexandros Tsaptsinos. ???? Music genre classification by lyrics using a hierarchical attention network.

Teh Chao Ying, Shyamala Doraisamy, and Lili Nurliyana Abdullah. 2012. Genre and mood classification using lyric features. In *2012 International Conference on Information Retrieval & Knowledge Management*. IEEE, pages 260–263.