Prediction of Genres and Emotions by Song Lyrics

Shiyu Li
Institute for Computational and Mathematical Engineering (ICME)
Stanford University
lishiyu@stanford.edu

Chunjiang Mou
Institute for Computational and Mathematical Engineering (ICME)
Stanford University
cmou2@stanford.edu

Chungfu Chang
Non Degree Option, Stanford Center for Professional Development
Stanford University
cfcalex@stanford.edu

Abstract

This project aims to apply Bidirectional Encoder Representations from Transformers (BERT) model to predict and classify the genres and emotions based on the Song Lyrics. We hope those predictions can facilitate the automation of the music industry.

1 Introduction

The importance of genre and emotion classification in music organization has long been recognized by the industry due to the explosion of music recordings online [1]. Some music player platforms such as Spotify are known to its music recommendation system, where they recommend music based on their customer historical or genre preferences individually. It will be a good idea if users get recommendations based on the mood of the lyrics. Lyrics-based analysis could provide benefits to the music industry by automatically tagging the genres and emotions of a song uploaded by an artist to improve user's experience when searching for songs. The objective of this study is to build an automatic classifier of the genres and emotions based on song lyrics.

In the study, we fine-tuned pre-trained BERT model and applied transfer learning for two classification tasks: genre prediction and emotion prediction. The input of the model is the song lyrics and the outputs are the labels of genres and emotions, both into 4 categories.

2 Related work

Past attempts to automatically tag the genres or emotions music mainly apply machine learning models such as NB, SVM, K-NN, and deep learning models such as LSTM for classification [2] [3] [4]. These models reads the lyrics sequentially and encodes it into single word embedding for each word. Therefore, when pre-trained representations are contextual and even bidirectional contextual, these methods would not work that well.

In comparison, Bert makes use of transformer to read the entire sequence of words at once. Therefore, it could uncover the contextual relationships based on all its surroundings (left and right).

3 Dataset and Features

The predicted labels for our dataset are one hot vectors that represent genres and emotions of the song. The feature for our model is the song lyrics.

3.1 Emotions

We found a lyrics dataset called labeled_lyrics_clearned.csv on Kaggle that contains full lyrics and labels of more than 150,000 songs [6]. The label is the Spotify valence attribute, ranging from 0 to 1. It describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive (happy, cheerful), while tracks with low valence sound more negative (sad, depressed).

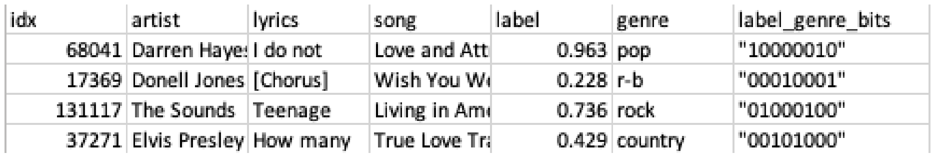
We encoded the Spotify valence into one hot vectors with dimension of 4. In other words, we have 4 emotion categories.

3.2 Genre

To get genre of each song in the lyrics dataset, first we tried Python library LyricsGenius [7] of Genius.com to search its lyrics pages. However, its search quality is not good enough to find lyrics pages because it searches everything on Genius.com and only limited number of results are returned per query. On the other hand, we observed that Genius.com has a template for its lyrics page url. By following this template to compose a possible lyrics page url for each song in the lyrics dataset, we successfully got 110,000 lyrics page urls from Genius.com. Then we used Python library Beautiful Soup [8] to build a web scraper to scrape these lyrics pages to extract its primary tag which contains its genre. Finally we also encoded the genres into one hot vectors with dimension of 4. Therefore, we have 4 genres: r-b, pop, rock, and country.

3.3 Data Processing

Given the emotion and genre encoding, we combined them together as a 8 character string column. Then We extracted 5000 samples from 150,000 samples because of the GPU limit on AWS. To ensure the performance of the model, we balanced the data by randomly extracting around 313 samples from each (genre, emotion) category. Then, we split the data into training data and test data with ratio of 9:1.



idx	artist	lyrics	song	label	genre	label_genre_bits
68041	Darren Haye:	I do not	Love and Att	0.963	pop	"10000010"
17369	Donell Jones	[Chorus]	Wish You W	0.228	r-b	"00010001"
131117	The Sounds	Teenage	Living in Am	0.736	rock	"01000100"
37271	Elvis Presley	How many	True Love Tr:	0.429	country	"00101000"

Figure 1: csv file of the lyrics dataset

4 Methods

4.1 Modeling

Before we apply the pre-trained BERT, we applied the encode_plus to tokenize each sentence, added the special [CLS] and [SEP] token to the start and the endand truncated sentence to maximum length of 512 which is allowed by BERT.

To fine tune our BERT model, we utilized AdamW optimizer and set batch size to be 10, which is the biggest possible batch size due to the limited RAM of GPU, although the recommended batch size is 16 or 32. We tried learning rates of (1 5e-5; 5 9e-6) respectively.

After getting outputs from BERT, We only use the output embedding of [CLS] token to predict emotions. We use FCs to shrink its size from 768 to 3, followed by a softmax activation. Finally, we use cross entropy as output loss function. We choose to use accuracy as our metric.

4.2 Transfer Learning

We started the modelling by focusing on classifying emotions. Following benign results, we applied Multi-Task Deep Neural Networks (MT-DNN) to predict both emotions and genres.[4] MT-DNN train the model in two stages. In the first stage, MT-DNN trained Lexicon Encoder and Transformer Encoder by masked language modelling and next sentence prediction. In the second stage, MT-DNN fine tuned the model, computed the loss across different task(prediction of emotions and genres), and then updated the parameters and learning rate at the same time.

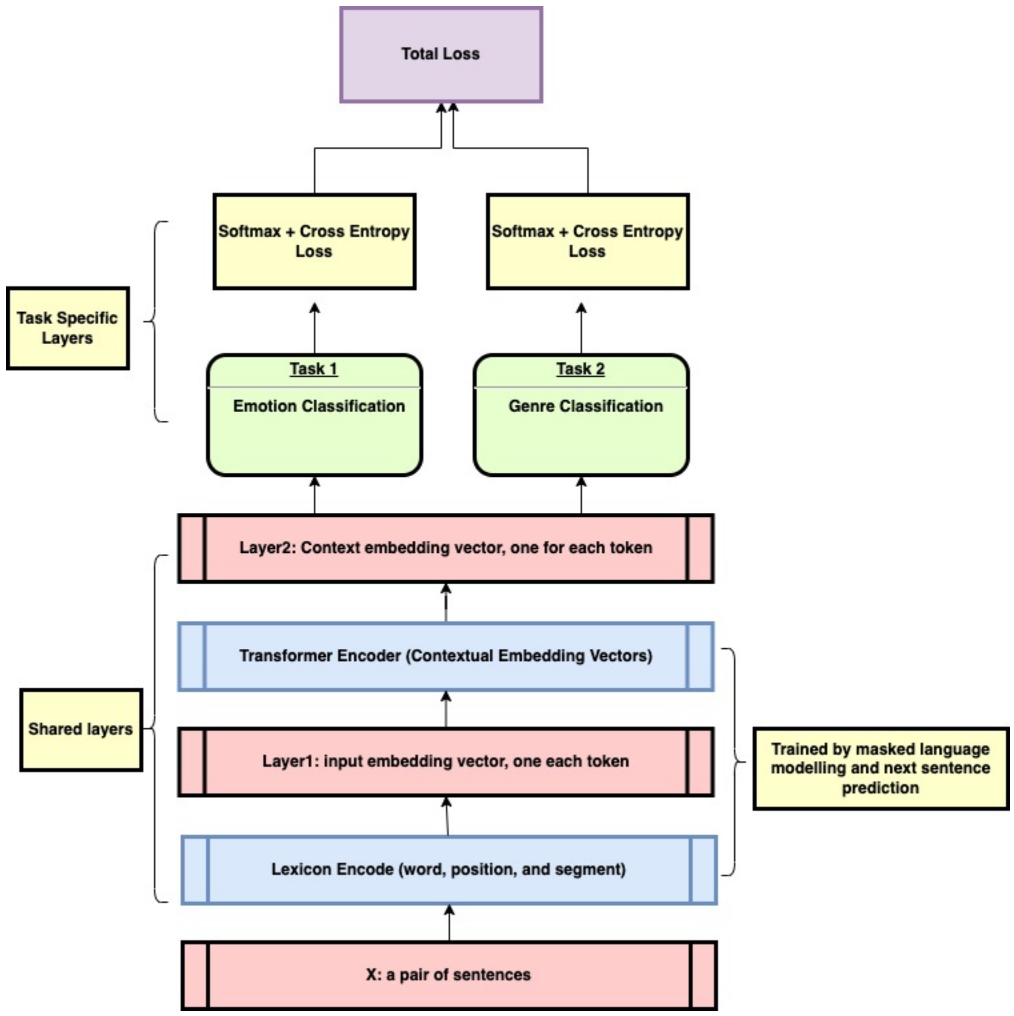

Total Loss
Softmax + Cross Entropy Loss
Softmax + Cross Entropy Loss
Task Specific Layers
Task 1
Emotion Classification
Task 2
Genre Classification
Layer2: Context embedding vector, one for each token
Transformer Encoder (Contextual Embedding Vectors)
Shared layers
Layer1: input embedding vector, one each token
Trained by masked language modelling and next sentence prediction
Lexicon Encode (word, position, and segment)
X: a pair of sentences

Figure 2: Architecture of our Model

5 Experiments/Results/Discussion

While training, we explore different hyperparameters, e.g., the number and dimension of layers of FCs after BERT, the dropout possibility, number of warmup steps and learning rate, and we use accuracy as our primary metric.

5.1 Result

Figure 3 and Figure 4 show the loss and accuracy of the two tasks during training. For genre prediction task, the loss begins to drop after some epochs, and the accuracy continues to grow, and finally reaches a level of 70%. For emotion prediction task, the loss barely decreases during training, and the accuracy remains around 30%. After training, the accuracy in test set for emotion prediction and genre prediction is 32.5% and 52.1%respectively.

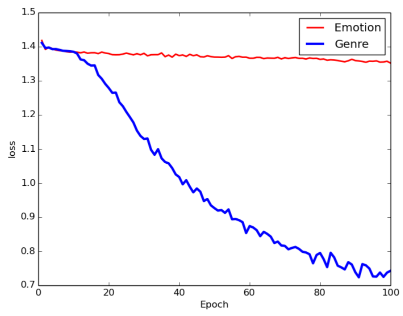


Figure 3: Loss of emotion prediction during training

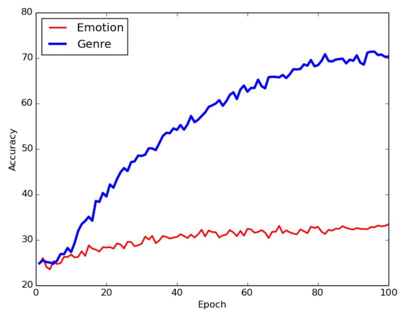
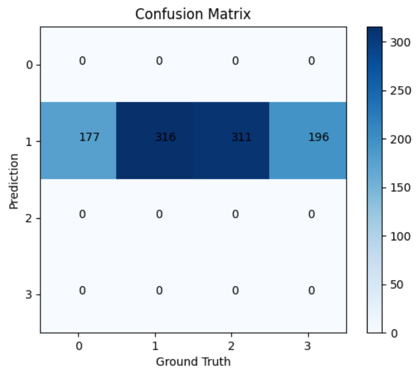


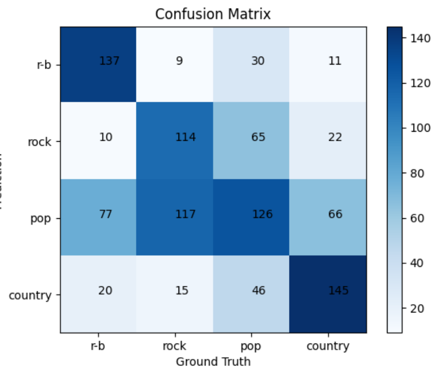
Figure 4: Accuracy of genre prediction during training

Figure 5 and Figure 6 show the confusion matrix of the two tasks in our test sets. We can find in Figure 6 that our model learns something in genre prediction task. However, even with the identical network structure and parameters as genre prediction task, in emotion prediction task the model doesn't learn anything, instead it just predict every sample to be a certain class. We first thought it is because the sample is unbalanced, so we resampled our input and ensure the among total 5000 samples, the number of sample of each class is equal. Unfortunately, since we split training set and test set randomly, the samples are not perfectly balanced in the two sets, so the model still tends to predict all the sample into one class, as shown in Figure 7.



Prediction \ Ground Truth	0	1	2	3
0	0	0	0	0
1	177	316	311	196
2	0	0	0	0
3	0	0	0	0

Figure 5: Confusion matrix of emotion prediction



Prediction \ Ground Truth	r-b	rock	pop	country
r-b	137	9	30	11
rock	10	114	65	22
pop	77	117	126	66
country	20	15	46	145

Figure 6: Confusion matrix of genre prediction

5.2 Discussion

We've encountered several problems in this project. The most serious problem is that the BERT model requires a lot of computation power and with our limited computation power, it takes about 15 minutes to train a epoch even we just input 5000 samples. The limited number of epochs we can train and the limited number of input samples we can handle cause our result to have both high-bias problem and high-variance problem. We tried to solve the high-bias problem by adding more layers to the FCs to improve it's expressive power, try different warmup steps and different learning rates. We find that among this methods, having a smaller initial learning rate (5e-6) can help most, but the

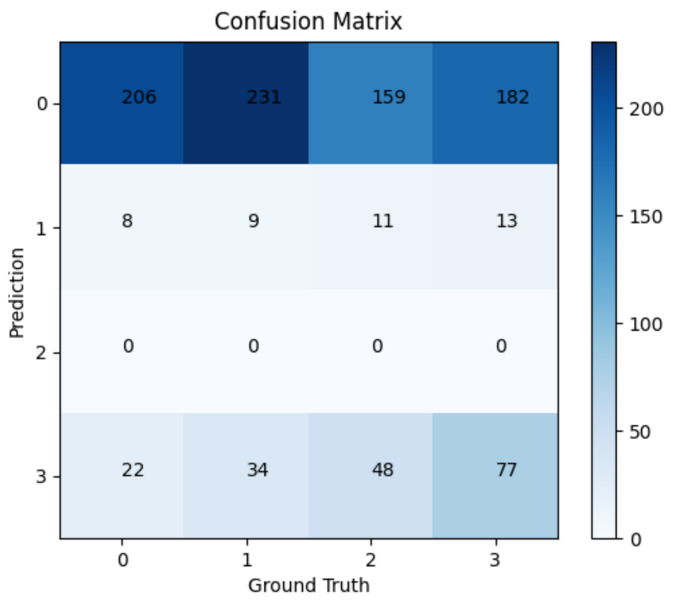


Figure 7: The confusion matrix of emotion prediction, after balancing input samples

benefit is still limited. We tried to solve the high-variance problem by increasing dropout possibility and weight decay, but it barely has any positive effects.

Another problem we've encountered is that, when conducting multi-task learning, we find that it is much easier for our model to predict the genre of the song than the emotion of the song. We try to address this problem by using higher learning rate for genre prediction and lower learning rate for emotion prediction. We also find class imbalance problem for emotion prediction, and tried to address it by re-sampling our input. Unfortunately, it barely has any effect, and the model still predicts all the sample into a single class.

6 Conclusion/Future Work

6.1 Conclusion

We fine-tuned pre-trained BERT model and applied transfer learning to predict genres and emotions. Due to the limitation of the computation power, we had to choose a relatively small proportion of our data so that we did not achieve a satisfying testing accuracy. The reports covers the model architecture and its limitations, as well as the possible remediations in the future.

6.2 Future Work

One of our difficulties is GPU limit on AWS, so one of our potential solutions will be Google's Colaboratory [9]. It allows users to write and execute their Python codes through a browser, and is especially well suited to machine learning, data analysis. It is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

.

7 Contributions

7.1 Shiyu Li

1. Created and fine tuned the Bert model, analyzed the results

2. Recorded the video

7.2 Chungfu Chang

Major contribution is lyrics dataset preparation including:

1. Searched bigger lyrics dataset and found the Spotify lyrics dataset which has 150000 songs with Spotify valence attribute.
2. Wrote several Python scripts to
    - build a web scraper to scrape lyric page of each song in the Spotify lyrics dataset in order to extract genre information from Genius.com.
    - encode each song's emotion and genre.
    - randomly choose 5000 samples and balance sample distribution.

Minor task is to find solutions for the issue of GPU limit on AWS. It seems Google Colaboratory will be the solution but we do not have enough time to complete the whole experiment.

7.3 Chunjiang Mou

1. Wrote the majority of final report and ppt
2. Performed literature review in Bert Model to locate resources
3. Built up baseline model

.

References

[1] Bert Language Model, TechTarget, `https://searchenterpriseai.techtarget.com/definition/BERT-language-model`

[2] Y. Hu, X. Chen, and D. Yang, "Lyric-Based Song Emotion Detection With Affective Lexicon and Fuzzy Clustering Method," *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, `https://archives.ismir.net/ismir2009/paper/000041.pdf`

[3] Y. Song, M. Pearce, "Evaluation of Musical Features for Emotion Classifications," *13th International Society for Music Information Retrieval Conference (ISMIR 2012)*

[4] E. Coutinho, G. Trigeogis, "Automatically Estimating Emotion in Music with Deep Long-Short Term Memory Recurrent Neural Networks," `http://ceur-ws.org/Vol-1436/Paper64.pdf`

[5] E. Ma, (2019) "When Multi-Task Learning Meet Bert," *Introduction to Multi-Task Deep Neural Networks for Natural Language Understanding*

[6] Edenbd, "150K Lyrics Labeled with Spotify Valence," *Kaggle*, 28 Apr. 2020, `https://kaggle.com/edenbd/150k-lyrics-labeled-with-spotify-valence`

[7] Miller, John W, "A Python Client for the Genius.com API." LyricsGenius, 2020, `https://lyricsgenius.readthedocs.io/en/master`

[8] Richardson, Leonard. "Beautiful Soup Documentation." Beautiful Soup Documentation - Beautiful Soup 4.9.0 Documentation, 2004, `https://www.crummy.com/software/BeautifulSoup/bs4/doc/#`

[9] Google Colab, Google, `https://colab.research.google.com/?utm_source=scs-index`