Sentence-BERT Embeddings for Music Genre Classification

Tianhao Qu
Vanderbilt University
tianhao.qu@vanderbilt.edu

ABSTRACT

This study investigates the feasibility of using Sentence BERT embeddings to classify music genres based solely on lyrics. By applying machine learning models, including random forest and multi-layer perceptron, we evaluated whether these embeddings could effectively capture the necessary semantic information for accurate genre classification. Our findings indicate that while Sentence BERT can distinguish between genres such as album rock, adult standards, alternative metal, alternative rock, and dance pop, the performance varies and is not optimal due to the intrinsic complexity of music genres. This complexity was further underscored by significant lyric similarities within and between genres. Future work could improve classification accuracy by incorporating more complex models and additional data types, such as artist backgrounds and release dates. This research lays the groundwork for advanced automated music genre classification using lyrical content.

Keywords

Sentence-BERT, Music Genre Classification, Machine Learning

1. INTRODUCTION

Music has been an indispensable component of human society, influencing cultural development since the early stages of civilization. As music evolved, so did its genres, a critical characteristic of music defined as “a category of artistic, musical, or literary composition characterized by a particular style, form, or content” [7]. In the digital age, genre has become crucial metadata for navigating musical databases. However, defining genre was a very difficult task because "genre is intrinsically ill-defined and attempts at defining precisely genre have a strong tendency to end up in circular, ungrounded projections of fantasies" [1]. Studies on music genres, especially those focusing on defining and classifying genres, had therefore been an important research area for decades. With the latest advancements in computer science, including artificial intelligence, machine learning, and natural language processing (NLP), new research possibilities have emerged. This study leverages these technologies, using NLP features such as tokens and embeddings, to explore whether music genres can be accurately classified based solely on lyrics. The technical details will be discussed in subsequent sections, with the primary goal to discover innovative methods for genre classification.

2. RESEARCH QUESTION AND PURPOSE STATEMENT

Research Question: **Could Sentence BERT lyric embeddings be used to classify genres among album rock, adult standards, alternative metal, alternative rock, and dance pop?**

This analysis leveraged lyrical and genre information retrieved from Kaggle, a dataset platform, and Spotify, a leading music streaming platform. In this research question, I selected the five genres because they were the most populous in the dataset, and there were enough samples to derive a meaningful answer to the question. The Sentence-BERT (Sentence Bidirectional Encoder Representations from Transformers) model was then used to extract and convert lyrics into embeddings, which are vectors of real numbers in a high-dimensional space, capturing semantic similarity and context. After generating embeddings, I passed the embeddings and corresponding genre labels of the songs into machine learning models such as random forests and multi-layer perceptron for training and testing. Ideally, I should have been able to analyze the performance of these models using metrics such as accuracy, error, precision, recall, and F1 score, etc., to verify if the research question was true.

This research question could have demonstrated the effectiveness of sentence embeddings on genre classification, offering insights into the relationship between lyrics and their corresponding musical genres. By exploring the capability of Sentence-BERT embeddings in capturing the semantic information of lyrical content and its correlation with specific genres, as well as the performance of different machine learning models on genre classification, this analysis could provide a new framework for automatic genre classification. It could also potentially assist future research in the field of music information retrieval, an interdisciplinary field focusing on methods and tools for processing, searching, and organizing music data using various features and metadata.

3. LITERATURE REVIEW

Understanding the fluidity of music genres and the impact of music lyrics embeddings for genre classification gave us a perspective on how music intersected with cultural, linguistic, and emotional aspects of human civilizations. Music is never static; its constant evolution has well-known effects on music genres. New genres appear frequently, and existing genres often undergo transformations, making modern music genre taxonomies extremely complicated [1]. With the advancement of recent technologies such as artificial intelligence and natural language processing, came innovative approaches, such as embeddings, to capture semantic information in text. Embeddings, as discussed by Mikolov et al. [9], were essentially dense vectors composed of floating-point values which were derived from textual data. These vectors were structured in such a way that the spatial relationships between them reflected the lexical relationships between words, capturing not just the words themselves but also the semantic information of sentences. This capability enabled a more sophisticated and nuanced machine understanding of text, facilitating tasks like sentiment analysis, machine translation, and document clustering by focusing on the deeper meanings rather than just the surface-level text.

Embeddings sounded very promising in terms of representing semantic information mathematically, but how were embeddings generated? Leveraging the Sentence-BERT model was an effective way to generate embeddings [11]. It was a modification of the pretrained BERT network that utilized Siamese and triplet network structures to derive semantically meaningful sentence embeddings which could be compared using cosine similarity. It significantly reduced hardware requirements for embedding generation with comparable accuracy to the original BERT model. SBERT models excelled in capturing the essence of lyrical content which contained cultural and emotional depth that was unique to genres. This characteristic made SBERT a good candidate for identifying lyrics similarities, which might also correlate with lyrical complexity, another important feature in music classification with an interesting ongoing trend leading to genre changes. Varnum et al. suggested that a preference for songs with simpler lyrics could be due to ease of memorization and transmission, with simpler, more repetitive music being perceived as more enjoyable, engaging, and memorable by listeners [14]. This observation implied that popular music genres may favor simpler lyrical content to align with listener preferences, leading to increased similarity within these genres due to a collective preference towards straightforward, catchy phrases and hooks. In short, embeddings generated by Sentence-BERT could be helpful in aiding in the classification process with higher accuracy and efficiency. More details on the Sentence-BERT model used will be discussed in later sections.

While embedding generation was an important pillar of this research, how did researchers analyze lyrics in the past? In the Music Information Retrieval (MIR) field, researchers prioritized tasks like genre classification algorithms and recommendation systems, considering lyrics as a crucial element alongside audio. MIR researchers had developed various machine learning approaches analyzing different aspects of music lyrics, with some focusing on keyword extractions. Tsaptisinos developed a Hierarchical Attention Network model for classifying music genres based on lyrics and demonstrated that this model could identify significant words or lines that contribute to genre classification [13]. This approach underlined the importance of lyrical content in distinguishing musical genres, pointing to specific themes and word choices as defining characteristics of each genre.

Some researchers focused on analyzing the correlation between audio and lyrics using robust neural network models. For example, Yu et al. proposed a deep learning solution that integrated temporal sequences to understand the correlation between audio and lyrics, thereby enhancing audio-lyrics cross-modal music retrieval (2018). This technique involved converting sequential audio and lyrics into a common semantic space, showcasing the possibility to combine textual information in lyrics with acoustic features for improving MIR systems [6]. Applications of such techniques led to more accurate music recommendations systems and search engines. It also helped with playlist generation that considered both the mood of music and theme extracted from lyrics, aligning with users' preferences and contexts.

Dialects and linguistic nuances were important in lyrics, and scientists were not ignoring them when it came to music classification tasks. Models such as the convolutional recurrent neural network (CRNN) model, as discussed by Choi, Fazekas, and Sandler (2016), combined the feature extraction capabilities of convolution layers of convolutional neural networks (CNN) with the sequential data processing capabilities of recurrent neural networks (RNN). This hybrid approach effectively handled complex lyric features such as dialects and other linguistic nuances, which were crucial for accurate music tagging in terms of cultural and regional differences [4]. Another solution that leveraged long-short term memory (LSTM) and SoftMax regression for semantic classification by Fourati, Jedidi, and Gargouri effectively captured emotional information in lyrics. They were able to encode complex emotional states in lyrics and facilitate a deeper level of genre classification beyond thematic and stylistic elements [5]. In summary, advancements in AI and machine learning made the solutions possible. They all demonstrated the potential in capturing the multifaceted nature of music lyrics and could lead to more powerful genre classification solutions that captured both linguistic diversity and emotions.

While the previous research was all leveraging state-of-the-art machine learning models extracting information from music lyrics, none of them mentioned using the Sentence-BERT model to preprocess the data. This research paper could serve as one of the first to leverage both sentence transformers and machine learning to see if the combination of the two created a competent solution in music genre classification. To answer the research question, I would implement the analysis in 3 steps: data loading & pre-processing, embedding generation, and analysis. The next section would discuss each of these three steps in detail.

4. METHODS

The research question aims to evaluate the efficacy of Sentence-BERT coupled with machine learning techniques in classifying music genres based solely on lyrics data. To address this question, the methodology is structured into three distinct stages:

- **Data Retrieval and Processing:** Initially, the study involves the collection and preprocessing of lyrics and genre data. This step ensures that the data is clean, well-organized, and suitable for further analysis.
- **Embedding Generation using Sentence-BERT:** The Sentence-BERT model is used to transform the preprocessed lyrics into semantic embeddings. These embeddings are designed to capture the nuanced semantic information inherent in the lyrics, which is critical for the genre classification task.
- **Model Training and Testing:** Finally, the prepared data is divided into training and testing subsets. Machine learning models are then trained on the training set to learn the relationships between the embeddings and their corresponding genres. The effectiveness of these models is then evaluated on the testing set, using standard performance metrics.

Please visit this GitHub repository for detailed implementation: https://github.com/thq12345/embeddings-music-genre-classification

4.1 Data

4.1.1 Primary Dataset

The primary dataset, named "57,650 Spotify Songs," was created by Joakim Arvidsson and retrieved from Kaggle, a well-known dataset platform [2]. Five genres (album rock, adult standards, alternative metal, alternative rock, and dance pop) were selected because they are the most populous genres in the dataset, and analyzing on them would derive more meaningful conclusions than genres with only a few songs in the dataset. Below are the column specifications after filtering:

- Artists: name of the artist, 167 unique values, no missing values.
- Song: title of the song, 19,105 unique values, no missing values.
- Link: link to the Spotify page for the song, 19,105 unique values, no missing values.
- Text: lyrics of the song, 19,105 unique values, no missing values.

In the context of answering the research question, I primarily used the “Artists” and “Text” columns from this dataset. The dataset was overall considered clean with minor duplications, which were discussed in the data preprocessing section.

4.1.2 Spotify Web API

Because genre information was missing in the primary dataset, another data source was required to complete the dataset needed for the research question. The Spotify Web API offers developers access to Spotify’s music database in a programmatic way. It allows users to integrate Spotify’s various functionalities such as playlist tracking, user data insights, and music information retrieval into their own applications [12]. This made it the perfect candidate for completing our dataset, as retrieving music metadata such as genre is one of its main functionalities. It is also free to use with a limited quota (more than what was needed for this project), which also made it the most cost-effective solution for this research. For details on the retrieval process, please refer to the Jupyter notebook. To summarize, I first looped through the dataset, extracting unique artist names, then I called the Spotify Web API for genre information on each artist. The Spotify Web API would return a list of genres associated with each artist, separated by commas. To simplify the analysis, the first genre in the list was extracted and used as the “main genre”. This decision will be further discussed in the limitations section.

4.2 Data Processing

4.2.1 Dataset Preprocessing

Due to the fact that the original dataset was mostly cleaned, there were only a few tasks that needed to be done to make the lyrics suitable for analysis. These tasks included converting all the words in the lyrics to lowercase and replacing all the “\n” (new line) characters with spaces. Additionally, columns that were deemed not useful for analysis, such as “Links,” were removed from the dataset. Standard natural language processing techniques such as stop word and punctuation removal were also performed to retain the essence of the music lyrics.

After preprocessing the lyrics data, joining the genre data was the next immediate step. As described in the Spotify Web API section, I used the API to retrieve genre information in a list, then extracted the first item from the list, and finally joined the genre information to the original data frame. For simplicity, I limited the research question to only classifying five genres; as a result, filtering was also used to eliminate songs that belonged to non-target genres. Below is an illustration of the number of songs in each of the selected five genres.

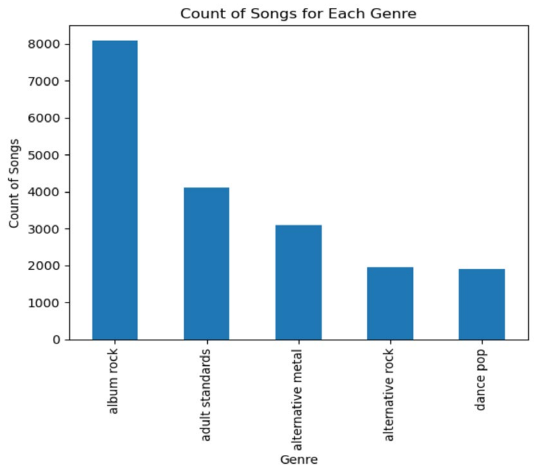

Figure 1. Count of songs for each of the selected music genres

Base on the chart, since the number of songs in each genre is heavily unbalanced, under sampling was done to balance the dataset prior to model training.

4.2.2 Embedding Generation

As described in the literature review, I used sentence-BERT, one of the latest technologies to convert lyrics into corresponding sentence embeddings which capture semantic information. In the past, leveraging Word2Vec was very popular for embedding generation [8]. However, using sentence-BERT in this research proved to be a more advanced, appropriate, and effective solution because Word2Vec only considered words on an individual basis without considering relationships with other words, whereas transformer models such as sentence-BERT are significantly more robust in considering the context of sentences when generating embeddings. This capability could possibly help us distinguish genres, as differences in semantic context are one of its key characteristics.

The sentence-BERT model deployed in this project was called “all-MiniLM-L12-v2”. It is a variant of MiniLM (Mini Language Model) developed by researchers at Microsoft with goals to create efficient and compact transformer models that retain the performance of larger models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) with significantly fewer parameters [15]. It was suitable for real-time applications and environments where computational resources were limited.

The “L12” in the model’s name indicated that this model had 12 transformer layers, maintaining a balance between efficiency and capability to capture semantic content from sentences. The model size was only 120 megabytes, containing 33 million parameters, fine-tuned from the base model “MiniLM-L12-H384-uncased” with a 1 billion sentence pair dataset containing text from Reddit, Stack Exchange, Wiki Answers, etc. While it was lightweight, performance-wise the base model performed better than the 109 million parameters BERT-base in numerous metrics such as CoLA (Corpus of Linguistic Acceptability), MNLI-m (Multi-Genre Natural Language Inference - matched), and RTE (Recognizing Textual Entailment).

In summary, this model was the perfect candidate for tasks such as text classification and sentence similarity assessments, which were tasks I was conducting in this research. Due to the hardware limitations for this research (which will also be discussed in the limitations section), the reduced size of this model also increased inference speed, making the embedding generation process feasible on personal computers. In this research, the embedding generation code was executed on a NVIDIA RTX 4090 GPU, and the tasks took less than 5 minutes to complete.

4.3 Machine Learning

In literature review, I discussed how researchers in the music information retrieval field implemented complex machine learning models analyzing music lyrics. In this research I am following a similar approach. Due to the hardware limitations and time constraints, after careful consideration I decided to deploy random forest and multi-layer perceptron model for genre classification. Both are signature models in their respective model groups (trees and neural networks) and should provide insight into how suitable machine learning models are in classifying music genres.

4.3.1 Random Forest

Random Forest is a bagging ensemble machine learning model that utilizes multiple decision trees' majority voting mechanism to complete classification or regression tasks. The reason Random Forest was a suitable model to train in this case is because it is well-suited in dealing with high-dimension datasets [3] and therefore should perform better than simpler models such as logistic regression. The Random Forest model mechanism includes randomness when selecting subsets of data and features for building each tree, making it even better at dealing with the nuances of genre classification.

In this research, I first under sampled the songs within each genre as mentioned in the previous section. Then, the pre-processed dataset was split into 70% training and 30% testing data. I used the sklearn package to initialize a random forest classifier [10]. Here are the hyperparameters I used:

- 'max_depth': maximum depth of each tree in the forest. The options were none, 10, 20, 30. The goal is to find the best trade-off between bias and variance since deeper tree could potentially capture more complex patterns but risk overfitting.
- 'min_samples_leaf': minimum number of samples required to be at a leaf node, controls the size of the tree. The options were 1, 2, 4. Finding the optimal value helps reduce model complexity and chance of overfitting by not allowing model to chase outliers.
- 'min_samples_split': minimum number of samples required to split an internal node, preventing a tree from overfitting and limiting growth. The options were 2, 5, 10. This set of options control how detailed a model should be, smaller numbers allow more complex trees whereas larger number restrict the model to more general patterns.
- 'n_estimators': number of trees in the forest, a higher number of trees could lead to better performance but also takes more time and computational resources. The options were 100, 200, 300. This set of options offers a scale to balance model performance and computational efficiency. More trees can reduce variance component of error but at the cost of increased memory and time.

Hyperparameter tuning via grid search was used to find the best set of hyperparameters for the Random Forest model. After training using the 70% data, I verified the performance of the model using the 30% testing data. See the results section for performance metrics and interpretation.

4.3.2 Neural Networks

Neural Network is an important type of model in machine learning that simulates human neurons in the brain. It consists of layers of interconnected nodes "neurons," each processing information received from its inputs and passing the output to the next layer of neurons. Each neuron carries a weight value, and the network gets trained by adjusting these weights. In this research, I implemented a multi-layer perceptron (MLP), a type of neural network that is specifically designed for supervised machine learning tasks. It consists of an input layer, multiple hidden layers, and one output layer and is capable of learning complex patterns, such as the ones in music lyric embeddings. Despite its relative simplicity among neural network architectures, the multi-layer perceptron provides an initial insight into the performance of neural network-based models for music genre classification tasks.

In this research, we used PyTorch to manually build an MLP network. Please refer to the notebook for specific code; here are the parameters we used:

- Input Layer Size ('input_size'): Number of input features, set to X_train.shape[1] so it matched the dimension of the input data.
- Hidden Layer Size ('hidden_size'): Set to 128 neurons, this represents the number of neurons in the hidden layer of the multi-layer perceptron. Choosing 128 neurons strikes a balance between computation efficiency and ability to learn from data.
- Number of Output Classes ('num_classes'): Set to the number of unique classes existing in the dataset. In the notebook, we set it to 'len(np.unique(y_encoded))', which should be 5 because we had 5 genres to classify.
- Learning Rate ('lr'): This represents how much the model changed per training iteration. I set it to 0.001 because it's small enough to avoid overshooting minima but also large enough to train efficiently.
- Number of Epochs ('num_epochs'): Number of times the entire training dataset was passed forward and backward through the neural network. In this case, I set it to 1000 because high number of epochs ensure deep learning and convergence when using a relatively small learning rate.
- Loss Function: Cross-entropy loss was used to evaluate errors. This is ideal for classification tasks because it measures the difference between predicted probabilities and actual distribution.

The model is then trained with pre-split training and testing set, same as random forest. Please see the results section for performance metrics.

4.4 Text Similarity Calculation

In addition to training machine learning models to classify genres, I implemented some visualizations demonstrating sentence similarities and differences within and between genres using cosine similarities, a way to measure how similar two sentences are. Embeddings convert sentences into high-dimensional vectors, and cosine similarity calculates the cosine of the angle between the two vectors, with value between -1 and 1. 1 means two sentences are literally the same while 0 means no similarity and -1 means complete dissimilarity. It's a great indicator for if two songs are similar, and please refer to next sections for details.

5. RESULTS

5.1 Random Forest

Below is a table for random forest model performance:

Table 1. Random Forest Performance

	Precision	Recall	F1-Score	Support
Adult Standards	0.55	0.56	0.56	608
Album Rock	0.34	0.18	0.24	582
Alternative Metal	0.5	0.49	0.49	566
Alternative Rock	0.37	0.47	0.42	544
Dance Pop	0.46	0.56	0.50	567
Accuracy			0.44	2867

I performed a grid search, as previously mentioned, for hyperparameter tuning. The best set of hyperparameters found was {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 300}. As we can see, the overall accuracy was 44% across all genres, indicating that it correctly predicted 44% of all 2,867 testing samples. Among each genre, the model was relatively better at predicting adult standards (F1 score 0.56) and dance pop (F1 score 0.5), while it performed poorly at classifying album rock (F1 score 0.24). Overall, even though the model did not perform well in terms of reaching 80% to 100% accuracy, it was still significantly better than random chance, which would be 20%.

5.2 Neural Network

Below are performance metrics for multi-layer perceptron with under sampling:

Table 2. Multi-Layer Perceptron Performance

	Precision	Recall	F1-Score	Support
Adult Standards	0.58	0.52	0.55	608
Album Rock	0.35	0.34	0.35	582
Alternative Metal	0.43	0.44	0.43	566
Alternative Rock	0.39	0.43	0.41	544
Dance Pop	0.46	0.47	0.46	567
Accuracy			0.44	2867

This multi-layer perceptron is a basic one, consisting of an input layer, a hidden fully connected layer with ReLU activation function, and another fully connected layer for output. The model was trained on a dedicated GPU. Compared to the random forest model, the overall performance was roughly the same. However, the performance in classifying each genre was more balanced. Adult standards performed the best (F1 score 0.55), while the lowest performing genre was album rock (F1 score 0.35), which was significantly better than the performance by the random forest model. With additional time, hyperparameter tuning could be conducted on this model to potentially enhance its effectiveness.

5.3 Lyrics Similarities & Visualizations

As observed in the model performance, while the models clearly were able to identify and classify genres based on lyrics, there was still significant room for improvement. In this research, cosine similarities provided a compelling reason as to why these machine learning models were not performing in the best possible way.

Below is a chart displaying the intra-genre mean similarity, which measures the lyric similarity across all songs within each genre:

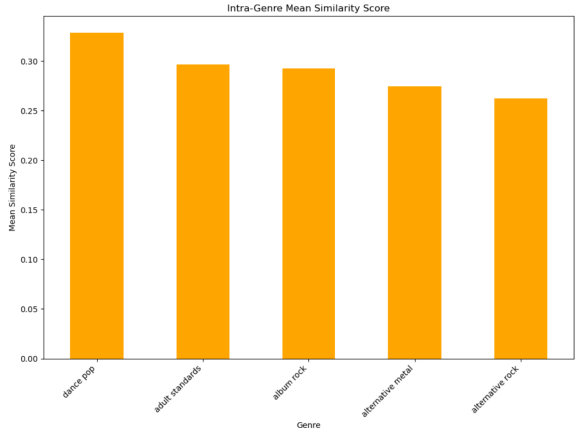

Figure 2. Intra-Genre Mean Similarity Score

As we observed, all five genres exhibited low intra-genre similarity, with the highest only around 0.41 (dance pop) and the lowest around 0.36 (alternative rock). This indicates that songs within each of these selected genres are quite different from each other.

Now, let's examine the inter-genre similarities, which measure the lyric similarities between songs across different genres:

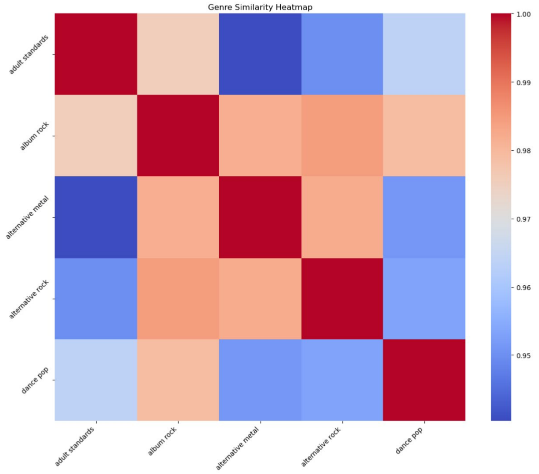

Figure 3. Inter-Genre Mean Similarity Score Heatmap

The above chart is a heatmap that represents the lyric similarities between each genre, along with a color scale where red indicates higher similarity and blue indicates less similarity. From the chart, we can see that dance pop is similar to album rock. Album rock is relatively more similar to other genres than any other genre, while alternative metal and adult standards are the least similar pair. Despite the color scale ranging from blue to red, visually representing a significant variation in cosine similarity, it's worth mentioning that the lowest value represented in this chart is still over 0.9. This indicates that all genres are somewhat similar to each other; it's only the relative degrees of similarity that differ.

With these visualizations, we can derive the reason behind the relatively low performance for both models: This task is extremely difficult when songs are very diverse within each genre but extremely similar between genres. It's difficult for models to capture a distinct pattern for each genre and use it to classify effectively.

6. CONCLUSION, LIMITATIONS & NEXT STEPS

Based on the machine learning models I trained and the visualizations, the answer is yes, Sentence BERT embeddings can be used to classify genres between album rock, adult standards, alternative metal, alternative rock, and dance pop. This research also verified the feasibility of pervious research on utilizing machine learning for music analysis. However, Sentence BERT embeddings alone might not yield good performance on genre classification because genres are very hard to define, as mentioned in the literature review.

With additional time, the next steps would be:

- Try more models: While I implemented a multi-layer perceptron model in this research, there are many more advanced models we can leverage such as deep neural networks. Hyperparameter tuning and adding additional layers, such as convolution layers to the existing MLP model, could be something worth trying.
- Add additional features: As we can see from the methods section, the only feature I am using is sentence embedding from lyrics. Since genre classification is a difficult task, additional features such as artist information, temporal information such as song release date could greatly help, especially on time-sensitive genres such as adult standards, which consists of American pop and jazz music from the early to mid-20th century.

There are also several limitations to this research:

- Hardware: More advanced models such as large language models are hard to run on personal computers, so basic models were selected in this research.
- Dataset: There are several limitations associated with this dataset.
    - The dataset owner did not disclose how the data were extracted from Spotify database as well as criteria for selecting songs to be included in the dataset. More transparency would greatly benefit the research.
    - I assumed that all songs in the dataset are in English. This limitation is because I cannot obtain language information for each song from Spotify Web API.
    - The numbers of songs for each genre in the dataset are heavily imbalanced, as a result, I had to under sample data prior to model training. With additional music data, it would be expected that model accuracy will increase.
    - While I was able to extract a list of genres associated with each song, I only used the first one in the genre list as the "main genre," which can be problematic when it comes to genre classification. This is an important assumption for simplicity, but it could also inevitably negatively impact our model performance.

7. ACKNOWLEDGMENTS

My thanks to Prof. Scott Crossley, Langdon Holmes, Wesley Morris and Jess Boyle for guidance during the research and paper-writing process.

8. REFERENCES

[1] Aucouturier, J. J., & Pachet, F. (2003). Representing Musical Genre: A State of the Art. *Journal of New Music Research*, 32(1), 83–93. DOI= https://doi.org/10.1076/jnmr.32.1.83.16801

[2] Arvidsson, J. (2024). 57,651 Spotify Songs [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/joebeachcapital/57651-spotify-songs/data

[3] Capitaine, L., Genuer, R., & Thiébaut, R. (2019). Random forests for high-dimensional longitudinal data. arXiv preprint arXiv:1901.11279. DOI= https://doi.org/10.48550/arXiv.1901.11279

[4] Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2016). Convolutional Recurrent Neural Networks for Music Classification. *arXiv preprint arXiv:1609.04243*. DOI= https://doi.org/10.48550/arXiv.1609.04243

[5] Fourati, M., Jedidi, A., & Gargouri, F. (2023). A deep learning-based classification for topic detection of audiovisual documents. *Applied Intelligence*, 53(4), 8776–8798. DOI= https://doi.org/10.1007/s10489-022-03938-x

[6] Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z., & Komarova, N. L. (2018). Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society open science*, 5(5), 171274. DOI= https://doi.org/10.1098/rsos.171274

[7] Merriam-Webster. (2023). Genre. In *Merriam-Webster.com dictionary*. Retrieved from https://www.merriam-webster.com/dictionary/genre

[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *arXiv preprint arXiv:1310.4546*. DOI= https://doi.org/10.48550/arXiv.1310.4546

[9] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781. DOI= https://doi.org/10.48550/arXiv.1301.3781

[10] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp. 2825-2830.

[11] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*. DOI= https://doi.org/10.48550/arXiv.1908.10084

[12] Spotify. (2023). Spotify for Developers: Web API Reference [Web API documentation]. Retrieved April 13, 2024, from https://developer.spotify.com/documentation/web-api/

[13] Tsaptsinos, A. (2017). Lyrics-Based Music Genre Classification Using a Hierarchical Attention Network. *arXiv preprint arXiv:1707.04678*. DOI= https://doi.org/10.48550/arXiv.1707.04678

[14] Varnum, M. E. W., Krems, J. A., Morris, C., Wormley, A., & Grossmann, I. (2021). Why are song lyrics becoming simpler? A time series analysis of lyrical complexity in six decades of American popular music. *PLOS ONE*, 16(1), e0244576. DOI= https://doi.org/10.1371/journal.pone.0244576

[15] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. *ArXiv*, abs/2002.10957