Effects of Word Embedding Generation Processes on Music Genre Classification

Carlos Calderon
Information and Data Science
University of California, Berkeley
ccal@ischool.berkeley.edu

Jorgiana Lopez
Information and Data Science
University of California, Berkeley
jorgiana@ischool.berkeley.edu

Abstract

Musical genres assist in organizing songs, artists, or albums into broader groups with shared musical properties. In particular, genre labels are characterized by parallels in form and style, both in their sound and lyrical content. Whilst previous works in genre classification have focused on multi-modal approaches, we sought to classify song genres given only a songs' lyrical content. We utilized the Genius API to scrape song lyrics based on five genres: Country, R&B, Rock, Pop, and Rap. Thereafter, we experimented with GloVe and DistilBERT-generated word-level embeddings to generate baseline and improved models. Our best models were trained on DistilBERT-generated embeddings, achieving 95.26% on training and 54.34% on test data.

1 Introduction

There has been plenty of academic research on song genre classification, however most of these studies come from the music information retrieval (MIR) field. As a result, prior work has largely focused on a multi-modal approach (Oramas et al.), that is, they usually pair lyrical content alongside auditory content of a song to inform their classification models.

Past NLP-related research on lyrics has had access to vast datasets obtained from websites such as LyricFind or databases such as the Wasabi Song Corpus (Tsaptsinos, 2017) and (Fell et al.) (2019). However, these lyrical databases have recently blocked access to lyrics due to copyright infringement laws. Due to this, we resorted to an alternate method where we scraped song lyrics from the well-known lyrics website, Genius, using their Genius API. From Genius, we extracted a total of 3,000 songs. To facilitate the task at hand, we removed any songs that obtained any non-English lyrics, resulting in a final set of 2,633 songs.

Unlike text in news articles or online user reviews, lyrical text does not resemble conventional prose text structure (Watanabe and Goto). Instead, lyrical text structure resembles that of poems, where each part of song is analogous to a poem stanza (i.e. verse, chorus, bridge). Due to this, common classification techniques in NLP often do not work as well for lyric-based classification tasks. One of the main goals of this paper is to explore how common algorithms in NLP today transfer to a music genre classification task.

Initially, our goal for this project was to summarize song lyrics solely based on their lyrics, reminiscent of (Fell et al.) (2019). Unfortunately, that proved to be an overly ambitious task considering it would have been an unsupervised learning task. We then came up with the idea to do a supervised learning task as we would now have labels for our training and test data to classify song lyrics into their respective genres. We chose to focus on the following five genres: Country, Pop, R&B, Rock, and Rap.

2 Background

This work draws on earlier work on text analysis, text classification, lyric-based music classification, more specifically on genre detection and classification. Caparrini et al. (2020) use decision trees and random forests trained on auditory song data to achieve a 48.2% accuracy when classifying amongst 29 electronic dance music subgenres. Similarly, Pimenta-Zanon et al. (2021) also use decision trees trained on auditory data to achieve 90% accuracy when classifying amongst 10 genres. Works in the MIR field have now started to implement deep learning techniques for genre classification with promising results. Sigtia and Dixon (2014) achieved an 83% accuracy in genre classification using a song's auditory data as input to a simple feed-forward neural network.

On the same token, Hsu et al. (2021) utilize convolutional neural networks (CNNs) to achieve a 60.6% accuracy when classifying among 30 genres. Moving away from the MIR and into the NLP domain, (Pandey and Dutta) (2014) used Support Vector Machines (SVMs), k-Nearest Neighbor (kNN), and Naive Bayes (NB) models to perform genre classification on a corpus of 116 songs, achieving a maximum accuracy of 85.7% when classifying amongst 3 genres. Their main approach was bag-of-words, in which the model inputs where the term frequency-inverse document frequencies for each word in the song corpus.

With this in mind, we sought to implement deep-learning techniques in NLP to analyze lyrical data and classify it based on four popular genres. More specifically, we were interested in how these deep-learning models would compare to similar genre-classification models. Coming in, we understood that accuracy through lyrical data might not be as high as models trained on auditory data, however, we wanted to explore the informational power of lyrics by themselves. Future work in the field might seek to incorporate deep learning with both lyrical and auditory components involved.

3 Methods

3.1 Data

As previously mentioned, our data was composed of scraped song lyrics from Genius, specifically, their Genius API. Our dataset consisted of song lyrics across five genres: Country, R&B, Rock, Pop, and Rap. The data was class imbalanced, thus, we randomly sampled 233 songs for each genre, ensuring that each class was well represented in each model's training. After balancing our dataset, we carried out a standard 80/20 train/test split. Our models were trained on batches of 20 songs. Preliminary exploratory data analysis revealed to us that the average token length per song was 200 words per song; therefore, we limited each song's vocabulary to solely include the top 200 words. Initially, these models were fed in lyrical text with positional tokens, but performance was found to be much better when these tokens were removed (i.e. chorus, verse).

Additionally, we experimented with removing repetitive words in a song, as the over representation of some word in a song would skew that word to be associated with the song's genre. Interestingly, we found that different models performed differently when given lyrics with and without repeated words.

3.2 Word Embeddings

A word embedding is a numerical vector that represents a given word. Word embeddings are an important notion in NLP, as they allow textual data to be mapped to numerical data, that can then be fed into common machine learning algorithms. A common approach in NLP research is to utilize pre-trained word vectors such `word2vec` or `GloVe`, but recent work in encoder-decoder architectures have seen the rise of transformers (Vaswani et al., 2017).

One such example is the BERT (Devlin et al., 2018) model, a multi-layer bidirectional Transformer encoder. In particular, BERT is jointly conditioned on both left and right context in all layers of the model's architecture (Vaswani et al., 2017). Simply put, BERT generates word representations that are actively informed by the words around them; these are BERT's contextual embeddings. A downside to generating embeddings through BERT and similar transformer-based models is that they take far too long to train when given a large corpus – due to the high number of parameters.

To mitigate this, we used DistilBERT, a lightweight version of BERT (Sanh et al.,

2019) which has a reduced number of 66 million trainable parameters compared to 110 million parameters in BERT. In addition to being smaller, DistilBERT is 60% faster than BERT at inference time. In light of this, we were interested to see how different word embeddings affected the performance of our models. Thus, we opted to use the 200-dimensional pre-trained GloVe word vectors and trained our own embeddings through DistilBERT.

3.3 Baseline Models

Long short-term memory neural networks (LSTMs) are common in NLP, however they have been improved on by the introduction of ”transformer” encoder-decoder architectures. These transformers-based models are computationally costly – even lightweight versions like DistilBERT. Clavié et al. (2021) were able to achieve similar accuracy rates in classification tasks using single layer LSTMs when compared to BERT and trained on relatively small datasets. Given that our dataset was small – relative to common corpus size for NLP tasks – we decided to test whether this was the case for our specific classification task as well. As such, our baseline models include the most-common classifier and a single-layer LSTM neural network with 200 non-trainable weights using 200-dimensional GloVe word embeddings for a total corpus size of $\approx$ 3000 words.

3.4 Improved Models

Given the aforementioned differences in prose and lyrical text, we took an iterative approach when improving our baseline models. To begin, we sought to improve the performance of the LSTM baseline model by implementing a bidirectional LSTM (BiLSTM) layer. BiLSTMs introduce a second layer in which information is able to be harnessed from two different directions; typically known as the past and future (Zhou et al., 2016).

We found no improvements when using the BiLSTM – with or without attention. We initially believed this model would be an improvement as it would capture backward textual relationships in lyrics, but their lackluster performance on test data proved that this was not the case. A similar trend was observed when we trained the BiLSTM with embeddings trained through DistilBERT.

Given the failure of BiLSTMs for our classification task, we decided to focus on refining the LSTM GloVe model. We found that adding a dropout layer greatly increased model performance. Through a hyperparameter grid search, we found the optimal dropout rate to be 0.5. Furthermore, we found that model performance improved when we allowed the embedding vectors to be retrained in the training step.

Our final improvement was to incorporate attention layers in each of our models. We implemented both additive and multiplicative attention layers. Additive attention is a mechanism originally proposed by Bahdanau et al. (2014) that uses a single layer, feed forward network and calculates the attention alignment:

$$f_{att}(h_i, s_j) = v_a^T \tanh(W_1 h_i + W_2 s_j)$$

where h_i is the current hidden state, s_j the previous hidden state, and W_1 and W_2 are randomly initialized matrices whose parameters are trained by the attention network.

The multiplicative attention mechanism was proposed by Luong et al. (2015); in short, this attention mechanism differs from additive attention only in that the operations are made through matrix calculations that make it more speed and space efficient.

We also incorporated a self-attention layer as proposed by Lin et al. (2017), here, we assume that we don’t have access to the previous hidden state and instead we attend to only the given sequence. Unfortunately, we found that this attention mechanism failed to aid our model in the classification task. This can be explained by the global importance of a word in the context of multiclass classification, specifically when a word can be present in multiple classes.

4 Results and Discussion

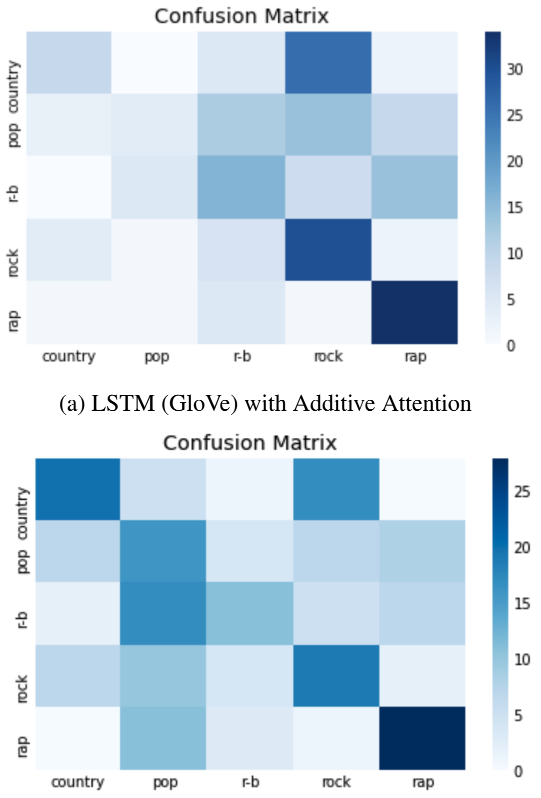
The results displayed below in Table 1, illustrate how each model performed against

Models	Train Accuracy	Test Accuracy	Dropout	Embeddings
LSTM w/ Additive Attention	95.26%	54.34%	Yes	DistilBERT
Single layer feed-forward NN	79.30%	44.34%	Yes	DistilBERT
LSTM w/ Additive Attention	44.71%	41.51%	Yes	GloVe
LSTM	55.77%	41.04%	Yes	GloVe
BiLSTM Multiplicative Attention	80%	41.04%	No	GloVe
BiLSTM w/ Additive Attention	66.67%	37.74 %	No	GloVe
BiLSTM	71.09%	33.02%	No	GloVe
LSTM	52.61%	32.55%	No	GloVe
Most Common Classifier	32%	32%	No	N/A

Table 1: Model performance (measured by % accuracy) against testing and training sets, sorted by descending order on test accuracy. Here, we can appreciate the improvements of using embeddings trained by DistilBERT.

training and test sets. Starting from the bottom up of Table 1, we can see how the models with no dropout layer display sub-optimal performances on test set. Moreover, we can also see how BiLSTMs seem to overfit the data – in turn – displaying some of the highest accuracy rates on training sets, but generalize poorly on the test set. Interestingly, when adding a dropout layer, the BiLSTM models performed even worse.

Models with additive attention outperformed those with multiplicative attention, with the exception of the BiLSTM. More importantly, however, models trained on embeddings trained by DistilBERT outperformed those trained on GLoVe embeddings. Figure 1 presents confusion matrices for the best models based on the input embeddings. Both LSTMs correctly classified a high number of rap lyrics. The GLoVe-trained LSTM misclassified a high number of Country songs as Rock. Meanwhile, the DistilBERT-trained LSTM exceeded the GloVe-trained LSTM in classifying Country, Pop, and R&B lyrics in performance, but also confused close to half of R&B lyrics as Pop, in addition to misclassifying some Country songs as Rock.



(a) LSTM (GloVe) with Additive Attention

(b) LSTM (DistilBERT) with Additive Attention

Figure 1: Confusion matrices for the best performing models.

5 Conclusion

With all of this in mind, we have seen how common algorithms in the NLP world today apply to a novel lyrical dataset. Although accuracies for this task were not as high, this was to be expected. Indeed, similar research in genre classification has only been able to achieve high accuracies when pairing lyrical and auditory data. Here, we took an NLP-heavy approach at genre classification, using only the textual information retrieved by lyrics site Genius. We found that embeddings trained by DistilBERT produced better results.

Next steps would be to take advantage of the robustness of the main BERT algorithm. Along the same lines, taking advantage of other transformers-based architectures and embeddings.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Antonio Caparrini, Javier Arroyo, Laura Pérez-Molina, and Jaime Sánchez-Hernández. 2020. Automatic subgenre classification in an electronic dance music taxonomy. *Journal of New Music Research*, 49(3):269–284.

Benjamin Clavié, Akshita Gheewala, Paul Briton, Marc Alphonsus, Rym Laabiyad, and Francesco Piccoli. 2021. Legalmfit: Efficient short legal text classification with lstm language model pre-training.

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

Michael Fell, Elena Cabrio, Fabien Gandon, and Alain Giboin. Song lyrics summarization inspired by audio thumbnailing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 328–337. INCOMA Ltd.

Wei-Han Hsu, Bo-Yu Chen, and Yi-Hsuan Yang. 2021. Deep learning based edm subgenre classification using mel-spectrogram and tempogram features.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation.

Sergio Oramas, Francesco Barbieri, Oriol Nieto, and Xavier Serra. Multimodal deep learning for music genre classification. 1(1):4–21. Number: 1 Publisher: Ubiquity Press.

Ayushi Pandey and Indranil Dutta. Bundeli folk-song genre classification with kNN and SVM. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 133–138. NLP Association of India.

Matheus Henrique Pimenta-Zanon, Glaucia Maria Bressan, and Fabrício Martins Lopes. 2021. Complex network-based approach for feature extraction and classification of musical genres.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Siddharth Sigtia and Simon Dixon. 2014. Improved music feature learning with deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6959–6963.

Alexandros Tsaptsinos. 2017. Lyrics-based music genre classification using a hierarchical attention network.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Kento Watanabe and Masataka Goto. Lyrics information processing: Analysis, generation, and applications. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, pages 6–12. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.