Cracking the code of musical genres:

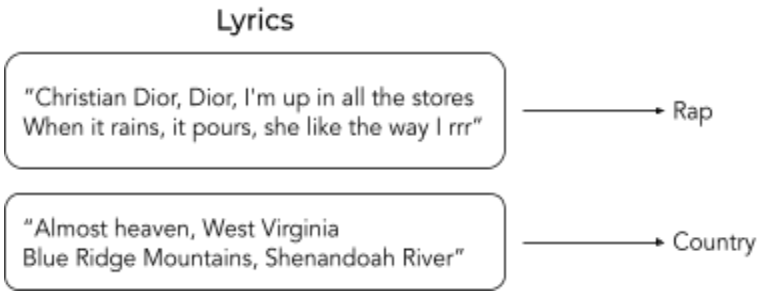
Unveiling the linguistic secrets hidden within lyrics using LSTMs!

By: James Nuci, Justin Liu, Yan Lashchev

Decoding Musical Artistry – Not a simple task!

Lyrics are much more than mere words in a song. They're a special cocktail of inventive narratives, day-to-day lingo, and subtle acknowledgments of societal trends. These components can differ drastically across various music genres. Our goal is to dig deeper and see if a specific type of learning model, known as a Long Short-Term Memory Network (or LSTM), can actually discern these unique genre-specific styles in song lyrics. But we're pushing it a step further, aiming to discover how important these lyrical styles really are when it comes to correctly identifying a song's genre. In this post, we'll delve into past work, our own experiments, and potential strategies for enhancing this intriguing AI challenge. The key driver behind this exploration is its potential to help create AI systems that are more sensitive to cultural nuances and free from biases. This could pave the way for innovative educational tools, using the power of AI to study the interplay between music and culture from a linguistic perspective. Additionally, fields that use technology to understand and interpret crucial information can benefit from effective language models.

A simple visual of the classification:



Past Attempts & Current Problems

Figuring out music genres from song lyrics alone is a tough task in natural language processing (NLP), mainly because audio features usually give us a lot of useful clues. Previous attempts to solve this puzzle have tried using methods like term frequency-inverse document frequency (TF-IDF) and simpler models like logistic regression, nearest neighbors, and support vector machines (SVMs). But these attempts haven't really hit the mark, with the best models barely scraping a 50% accuracy rate at best when classifying five or more genres.

Upon inspecting the current state of the field, it's clear that LSTMs, with their proficiency for handling sequential data - much like the stream of words in a song, seem to hold the most promise for this task. However, we've noticed some considerable stumbling blocks in the existing work. A major problem stems from the disparate distribution of songs across different genres. For example, within certain datasets, the pop genre might be represented by more

than 100,000 songs, while country songs might only number around 15,000. This uneven balance could be throwing off the performance of these models. On top of that, we've got a hunch that trying to classify songs into a wide array of genres spanning multiple decades might be overcomplicating the task, thereby negatively impacting accuracy. Streamlining the task to focus on roughly four genres across a single decade might make the job more manageable for these models. Lastly, and most crucially, previous attempts appear to overlook an important factor - diving deep into a linguistic analysis to get to the heart of the problem. That's where our approach brings something new to the table.

Data and Preparation

The lyrics dataset we're working with was obtained from Kaggle, thanks to a dedicated user who collected every song from genius.com. The dataset is rich in detail, providing information such as the artist, number of views, lyrics, and most often, the genre. These details helped us craft a balanced dataset for our task. Since we're focusing on English lyrics, any non-English songs were filtered out. Our attention is specifically on the decade spanning 2010 to 2020, and we've curated the top 25,000 songs from each of the following genres: Pop, Rap, R&B, Rock, and Country, based on view counts. To better process the data, we created two versions of the dataset. In both versions, we undertook the usual NLP cleanup tasks, like removing non-ASCII characters, newline characters, and musical markers such as "chorus", "verse", "hook", and so on. The second version of the dataset underwent additional steps - removal of common 'stop words' and lemmatization, a process that reduces words to their base or dictionary form.

Musical Model Face-Off: Mood or Sound?

Model 1: Word sequence Bi-LSTM

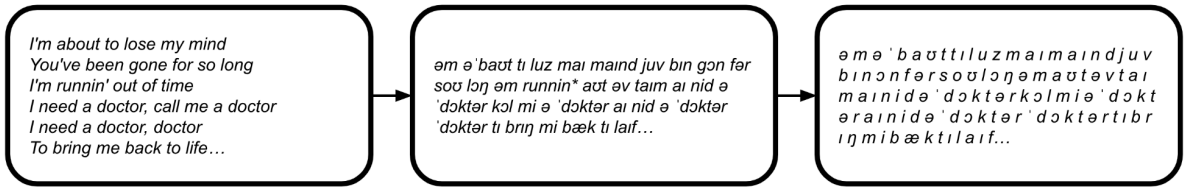
We trained a genre classification model using song lyrics with the LSTM architecture. The Keras library was used for preprocessing input sequences and encoding genre tags into numerics like: {Country: 0, Rap: 1, Pop: 2, R&B: 3}. The first layer, an embedding layer, processed words within the sequences via a 100-dimensional vector. Cross-validation showed this was superior to 200 or 300 dimensions in terms of training efficiency and overfitting. Training our embeddings as part of the model gave us faster training time and higher validation accuracy than using pre-trained glove embeddings. A bidirectional LSTM (biLSTM) was chosen over a standard LSTM as it provided better results, understanding both past and future context. A 300-hidden unit biLSTM was eventually selected, providing the best performance on a separate validation set. We experimented with dropout, but even low levels slowed training and reduced accuracy, so it was discarded. The model also includes a global max pooling layer for dimensionality reduction and computational efficiency and a final dense layer for output. The softmax function then converted the output to probabilities for each genre.

The highest validation accuracy achieved was 73.34% with training accuracies between 77% and 80%.

Model 2: Phonological character LSTM

Our second method focuses on the phonological representations of the lyrics (i.e., how the lyrics *sound*) to classify the song genres. For example, the lyrics of a pop song may sound different from the lyrics of a rap or country song due to rhyme patterns and/or the types of words being used.

In order to retrieve the pronunciations of our lyrics, we need to convert the lyrics to IPA (short for International Phonetic Alphabet) form. The IPA is a standardized way of representing the speech sounds (called phonemes) of any language – you can check out this link to play around with these sounds. The Carnegie-Mellon University Pronouncing Dictionary provides pronunciations for over 134,000 words, and the *eng_to_ipa* library uses this dictionary to transcribe text to IPA. An example of our pipeline is shown below.



After transcribing raw lyrics to phonemes (speech sounds in IPA form), we cleaned the data by removing words without pronunciations and non-IPA symbols. We then encoded each phoneme as an index, converting them into a numerical form suitable for our model.

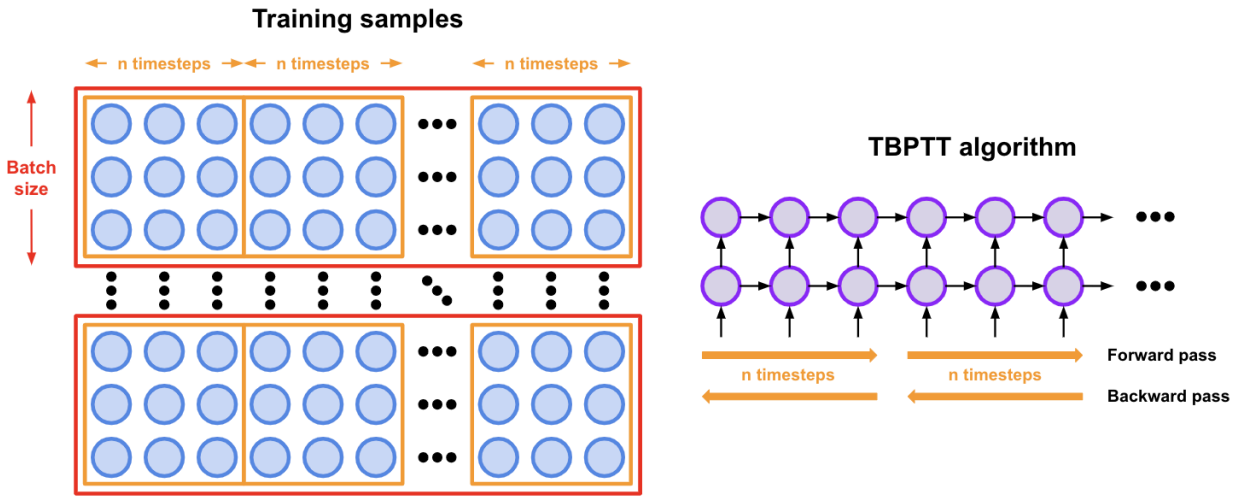
Due to the extensive length of phoneme sequences (some songs have thousands of phonemes, with the longest over 23,000), character-based models posed memory issues and vanishing gradient problems. To mitigate these, we limited each sequence to a maximum of 2,400 phonemes (using the IQR method and rounding to the nearest hundred). This helped retain song information while making the training process manageable. We also used truncated backpropagation through time (TBPTT) to process multiple characters at once during each forward pass and update weights during backpropagation, reducing memory demands.

Our phoneme model had three key layers:

- 50-dimensional embedding layer for identifying influential phonemes
- 64-unit bidirectional LSTM layer for processing characters from both directions
- 4-unit output layer with softmax activation for genre probability outputs

This model, owing to the larger size of phoneme sequences, had significantly fewer parameters (31,650) than the word-based model (over 12 million).

To train the model, we divided our training samples into batches (in red), with each batch containing mini-batches (in orange) of characters or timesteps (in blue) of size *n*. The model is trained on each mini-batch until it has gone through all of the timesteps. Then we repeat this process for every batch.



In our case, we trained the model in batches of 32 songs, processing 100 characters/timesteps at a time until all 2,400 characters/timesteps were covered. It's important that the total timesteps be divisible by the number of characters processed at a time, hence the rounding of the maximum characters to a multiple of 100.

Each training epoch took over 30 minutes, achieving a convergence training accuracy of approximately 41.5% after five epochs. However, the test accuracy on 20,000 song lyrics was a disappointing 25.1%, barely better than random guessing.

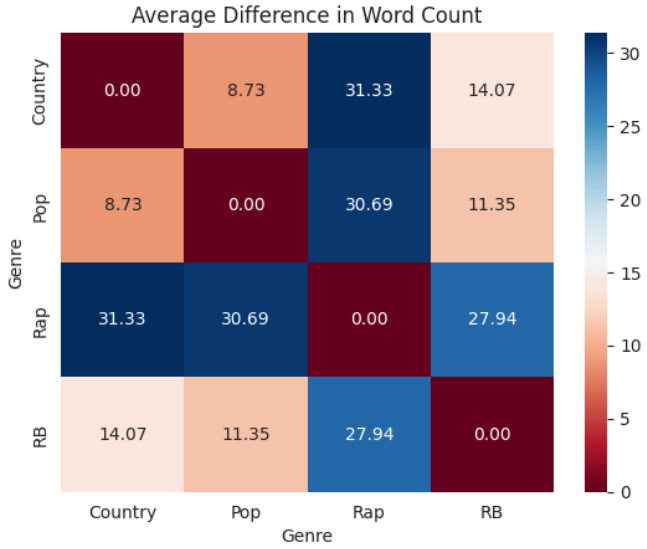
Uh-Oh, Rock n Roll!

Following a thorough analysis, we made the decision to exclude the Rock genre from our study. The primary rationale behind this was the broad classification approach adopted by Genius, where all rock sub-genres - including metal, punk, grunge, and others - are simply labeled as 'rock'. Among all the genres and their respective subcategories, rock exhibited high degrees of variation and uniqueness from observation. We initially thought that the pop genre would have this issue – but it did not. Further complicating the issue, rock lyrics didn't appear to have any strong, unique identifiers. In fact, much of the rock lyric content we examined seemed versatile enough to fit into almost any other genre category, with the possible exception of country. Moreover, rock music frequently borrows elements from other genres. For instance, the rhythm and rhyme schemes often found in rock songs bear a striking resemblance to those in rap. Given these factors, including rock in our study was creating more confusion than clarity, leading us to focus on the remaining genres.

Other Genres

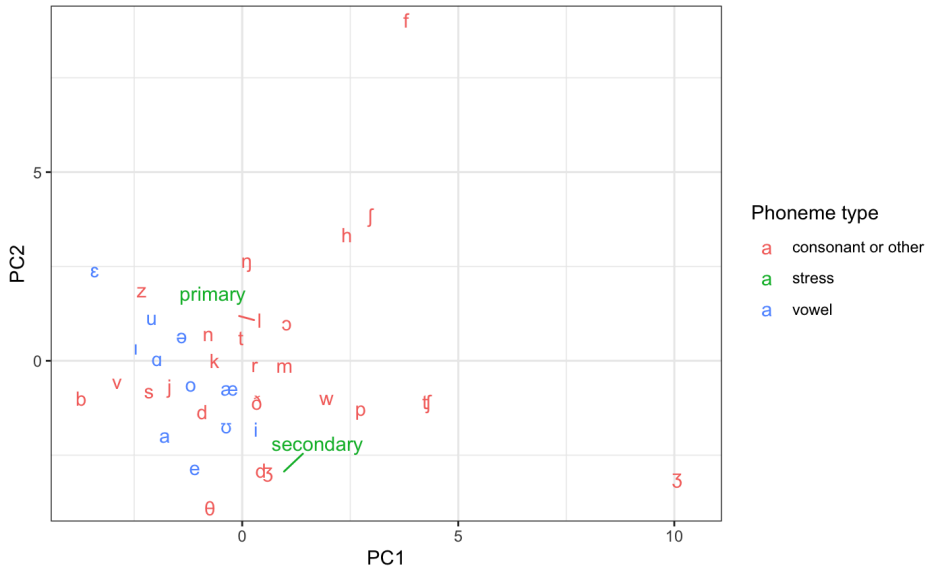
We picked Pop, Rap, and R&B as our main focus, then added Country into the mix to give the model a real test. We initially thought that Country and Rap would be the easiest to classify. The country genre often uses specific words like 'river', 'mountain', 'fishing', 'horse', and 'woods', which aren't commonly found in other genres. Meanwhile, Rap stands out due to its heavy use of slang and profanity. This made for a fascinating challenge and an interesting mix of genres to explore.

Diving into the Linguistic Depths of Lyrics



One of the simplest ways to analyze differences between song lyrics from a linguistic perspective is through word counts. In order to check for significant differences, the word counts by genre were collected and compared with each other. Differences in word count (for the same word) were then averaged and displayed in the aside heatmap. The largest differences were between rap and all other genres, suggesting that rap is most easily distinguishable from any other genre when considering language, as we predicted. This lines up well with our model performance. When analyzing accuracy by genre, rap songs had an accurate classification rate of 0.94, while country, R&B, and pop had rates of 0.78, 0.66, and 0.59, respectively. Taking the average of the averages displayed in the heatmap by genre gives us a useful statistic, quantifying how dissimilar each genre is from the others. The values are as follows. Rap: 29.99, Country: 18.05, R&B: 17.79, Pop: 16.93. Our model uses words as part of a sequence, not as individual tokens, but analysis of word counts alone proved to be an accurate predictor for our model's performance.

Although the phonological character-based model didn’t perform nearly as well as the word-based model, we can still look at the embeddings that it learned for each phoneme. To visualize this, we take the 50-dimensional embeddings and use Principal Component Analysis (PCA) to reduce it down to 2 dimensions. This allows us to visualize each phoneme’s embeddings on a Cartesian plane.



The plot above shows how the embeddings are (roughly) distributed in space, where embeddings close to each other are similar and embeddings far away from the origin (0, 0) tend to have more influence on the model's predictions. We note that the vowels (in blue) are more clustered toward the origin compared to the consonants (in red). Vowels appear frequently in the data and have similar frequency distributions across genres, which could lessen their influence when distinguishing different types of music.

Taking a closer at the consonants, some of the fricatives (*ʒ*, *f*, *h*, *ʃ*, *θ*) and affricates (*ʤ*, *ʧ*), which are characterized by hissing sounds, tend to be further away from the origin. These phonemes appear very infrequently across all genres, with *f* occurring in approximately 1.5% of all phonemes and the others appearing less often. Interestingly, some of these phonemes are more likely to be used in rap songs compared to other genres, though we cannot draw any concrete conclusions beyond this.

Both of the results above seem to point to the idea that phonemes that appear infrequently in the data generally have larger embeddings. To test this hypothesis, we carried out a Pearson's correlation test between the distances from the origin to each 2-D phoneme embedding and the mean frequency of each phoneme and found a significantly negative correlation (r = -0.380, p = 0.02). The sizes of our embeddings may be more related to how often the individual phonemes occur rather than meaningful patterns between co-occurring phonemes.

Given the poor performance of our results, we would need to modify our phoneme model. One thing, in particular, is that we removed all words that did not have pronunciations in the CMU dictionary. The model may have overlooked adjacent patterns between phonemes since these words were omitted. Another modification could be integrating the phonemic properties alongside the words in one model, possibly improving predictions and getting more informative embeddings.

Pros & Cons

Pros

- We curated a relatively balanced dataset and were able to achieve better accuracy than previous attempts at this task.
- We were able to figure out that some genres such as rock should be handled in a more robust manner to handle all the subcategories.
- We shed some light on some of the linguistic features that previous work failed to do.

Cons

- We were hoping to predict on more genres, but all in all, we felt successful.
- We were hoping the phonological model would be stronger, but we likely did not consider the best ways to implement it.
- We likely could have had more success if we had more powerful machines.

Final Thoughts

Our problem of genre classification was approached in two ways. Both methods, using sequences of words or sequences of phonemes, were able to give insight into the linguistic nature of song lyric data. Our first model's performance reinforced the idea that LSTM architectures are suited for classifying lyric data and sequences of text in general. Specifically considering lyric data, our results make it apparent that over time, lyrics of a certain genre can develop patterns within their language. It would be interesting to analyze the results of a similar model on data from other decades. We could explore how the language of 2 genres compare to each other over a long period of time. Sound is always going to be an important factor in determining the genre of a song. Our chosen model, biLSTM, was not effective in classifying songs by their phonology, but another architecture may be able to. Considering our analysis of embeddings and major components, patterns in sound may be more prevalent within different parts of a song's composition. An RNN architecture may be suited for learning and classifying a chorus, intro, bridge, etc., or even classification by phonology as we attempted. There is much more to be discovered within the sounds and syntax of songs, and we hope our work will inspire others to conduct research of their own!

References:

[1] 5 Million Song Lyrics Dataset
[2] Genre Classification using Word Embeddings and Deep Learning
[3] Deep Neural Networks: A Case Study for Music Genre Classification
[4] Lyrics-based Analysis and Classification of Music
[5] Music Genre Classification using Song Lyrics
[6] The CMU Pronouncing Dictionary
[7] Stateful LSTM in Keras