Jai Lunkad, Sam Anderson, Makar Dubovskov

Analyzing Lyrical Diversity by Artist

Introduction:

The main objective of this project is to assess the lyrical diversity of musicians who have published anywhere from a few, to hundreds of songs. While many musicians have cultivated a specific audience that likes to listen to a very specific kind of song, or a particular musician focuses on a narrow set of topics. Thus, these musicians make very lyrically similar songs again and again and gain popularity. While there are many aspects of a song such as the sound (overtone, timbre, pitch, amplitude, duration), melody, harmony, rhythm, and more, we will strictly focus on analyzing lyrics during this project. In doing so, we will try to understand if artists use similar words consistently in their songs. We will use natural language processing methods to analyze the similarity between songs by a given artist.

This measurement for lyrical diversity will be calculated as follows: for each song made by an artist, we will calculate the cosine similarity with every other song. After which we will sum up all of these figures then dividing by the amount of comparisons, effectively finding the mean similarity. In addition we will track the two most similar songs for every artist. Although there are several conceptual problems when solely using lyrics to get information about a song. The main problem is that the lyrics may not be indicative of any other characteristic of a song, like genre and rhythm. While a human listener can see two distinct songs, the machine may see surprisingly similar songs even if the ordering of words, and the emotional ambiance is

completely distinct.For example, a pop song might get a techno makeover where the techno song is mostly only beats but the few lyrics that it has, are from the pop song. In this case, the machine would think that these songs are pretty much the same, whereas somebody who loves that pop song might hate the techno remix. Another problem is that a lot of lyric intensive songs, like pop, rap, and sometimes rock, have similar topics yet, because words are not strictly matching, the difference can be exaggerated in some cases and songs may appear original even if they are not lyrically. There is also the intrinsic fault within tf-idf in that two songs can appear similar in cosine similarity, but that does not reflect the order of those words. The role of stop words in lyrics may also be significant to the flow and structure of lyrics. Many songs in the dataset are also not in English. There is also the question of what to do with remixes, since they will be similar to one another and the original in terms of lyrics. Lastly, some lyrics have parts that include repeated use of onomatopoeia-like phrases, such as ‘la la la’. These are very common in many pop songs and sometimes represent a large part of the lyrics for a given song.

Due to our decision to use tf-idf and cosine similarity, there is no easy way to alleviate or control all of these problems without using other machine learning tools. A lot of these factors will also impact our results simultaneously. That said, we do have some control over the latter two problems. For our model, we decided to be quite harsh in terms of originality and keep remixes and the onomatopoeia-like phrases in our model. This will lower the originality of artists that overuse those, and may impact artists who specialize in a language other than english.

Input Data

All data used during this project is sourced from a dataset containing about forty thousand songs, and over a thousand artists from this Kaggle page, we will form the basis for all the songs

we will use in this project. From this dataset, the data is stored in a .jl file containing a list of json objects where each object has keys for an individual song, conveniently with the key “song”, and the key mapping to the lyrics is called “lyrics”. We define an artist’s “vocabulary diversity” as 1-(average similarity) between songs for an artist.

It is also worth noting any additional song can be added to this data set by downloading the lyrics with the Genius Lyrics API. Using the artist name and sont name, any additional song in their repository can be included since forty thousand songs is by no means comprehensive, and will not be enough to satisfy everyone’s curiosity.

Methodology

The first order of business was creating a TF_IDF matrix among all songs, so we could compare any song with any other song. For our TF-IDF, it was decided not to normalize the TF values. The reason behind this is that the length of these songs at 2-5 minutes means that the TF values will not be massive even if they are not normalized. After that, it was clear the data set requires some cleaning because the actual name of the song is appended with the artist name, so we attempt to compile a list of all the artists by comparing the amount of songs with a longer name, with a shorter name. This allows us to differentiate between a key starting with “Lil-yachty” and “Lil-wayne” because there are more song keys that start with “Lil” than either “Lil-yachty” and “Lil-wayne”. Similarly, the amount of songs that start with "Bebe-rexha” and “Bebe” are equal, so these must be the same artists, or this artist only has one song. While this will normally be an issue, since we are targeting originality, we chose to ignore artists with less than 5 songs. This seems to eliminate the issue of an artist having too few songs, or songs that start with the same word being treated as different artists. For example, if Kendrick Lamar has 2 songs that start with “money” the algorithm would otherwise think “Kendrick-Lamar-money” is

the name of an artist sowe bubble these songs with the shorter artist name, and artists with too few songs are still removed.

Finally, while there were a few exceptions that were easy to manually remove, we have a cleaned list of 713 artists and well over 10,000 songs, where each song is organized by artist, and ready to be compared. One problem that occurs with using lyrics to determine lyrical diversity is the dilemma of remixes, extended versions, radio edits and other small changes to a song that causes a re-release. These songs normally have nearly identical tf-idf therefore, the question is whether or not to count them in or not. Removing remixes has the benefit of eliminating the noise caused by a remix being released, where the original artist was not involved in the remix, but if the artist has some influence with the remix, their score would be artificially inflated. While we did not remove altered versions, there could be additional research if the original lyrics were strictly observed.

Next we can calculate similarity between songs already sorted by artist. In addition to the average similarity by artist, we tracked the two most similar songs by each artist. From here it is possible to use the created data model to compare any song, or run a textual search among this dataset.

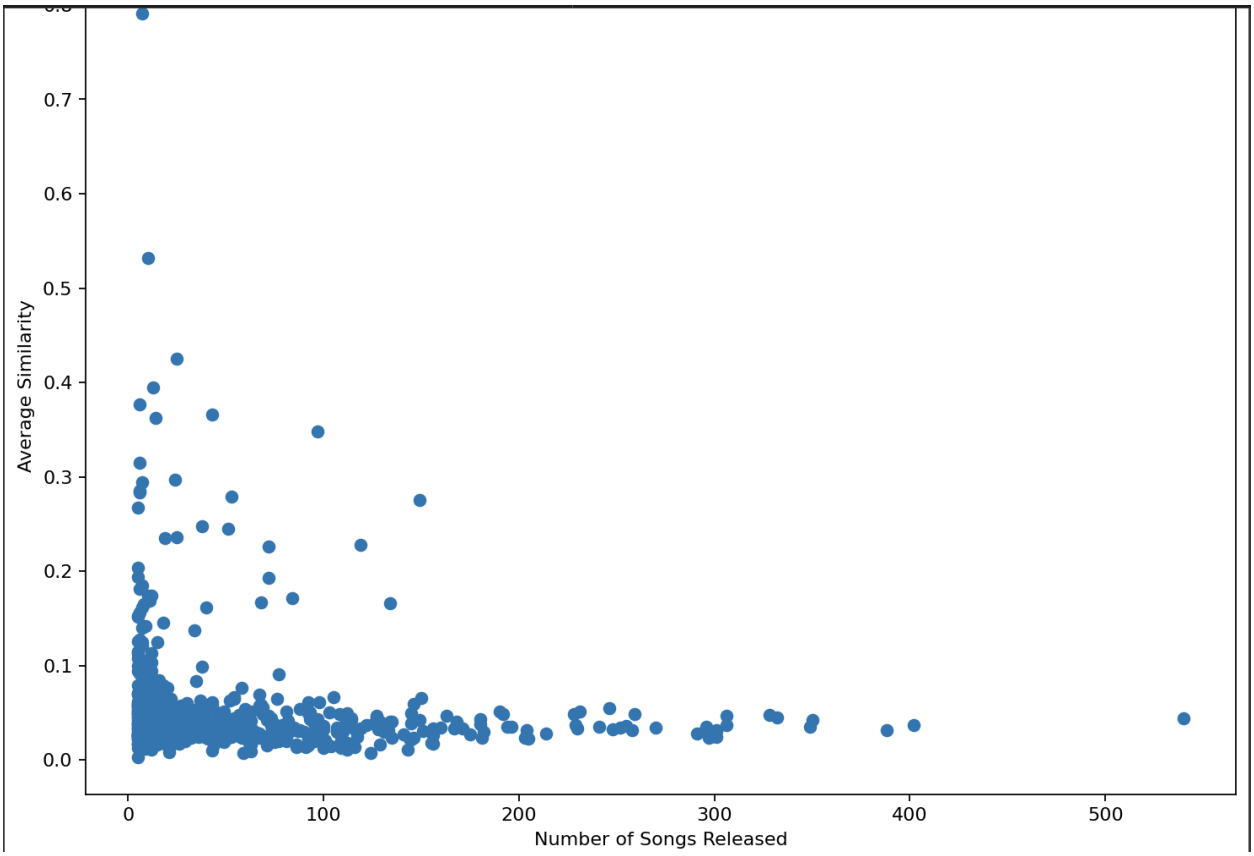
Lastly, with some final processing, such that data is appropriate for graphing, to calculate trends within the originality results. From there we will look at trends, interesting findings and observations during the course of this project.

Findings

Several initial hypotheses were confirmed and validated. For instance, two songs by Tinashe, namely ‘Joyride’ and ‘Ooh La La’ yielded ~95% for cosine similarity between two of their

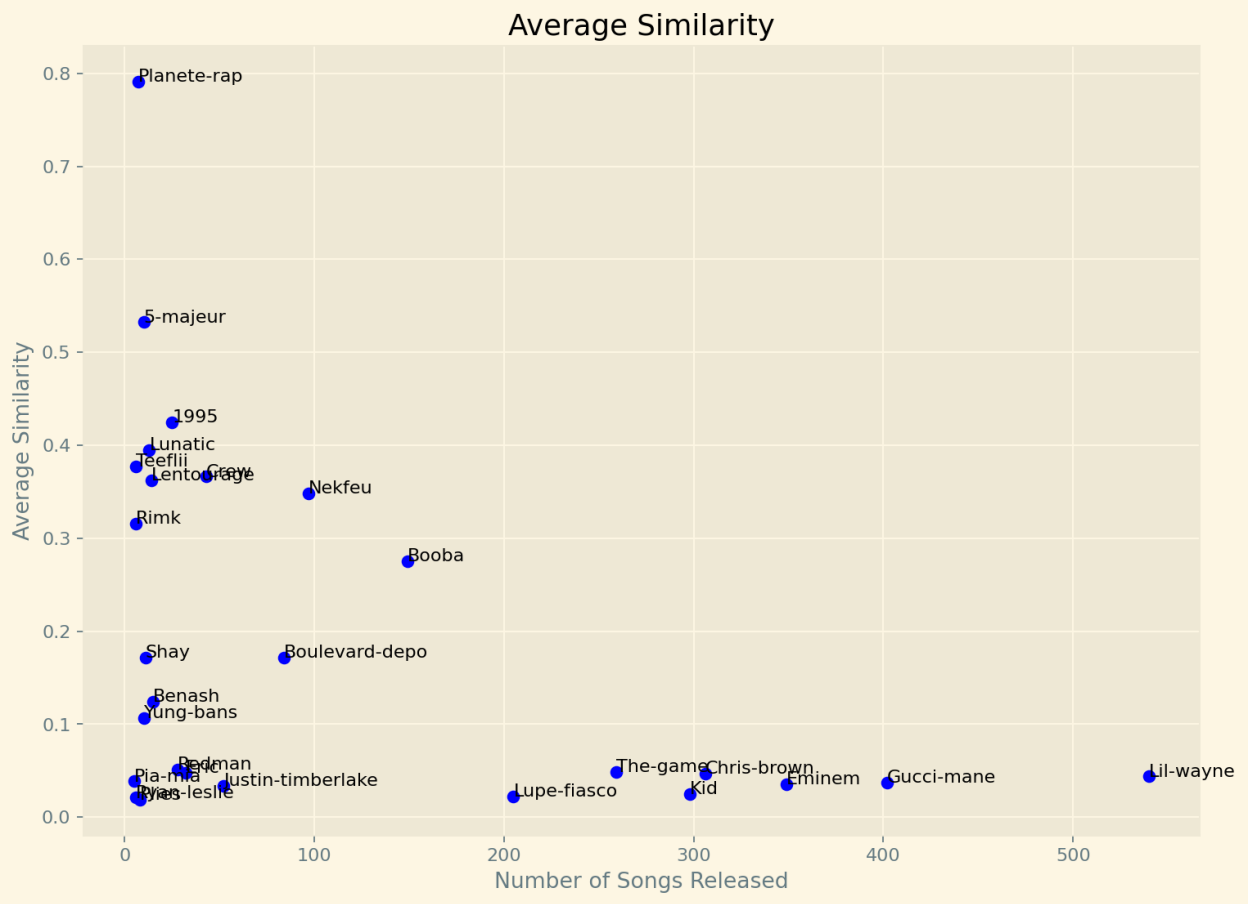
songs despite them being two distinct songs. This appears to be in no small part because both songs use the phrase ‘la’ for its chorus excessively, as predicted.

Below is the full output. It is not too useful in its current state apart from the trend that is presented, which will be discussed later. However, an initial finding is that not that many artists have an average similarity of above 0.2. This would indicate that overall the artists presented in this dataset have fairly original lyrics to a degree and at least relative to each other. Additionally, there appears to be a negative trend in the data although it does not appear to be linear. Later we will explore an exponential trend.

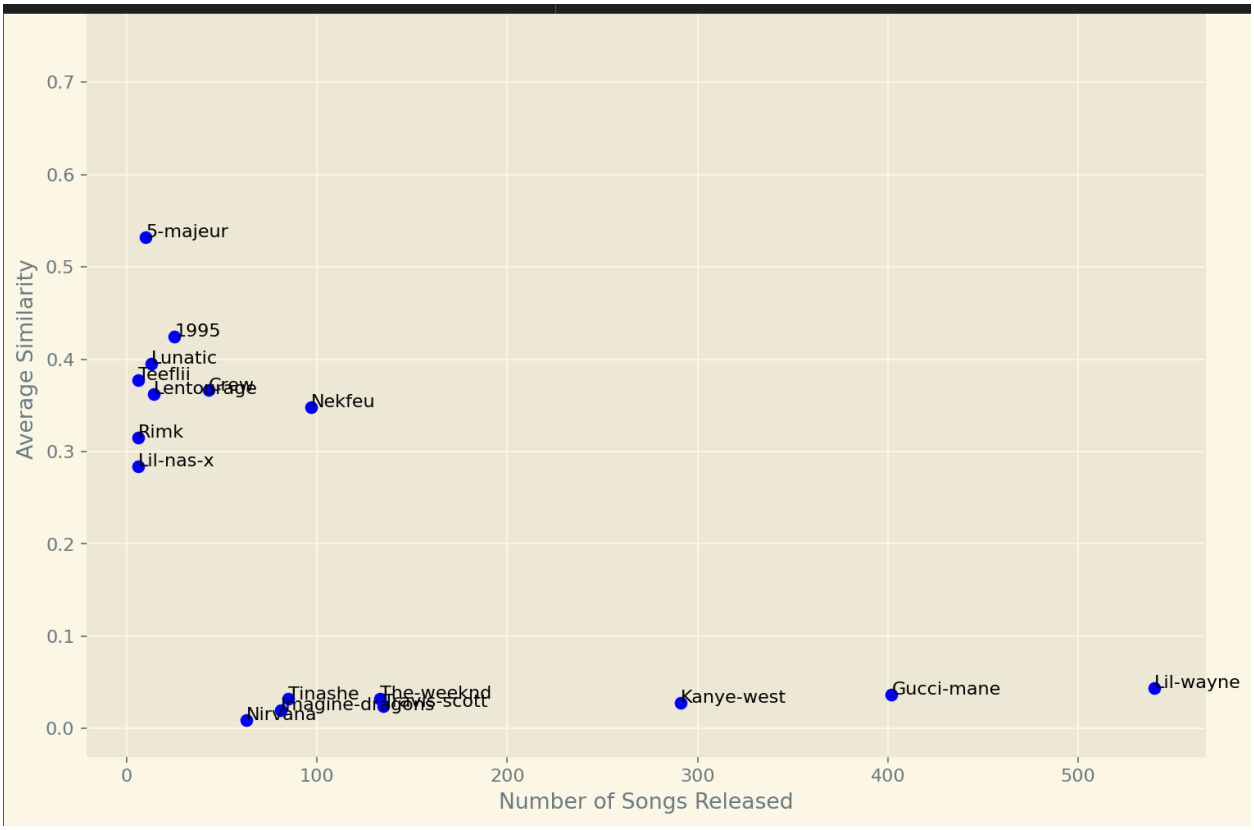


Below is the same graph but with labels, and randomly selected data points in the densely clustered region from the graph above . The two artists with the highest average similarity, have written their songs in French, whose words almost never appear in songs written in English. Since we computed tf-idf over the entire corpus of lyrics, this meant that all the words in French songs have an abnormally high idf of at least log(40k/5). This in turn increased the weights that each word in their lyrics got which meant that their cosine similarity would be much larger than those of other songs. The fact that we only cleaned English stop words probably also increased the similarity between the French songs, since a lot of repeating articles like ‘le’ and ‘un’ would remain and increase similarity.

This trend does not seem to hold up with Spanish songs, likely because they have a much larger footprint in this dataset. On the other hand, as an artist has written more songs, all of them have diverse lyrics. We believe this is because all of these have long careers with varied inspirations so, they have a need to diversify their lyrics.

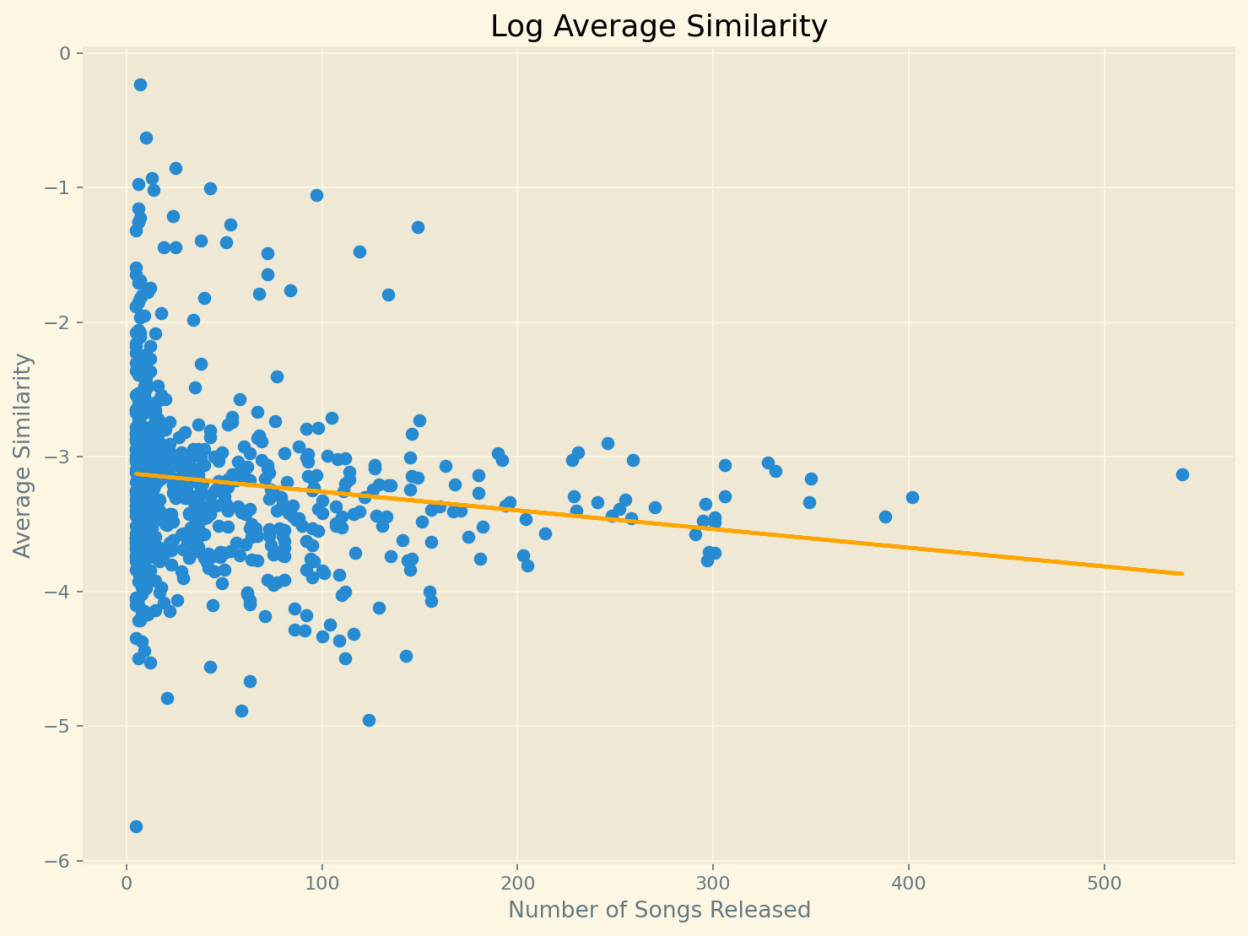


The graph below is slightly changed from the previous example . Here we see that, contrary to earlier expectations, “Tinashe” achieved an average cosine similarity far below 0.1 for about 90 songs. This shows that in our model, even if two songs are very similar, it does not necessarily mean that the originality of an artist overall will be low. “Nirvana” also features on this graph. They are very well known for their lyrics and they are approaching 0 in terms of average similarity, which is the expected result.



While not in the previous graph there are other artists with lower scores like Skrillex RadioHead, and with the least similarity, Iggy Pop has an average similarity of 0.0032.

The next graph uses the log() of the artist's originality so we can more confidently use a linear trend on what might be an exponentially decaying trend. We perform this by taking the log for all values on the Y axis, by first taking the log(), then plotting all points and finally, we calculate the line of regression.



Taking the logarithm of the average similarity points to the more original artists having more songs. If we accept such a finding, then the people who produce a lot of music are able to also make their lyrics relatively original each time. The trend is however weak, with an R value of -0.1186, and this could have been influenced by the assumptions we made in our model. This could mean that in reality all of the artists investigated have similar originality on average, and the outliers are the ones influencing the trend. One finding from the first graph is that all artists with more than 100 compositions are below 0.3 in average similarity, which supports the trend expressed earlier. In fact, all composers with more than 200 songs are below 0.1 in average similarity, and most are below .05.

The artists below that trend line but with fewer songs than those artists along the line are not necessarily more original: the artists on the trend line make enough songs that there are bound to be recurring words, which brings their similarity up. Perhaps a point of evaluation could be to consider penalizing the originality of the artists with fewer songs.

Conclusion

With this code and analysis, we believe we applied a useful tool to gain a series of interesting metrics. Our methodology is simple in concept and application, though that does leave a lot of room for improvement. There are quite a few limitations to using lexical analysis in general in regards to songs as was mentioned earlier, but probably the single largest source of simplification is the order unaware characteristic of tf-idf and cosine similarity. Clearly if we were able to take order into account, many songs would be recognized as much more different.

One possible vector for future analysis of songs is to analyze not words, but the actual sound. This can be done by applying weights to different discrete representations of sound, such as a note or pitch, in a manner similar to tf-idf. The problems with doing so would carry over from our methodology, but it could provide another way to look at originality. An additional problem with this application is that there are probably far fewer datasets that have this kind of information than for song lyrics.

To summarize our findings, we found that overall, a lot of the artists we investigated have on average original lyrics relative to each other, and artists with extensive careers have more diverse vocabulary. It is still unknown if diverse lyrics imply long careers, or long careers force the artist to explore different lyrics to avoid burnout but, either way, diverse lyrics bode well for long careers.