Analyzing Lyrical Content to Classify Popularity of Modern Pop Music

David Smart, Naman Agrawal, and **Vaidehi Dalmia**

ds3361@columbia.edu, na2603@columbia.edu, vd2302@columbia.edu

Abstract

Lyrical content is one of the most important characteristics of a popular song, as it adds semantic meaning, context, and a method of social sharing to the instrumental. In this study, we sought to explore the relationship between lyrics and popularity, using presence on the Billboard Hot 100 chart as the metric of popularity. We found that there is a non-trivial relationship between the lyrical features of songs – such as word count, complexity, and top lemmas – and their popularities. Furthermore, we discovered that predicting popularity within-artist can be achieved by analysis of lyrical content, but genre may be a confounding factor. Finally, we identified significant correlations between certain lyrical features and their presence in popular songs over time.

1. Introduction

In our current era of Instagram-caption obsession, trending Twitter moments, and instantly viral memes, the propensity for a song's lyrical content to lead it to success seem higher than ever. Even if you've never heard the song itself, lyrics like "started wearing less and going out more" and "rain drop, drop top" (from *Hotline Bling* by Drake and *Bad and Boujee* by Migos, respectively) might ring a bell. Catchiness, relatability, and/or attachment to online trends can all play a factor in boosting the reachability of a song's lyrics.

In fact, lyrics have become such a focal point in recent years, that an extremely comprehensive lyrical-analysis website, Genius.com, has become the go-to place to delve into the deeper meanings behind almost any song on the market. Common listeners, musicians, and literary experts alike collaborate to explore the cultural and personal significance of words and phrases.

Yet, while the ability of lyrics to be very digestible and incredibly intriguing is easily observable, identifying the exact features that make a song popular is a trickier subject. This is mostly because a song's lyrics are closely tied with its musical composition. Is *Hotline Bling* popular because of its catchy chorus, Drake's voice, or the danceable instrumental? While we understand that each of these contributes to the popularity of a song, we hypothesized that lyrical content alone is enough to be a significant predictor of popularity.

For our research, we decided to explore three questions related to the relationship between popularity and lyrical content. The intent was to build non-trivial binary classifiers for popularity and to discern significant trends in the lyrical features (text only) of popular songs across the span of a recent decade.

(1) Can we classify a song's popularity from its lyrics, in general?
We sought to train a classifier that could use lyrical features to classify any recent song as popular or unpopular.

(2) Can we classify a song's popularity from its lyrics, for a given artist?
We sought to train a classifier that could use lyrical features of an artist's discography to classify the same artist's song as popular or unpopular.

(3) How have popular song lyrics changed over the decade of 2006-2015?

We sought to discover significant correlations between the presence of specific lyrical features and the passing of time over a recent decade.

In the upcoming methodologies and analysis, we hope to gain insight into how a deeply emotional and extremely lucrative industry is affected by the bars and stanzas that make its heart beat.

2. Previous Work

There has not been a lot research focusing on predicting popularity of music. Mussmann et al. [2] find that lyrical information can be useful in predicting song rankings. They combine their set of trajectory labels with KNN and regression to predict song rankings on the Billboard 100. After an initial feature selection to find the most predictive features, the authors predict a set of trajectory labels. Their results are slightly better than their random walk baseline.

Dhanaraj and Logan[3] extract both acoustical and lyrical information from songs and use SVM and boosting classifiers to identify popular songs. They conclude that lyric-based features are slightly more useful than the acoustic features in predicting popularity.

Thierry Bertin-Mahieux, Daniel P.W. Ellis et al.[7] predict the year a song was released based on audio features. Though they don't do lexical analysis, their approach is helpful for us to answer our third question. They use Vowpal Wabbit(VW) to perform regression by learning a linear transformation using gradient descent. Their results are significantly better than their benchmark prediction using k-NN classification.

The approach taken by Gauvin[1] most closely relates to ours. He tracks how the popular music tends to sway with the attention economy. Gauvin studies how the composition of hit singles has changed over the last 30 years, as well as the difference in characteristics exhibited by an artist's popular and unpopular songs. Both these approaches provide useful reference to answer our second and third questions. Unlike our approach however, Gauvin uses acoustic information instead of lyrical information to predict song popularity. He finds significant correlations between the year that a song is on the Billboard 100 and four features: number of words in title, main tempo, time before voice, and time before title. For any given artist, there was no significant difference between attention economy characteristics exhibited by popular and unpopular songs.

3. Data Used

We used three different datasets of our own making to approach the three research problems stated above. The general method was to first create a list of song titles that the corpus would be made of, with about half corresponding to each label of "popular" or "unpopular." The song title and artist was then fed into the Genius API to retrieve the web page with the lyrics for that particular song. The lyrics were then taken from the page using Beautiful Soup, a Python library for web scraping. Finally, the lyrics were cleaned by removing extra spaces and informational strings such as "[Verse 1]" with a regular expression that removed parts of the text with brackets.

3.1 10-Week Corpus

The 10-Week Corpus was created for the first question, whether we could classify a song's popularity from its lyrics. This consisted of 351 songs: 186 popular and 165 unpopular. We wanted to look only at recent songs in order to hold time period constant and find any patterns culturally relevant to today's music in order to be most useful for future marketing.

To get recent popular songs, we mined Billboard's Hot 100 charts, the most comprehensive measure of the success of individual singles. We defined recent as being on the charts in the past 10 weeks, between February 18, 2017 and April 22, 2017, as this was a large enough time gap to give a substantial dataset yet also not too far back in time to mix any potential time-sensitive trends.

To mine the song titles and artists, because Billboard does not have an official API, we used an unofficial Python API created by Github user *guoguo12*.

To get the unpopular songs, for each popular song picked via Billboard, we manually picked out a recent unpopular song created by the same artist. This method was also used by Gauvin in "Drawing Listener Attention in Popular Music" [1] when

comparing attention economy characteristics of popular songs to unpopular songs. We defined unpopular as having the lowest play count on Spotify among possibilities that fit the criteria above. Furthermore, if the popular song was on an album, then we took the song with the least play count from that same album. This ensures a degree of consistency throughout the dataset between unpopular and popular songs under the assumption that taking similar songs by the same artist maintains a similar breakdown among genres, artistic styles, and lyrical characteristics, thus reducing the effects of confounding variables.

Among the 186 popular songs, 165 counterparts were found, with some missing due to impeding reasons such as the artist only having one song or the artist not having any recent songs.

3.2 Influential Artist Corpus

The Influential Artist Corpus was created for the second question, whether we could classify, for a given artist, the popularity of a song from their discography. We intended to create datasets for multiple artists and compare them, so we picked three artists to begin with. The criteria for an influential artist was to be musically and culturally relevant for a long period of time (up until now) and to have a large body of work to increase the size of our sample.

We looked at Billboard's Artist 100 chart and picked Drake, Rihanna, and Coldplay, three prolific artists from different genres who have all enjoyed time on the charts for years. We used their entire discographies and labeled them as popular if they had ever been on the Billboard Hot 100 and unpopular if not. This resulted in 81 songs (37 popular, 44 unpopular) for Rihanna, 112 songs (68 popular, 44 unpopular) for Drake, and 66 songs (17 popular, 49 unpopular) for Coldplay. The Coldplay dataset was not analyzed further due to the small sample size and the significant difference between number of popular and unpopular songs.

3.3 10-Year Corpus

The 10-Year Corpus was created for the third question, whether popular song lyrics had changed over the past decade, which encapsulates the era of radical changes to the music industry with the rise of social media, YouTube, and streaming. We decided to use the Billboard Year-End charts to include the 100 most popular songs of each year between 2006 and 2015. Because the unofficial Billboard API being used did not feature access to these Year-End charts, we sourced our data from a 50-year (1965-2015) dataset created by Github user *walkerkq*. We filtered out the songs before 2006 so we could focus on just the past 10 year. Out of 1000 songs from the charts, we ended with 977 that had lyrics, labeled with the year of their success.

4. Method

To gauge the viability of lyrical content as a predictor of popularity, we used Linguistic Inquiry and Word Count (LIWC) features, as well as features obtained using the Natural Language Toolkit (NLTK), to train and test a classifier. To analyze lyrical trends in popularity over time, we fit the LIWC features to a linear regression model (feature value vs. time). In the subsections below we discuss the differences in LIWC and NLTK feature analysis, the training and validation of the classifier, and the longitudinal analysis of the Decade Lyrical Corpus.

4.1 Feature Analysis with LIWC

The Linguistic Inquiry and Word Count (LIWC) features were extracted using the LIWC2015 program on the lyrics field of the data, resulting in more than 80 features that were evaluated for best performance. Relevant features included word count, swears, netspeak, and words related to biological processes and power, and these are discussed further in the results and discussion sections. For determining the best features among the many features, we used a feature selection function from the Python module sklearn called SelectKBest, which reduced the feature set to the K best features based on which resulted in the highest ANOVA F-values for each of our datasets. After trying out various values for K, the optimal performance was achieved at K=9 for all three datasets.

4.2 Feature Analysis with NLTK

The Natural Language Toolkit is a Python module that provides tools for analyzing text – lyrics, in our case. After obtaining a clean corpus (described in the previous section), we coded a feature extractor

that read strings of text (the lyrics of the songs) and calculated counts for various features. The features included in our NLTK extractor were:

- *Lemmas:* Lemmatization is a technique wherein instead of counting the number of occurrences of a word – such as 'dogs' – we count the number of occurrences of the stem of a word. For instance, "The dog was better than the other dogs" would return a count of two (2) for the lemma 'dog'. We counted occurrences of the top 30 lemmas across each corpus.
- *Syntax:* For measures of syntax, we counted the occurrences of 10 parts of speech[list here]. NLTK has built in part-of-speech detection.
- *Unigrams:* A unigram is a one-word phrase – such as "dog" or "man". We chose to focus on unigrams instead of other n-grams (an example of a bigram is "oh man". As with lemmas, we counted occurrences of the top 30 unigrams across each corpus.
- *Complexity:* We combined the following three features into one umbrella feature. *Average characters* is a count of the average number of characters per word. *Uniqueness* is the number of unique words in a text. For instance, the sentence "I need you to need me" has five (5) unique words: 'I', 'need', 'you', 'to', and 'me'. *Long words* is a count of words with 6 or more characters.

It is important to mention that *preprocessing* and *normalization* measures were taken to ensure the most accurate collection of counts. Before feature extraction, our code removed *stopwords* (words such as 'whom', 'her', 'him', certain prepositions, etc.) that are extremely common in English and thus add a lot of noise in natural language analysis. Additionally, the counts above (except for *average characters*) were normalized by dividing them by the number of tokens (words) in each song.

After we obtained feature counts for all three corpuses, we exported them to Comma Separated Vector (CSV) files for further analysis and the training of our classifiers.

4.3 Classifying Songs as Popular or Unpopular

One of the major goals of this project was to build a nontrivial (better than naïve) binary classifier to predict the popularity of contemporary popular songs, as well as to use an artist's discography to predict the popularity of their own songs. The criterion for popularity is whether or not the song reached the Billboard Hot 100 chart.

Using the features from section 4.1 and 4.2, we trained and tested a Gaussian Naïve Bayes classifier using 10-fold cross-validation. The choice to use 10-fold cross validation – meaning that the corpus was used for training and testing in 10 different train/test configurations – over a train-then-test method was that we felt our dataset was not sufficiently large enough for the latter option.

To build our classifier, we used Sci-Kit Learn (SKLearn), a Python module that serves as toolkit for machine learning and data analysis. With the feature matrices obtained from NLTK, and the labels of popular or unpopular obtained from our corpus, we ran 10-fold cross-validation on SKLearn's Gaussian Naïve Bayes Classifier. We used the mean of the 10 scores to represent the predicted accuracy of our model on an independent dataset.

We used several different combinations of features to create numerous classifiers, using *forward stepwise selection*. We started by only including one feature at a time, then added features to form groups of two, three, and four. The combinations of features that worked best will be discussed in section five.

It is important to remember that when assessing the results of a classifier, the validity is based on comparison to another classifier. In our case, the baseline classifier was that in which picking the majority label in the corpus was the only strategy (picking all unpopular or all popular).

4.4 Linear Regression for the 10-Year Corpus

For exploring the evolution of lyrical features over time, we used R to find significant correlations between specific features and the year that the song was on the Billboard charts. We ran R's correlation tests between individual features and year to observe which specific ones correlated the most strongly, given a p-value < 0.05. Furthermore, we noted the direction of correlation over time based on the sign value of the correlation coefficient returned by these tests. Finally, using combinations of the

best features, we ran a linear regression model using SKLearn and measured performance of these results by the coefficient of determination, which indicates the proportion of the variance in the year of popularity that is predictable from the features being used in the model.

5 Discussion and Results

The following tables include our most significant results from each of the three studies. For the 10-week and Influential Artist studies, we have included classifier accuracies. The accuracies represent the mean of the 10 scores obtained from 10-fold cross-validation. In Tables 1-4, the baseline classifiers are defined as that which would pick the majority label for a corpus. For instance, if a corpus was 30% popular, a baseline classifier would pick unpopular for every test example.

NLTK Features	**Prediction Accuracy**
baseline	53.4%
complexity, syntax	56.5%
complexity	53.6%
syntax	52.9%

Table 1: 10-Week NLTK Results

Using complexity and syntax as the features for our NLTK classifier resulted in a 3.1% accuracy increase from the baseline., thus song lyrics alone can improve the prediction of popularity. It is interesting that syntax was able to improve the 53.6% accuracy of the complexity classifier alone. The results of these and several other classifiers can be seen in Table 1.

LIWC Features	**Prediction Accuracy**
baseline	53.4%
All features	53.9%
Word count	57.4%
Work, word count	58.6%
Work, body	59.2%
Work, word count, body	61.6%

Table 2: 10-Week LIWC Results

Table 2 shows the classifier performance for Problem 1 (10-Week Corpus) with various combinations of features, compared to a majority-vote baseline of 53.4%. All features brought only a slight increase, which is expected since many features would be irrelevant and add noise to the data. Word count proved to be the most significant single feature, and the nine best features resulting from the SelectKBest function were: word count, drives, work, netspeak, all punctuation, period, comma, and dash.

Using the nine best LIWC features resulted in an 8.2% improvement from baseline in classification accuracy, proving that song lyrics alone can improve prediction of its popularity. Word count was the most important feature for the model, providing a 4.0% increase in accuracy by itself. After a cursory review of the dataset, one explanation for this may be that hip-hop and rap songs are currently the most popular genre and inherently include many more words than other genres due to the nature of their lyricism. This is exemplified by the fact that a song with one of the highest word counts (1000+ words) was “Bad and Boujee” by Migos, a work that topped the charts for weeks. In this way, word count could be a proxy for deciphering the most popular genre to find popular songs. Among other interesting features was “netspeak,” which relates to words used while texting and on social media. This would be a culturally relevant linguistic category that may be more relatable for young teens driving the success of Hot 100 songs. An example of prolific artists using netspeak is Drake, who had multiple popular songs among the highest scoring in this category.

NLTK Features	**Prediction Accuracy**
baseline	54.4%
complexity	61.5%
syntax	55.2%
lemmas	55.2%
lemmas, complexity	56.4%
lemmas, syntax	55.2%
complexity, syntax	55.1%

Table 3: Rihanna NLTK Results

Using complexity alone, the NLTK classifier was able to improve upon the baseline accuracy for predicting the popularity of Rihanna’s songs by

7.1%. This is much more significant than the improvement for the 10-week corpus - and interestingly, complexity seems to play a huge role. The results of this and several other classifiers can be seen in Table 3.

LIWC Features	Prediction Accuracy
None	54.4%
All features	55.6%
Word count	61.7%
Negations	64.3%
Word count, negations, differentiation	69.8%
9 best features	73.7%

Table 4: Rihanna LIWC Results

Table 4 shows the classifier performance for Problem 2 (Influential Artist Corpus) with various combinations of features, compared to a majority-vote baseline of 54.4%. We only present the data from the Rihanna dataset, as it had the best balance of popular and unpopular songs and proved to show the highest performance between Rihanna, Drake, and Coldplay. All features brought a slight increase, and word count was the most significant single feature, similar to the result in Table 2. The nine best features resulting from the SelectKBest function were: word count, function, they, negations, verb, number, quantifiers, male, differentiation.

Using the nine best LIWC features resulted in a 19.3% improvement from baseline in classification accuracy for Rihanna songs, proving to be the best performing classifier across all problems and datasets. Word count again was the most influential feature, leading a 7.3% improvement in accuracy by itself. Another potential reason for popular songs having more word counts would be that many artists have shorter "Intro" songs or instrumentals that are more artistic and not marketed for the radio, which would inherently have less words in their lyrics. Another interesting feature among the nine best was the use of the "male" linguistic category, with hits such as "Unfaithful" and "Rude Boy" topping the list in highest scores in this category. This could be attributed to the fact that Rihanna's audience mainly consists of straight females and talking about males and relationships with them could be the most relatable content.

Positive Correlation with Years	Negative Correlation with Years
len(word)>6**	word count**
power	she/he**
biology	words/sentence**
death	male**
swear	ipron**
space	insight
affiliation	netspeak
we	cause
authentic	clout

Table 5: 10-Year LIWC Results **= $p < 0.005$

Table 5 shows the results for Problem 3 (10-Year Corpus) and details the most significant features that resulted from our correlation tests over time along with the direction of correlation. Among the entire feature set including both LIWC and NLTK, these 18 features all had p-values below the threshold of 0.05, with nine positive and nine negative correlations. Three of the features - word count, words with length greater than six, and words per sentence - were more strongly correlated, with p-values < 0.0005.

Among the 18 significant features that correlated with year of song popularity, there was a combination of surprising and expected results. Word count, a very strongly correlated feature ($p < 0.005$), decreased over the decade, which was counter-intuitive to what we hypothesized. We believed that due to the increasing popularity of hip-hop in the mainstream sphere in the last decade, word count would increase over time in popular songs. One reason would be that songs have increasing portions of only instrumental danceable beats with the rise of electronic music and artists such as The Chainsmokers. Swear words, on the other hand, increased over time, which could be attributed to a liberalization in open speech and the increase in explicit content from hip-hop. Another very strongly correlated feature was the use of "he" and "she," which decreased over time, while "we" increased over time. This could be attributed to more unifying, communal songs that, rather than telling individual stories, bring everyone together. Finally, the use of "male" references ("boy", "his",

"dad") was a very strongly correlated feature that decreased over time. The reasoning behind this remains unclear, but it could be attributed to the fact that there are more females writing songs and writing about themselves and other females. There are numerous other strongly correlated features, as shown in Table 5, and many need to be further analyzed to gain more insight into the reasoning behind these statistical findings.

6 Conclusion

In this work, we show that songs can be classified by popularity using lexical analysis of lyrics. The popularity of a song can be predicted relative to other songs by that same artist or relative to all songs. By experimenting with a number of different features and classifiers, we analyze how lexical-based features of lyrics have changed over the past decade. The highest classifier accuracies came from SelectKBest and Forward Stepwise Selection for feature selection, Gaussian Naive Bayes for classification, and Linear SVM for regression. We find that there are statistically significant lexical differences between popular and unpopular songs.

Music forms one of the largest subdivisions of the entertainment industry. By predicting song popularity, record labels could formulate business strategies that allow them to optimize the order in which songs are released. Lyrical analysis can also help music producers adapt their songs to the continuously changing public tastes of their fans.

7 Future Work

For future work, we could potentially increase the size of our datasets, such as looking at more than the past ten weeks for our 10-Week Corpus and finding artists with larger bodies of work for our Influential Artist Corpus. This may reduce variability resulting from a smaller sample size, especially since our artist-focused corpus can only work with a limited number of songs.

Furthermore, we could narrow our definition of popularity and perform regression to find a finer linear relationship between our features and popularity. This could be done using Spotify's popularity scores, which are values between 0 and 100 that are based on play counts and recency of success.

Finally, there may be patterns that are only relevant to specific genres, as the lyrical style of hip-hop would greatly differ from that of soft rock. Except for our Influential Artist corpus, our datasets included all relevant genres, so analyzing by genre may reveal unseen relationships that could then improve classification accuracy further.

8 References

[1] Hubert Léveillé Gauvin, “Drawing listener attention in popular music: Testing five musical features arising from the theory of attention economy”, 2017
http://journals.sagepub.com/doi/pdf/10.1177/1029864917698010

[2] Stephen Mussmann, John Moore, Brandon Coventry, “Using Machine Learning Principles to Understand Song Popularity”, 2014
https://www.cs.purdue.edu/homes/moore269/docs/music.pdf

[3] Ruth Dhanaraj, Beth Logan, “Automatic Prediction of Hit Songs”, 2005
https://pdfs.semanticscholar.org/2de2/34f32c268879e0aa331f286f50c3426837ad.pdf

[4] James Pham, Edric Kyauk, Edwin Park, “Predicting Song Popularity”
http://cs229.stanford.edu/proj2015/140_report.pdf

[5] Brian McFee et al., “The Million Song Dataset Challenge”, 2012
https://www.ee.columbia.edu/~dpwe/pubs/McFeeBEL12-MSDC.pdf

[7] Thierry Bertin-Mahieux, Daniel P.W. Ellis et al., “The Million Song Dataset”, 2011

[8] Allen Guo (guoguo12), “billboard.py”
https://github.com/guoguo12/billboard-charts

[9] Kaylin Walker (walkerkq), “musiclyrics”
https://github.com/walkerkq/musiclyrics

[10] Pennebaker, J. W., Booth, R. J., & Francis, M. E, Linguistic Inquiry and Word Count: LIWC [Computer software], 2007, Austin, TX: LIWC.net.
https://s3-us-west-2.amazonaws.com/downloads.liwc.net/LIWC2015_OperatorManual.pdf

9 Contributions

David Smart -
David wrote the abstract, introduction, the method and results for NLTK features. For data collection, he got the names for the decade dataset. David analyzed the NLTK features for all datasets.

Naman Agrawal -
Naman wrote the data used section, the method and results for LIWC features and also future works of the paper. For data collection, he extracted most of the lyrics. Naman, along with Vaidehi, analyzed the LIWC features for all datasets.

Vaidehi Dalmia -
Vaidehi wrote the related works, conclusion and found all the references for the paper. For data collection, she found the names for the popular songs in the last 10 weeks and helped extract lyrics for all datasets. Vaidehi, along with Naman, analyzed the LIWC features for all the datasets.

Our code and datasets can be found on: *github.com/namanaman/cmsl-music*