Topical Song Suggestion

CS410 Research Report

Amundeep Singh (singh78)

Introduction

Topical Song Suggestion is a system that allows you to discover new songs based on your interests in the songs lyrics and what the song is discussing, rather than the song's genre. So by looking for certain topics, you can find different songs that discuss that topic as well. The following is a research report oriented around the use of this system.

Utility & Impact

Most music services and song recommendation radios currently only suggest new music based on the genre of the songs you like and they do not really delve beyond that surface level search. The music industry has been forever evolving and now "on-demand music streaming accounts for 54% of total audio consumption" with 100s of billions of streams occurring each year, and that number is constantly growing (Perez).

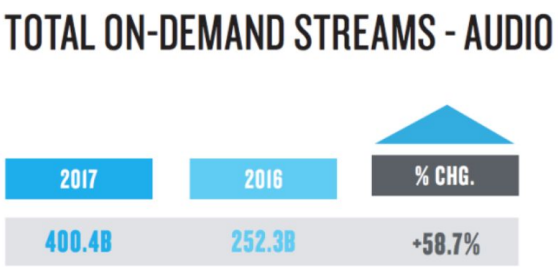

Figure 1: Percent change/increase in on-demand streams

For the past several years, music recommendation has been one of the main reasons streaming services have grown so much, and since 2014 Spotify "also looked into deep learning quite early, with Sander Dieleman interning in 2014 focusing on pure audio based methods" (Bernhardsson). Since then, audio-based methods have progressed over time and have gotten to the point where it is essential to have good audio-based recommendations in a music streaming service today. This leaves so much more room to grow when incorporating lyric-based recommendations **as well**, since it's been a relatively untapped path in terms of innovation.

Giving users more flexibility for how they search for new music is in demand with so many listeners and rise of streaming services, so using this TSS system as an extension to current streaming services would be a reputable feature that users can take advantage if they decide to.

Related Works

When this idea first came up, I searched online for how different companies or websites tried to execute a similar idea but could not find anything conclusive or specific enough to the implementation that I wanted.

Most websites are simply advertised and used as lyric search engines, for when users know certain lyrics to a song but do not necessarily know the song name, so they search the lyrics they know and hopefully the song that they are thinking of is displayed. However, this implementation does not seem to work well when searching for topics of songs that are more general because of how each site seems to prioritize different songs.

For example, Genius.com is a popular, well-renown site that is used to search and annotate lyrics. However, searching for topics such as “love” or “break up” returns a popularity-sorted list of songs with either the *title* of the song as what was searched for, or with very few lyrics containing the searched topic (i.e. searching “break up” on Genius returned “G.O.M.D” by J.Cole which only has the term “break up” **once** in the entire song). Granted, the website is most likely not optimized for generic one-worded topics like these, but this just goes to show that it’s not the implementation as intended with the TSS.

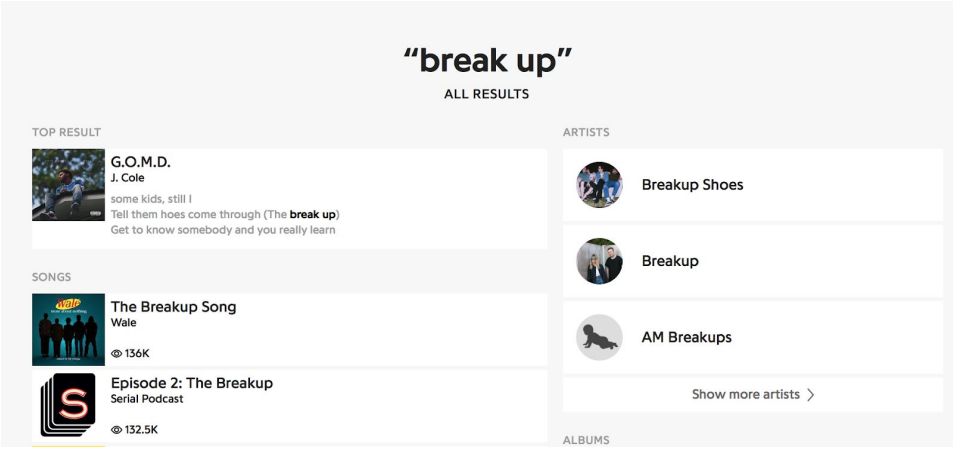

Figure 2: Genius.com “break up” search results

Another website called “Spotalike” allows users to enter a track they would enjoy during the summer, and the result is a Spotify playlist of “similar songs”. The result however has nothing to do with the lyrics and plays off the fact that the song **sounds** similar.

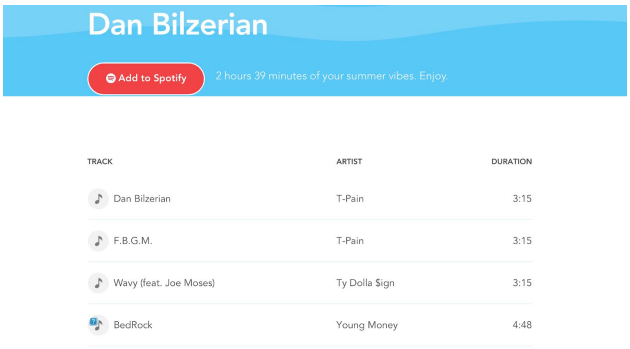

Figure 3: Spotalike.com results after entering “Dan Bilzerian - T-Pain”

Numerous different applications have gone further and further into what most music applications already do today, which is share and discover more music that sounds similar to what you already like, however the path that has not been touched on as much is sharing similarities between songs based on what the artist expresses through his or her lyrics.

Implementation – Successes and Difficulties

With this web application, you can simply search for a certain song topic/theme in the search box and, based on the current scraped dataset, it will return the songs that are most relevant to the user specified topic.

In order for this application to work, there had to be a system that can scrape for songs and analyze the song lyrics and perform the right functions that can help determine which songs match to what topics. Several problems stood in the way of fully implementing this backend, but I was able to figure out a workaround in order to still test this project’s implementation and conduct research.

There were several hiccups on the way to implement the scraping of songs off the web, especially with restrictions on certain API calls and how the web server for the website was set up in response to those API calls. The first try at implementation I chose was the Musixmatch API, but it would only let you view 30% of the song lyrics at time unless you paid for a license. So the best solution given the time was to scrape and format the lyrics

part-manually and part-Python scripting in order to format it in a way that can be used by MetaPy/MeTA and all the functions associated with it. I started to choose random songs that pertained to random topics and began to compile them into a .dat file that can be used by MetaPy, with each song's lyrics on each line of the file, formatted to remove any illicit syntax/spacing. Another part of scraping the lyrics included removing too many repeated segments/choruses where in some songs, lines of the chorus were repeated many times and some were removed based on certain songs at my discretion.

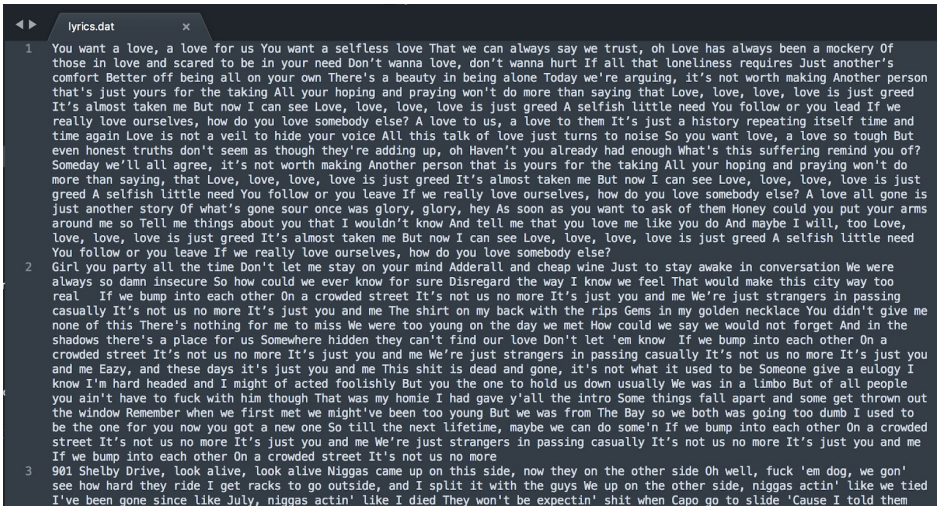

Figure 4: Part of the lyrics.dat file used for analysis (contains expletives)

In order to parse through the lyrics and find out what the most common lyrics were, I decided to basically rank them by splitting them into topics through the Latent Dirichlet Allocation (LDA) topic model. This LDA topic model is used for testing methods in order to find out whether this is a sufficient way of ranking the lyrics in the song or not.

After running into many issues running the LDA topic model Python MeTA backend onto a web server, specifically HTTP protocol and runtime, I decided to use the LDA topic model to conduct research into whether or not the topic corresponded to the songs that I used for the dataset.

Methods & Results

In order to test and see whether or not this method of finding similarities between songs will work effectively or not, running LDA topic models allows us to group all the text in the lyrics into distinct topics and be able to assign them to certain songs/documents. However, grouping the lyrics into too many or too few topics can render inaccurate

results, so with the lyrics dataset I compiled, I decided to run LDA with 4 topics for this run.

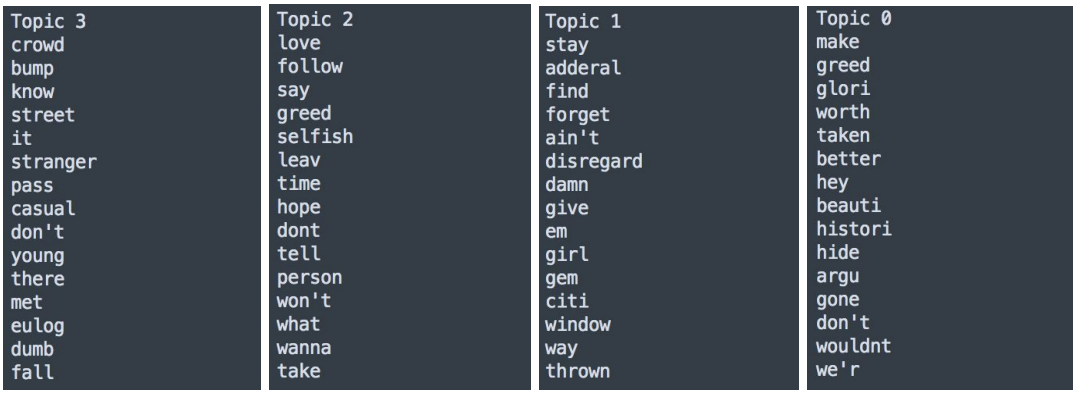

Figure 5: The output-topic.txt file output with 4 topics

Above in Figure 5 are the results of running LDA with 4 topics on the song lyrics in the dataset (exact song names/artists can be visible via the songs-key.txt file in the GitHub). The problem that occurs when running too many topics is that we start to see more duplicates among each topic set, such as 'greed', which is present with pretty high probability in both topic 2 and topic 0. This makes it a little more difficult when processing topic rankings with respect to documents because then the results become more vague and less succinct.

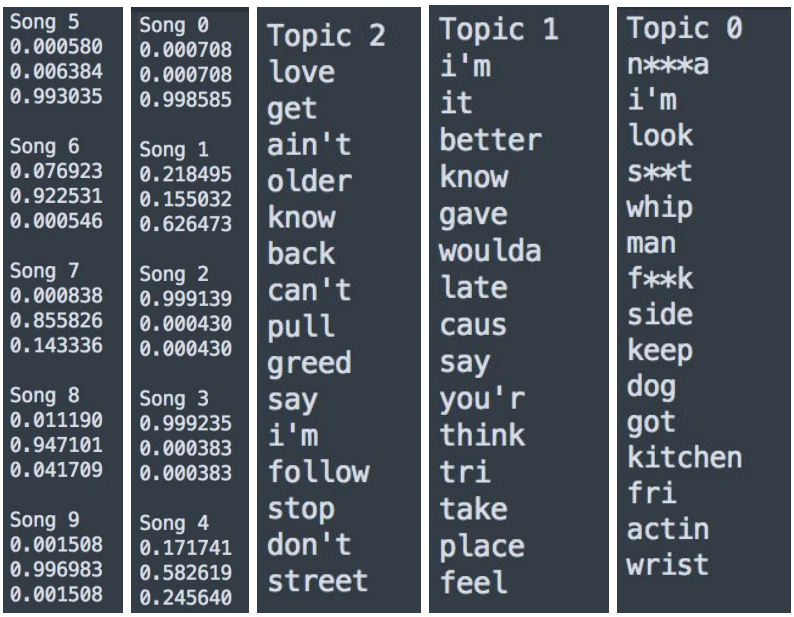

Figure 6: Topic distributions with 3 topics, and relevant song topic distribution (expletives censored)

Above in Figure 6 is the output when running with 3 topics and some of the document topic rankings and the results between all of them are clearly more distinct than that of the figure above. When looking at one of the songs in the dataset "Love is Greed", with the topics in Figure 5, having multiple topic distributions with 'greed' in it is unnecessary and adjusting the number of topics with respect to the number of songs in the dataset is an important part of retrieving relevant data.

Another one of the caveats that I ran into when analyzing the results included how different genres of songs either contained more lyrics (rap) or less lyrics (pop), so the dataset compiled is composed of a variety of all types from rap, to pop, and even country. While most music recommendation software looks at just the genre or just the tone of the music, looking at and incorporating them **both** can make the experience even more beneficial to users of music streaming/recommending services.

According to a paper by Stanford Masters student Alexandros Tsaptsinos, we can see these differences in lyrics by genre through Hierarchical Attention Networks (HAN). HAN allows us to view sentence structure as a hierarchy, for example words form sentences and sentences form a document (Yang). In the case of song lyrics it allows us to see the discrepancy of lyrics between genres. The table below shows how many times the HAN guessed the genre correctly or not. Rows correspond to the true genre, and columns correspond to the genre predicted based on the lyrics.

	Rock	**Pop**	**Alternative**	**Country**	**Hip-Hop/Rap**
Rock	6879	1198	2460	561	91
Pop	1892	2620	738	324	157
Alternative	2385	534	2866	150	67
Country	534	304	90	2199	4
Hip-Hop/Rap	71	97	63	11	2629

Figure 7: Confusion matrix for running HAN on song lyrics (Source: Alexandros Tsaptsinos)

With this data, we can observe and separate topic distributions based on genres and give users an even spread of suggestions among different genres, or if the user specifically likes certain genres, we can narrow it down and use the LDA topic models to more precisely cater to whatever a user desires.

Discussion

In this paper, we've shown that generating LDA topic models is a viable way of finding out songs' relevance to certain themes, and the results from the HAN can further narrow this based on genre. The audience for music recommendation has been constantly growing over the last several years and incorporating lyrical relevance and text analysis on these lyrics can further improve user experience in this area.

Besides musical instruments and genre, lyrics can have an even deeper connection to listeners because of themes and messages songs can discuss, and running text analysis and language processing on these lyrics allows us to learn more than just what a genre of music can tell us, and according to the Stanford paper, "Lyrics typically perform stronger on the sentiment analysis classification task." (Tsaptsinos). Lyrics can show much more about a song rather than just the genre, and it's proven based on LDA topic models and the data that the HAN study has shown.

References

Bernhardsson, Erik. "How Did Spotify Get So Good At Machine Learning?" *Forbes*, Forbes Magazine, 20 Feb. 2017, www.forbes.com/sites/quora/2017/02/20/how-did-spotify-get-so-good-at-machine-learning/#20f45b25665c.

Perez, Sarah. "On-Demand Streaming Now Accounts for the Majority of Audio Consumption, Says Nielsen." *TechCrunch*, TechCrunch, 31 Jan. 2018, www.techcrunch.com/2018/01/04/on-demand-streaming-now-accounts-for-the-majority-of-audio-consumption-says-nielsen/

Tsaptsinos, Alexandros. *Music Genre Classification by Lyrics Using a Hierarchical Attention Network*. *Music Genre Classification by Lyrics Using a Hierarchical Attention Network*, http://web.stanford.edu/class/cs224n/reports/2728368.pdf

Yang, Zichao, et al. *Hierarchical Attention Networks for Document Classification*. pp. 1–10, *Hierarchical Attention Networks for Document Classification*, http://www.cs.cmu.edu/~./hovy/papers/16HLT-hierarchical-attention-networks.pdf