INFORMATION RETRIEVAL & TEXT MINING

TEXT MINING MUSIC LYRICS

Amit Krishna Jadhav (self)
Advaith Jaishankar (Group Member)

Supervisor - Dr. Jan Scholtes
Teaching Assistant - Thales Bertaglia

Department of Data Science and Knowledge Engineering
Maastricht University

May 25, 2022

1 Introduction

Language has always been the fastest mode of physical communication. The rapid progress in machine learning has led to the development of a multitude of highly effective techniques that enables us with analysing various forms of written textual language. Human beings are superior in terms of their background knowledge of language and its comprehensibility/ understanding. But they lack the quantum of memory and lightening fast mathematical processing strength or the ability to pseudo-parallel processing of numerous sophisticated mathematical operations. A rational human being who is quick at reading can very well understand say a single 500 or even a 1000 pager book in a day. But the amount of textual content that is easily accessible today and that is necessary to derive meaningful insights about any subject is much more than the human capacity. This is where a machine comes in handy. A personal laptop of any average person is capable of scanning through Millions of pages of textual content in a single moment. This makes it ever more motivational, a cause, to understand and enhance the capability of machines to understand, and generate popular human languages.

This project aims to apply a very tiny set of such techniques which fall under the umbrella term of text mining on a dataset of Music lyrics of the top grossing songs worldwide over the span of last 15 years. It is an attempt to deduce the characteristics of the most popular music preferences. The methodology employed to achieve this objective is discussed in the following sections.

2 Breakup of Work Done

The table 1 below summarizes the work done by each of the two members for this Practical Assignment.

3 Methodology and Results

3.1 Dataset

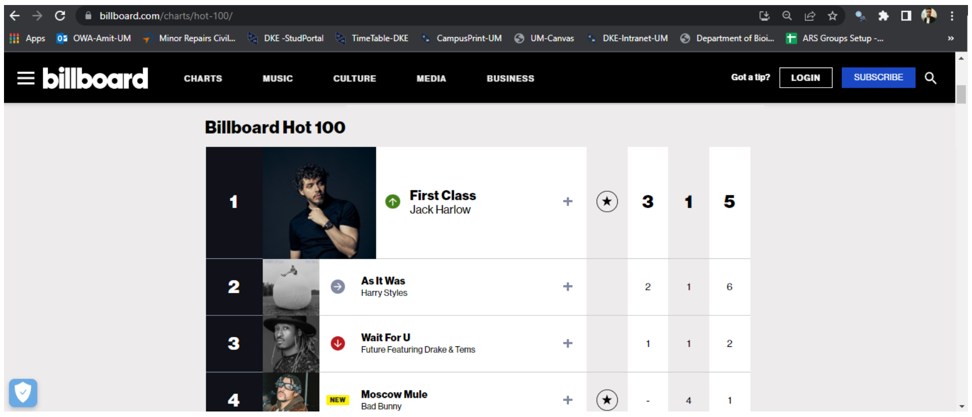
Perhaps it is best to start with the dataset and the information that we had initially to let the reader get a perspective of the problem at hand. Attached below are the snapshots of the top ranking songs of each year from 2006 to 2021 on the billboard website [1]. These were merged with the lyrics and genres of each of these songs extracted from Genius website [2]. A snapshot of these two websites is shown in figure 1 and figure 2.

3.2 Scraping

The dataset was scraped by the group member of the author using scrapy and requests. The songs names were first picked up from the 15 years of top 100 billboard charts and later the Genres,

Scraping Lyrics	Advaith J
Regular Expressions Cleaning	Amit J
Preprocessing - Language Detection, POS Tagging	Amit J
Preprocessing - Tokenization	Advaith J
Preprocessing - Resolving Issue with Joined sentences (sentences without separator)	Amit J
Named Entity Recognition - NEChunk	Amit J
Named Entity Recognition - Stanford [5]	Advaith J
Data prep for Visuals - Artistwise and Yearwise NER summary for visuals	Amit J
NER Visuals Artists Network	Advaith J
Entity Linking - Artists to entity link networks	Advaith J
Named Entity Visuals - Bar chart race	Advaith J
Emotion Quantification using Empath [4] categories	Amit J
Emotion Quantification using EmoRoBERTa	Amit J
Sentiment Analysis - Textblob, Vader, Emotion Sentiment	Amit J
Word Embeddings using Gensim[7] word2vec	Amit J
Spacy Lemmatization	Amit J
TFIDF Word Vectors with Kmeans clusters	Amit J
Topic Modelling using LDA with TFIDF filter of phrases	Amit J
Topic modelling visuals using LDAVis [8]	Amit J
Topic modelling using BERTopic	Amit J

Table 1: Summary of Work Done

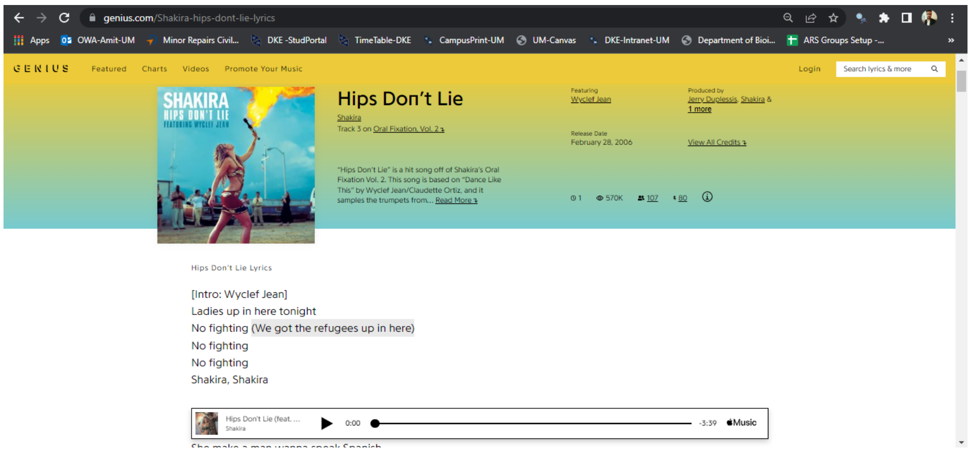


Billboard Hot 100

1	First Class Jack Harlow	3	1	5
2	As It Was Harry Styles	2	1	6
3	Wait For U Future Featuring Drake & Tems	1	1	2
4	NEW Moscow Mule Bad Bunny	-	4	1

Figure 1: Billboard Top 100 Songs

Artists and Lyrics were scraped from Genius website. This data was later pushed to a SQL database for easy access from Google Colab notebooks. The pushed data had a schema as: SONG NAME, YEAR, ARTISTS, PRIMARY ARTIST, LYRICS and GENRES. This study did not focus on the ranking of Songs and hence the song Ranking was not scraped.



Hips Don't Lie

Shakira

Track 3 on Oral Fixation, Vol. 2

Featuring
Wyclef Jean

Produced by
Jerry Duplessis, Shakira & 1 more

Release Date
February 28, 2006

View All Credits

"Hips Don't Lie" is a hit song off of Shakira's Oral Fixation Vol. 2. This song is based on "Dance Like This" by Wyclef Jean/Claudette Ortiz, and it samples the trumpets from... Read More

Hips Don't Lie Lyrics

[Intro: Wyclef Jean]
Ladies up in here tonight
No fighting (We got the refugees up in here)
No fighting
No fighting
Shakira, Shakira

She make a man wanna speak Spanish

Figure 2: Genius Website - Sample Lyrics

3.3 Data Issues - Music Lyrics

The lyrics after scraping were hidden inside containers. Beautiful Soup library was used to extract the lyrical sections out of these html containers.

A few songs for which no lyrics / no genre could be obtained or in cases where the genres were non-music genres were eliminated. The musical dataset is highly tuned to the culture of the artists and preferences of people. Thus there are a lot of musical and language slangs used in the lyrics. The lyrics themselves also contained the indicators of parts of the song such as CHORUS, PRECHORUS, VERSES, INTRO and OUTRO. These indicators and also the language slangs were removed by using more than 25 conditions in python regular expressions. A brief snapshot of these regular expressions to remove the language is depicted in figure 3

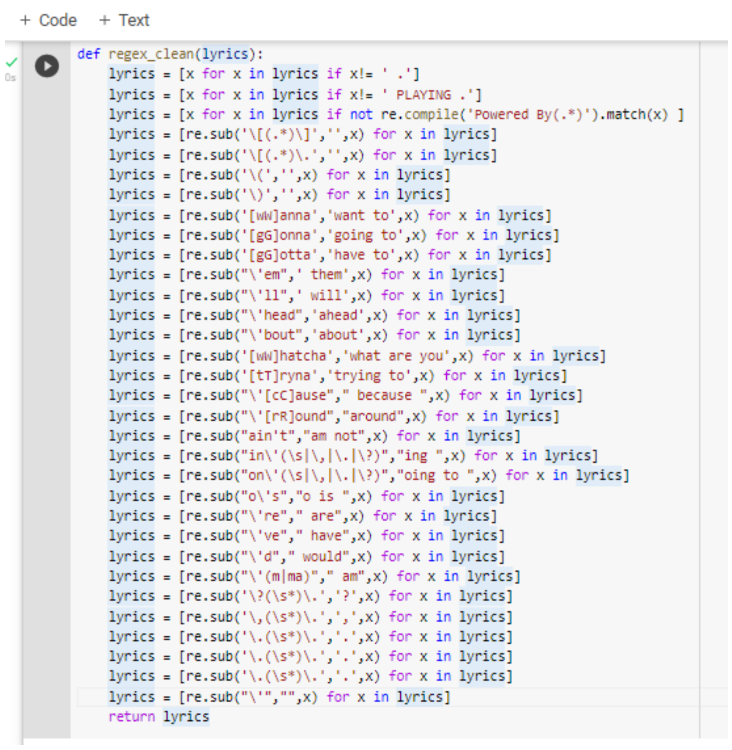
3.4 Preprocessing

The lyrics so obtained after cleaning were fed to language detection snippet and about 15 out of 1507 songs not belonging to the English language were removed from the data to limit the scope to only English songs.

The lyrics were then tokenized and any stop-words were removed using nltk library's [3] stopword list for English language.

3.5 Song Repititiveness check

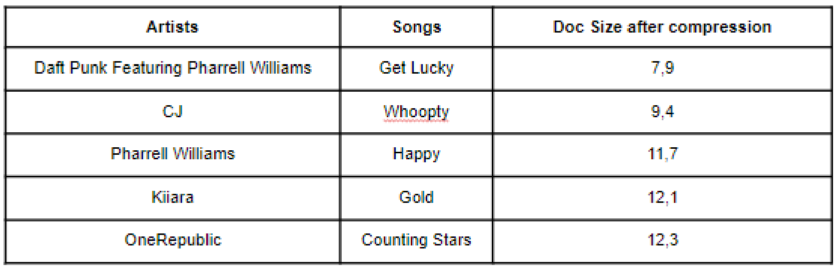
Upon cleaning the lyrics, it was observed that there is highly repetitive content in a considerable amount of songs. To give better insghts into this, a song lyric repetitiveness check was performed using using Lempel-Ziv-Markov Algorithm. The result revealed drastic amounts of repetitiveness in some chart topping songs. Figure 4 shows the compression factor or the size of the vocabulary left



```python
def regex_clean(lyrics):
    lyrics = [x for x in lyrics if x!= ' .']
    lyrics = [x for x in lyrics if x!= ' PLAYING .']
    lyrics = [x for x in lyrics if not re.compile('Powered By(.*)').match(x) ]
    lyrics = [re.sub('\[(.*)\]','',x) for x in lyrics]
    lyrics = [re.sub('\[(.*)\.','',x) for x in lyrics]
    lyrics = [re.sub('\(','',x) for x in lyrics]
    lyrics = [re.sub('\)','',x) for x in lyrics]
    lyrics = [re.sub('[wW]anna','want to',x) for x in lyrics]
    lyrics = [re.sub('[gG]onna','going to',x) for x in lyrics]
    lyrics = [re.sub('[gG]otta','have to',x) for x in lyrics]
    lyrics = [re.sub("\'em",' them',x) for x in lyrics]
    lyrics = [re.sub("\'ll",' will',x) for x in lyrics]
    lyrics = [re.sub("\'head",'ahead',x) for x in lyrics]
    lyrics = [re.sub("\'bout",'about',x) for x in lyrics]
    lyrics = [re.sub('[wW]hatcha','what are you',x) for x in lyrics]
    lyrics = [re.sub('[tT]ryna','trying to',x) for x in lyrics]
    lyrics = [re.sub("\'[cC]ause"," because ",x) for x in lyrics]
    lyrics = [re.sub("\'[rR]ound","around",x) for x in lyrics]
    lyrics = [re.sub("ain't","am not",x) for x in lyrics]
    lyrics = [re.sub("in\'(\s|\,|\.|\?)","ing ",x) for x in lyrics]
    lyrics = [re.sub("on\'(\s|\,|\.|\?)","oing to ",x) for x in lyrics]
    lyrics = [re.sub("o\'s","o is ",x) for x in lyrics]
    lyrics = [re.sub("\'re"," are",x) for x in lyrics]
    lyrics = [re.sub("\'ve"," have",x) for x in lyrics]
    lyrics = [re.sub("\'d"," would",x) for x in lyrics]
    lyrics = [re.sub("\'(m|ma)"," am",x) for x in lyrics]
    lyrics = [re.sub('\?(\s*)\.','?',x) for x in lyrics]
    lyrics = [re.sub('\,(\s*)\.',',',x) for x in lyrics]
    lyrics = [re.sub('\.(\s*)\.','.',x) for x in lyrics]
    lyrics = [re.sub('\.(\s*)\.','.',x) for x in lyrics]
    lyrics = [re.sub('\.(\s*)\.','.',x) for x in lyrics]
    lyrics = [re.sub("\'","",x) for x in lyrics]
    return lyrics
```

Figure 3: Cleaning Language Slangs using Regular Expressions

after eliminating the repetitive lyrics from the song.



Artists	Songs	Doc Size after compression
Daft Punk Featuring Pharrell Williams	Get Lucky	7,9
CJ	Whoopty	9,4
Pharrell Williams	Happy	11,7
Kiiara	Gold	12,1
OneRepublic	Counting Stars	12,3

Figure 4: Most Repeating Songs among past 15 year Chart toppers

3.6 POS Tagging

The nltk [3] POS tagger was used to classify the tokens generated in the pre processing step under 36 Part of Speech tag categories with different part of speech bifurcations. Part of Speech tagging is a crucial step as it acts as an input to the Named Entity Recognition algorithm since the Entities are associated with Noun types of tokens.

3.7 Named Entity Recognition

Named entity is any uniquely identified person, organization, geo-political entity, geographical location, temporal moment or interval thats identified with a name, Amount of Money in a named Currency or even a relative proportion expressed as percentage. For our study, Named Entity recognition was performed using two well known libraries that are a part of NLTK [3]. The visualization of Artists' collaboration and Entities referred by top artists is depicted in figure 5 and figure 6 respectively. One interesting from the collaboration graph was that the most well known artists are the ones who have collaborated a lot, whereas there are few other artists who have rarely collaborated with others as seen in figure 7. The visualizations can be completely observed on web page **1** and **2**. The most referred named entities were observed chronologically as well in the form of a bar chart race which is hosted here.

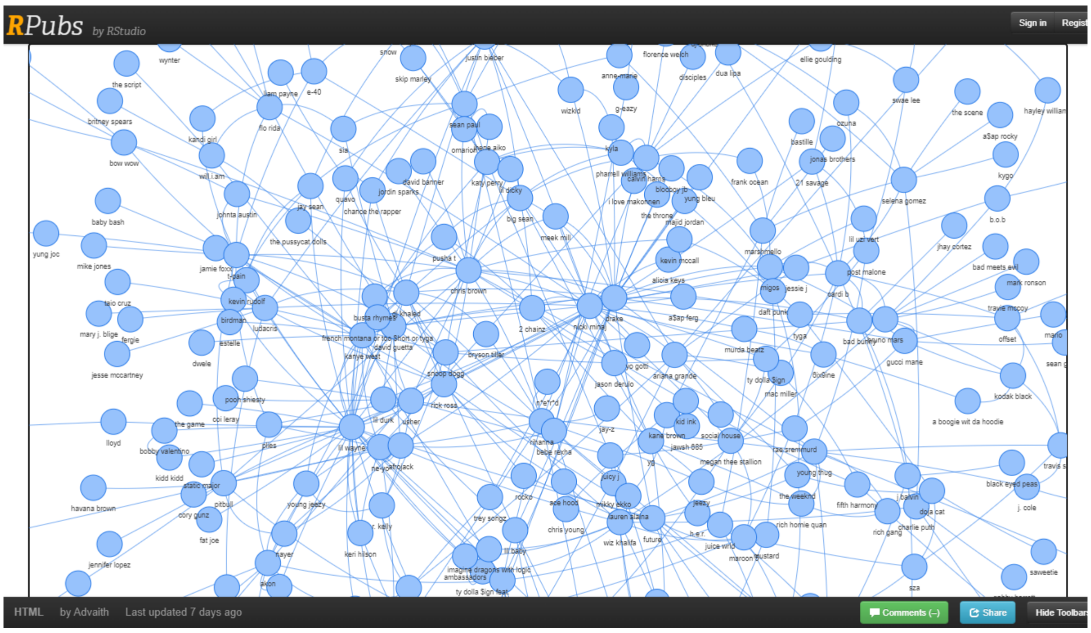


Figure 5: Collaboration network of artists

3.7.1 Named Entities Using NLTK NeChunk

Nechunk uses the existing POS tags generated to recognise and separately tag the various Named entities detected in the POS tagged list of tokens and also the type of these named entities. The detected named entities are separately enclosed in the tuples in the final output.

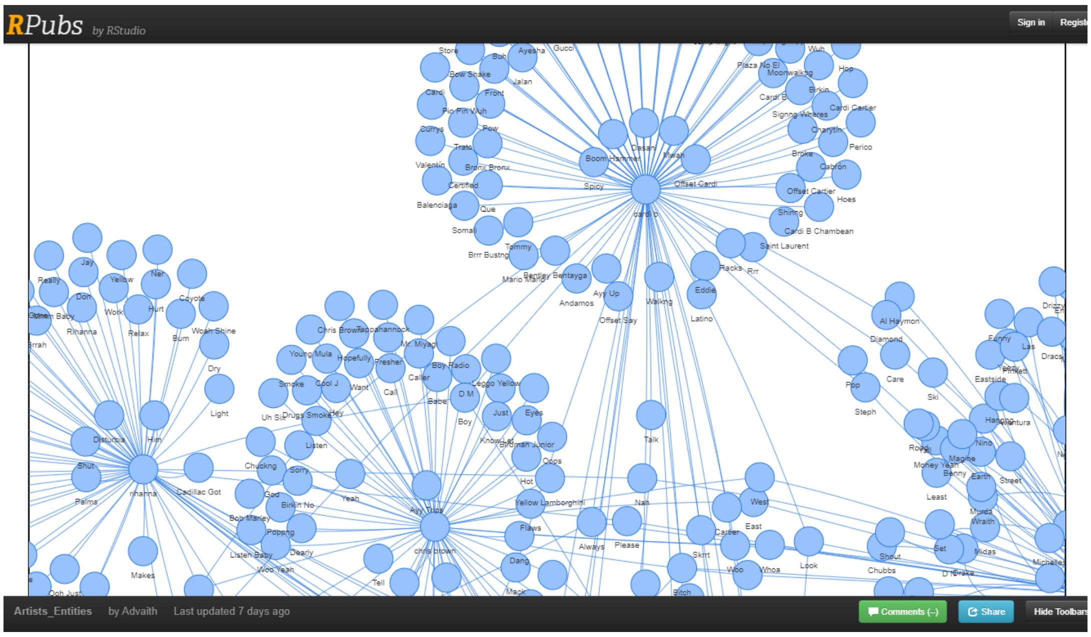


Figure 6: Named Entities most referred by few popular artists

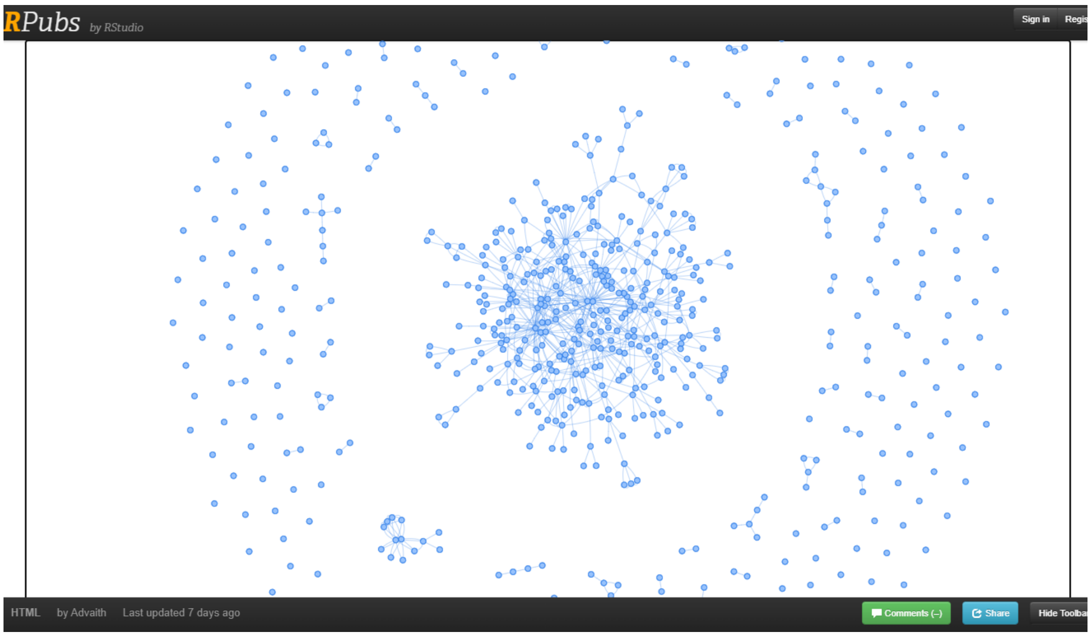


Figure 7: Result showing isolated versus highly collaborating artists

3.7.2 Named Entities Using NLTK Stanford NER

Stanford Named Entity Recogniser [5] unlike NLTK nechunk, draws the entities using internal POS tagging instead of taking the POS tagged tokens as input. So the input was plain tokens generated using NLTK word tokenize function.

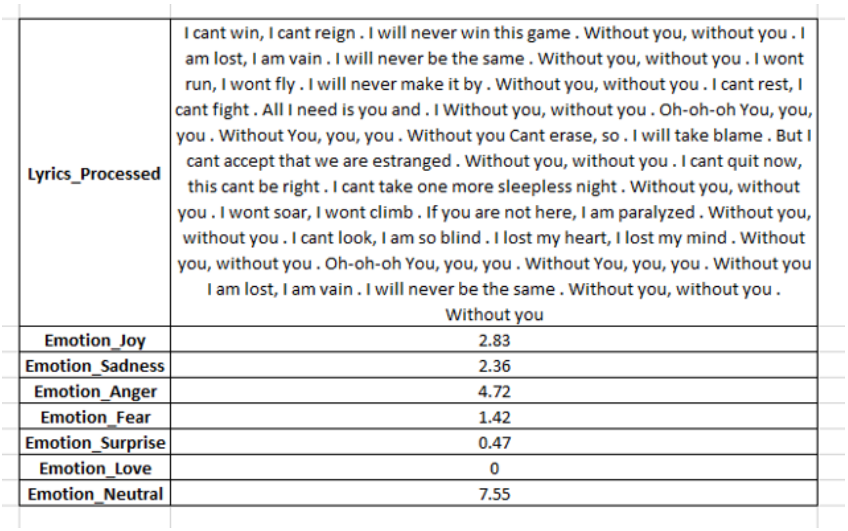
The comparison of results of both these techniques was performed to relate their effectiveness in recognising the named entities and this is listed in the 'Validation of Results' section below.

3.8 Emotion Quantification

The lyrics in almost all the songs are meant to intrigue, inspire, console, justify, comfort, cheer up, satisfy, engage people. This makes the presence of emotions in the songs inevitable. In an attempt to understand what emotions are present in the top charting songs, an emotion quantification exercise was performed using two approaches.

3.8.1 Emotion classification by leveraging Empath

Empath [4] is another interesting library which attributes each word in the english language text to 196 lexicon categories which are not necessarily mutualy exclusive, meaning one word can be a part of ar contribute to multiple lexicon categories. These 196 Empath lexicon categories were further bucketed manually into 7 emotion types (Joy, Love, Sadness, Anger, Fear, Surprise, Neutral) by the Authors and emotions were quantified as a normalised score using the sum of scores of all words across each of the 196 categories to which the word contributed. This way a score in the range of 0 to 100 was obtained for each emotion bucket. A sample screenshot of the emotion is listed in figure 8 The performance issues in empath [4] emotion quantification led to more exploration and finally the



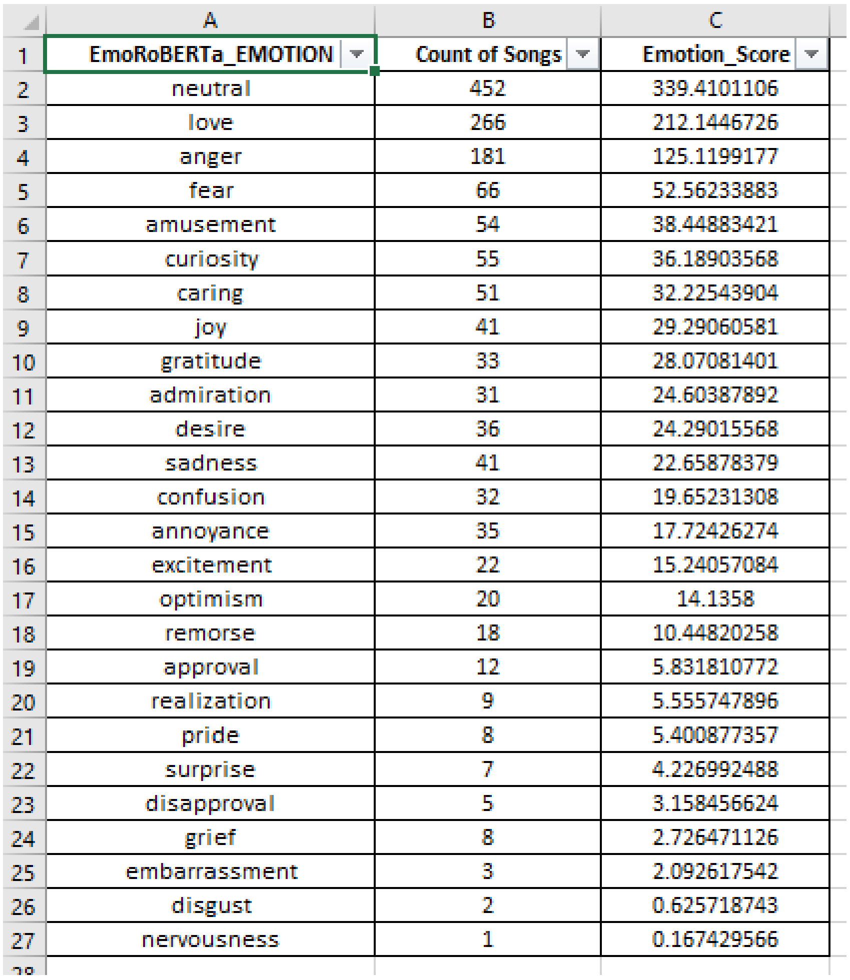
Lyrics_Processed	I cant win, I cant reign . I will never win this game . Without you, without you . I am lost, I am vain . I will never be the same . Without you, without you . I wont run, I wont fly . I will never make it by . Without you, without you . I cant rest, I cant fight . All I need is you and . I Without you, without you . Oh-oh-oh You, you, you . Without You, you, you . Without you Cant erase, so . I will take blame . But I cant accept that we are estranged . Without you, without you . I cant quit now, this cant be right . I cant take one more sleepless night . Without you, without you . I wont soar, I wont climb . If you are not here, I am paralyzed . Without you, without you . I cant look, I am so blind . I lost my heart, I lost my mind . Without you, without you . Oh-oh-oh You, you, you . Without You, you, you . Without you I am lost, I am vain . I will never be the same . Without you, without you . Without you
Emotion_Joy	2.83
Emotion_Sadness	2.36
Emotion_Anger	4.72
Emotion_Fear	1.42
Emotion_Surprise	0.47
Emotion_Love	0
Emotion_Neutral	7.55

Figure 8: Empath [4] Emotion Scores for One sample song

alternate approach of EmoRoBERTa was finalized for better insight into song lyric emotions.

3.8.2 Emotion Classification using EmoRoBERTa

A bert based pretrained model namely EmoRoBERTa which has been trained on 58k Reddit comments as a part of the Go-Emotions dataset, was used to classify the music lyrics into the most majorly evident emotion in the lyrics. Each song lyric was, as a result, labelled with one of 28 available emotion labels and was assigned a saturation score associated with this labelled emotion. A brief glimpse of the sums of scores of all the recognized emotions and the number of songs the corresponding major emotion are listed herewith in figure 9



	A	B	C
1	EmoRoBERTa_EMOTION	Count of Songs	Emotion_Score
2	neutral	452	339.4101106
3	love	266	212.1446726
4	anger	181	125.1199177
5	fear	66	52.56233883
6	amusement	54	38.44883421
7	curiosity	55	36.18903568
8	caring	51	32.22543904
9	joy	41	29.29060581
10	gratitude	33	28.07081401
11	admiration	31	24.60387892
12	desire	36	24.29015568
13	sadness	41	22.65878379
14	confusion	32	19.65231308
15	annoyance	35	17.72426274
16	excitement	22	15.24057084
17	optimism	20	14.1358
18	remorse	18	10.44820258
19	approval	12	5.831810772
20	realization	9	5.555747896
21	pride	8	5.400877357
22	surprise	7	4.226992488
23	disapproval	5	3.158456624
24	grief	8	2.726471126
25	embarrassment	3	2.092617542
26	disgust	2	0.625718743
27	nervousness	1	0.167429566

Figure 9: Emotionwise No. of Songs and Total Emotion Scores

3.9 Sentiment Analysis

As a natural extension to the emotion analysis and to also induce polarity in the understanding a sentiment polarity analysis was done to gauge the number of songs that display positive, negative and neutral sentiment polarities respectively. Three methods were used to draw the sentiment polarities and classify the songs into one of the above three sentiments. The count of songs categorised under positive, negative and neutral sentiment polarity has been depicted in the bar graphs in figure 10 for Emotion sentiment analyser, TextBlob sentiment analyser and Vader sentiment analyser.

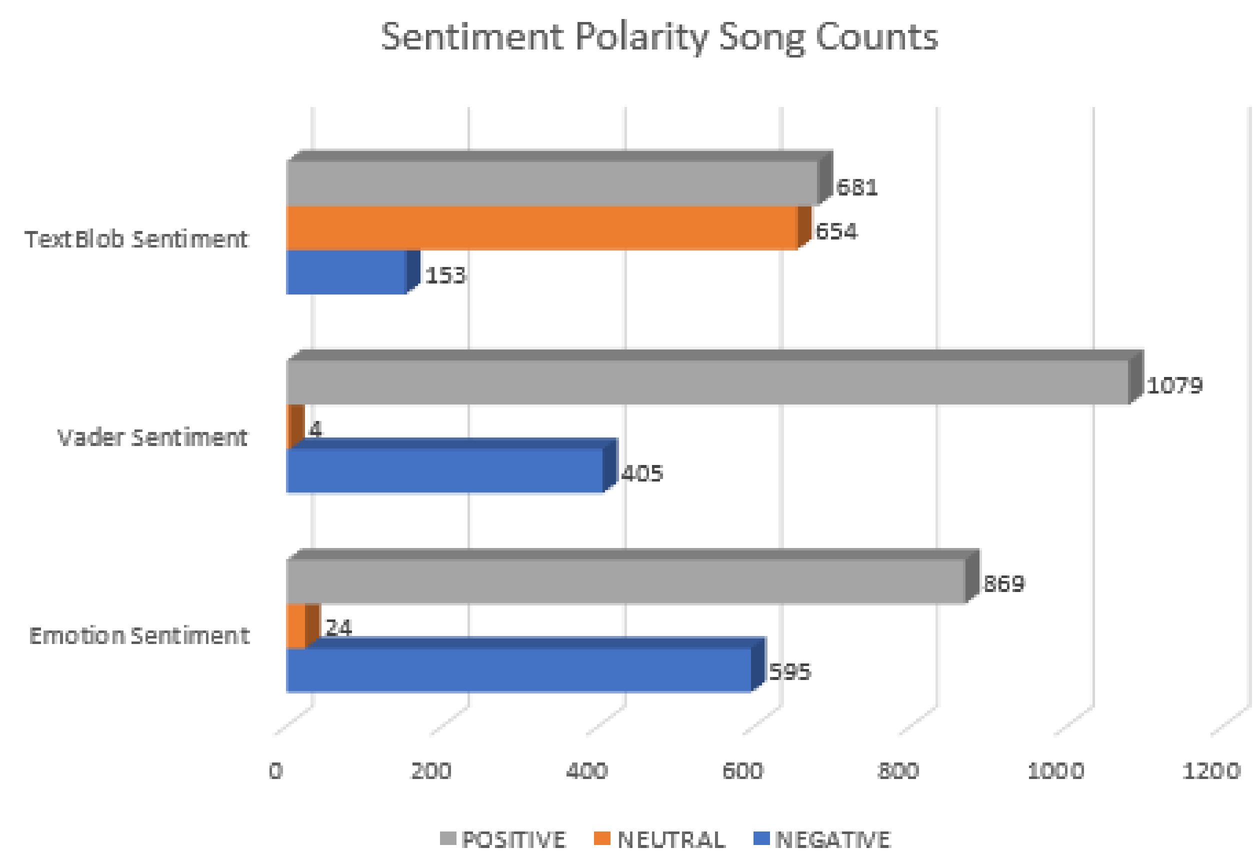


Figure 10: Songwise sentiment polarity counts

3.9.1 Sentiment Analysis using Empath Quantified Emotions

As an initial attempt, the sentiment polarities were calculated using one assumption that certain emotions are positive emotions (i.e. the emotional states in which a human being is inclusive and conducive to any changes in the environment), whereas a few others are negative emotions (i.e. the emotional states in which a human being is exclusive and explosive to even minute changes in the environment) and some are neutral. This assumption was based on the fact that most sentiment polarity calculation tools base the scoring on the positive word and negative word lists. These allowed for calculation of the sentiment as per the below equation:

$$Sentiment = Joy + Love - Anger - Fear - Sadness \tag{1}$$

Surprise and Neutral were treated as not contributing to positive or negative emotions. Note here that surprise can be a negative as well as a positive surprise.

3.9.2 Sentiment Analysis using TextBlob

TextBlob is a well documented text processing package which has a sentiment polarity computation method. This was used to calculate the sentiment polarities.

3.9.3 Sentiment Analysis using Vader

NLTK's [3] Vader sentiment intensity analyzer is a robust tool to effectively analyse the sentiment polarity of songs while considering multitude of different cases in which the semantics and structure of a sentence may give rise to varying sentiments. **Few of the possibilities handled by vader include sentiment boosting or sentiment eroding adverbs, sentiment laden idioms, special case sarcastic phrases, negating word contractions, emoticons and words together with emoticons, precedence of "no" or "but" before the terms in the vader lexicon dictionary**, etc.

3.10 Additional Pre-Processing to adjust input to Word2Vec, TF-IDF and Topic Modelling

The data had a few songs which had topped the charts for more than one year and hence were having duplicate entries in the lyrics data. This redundancy was eliminated by dropping the duplicates across having Artist and Song name as the key and retaining the oldest year that the song grossed the billboard charts. This was done to avoid bias in the TF-IDF (specifically the Inverse document frequency) calculations part and to not affect the topic-to-word frequency in topic modelling using LDA. Additionally the lyrics of the songs were lemmatized using SpaCy lemmatizer to avoid feeding different forms of the same words to the subsequent techniques.

3.11 Word Vectorization - Word2Vec

In an attempt to study and understand word vectorization using the word2vec vanilla approach, the word embeddings were drawn leveraging the gensim library[7]. The CBOW or continuous bag of words approach was followed with a window size of 3 and the size of word embeddings as 20 dimensional. To eliminate very scare words, a minimum count of 5 was set as the word count threshold for a word embedding to be generated. After running about 1000 iterations over the data, the word vectors seemed to make partial sense over such a small corpus. The code contains some example tests on the output of the word vector similarity, where the embeddings have been tested to draw analogies, to find most similar words to the queried word and to find the word that is most different among a group of words pass to the trained model. The distance metric used here is Cosine similarity.

3.12 Word Importance Clusters - TF-IDF, Kmeans, PCA

To observe how the different in music align together in terms of the most importance terms referred to in the music lyrics, it was decided to perform word Vectorization using TF-IDF and then cluster the words that are listed at least 5 times in each document and are present in not more than 80% of the documents. In case the number of terms satisfying this condition exceeded 100 terms, then only the top 100 terms per document were retained The document-to-word importance scores so obtained were used create clusters of documents with similar important words. K means clustering was used for this purpose and was fed with 10 initial cluster centers initiated optimally using kmeans++ initiation. Since each document was described using the TF-IDF vectorized at most 100 words, PCA was employed to reduce the dimensionality from 100 to the most information intensive two principal dimensions. This converted the 100 dimensional 10 Kmeans cluster space to a 2 dimensional PCA cluster space. The genres of each song were then visually examined using the so formed PCA clusters. A zoomed out and cropped snapshot of the lyric-wise clusters is depicted in figure 11.**Note that the PCA clusters would have an overlap due to the drastic dimensionality reduction from 100 to 2. Even after this, it was observed that the RAP songs align more extremely than all other genres, implying that there are more terms with higher TF-IDF scores, i.e. pointing towards a richer vocabulary in a**

crude sense.

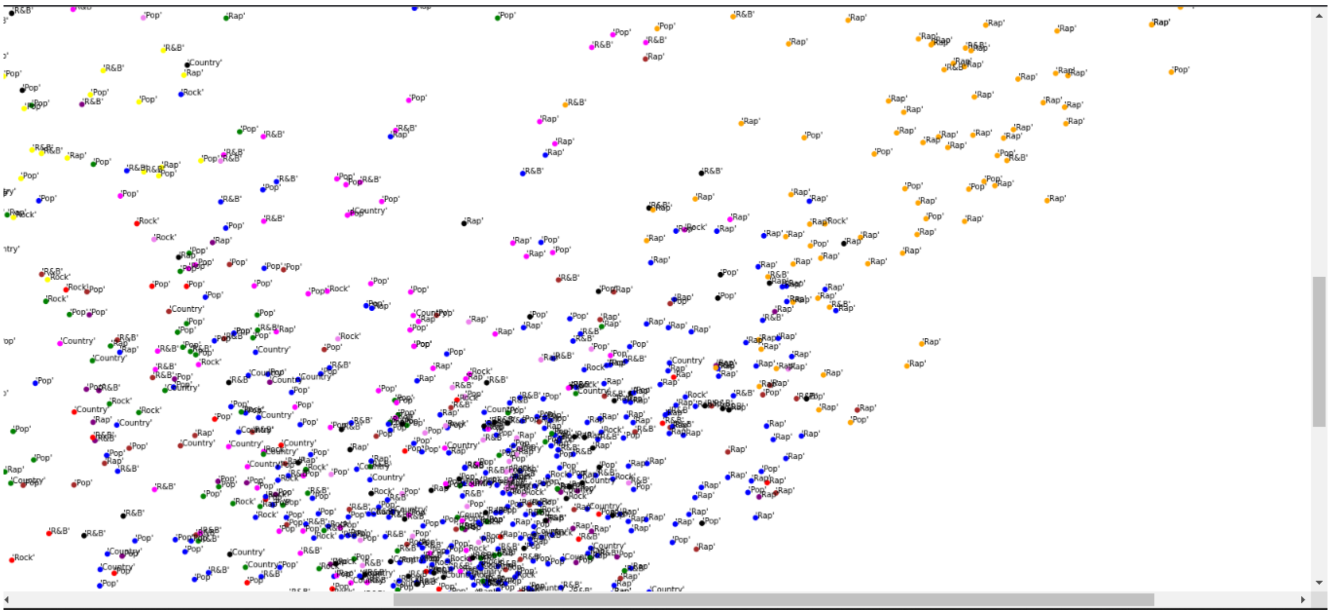


Figure 11: Song Clusters by Genre Basis TFIDF-KMeans-PCA

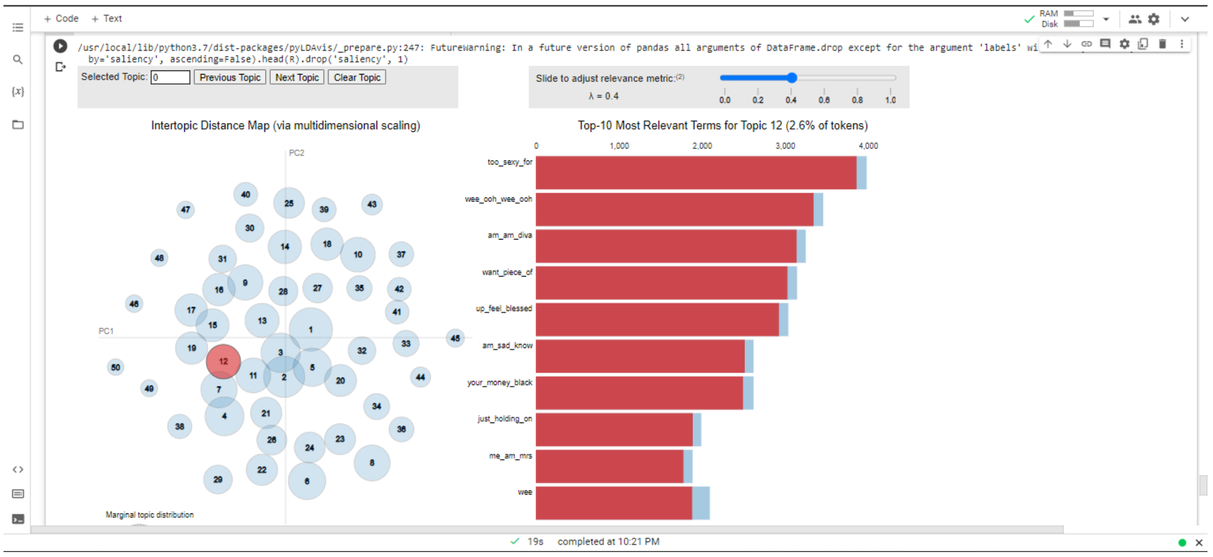
3.13 Topic Modelling on all songs' lyrics

3.13.1 Topic Modelling using LDA

The text in the musical lyrics also normally conforms to certain trending topics. Latent Dirichlet Allocation or LDA is a widely used technique which aligns the text in a document to topic and topic to word matrix to derive the trending / most important topics in the text. LDA was performed using the LDAmodel as a part of Gensim library[7]. A value of 20 as the number of topics to be discovered was provided to the model. Initial results of LDA after passing single words (unigrams) to the model did not make much sense since it is hard to visualize topics with single words. To address this, bigrams of bigrams, trigrams of bigrams of trigrams of bigrams were fed to the model as phrases to compute and uncover topics. To refine the deduction of most important topics only, phrases where the TFIDF scores were lower than a threshold of 0.08 were were dropped from the words / phrases supplied to LDA. The results of LDA after this were impressive at a relevance balancing threshold $\lambda = 0.4$. The LDAVis library [8] was chosen for visualizing the result of LDA showing 50 topics and 10 most relevant words to each topic. Another attempt was made using 20 topics and 20 most relevant words to each topic. Increasing the value of the relevance balancing threshold λ implies that more importance is given to the relevance of the word given the topic i.e. the topic - word association than the marginal probability of the occurance of the word itself i.e. the term frequency. Thus we can see that decreasing λ allows for discovery of topics which have most words relevant to a particular topic while being simultaneously irrelevant to another topic. This further means that increasing λ decreases the inter topic distance and increases confusion between separating the topics whereas decreasing λ increases the inter topic distance thereby defining clearer boundaries for each topic.

3.13.2 Topic Modelling using BERTopic

To check the difference in the results of topic modelling achieved by LDA with those achieved another technique, BERTopic was chosen as an alternate way to discover topics in the song lyrics. The scope of this exercise was however limited to single word topics using BERTopic[6] and without multi-word phrases due to time constraints. About 500 out of 1490 song lyrics could not be categorized under any topic. BERTopic discovers the total number of topics on its own without providing the number of topics to it. It could be observed that the LDA topics made more sense than those discovered by BERTopic. BERTopic would probably give more interesting results if multi word phrases are considered as well. The visuals of LDA with 50 and 20 topics and the Topics output by BERTopic are depicted in figure 12, figure 13 and figure 14 respectively.

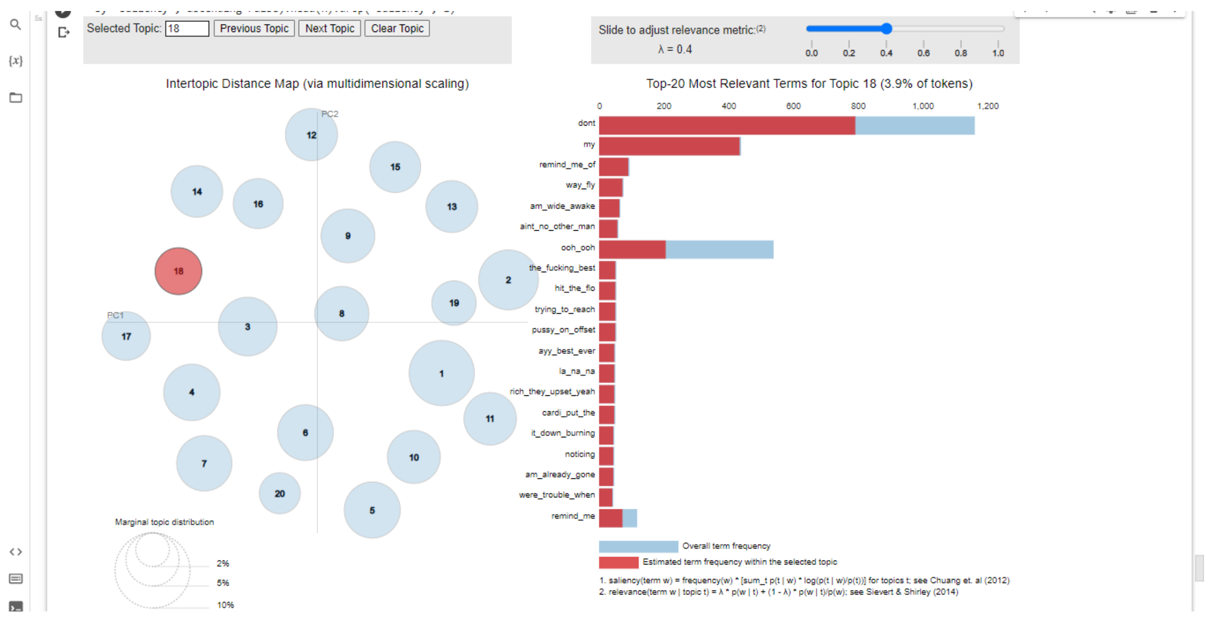


```
/usr/local/lib/python3.7/dist-packages/pyLDAvis/_prepare.py:247: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' wi
  by='saliency', ascending=False).head(R).drop('saliency', 1)
```

Figure 12: LDA Visualization with 50 Topics - 10 best words per topic

4 Validation of Results

1. : **Named Entity Recognition:** The named entity recognition as stated earlier was performed using NLTK [3] Nechunk and using Stanford NER Tagger [5] methods. For validation of the results, a random sample of 10 songs was picked and the entities recognised / missed by both the above methods were bifurcated into 4 categories of prediction as True Positives, False Positives, True Negatives and False Negatives. Basis this the precision, recall, F1 score and accuracy was plotted. The validation outcomes are shown in figure 15 and the corresponding calculations

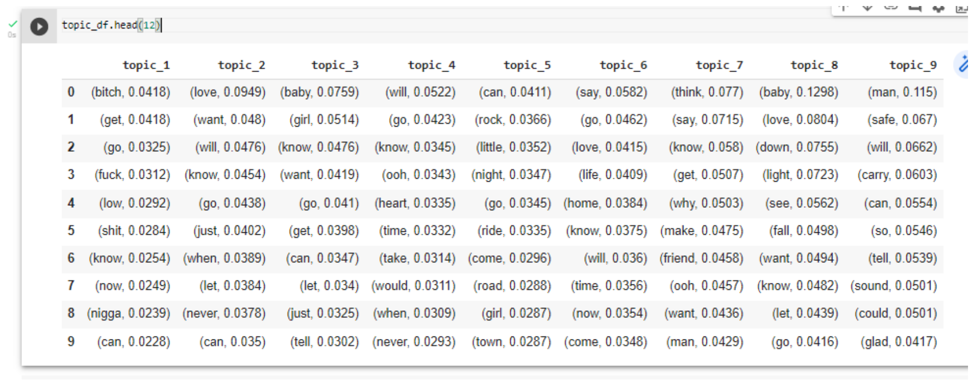

Selected Topic: 18

Slide to adjust relevance metric:(2)

λ = 0.4

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 13: LDA Visualization with 20 Topics - 20 best words per topic

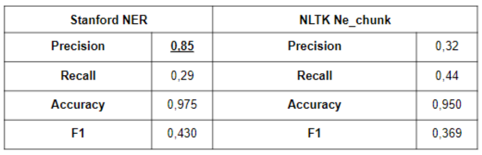


```
topic_df.head(12)
```

	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8	topic_9
0	(bitch, 0.0418)	(love, 0.0949)	(baby, 0.0759)	(will, 0.0522)	(can, 0.0411)	(say, 0.0582)	(think, 0.077)	(baby, 0.1298)	(man, 0.115)
1	(get, 0.0418)	(want, 0.048)	(girl, 0.0514)	(go, 0.0423)	(rock, 0.0366)	(go, 0.0462)	(say, 0.0715)	(love, 0.0804)	(safe, 0.067)
2	(go, 0.0325)	(will, 0.0476)	(know, 0.0476)	(know, 0.0345)	(little, 0.0352)	(love, 0.0415)	(know, 0.058)	(down, 0.0755)	(will, 0.0662)
3	(fuck, 0.0312)	(know, 0.0454)	(want, 0.0419)	(ooh, 0.0343)	(night, 0.0347)	(life, 0.0409)	(get, 0.0507)	(light, 0.0723)	(carry, 0.0603)
4	(low, 0.0292)	(go, 0.0438)	(go, 0.041)	(heart, 0.0335)	(go, 0.0345)	(home, 0.0384)	(why, 0.0503)	(see, 0.0562)	(can, 0.0554)
5	(shit, 0.0284)	(just, 0.0402)	(get, 0.0398)	(time, 0.0332)	(ride, 0.0335)	(know, 0.0375)	(make, 0.0475)	(fall, 0.0498)	(so, 0.0546)
6	(know, 0.0254)	(when, 0.0389)	(can, 0.0347)	(take, 0.0314)	(come, 0.0296)	(will, 0.036)	(friend, 0.0458)	(want, 0.0494)	(tell, 0.0539)
7	(now, 0.0249)	(let, 0.0384)	(let, 0.034)	(would, 0.0311)	(road, 0.0288)	(time, 0.0356)	(ooh, 0.0457)	(know, 0.0482)	(sound, 0.0501)
8	(nigga, 0.0239)	(never, 0.0378)	(just, 0.0325)	(when, 0.0309)	(girl, 0.0287)	(now, 0.0354)	(want, 0.0436)	(let, 0.0439)	(could, 0.0501)
9	(can, 0.0228)	(can, 0.035)	(tell, 0.0302)	(never, 0.0293)	(town, 0.0287)	(come, 0.0348)	(man, 0.0429)	(go, 0.0416)	(glad, 0.0417)

Figure 14: 10 words obtained in 9 BERTopic[6] Topics

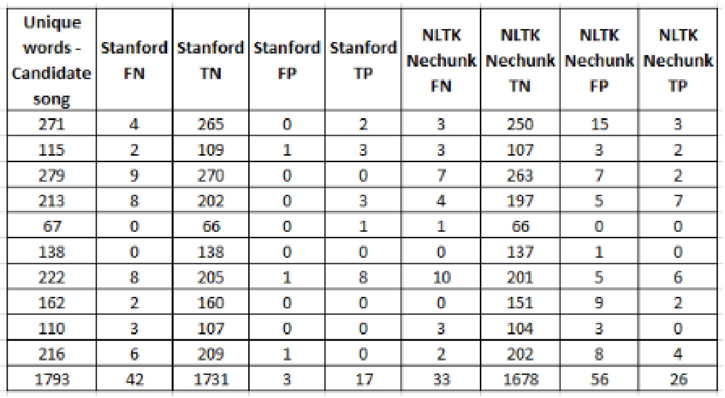
are listed in figure 16. It can be observed clearly that Stanford NER tagger has a much better precision than NLTK Nechunk NER tagger.



Stanford NER		NLTK Ne_chunk	
Precision	0,85	Precision	0,32
Recall	0,29	Recall	0,44
Accuracy	0,975	Accuracy	0,950
F1	0,430	F1	0,369

Figure 15: Outcome of NER validation basis 10 random songs

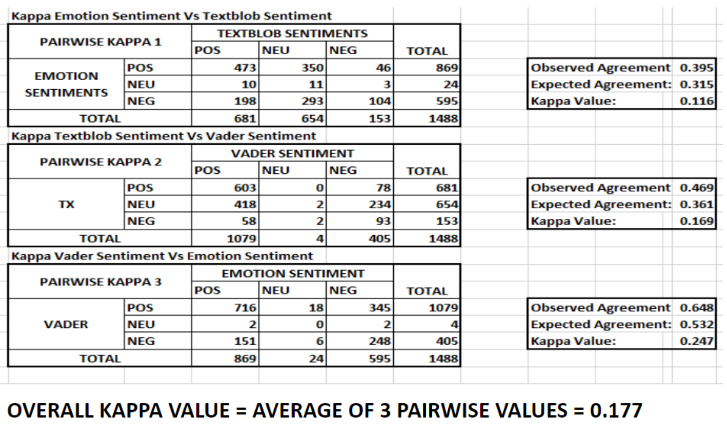
2. **Sentiment Polarity:** Sentiment polarity was analysed using three techniques of TextBlob sentiment, Vader Sentiment and Sentiment polarity computed from quantified emotions. These three predictions of sentiment polarity were treated as three judges and a Cohen's Kappa score



Unique words - Candidate song	Stanford FN	Stanford TN	Stanford FP	Stanford TP	NLTK Nechunk FN	NLTK Nechunk TN	NLTK Nechunk FP	NLTK Nechunk TP
271	4	265	0	2	3	250	15	3
115	2	109	1	3	3	107	3	2
279	9	270	0	0	7	263	7	2
213	8	202	0	3	4	197	5	7
67	0	66	0	1	1	66	0	0
138	0	138	0	0	0	137	1	0
222	8	205	1	8	10	201	5	6
162	2	160	0	0	0	151	9	2
110	3	107	0	0	3	104	3	0
216	6	209	1	0	2	202	8	4
1793	42	1731	3	17	33	1678	56	26

Figure 16: Calculations of NER validation basis 10 random songs

was calculated for checking agreement among the three methods. This was done by averaging the three pairwise kappa values of these methods so obtained by comparing two at a time. Figure 17 shows the outcome of Validation of the results by means of kappa agreement scores. Notably there is lesser agreement between



Kappa Emotion Sentiment Vs Textblob Sentiment

PAIRWISE KAPPA 1		TEXTBLOB SENTIMENTS			
		POS	NEU	NEG	TOTAL
EMOTION SENTIMENTS	POS	473	350	46	869
	NEU	10	11	3	24
	NEG	198	293	104	595
TOTAL		681	654	153	1488

Observed Agreement	0.395
Expected Agreement:	0.315
Kappa Value:	0.116

Kappa Textblob Sentiment Vs Vader Sentiment

PAIRWISE KAPPA 2		VADER SENTIMENT			
		POS	NEU	NEG	TOTAL
TX	POS	603	0	78	681
	NEU	418	2	234	654
	NEG	58	2	93	153
TOTAL		1079	4	405	1488

Observed Agreement	0.469
Expected Agreement:	0.361
Kappa Value:	0.169

Kappa Vader Sentiment Vs Emotion Sentiment

PAIRWISE KAPPA 3		EMOTION SENTIMENT			
		POS	NEU	NEG	TOTAL
VADER	POS	716	18	345	1079
	NEU	2	0	2	4
	NEG	151	6	248	405
TOTAL		869	24	595	1488

Observed Agreement	0.648
Expected Agreement:	0.532
Kappa Value:	0.247

OVERALL KAPPA VALUE = AVERAGE OF 3 PAIRWISE VALUES = 0.177

Figure 17: Cohen's Kappa Agreement - Sentiment Polarities - Compared for Emotion sentiment, TextBlob sentiment and Vader Sentiment

5 Conclusion - What could be better?

Multiple text mining and information retrieval techniques were tried in this study to give insights into the characteristics of popular music. Though the results are not as good as the readers may have liked them to be, this study is meant to be a kick-starter for triggering thoughts in applying text mining to a somewhat unstructured data. Many interesting insights were revealed that would not have otherwise been observed had the analysis been done using traditional text search tools and manipulation methods. As an extension of this study an attempt can be made to have a larger corpus of music and better word embedding techniques can be employed to give better accuracy across techniques employed here.

References

[1] Billboard website top charts.

[2] Song lyrics on genius.com website.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O'Reilly, Beijing, 2009.

[4] Ethan Fast, Binbin Chen, and Michael S. Bernstein. Empath: Understanding topic signals in large-scale text. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016.

[5] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[6] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[7] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[8] Carson Sievert and Kenneth Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.