

Cav Them Jewels Fast: A Lyrical Study of Run the Jewels



DS5001: Exploratory Text Analytics Final Project

University of Virginia

MSDS, Summer 2020

Travis Vitello

tjv9qh

INTRODUCTION

Recognizable by their finger gun-and-fist logo, Run the Jewels is a Grammy-nominated hip hop duo consisting of rappers El-P (New York-based Jaime Meline) and Killer Mike (Atlanta-based Michael Render), a pair whose relationship was forged in the early 2010s. Upon the underground success of their 2013 self-titled debut album, *Run the Jewels*, the pair would go on to release subsequent albums *Run the Jewels 2*, *Run the Jewels 3*, *RTJ4*, and a crowd-funded remix album entitled *Meow the Jewels*. This latter album reimagines much of the music of *Run the Jewels 2*, but with its music overlayed or otherwise supplanted by samples of meowing cats, an unconventional album by a group that attempts to challenge the concepts of the modern rap genre.

The band has been dubbed a “musical militia” by the New York Post for their often-politically charged and unconventional lyrics. It is these very lyrics which this study attempts to explore. Through Python code provided as a supplement to this write-up, methods including sentiment analysis, Principal Component Analysis (PCA), Latent Dirichlet Allocation (LDA), and Word2Vec + Doc2Vec are used to explore each of the aforementioned five Run the Jewels albums. The intent of this study is to derive topics and word frequencies of the Run the Jewels discography, to observe the similarities / differences at the song and album level, and to ultimately attempt to train language models based on the Run the Jewels lyrics.

METHOD

All lyrics used in this study were accessed from the website genius.com. Beautiful Soup was employed to scrape the 57 songs comprising the corpus, whereby each song was categorized according to an Ordered Hierarchy of Content Objects (OHCO) model whose index was based on the album level, song (track) level, and lyric level; this lattermost level is the per-line value of the song, ignoring any empty lines or bracketed items by way of regular expressions. For example, in the genius.com rendering of the song “Lie, Cheat, Steal” there exists text items like “[Hook: Killer Mike]” and “[Verse 1: El-P]”; it is text of this form which will be ignored, as such items do not represent actual lyrical content.

It is recognized that the language used by Run the Jewels may be considered offensive to sensitive listeners. To ameliorate this condition, a censorship function was written that attempts to reproduce the effects of a grawlix. Note that all analysis is performed on the uncensored lyrics, however the censored content will appear with substantial lettering replaced by asterisks. For example: a certain four-letter word will be represented as “fu*k” in displayed content, and so forth.

Further, censorship will be based on the wordlist contained at:

<https://github.com/RobertJGabriel/Google-profanity-words/blob/master/list.txt>

WARNING: The word list at the above link contains very explicit language!



However, in their reproach to conventionalism, Run the Jewels has employed a number of novel words which may still be considered offensive. To this end, additional censorship methods were employed with the intent to capture any such escapees.

It is also noted that lyrics from genius.com are transcribed by users who make a best-effort attempt to capture the lyrics of a song. This process is not perfect. There are a number of examples where lyrics may contain typos (such as “Jamie” in lieu of “Jaime” for some lines referencing rapper El-P, or “tounge” for “tongue”, etc.) or unconventional formatting (“hi-i-i-i-igh” for “high”). In all such cases, a “fix_word” function was written to manually adjust such discrepancies; however, this is not exhaustive and will not necessarily have captured every such instance of user-based typography.

Finally, as the Natural Language Toolkit (**nltk**) is employed in this study for classifying text for part of speech and other methods, all punctuation and capitalization will be stripped from words in the corpus. This results in some ambiguities, as words like “she’ll” becomes “shell” or “he’ll” becomes “he*I” (notice the censorship). In many cases, such contractions are appended to the “English” nltk stop-word list employed in this study. It is recognized that words like “won’t”, “can’t”, and “we’d” can become “wont”, “cant”, and “wed” respectively; discrepancy was used to determine that the inclusion of such words to the stop-word list would not appreciably affect the study. However, “he*I” was not included in the stop-word list so there will be some additional frequency of this term when originally used as “he’ll”.

NOTE: The song “Banana Clipper” features the single lyric “Cause can’t nan”. Such lyric will be tokenized as [cause], [cant], [nan]; however, Python interprets “nan” as equivalent to “NaN” or “not a number”, a float value which cannot be operated upon using string methods. To remedy this condition, it was determined that replacing all occurrences of “nan” as “nann” was a pragmatic solution for the purposes of this study.

In [2531]:			Out[2531]: RTJ_cens.head(15)										
Album_num	Track_num	Lyric_num	Token_num	Album	Title	Lyrics	POS_Tagged	POS	Token				
1	0	1	0	Run the Jewels	Run the Jewels	run run run run run	[(run, VB)]	VB	run				
		1	1	Run the Jewels	Run the Jewels	run run run run run	[(run, VB)]	VB	run				
		2	2	Run the Jewels	Run the Jewels	run run run run run	[(run, VB)]	VB	run				
		3	3	Run the Jewels	Run the Jewels	run run run run run	[(run, VB)]	VB	run				
		4	4	Run the Jewels	Run the Jewels	run run run run run	[(run, VB)]	VB	run				
		0	0	Run the Jewels	Run the Jewels	ay ay ay ay ay ay	[(ay, NN)]	NN	ay				
		1	1	Run the Jewels	Run the Jewels	ay ay ay ay ay ay	[(ay, NN)]	NN	ay				
		2	2	Run the Jewels	Run the Jewels	ay ay ay ay ay ay	[(ay, NN)]	NN	ay				
		3	3	Run the Jewels	Run the Jewels	ay ay ay ay ay ay	[(ay, NN)]	NN	ay				
		4	4	Run the Jewels	Run the Jewels	ay ay ay ay ay ay	[(ay, NN)]	NN	ay				
		5	5	Run the Jewels	Run the Jewels	ay ay ay ay ay ay	[(ay, NN)]	NN	ay				
		6	6	Run the Jewels	Run the Jewels	ay ay ay ay ay ay	[(ay, NN)]	NN	ay				
3	0	0	Run the Jewels	Run the Jewels	oh dear what the fu*k have we here	[(oh, NN)]	NN	oh					
3	1	1	Run the Jewels	Run the Jewels	oh dear what the fu*k have we here	[(dear, NN)]	NN	dear					
3	2	2	Run the Jewels	Run the Jewels	oh dear what the fu*k have we here	[(what, WP)]	WP	what					

Figure 1. Typical Lyric Dataframe

Finally, it’s recognized that the study has involved lyrics only. It is recognized that the Run the Jewels albums exist beyond just the lyrical content itself, and that the musicality is an essential element to their songs.



GENERAL SURVEY AND PREPARATION

The corpus consists of 57 songs across 5 albums, providing a total of 3,854 lyrical lines. These lines yielded 4,842 unique tokens, of which 191 were considered stop-words for the purposes of this study. Upon applying the aforementioned OHCO structure, a typical token dataframe will appear per the following, noting that **pos_tag_sents** from the nltk library may be applied to go give parts of speech per token. It is noted that the lyrics do not follow conventional sentence structure or organization, the **pos_tag_sents** method is a best-attempt at classifying the tokens; however, the classification of “NN”

(common noun) is often applied to words which are not otherwise easy for nltk to parse. The downstream impact of this is that the noun count will be somewhat inflated when viewing a corpus overview. For the purposes of this study, such a condition is recognized, noting that an improvement to this study would be to re-classify any such words listed as the default NN value but which would be better denoted as some other part of speech.

In [821]:	RTJ_cens.sample(11)							
Out[821]:								
	Token_num	Album	Title	Lyrics	POS_Tagged	POS	Token	term_id
Album_num	Track_num	Lyric_num						
2	3	56	0	Run the Jewels 2	Close Your Eyes (And Count to FU'K)	its de la on the cut lifting six on your snitc...	[(its, PRP\$)]	PRP\$ its 2168
	1	60	6	Run the Jewels 2	Oh My Darling Don't Cry	been a bit a bully talk beating on my chest in...	[(beating, NN)]	NN beating 322
	6	15	2	Run the Jewels 2	Early	cause i respect the badge and the gun	[(respect, NN)]	NN respect 3473
5	8	52	3	Meow the Jewels	Snug Again	im about my money hoe	[(money, NN)]	NN money 2691
	2	0	17	Run the Jewels 2	Jeopardy	and i want it i need it see it i take it	[(i, NN)]	NN i 2079
	1	86	2	Run the Jewels 2	Oh My Darling Don't Cry	delivery dope like a dosage of dope	[(like, IN)]	IN like 2392
3	12	73	0	Run the Jewels 3	Thursday in the Danger Room	still talk to you like youre around like youre...	[(still, RB)]	RB still 4064
2	1	75	4	Run the Jewels 2	Oh My Darling Don't Cry	you got the cash ill make the trip	[(ill, NN)]	NN ill 2091
4	7	16	3	RTJ4	never look back	got fortyfour bucks on the smoke whatchu got	[(on, IN)]	IN on 2899
3	5	71	44	Run the Jewels 3	Stay Gold	she got a mean bop i got a lean to the way i walk	[(a, DT)]	DT a 19
5	1	21	5	Meow the Jewels	Oh My Darling Don't Meow	my business card says youre in luck i do two t...	[(in, IN)]	IN in 2113

Figure 2. Typical Dataframe with POS Applied

Stemming of the derived vocabulary was done using Snowball (also known as Porter2). This decision was based on positive feedback found online on the Snowball stemmer as an improvement over Porter, while not being as aggressive as also-popular the Lancaster stemmer. Further study may adjust the stemming method to see the impact on all downstream results.



	term_rank	term_str	n	stop	p_stem	pos_max
term_id						
4277	1	the	1695	1	the	DT
19	2	a	999	1	a	DT
2079	3	i	830	1	i	NN
140	4	and	775	1	and	CC
4818	5	you	669	1	you	PRP
4362	6	to	539	1	to	TO
2113	7	in	473	1	in	IN
2885	8	of	433	1	of	IN
2769	9	my	419	1	my	PRP\$
2098	10	im	369	1	im	NN

Figure 3. Most Common Words in Corpus

TFIDF AND BAG-OF-WORDS

With the tokenized vocabulary established, the TFIDF (term frequency / inverse document frequency) can be calculated to determine the most significant words in the corpus according to the level of detail of interest. For this study, TFIDF calculations were performed at the album and song levels, respectively. This is also the structure of the corresponding bag-of-words models, as shown in the following figures. Upon determining the bag-of-words models, document-term matrices were then computed to determine the frequency of each token according to the respective bag levels; here, the number of columns per matrix was equal to the number of unique tokens (4,842), while the number of rows corresponded to the size of the bag (either 57 for the song level or 5 for the album level).

In [874]:	Out[874]:	Out[875]:																																																																		
<code>1 BOW_1.sample(10)</code>	<table border="1"> <thead> <tr> <th>Title</th> <th>term_id</th> <th>n</th> </tr> </thead> <tbody> <tr> <td>JUST</td> <td>1335</td> <td>2</td> </tr> <tr> <td>holy calamafu'k</td> <td>1201</td> <td>1</td> </tr> <tr> <td>Jeopardy</td> <td>1607</td> <td>1</td> </tr> <tr> <td>A Report to the Shareholders / Kill Your Masters</td> <td>2926</td> <td>1</td> </tr> <tr> <td>yankee and the brave (ep. 4)</td> <td>1184</td> <td>1</td> </tr> <tr> <td>Blockbuster Night, Pt. 1</td> <td>1375</td> <td>1</td> </tr> <tr> <td>never look back</td> <td>4530</td> <td>2</td> </tr> <tr> <td>Don't Get Captured</td> <td>3848</td> <td>1</td> </tr> <tr> <td>a few words for the firing squad (radiation)</td> <td>4772</td> <td>1</td> </tr> <tr> <td>All Meow Life</td> <td>2821</td> <td>4</td> </tr> </tbody> </table>	Title	term_id	n	JUST	1335	2	holy calamafu'k	1201	1	Jeopardy	1607	1	A Report to the Shareholders / Kill Your Masters	2926	1	yankee and the brave (ep. 4)	1184	1	Blockbuster Night, Pt. 1	1375	1	never look back	4530	2	Don't Get Captured	3848	1	a few words for the firing squad (radiation)	4772	1	All Meow Life	2821	4	<table border="1"> <thead> <tr> <th>Album</th> <th>term_id</th> <th>n</th> </tr> </thead> <tbody> <tr> <td>Run the Jewels 2</td> <td>2025</td> <td>1</td> </tr> <tr> <td>Run the Jewels 3</td> <td>4370</td> <td>11</td> </tr> <tr> <td>Meow the Jewels</td> <td>2171</td> <td>7</td> </tr> <tr> <td>Run the Jewels</td> <td>1</td> <td>1</td> </tr> <tr> <td>RTJ4</td> <td>3156</td> <td>2</td> </tr> <tr> <td>Run the Jewels 2</td> <td>1360</td> <td>2</td> </tr> <tr> <td>Run the Jewels</td> <td>811</td> <td>1</td> </tr> <tr> <td>Run the Jewels 3</td> <td>3960</td> <td>1</td> </tr> <tr> <td>Run the Jewels 2</td> <td>958</td> <td>1</td> </tr> <tr> <td>RTJ4</td> <td>4266</td> <td>2</td> </tr> </tbody> </table>	Album	term_id	n	Run the Jewels 2	2025	1	Run the Jewels 3	4370	11	Meow the Jewels	2171	7	Run the Jewels	1	1	RTJ4	3156	2	Run the Jewels 2	1360	2	Run the Jewels	811	1	Run the Jewels 3	3960	1	Run the Jewels 2	958	1	RTJ4	4266	2
Title	term_id	n																																																																		
JUST	1335	2																																																																		
holy calamafu'k	1201	1																																																																		
Jeopardy	1607	1																																																																		
A Report to the Shareholders / Kill Your Masters	2926	1																																																																		
yankee and the brave (ep. 4)	1184	1																																																																		
Blockbuster Night, Pt. 1	1375	1																																																																		
never look back	4530	2																																																																		
Don't Get Captured	3848	1																																																																		
a few words for the firing squad (radiation)	4772	1																																																																		
All Meow Life	2821	4																																																																		
Album	term_id	n																																																																		
Run the Jewels 2	2025	1																																																																		
Run the Jewels 3	4370	11																																																																		
Meow the Jewels	2171	7																																																																		
Run the Jewels	1	1																																																																		
RTJ4	3156	2																																																																		
Run the Jewels 2	1360	2																																																																		
Run the Jewels	811	1																																																																		
Run the Jewels 3	3960	1																																																																		
Run the Jewels 2	958	1																																																																		
RTJ4	4266	2																																																																		

Figure 4. Typical Song (or Track)-Level Bag of Words / Typical Album-Level Bag of Words



In [877]: 1 DTCM_1.sample(6)

Out[877]:

term_id	0	1	2	3	4	5	6	7	8	9	...	4832	4833	4834	4835	4836	4837	4838	4839	4840	4841
Title																					
Legend Has It	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
pulling the pin	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Snug Again	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
JUST	0	0	0	1	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	1
Sea Legs	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
Oh My Darling Don't Meow	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0

6 rows x 4842 columns

Figure 5. Example Matrix of Term Occurrence per Song

In [879]: 1 DTCM_2

Out[879]:

term_id	0	1	2	3	4	5	6	7	8	9	...	4832	4833	4834	4835	4836	4837	4838	4839	4840	4841
Album																					
Meow the Jewels	0	0	1	0	0	0	1	0	0	0	...	0	0	0	0	1	0	0	0	0	1
RTJ4	0	0	0	1	1	0	0	0	0	0	...	0	1	0	0	0	0	1	1	1	0
Run the Jewels	1	1	0	0	0	0	0	3	2	1	...	1	0	1	1	1	1	0	0	0	0
Run the Jewels 2	0	0	2	0	0	0	1	0	0	0	...	0	0	0	0	1	0	0	0	0	1
Run the Jewels 3	0	2	0	0	0	1	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0

5 rows x 4842 columns

Figure 6. Example Matrix of Term Occurrence per Album

Details of the TFIDF calculations can be found the supplemental code, with typical results per the following:

In [889]: 1 TFIDF_1.sample(6)

Out[889]:

term_id	0	1	2	3	4	5	6	7	8	9	...	4832	4833	4834	4835	4836	4837	4838	4839	4840	4841
Title																					
the ground below	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0
Oh My Darling Don't Meow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.000000	0.0	0.0	0.000000	0.001658	0.000000	0.0	0.0	0.0	0.0
36" Chain	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002792	0.00674	0.0	...	0.000000	0.0	0.0	0.000000	0.000000	0.00337	0.0	0.0	0.0	0.0
Get It	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0
All Meow Life	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0
Run the Jewels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.003152	0.0	0.0	0.003152	0.000000	0.000000	0.0	0.0	0.0	0.0

6 rows x 4842 columns

Figure 7. TFIDF Applied to Songs

In [890]: 1 TFIDF_2

Out[890]:

term_id	0	1	2	3	4	5	6	7	8	9	...	4832	4833	4834	4835				
Album																			
Meow the Jewels	0.000000	0.000000	0.000076	0.000000	0.000000	0.000000	0.000000	0.000076	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
RTJ4	0.000000	0.000000	0.000000	0.000104	0.000104	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000104	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Run the Jewels	0.000122	0.000070	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000367	0.000245	0.000122	...	0.000122	0.000000	0.000122	0.000000	0.000000	0.000000	0.000000
Run the Jewels 2	0.000000	0.000000	0.000116	0.000000	0.000000	0.000000	0.000000	0.000058	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Run the Jewels 3	0.000000	0.000094	0.000000	0.000000	0.000000	0.000082	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

5 rows x 4842 columns

Figure 8. TFIDF Applied to Albums



The top tokens according to the TFIDF calculations are given per the following for the song-level bag:

		n	tf	tfidf	term_str	pos_max
Title	term_id					
ooh la la	2947	44	0.085437	0.150016	oui	NN
	2295	52	0.100971	0.098722	la	NN
Stay Gold	1773	59	0.083688	0.088450	gold	NN
DDFH	1960	44	0.118280	0.072121	hey	NN
Meowpurrdy	2609	26	0.040752	0.071556	meow	NN
Call Ticketron	1702	26	0.035568	0.062452	garden	NN
a few words for the firing squad (radiation)	497	40	0.053619	0.061867	brave	NN
ooh la la	89	25	0.048544	0.056010	ah	NN
Don't Get Captured	651	14	0.037940	0.055197	captured	VBN
Run the Jewels	246	23	0.041293	0.052803	ay	NN
ooh la la	2912	27	0.052427	0.051259	ooh	NN
	969	14	0.026975	0.039244	crown	NN
Crown	1994	18	0.034682	0.036656	holding	VBG
	3092	13	0.025048	0.032030	pick	NN
A Report to the Shareholders / Kill Your Masters	2257	93	0.095092	0.030858	kill	NN
Meowrly	1323	16	0.028269	0.029877	early	RB
Oh Mama	2515	19	0.040773	0.027591	mama	NN

Figure 9. Top Tokens per TFIDF for Song-Level Bag

For the album-level bag, the corresponding table appears as the following:

		n	tf	tfidf	term_str	pos_max
Album	term_id					
RTJ4	2947	44	0.006578	0.004598	oui	NN
Meow the Jewels	2609	26	0.004959	0.003466	meow	NN
RTJ4	250	42	0.006279	0.002499	ayy	NN
Run the Jewels 3	1702	26	0.003061	0.002140	garden	NN
RTJ4	2912	31	0.004634	0.001844	ooh	NN
Run the Jewels	246	25	0.004374	0.001740	ay	NN
RTJ4	497	42	0.006279	0.001393	brave	NN
Run the Jewels 3	651	15	0.001766	0.001234	captured	VBN
Run the Jewels	3679	9	0.001575	0.001101	sea	NN
	2361	9	0.001575	0.001101	legs	NNS
	827	18	0.002691	0.001071	cold	NN
RTJ4	4499	10	0.001495	0.001045	uh	NN
	89	26	0.003887	0.000862	ah	NN
Run the Jewels 2	969	14	0.002045	0.000814	crown	NN
RTJ4	2295	52	0.007774	0.000753	la	NN
Run the Jewels 3	979	9	0.001060	0.000741	crying	VBG
RTJ4	2896	7	0.001046	0.000731	ol	NN
	709	7	0.001046	0.000731	charlatans	NNS

Figure 10. Top Tokens per TFIDF for Album-Level Bag



It is observed that in the top word lists according to TFIDF score, there is some overlap with words like “oui”, “meow”, and “garden” near the tops of both lists. These words appear frequently in the songs “ooh la la”, “Meowpurrdy”, and “Call Ticketron” respectively (“oui” is used 44 times, “meow” is used 26 times, and “garden” is used 26 times on said corresponding songs).

These words never appear outside of those tracks (and subsequently, never outside the higher-level albums). Interestingly, at the song-level, the word “gold” ranks highly as it is used a total of 59 times on the track “Stay Gold”, however the word also appears in usage on the tracks “pulling the pin” (*RTJ4*), “Paw Due Respect” (*Meow the Jewels*), “Call Ticketron” (*Run the Jewels 3*), and “All Due Respect” (*Run the Jewels 2*), hence lowering its TFIDF score on the album-level.

The word “kill” can be observed in the song-level table having a count of 93 uses on the track “A Report to the Share Holders / Kill Your Masters”, also giving it one of the highest TF scores; however, its usage (142 times total) across multiple songs (and across multiple albums) dramatically reduce its TFIDF score at both the song and album levels. This word will be revisited for its impact on sentiment later in this document.

One key takeaway is that despite the *Meow the Jewels* album being focused on cat-sounds, it is perhaps unexpected that the word “meow” only appears as a lyric on said album’s opening track!

CLUSTERING

In this report, the results of some basic clustering methods will be shown whereby like-songs and like-albums are found using the TFIDF calculations. This will be presented as select dendograms, the full extent of which, including tables, is available in the supplementary Python code provided with this report. To develop the clustering, pairwise distances were calculated for tokens using the **scipy pdist** method. In the supplementary Jupyter Notebook, “cityblock”, “Euclidean”, “cosine”, “jaccard”, “dice”, “euclidean”, and “sokalsneath” were the arguments considered, however a further study could consider alternative **pdist** arguments such as “canberra” or “russellrao”. Here, only the “sokalsneath” approach will be shown (with others provided, as noted, in said notebook).



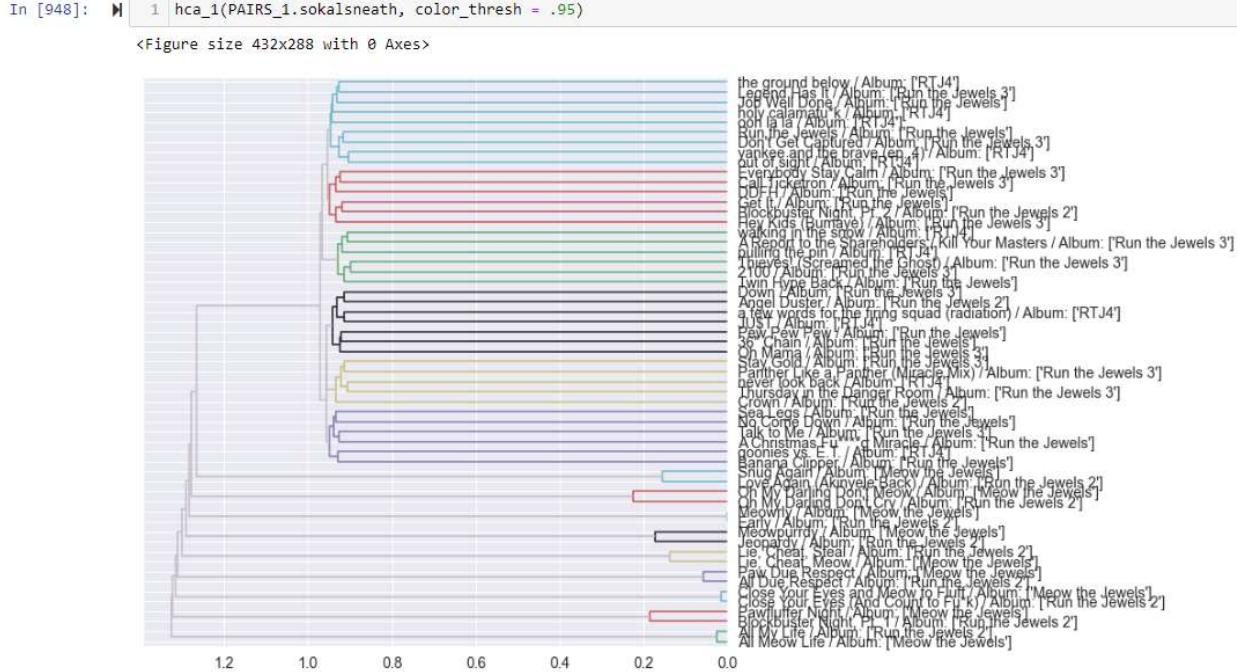


Figure 11. Songs Clustered by pdist

The Sokal-Sneath method computes the dissimilarity between two Boolean 1-D arrays. Here, it is observed that at the song level, this method groups the Run the Jewels songs into 6 major, multi-song clusters (blue, red, green, black, yellow, and purple, respectively), followed by a set of 9 pairs of songs. These 9 pairs of songs represent each track from *Meow the Jewels* and the corresponding track from *Run the Jewels 2* upon which the former was based. A shocking result would have been if the remixed album's tracks did not closely cluster with the original album; however, such a result is fortunately not evident in the data. In each of the major, multi-song clusters there is no immediate pattern as each cluster contains songs from each of the remaining other albums. It is noted that the purple cluster appears to contain "Sea Legs", "No Come Down", "A Christmas Fu****g Miracle", and "Banana Clipper", four of the thirteen tracks off the band's debut, self-titled album. Also observed is the blue group having "the ground below", "holy calamafu*k", "ooh la la", "yankee and the brave (ep. 4)", and "out of sight", five of the RTJ4 album's eleven total tracks.

Because of the very high correlation between the *Meow the Jewels* and the *Run the Jewels 2* tracks, the 9 aforementioned clusters appear to be quite different from the rest of the Run the Jewels discography, which does not appear from the above dendrogram to have any particularly outlying or otherwise unusual songs. However, it's noted that of the tracks on *Run the Jewels 2* which do not have a *Meow the Jewels* counterpart (that is, "Angel Duster" and "Blockbuster Night Pt. 2") can be found in the black and red clusters, respectively. This may suggest that had *Meow the Jewels* not been included in this corpus, the rest of the *Run the Jewels 2* tracks would be expected to cluster well with the rest of the discography.

At the album level, again the Sokal-Sneath method is considered. The resulting clustering is observed per the following dendrogram, noting that dendrograms for each of the other **pdist** approaches are available in the provided Jupyter Notebook.



```
In [974]: 1 hca_2(PAIRS_2.sokalsneath, color_thresh = .93)
```

```
<Figure size 432x288 with 0 Axes>
```

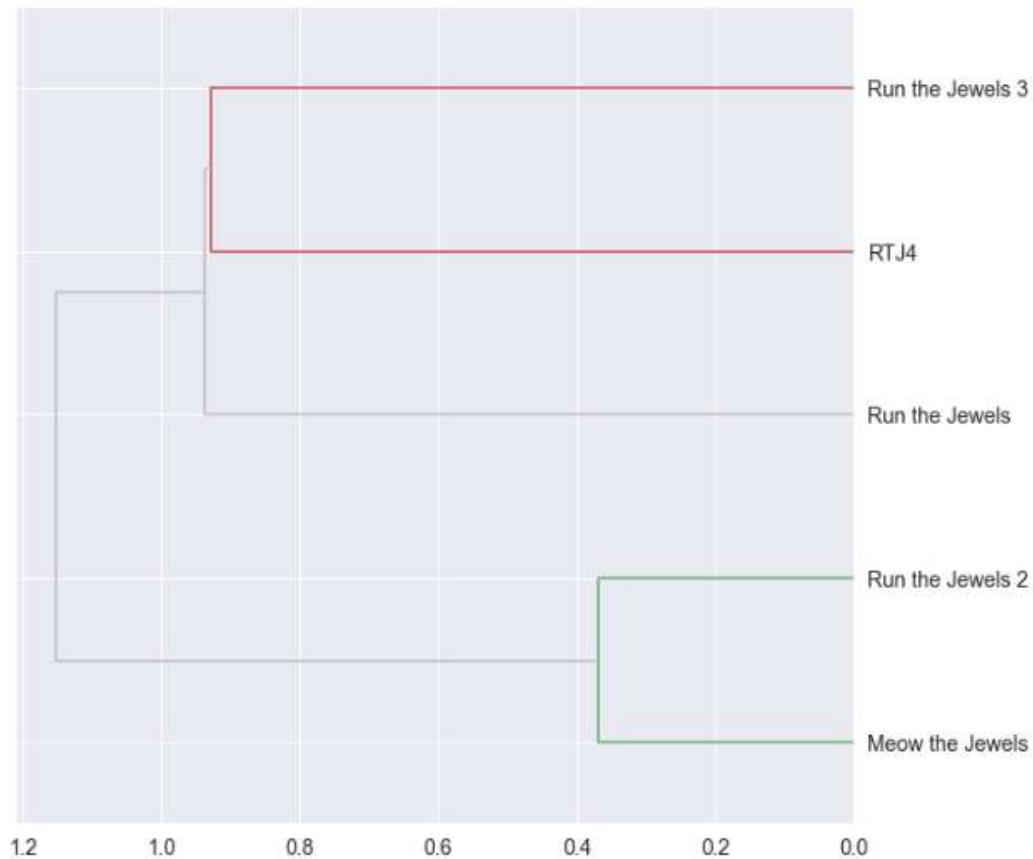


Figure 12. Albums Clustered by pdist

It is observed that *Meow the Jewels* and *Run the Jewels 2* have clustered together closely, with some slight divergence likely in part due to said songs on *Run the Jewels 2* which do not exist analogously on *Meow the Jewels*. Lyrically, the two albums' tracks are similar but not identical (as the previous dendrogram illustrates); of the songs, "Meowrly" and "Early" are clearly the two most closely related songs. Here, it is observed that *Run the Jewels 3* and *RTJ4* have been clustered together. This result may have some basis chronologically, as these albums are the most recently released of the band's discography.

According to *Billboard*, these latter two albums have received more mainstream and international success for Run the Jewels, peaking at 13 and 10 on the Billboard 200 (US) charts respectively (as of the time of this report). These latter two albums, released in December 2016 and June 2020 (again, respectively) may have also been shaped by a similar political climate given the build-up to the 2016 US Presidential Election through the fourth-year of the Trump administration. Also recognized is the timeline of African-American deaths by law enforcement during or upon completion of the *Run the Jewels 2* album, including that of Michael Brown, Eric Garner, Philando Castile, Breonna Taylor, and George Floyd, among others. For example, songs like "walking in the snow" reference the shared Eric Garner and George Floyd quote of "I can't breathe" prior to their respective deaths by law enforcement.



WORD CLOUDS

The following figures represent word clouds generated from the total Run the Jewels discography corpus, along with each of the album-level corpuses. Note that stop-words have been removed in all cases. Also in call cases, the clouds consist of the top 700 words from each of the respective corpuses, with words scaled by frequency according to the **WordCloud** Python method.

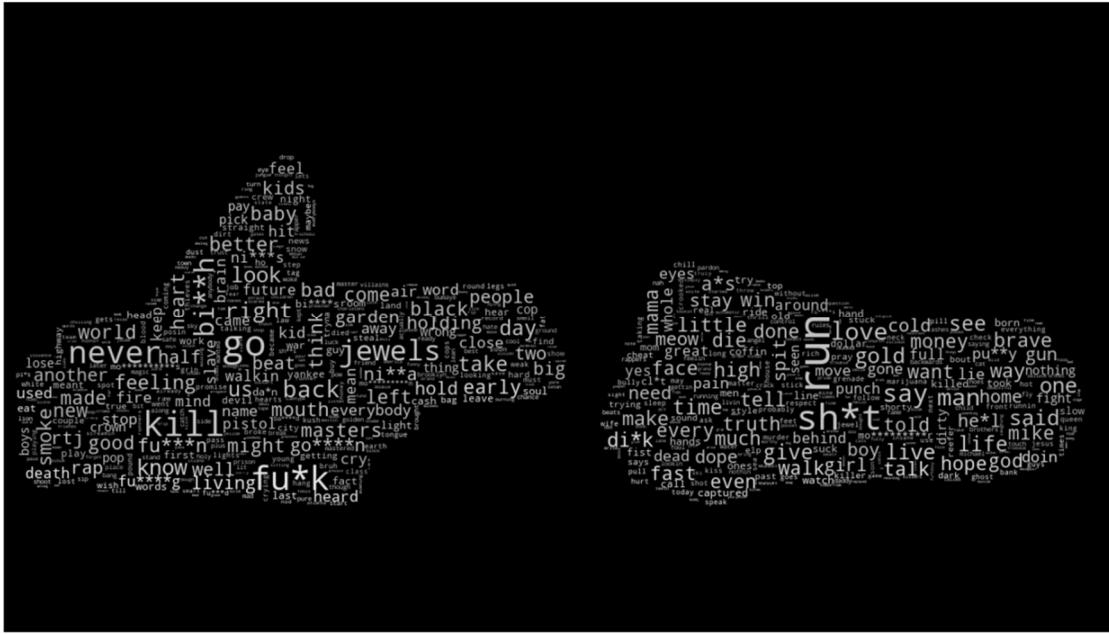


Figure 13. Corpus Word Cloud

At the discography level, immediate words that jump out include “run”, “go”, “kill”, and “jewels” are among the most frequently occurring words, along with obscenities like “fu*k” and “sh*t”. An early sense of themes begin to emerge, where one can observe words associated with violence (“beat”, “gun”, “pistol”, “die”), as well as words which reference sexual topics. By contrast, there are more positively connotated terms like “love”, “life”, “truth”, “live”, and “God”. Notably not observed are lyrics referencing material or commercial brands or products, nor are there recurring themes on topics suggesting drug usage (“smoke”, “high”, and “dope” can each be observed, but generally the word cloud shown supports that such topics are not dominant in the Run the Jewels lyrics).



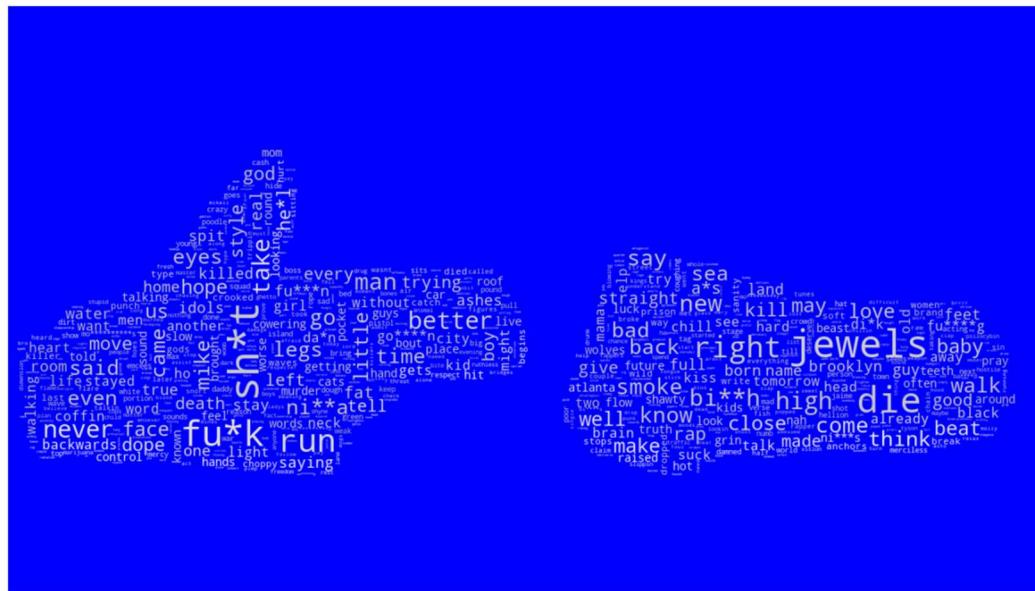


Figure 14. Run the Jewels Word Cloud

For *Run the Jewels*, the band's self-titled debut album, dominant words include "run" and "jewels", as well as obscenities like "fu*k" and "sh*t". The word "die" is dominant, and to a lesser extent other violent words like "kill" and "death" are also evident. Sexual themes do not appear as prominently on this album.

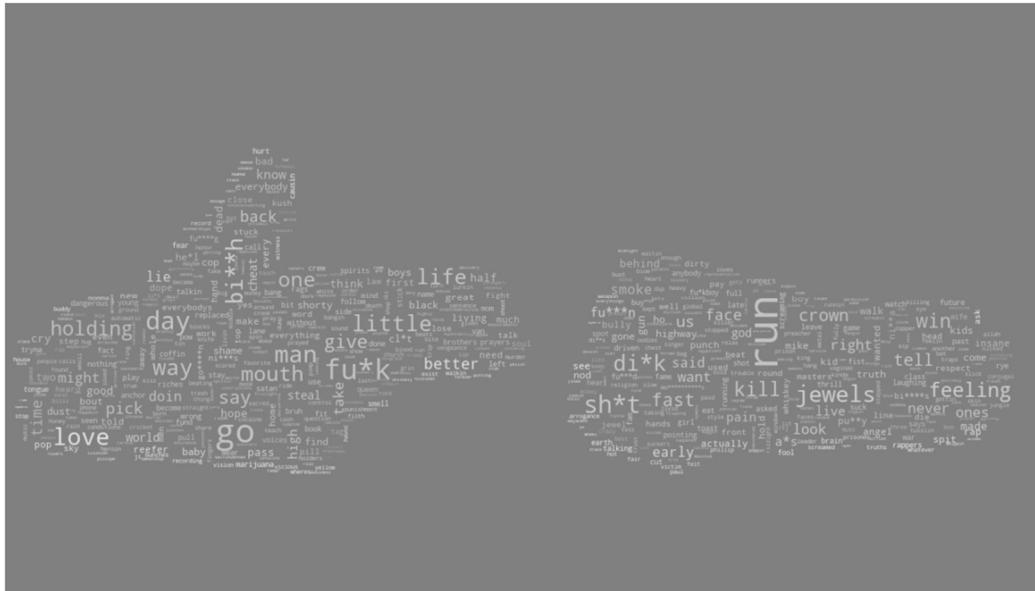


Figure 15. Run the Jewels 2 Word Cloud



On their *RTJ4* album, Run the Jewels appears to have scaled back their use of the word “kill”. Also reduced is the term “jewels” (however the word “run” is shown prominently). Songs like “never look back” inflated the usage of the word “never” for this album, while “a few words for the firing squad (radiation)” helped boost the number of uses of the words “yankee” and “brave”. In general, there does not appear from the word clouds to be as much emphasis on violence or sexual themes as found in the past albums.

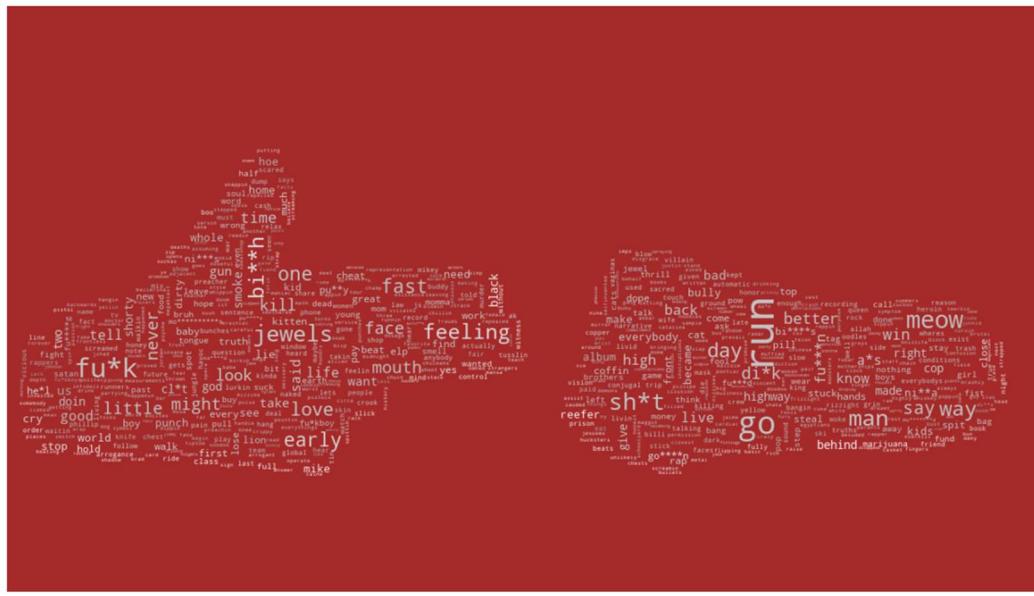


Figure 18. Meow the Jewels Word Cloud

Like the word cloud for *Run the Jewels 2*, terms like “run”, “jewels”, and obscenities like “f*ck” and “sh*t” are prominent on *Meow the Jewels*. There is a strong resemblance to said other word cloud, with the most distinct difference being the presence of the term “meow”.

SENTIMENT ANALYSIS

From the word clouds observed, it is evident that much of the lyrical content within the Run the Jewels discography appears to have negative sentiment due to themes like violence. Sentiment analysis was performed to investigate this topic on both an album level and on the corpus level, with a selection of songs considered based on the findings of such investigation. To determine sentiment, the Sentiment Analysis NRC Word-Emotion Association Lexicon (EmoLex) was used, which considers sentiments associated with negative sentiments (anger, disgust, fear, sadness) and positive sentiments (trust, joy, anticipation, surprise).

Words may be classified by one or more sentiments, with the net positive or net negative aggregate contributing to either a +1 (positive), -1 (negative), or 0 (neutral) polarity score. For



additional perspective, sentiment was also considered using the so-called Bing Liu lexicon, whereby positive and negative word lists were used; this method classifies a words as either +1 (positive), -1 (negative), or 0 (neutral → occurs if the word is not present on either list). Note that the Bing Liu results can be reviewed in the Jupyter Notebook.

Using the EmoLex approach, it is found that the aggregate emotions for the five albums considered in this study has the following distribution, reinforcing the idea that the Run the Jewels discography is generally “negative” sentiment-wise.

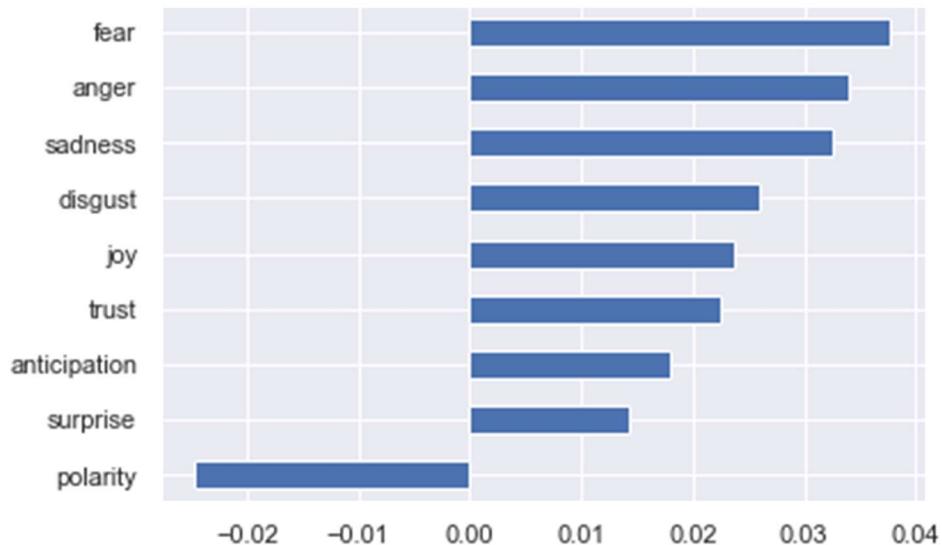


Figure 19. Corpus Sentiment by Category

Next, we consider the equivalent plots for each of the individual albums. For *Run the Jewels*, the sentiments appear as:

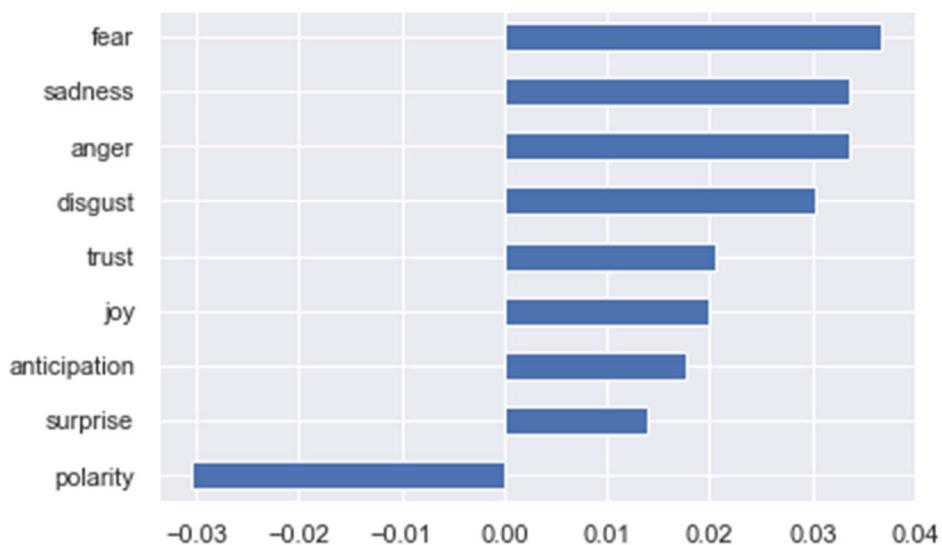


Figure 20. Run the Jewels Sentiment by Category



For *Run the Jewels 2*:

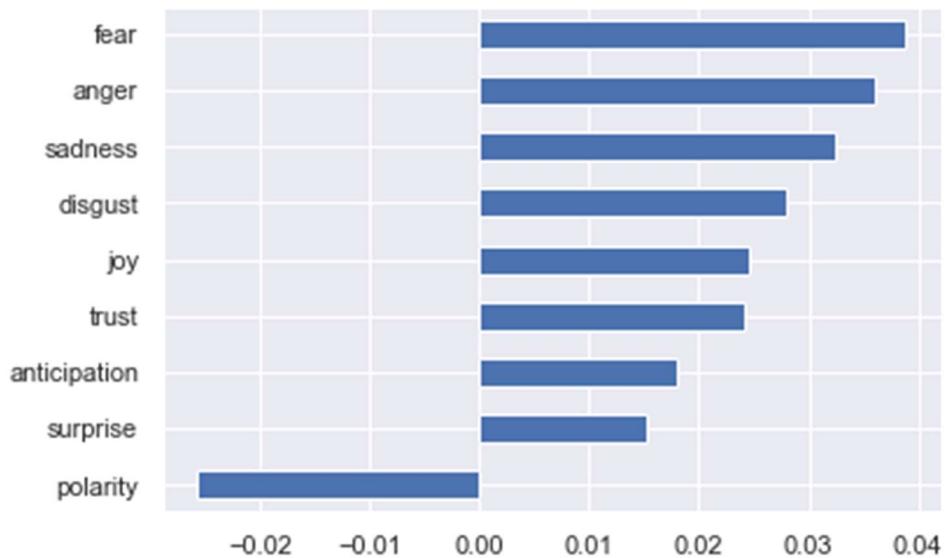


Figure 21. Run the Jewels 2 Sentiment by Category

For *Run the Jewels 3*:

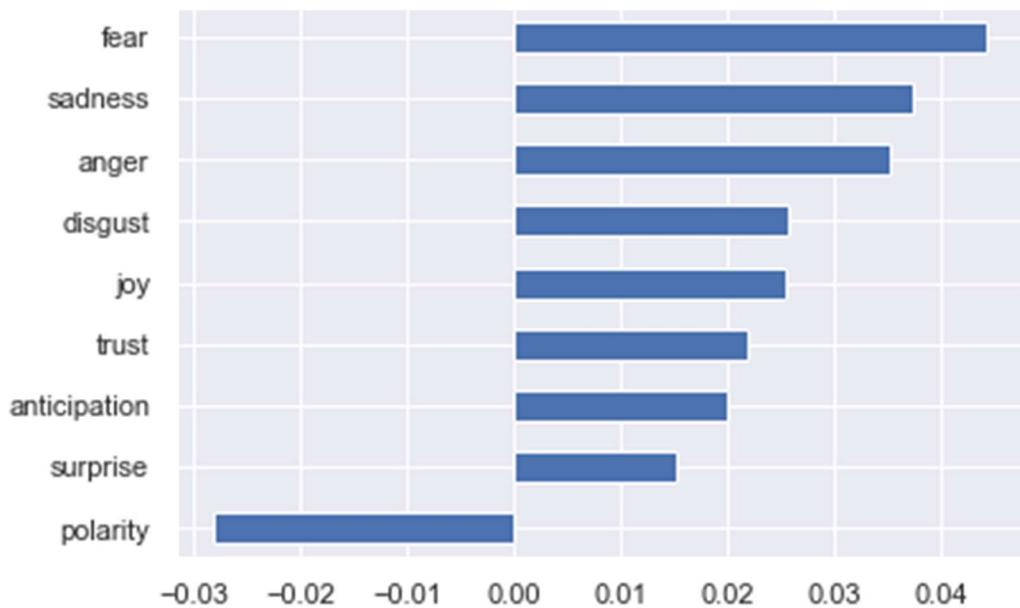


Figure 22. Run the Jewels 3 Sentiment by Category

For *RTJ4*:

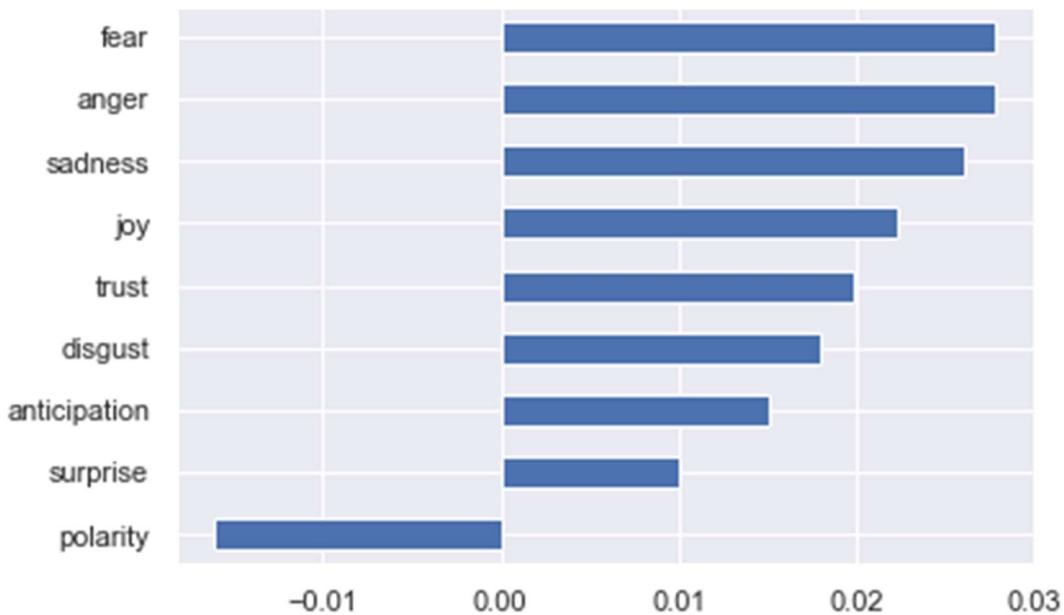


Figure 23. RTJ4 Sentiment by Category

For *Meow the Jewels*:

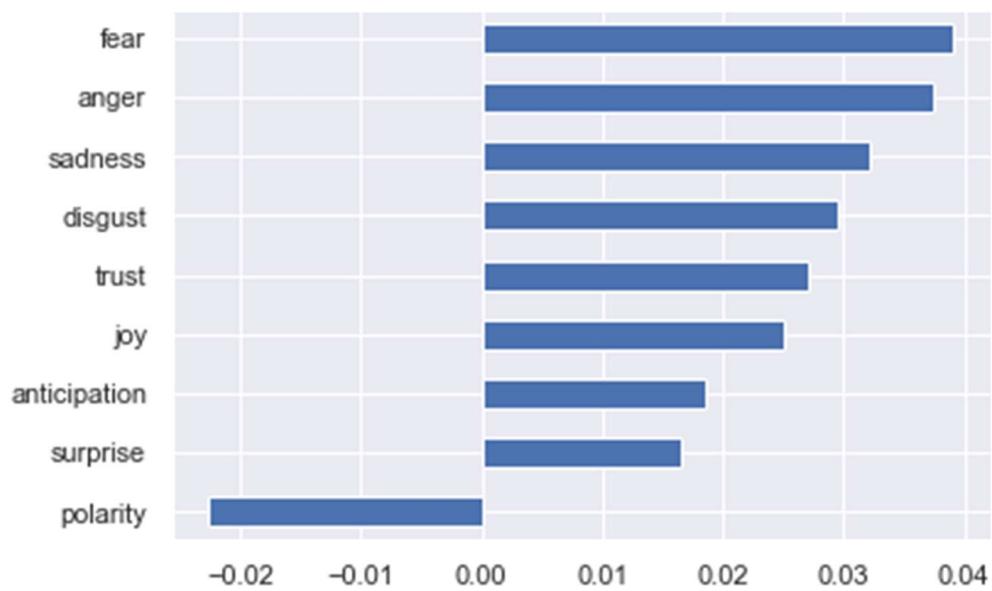


Figure 24. Meow the Jewels Sentiment by Category

It can be observed that not only is the overall discography having a negative polarity, but so too do each of the individual albums. Fear, anger, and sadness are consistently the top 3 emotions



per album, while surprise and anticipation are consistently the bottom 2 emotions. Interestingly, on *RTJ4*, joy is the fourth-highest ranking emotion, having overtaken disgust alongside trust.

It should not be surprising then to observe that the net sentiment per album would see *RTJ4* as the most-positive (or, perhaps better phrased, “least-negative”) album. That is observed here, knowing that Album 1 = *Run the Jewels*, Album 2 = *Run the Jewels 2*, Album 3 = *Run the Jewels 3*, Album 4 = *RTJ4*, and Album 5 = *Meow the Jewels*:

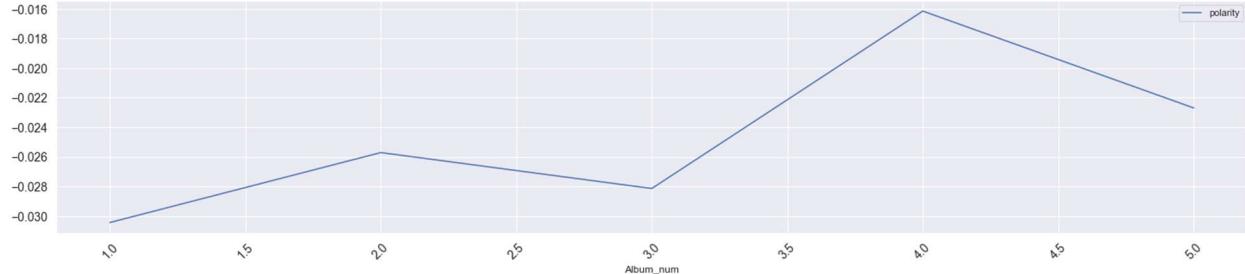


Figure 25. Polarity by Albums

From this chart, the band’s debut album was the most negative, followed by the band’s third album. *Run the Jewels 2* scored a lower sentiment polarity than *Meow the Jewels*, which may be surprising as these two albums are closely related (recalling that *Meow the Jewels* is a remix of *Run the Jewels 2*); however, it is recognized that *Run the Jewels 2* contains two additional tracks which do not have *Meow the Jewels* counterparts --- hence, more opportunity to accumulate negative sentiment.

Across the individual albums, sentiment transitioned by track per the following, starting with *Run the Jewels*:

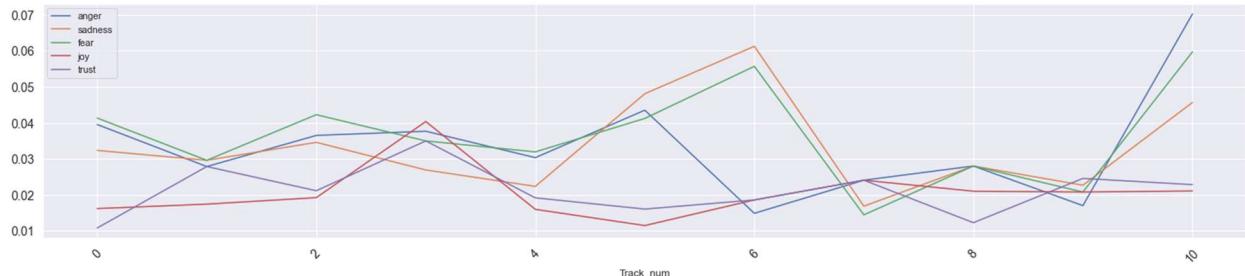


Figure 26. Sentiment for Run the Jewels

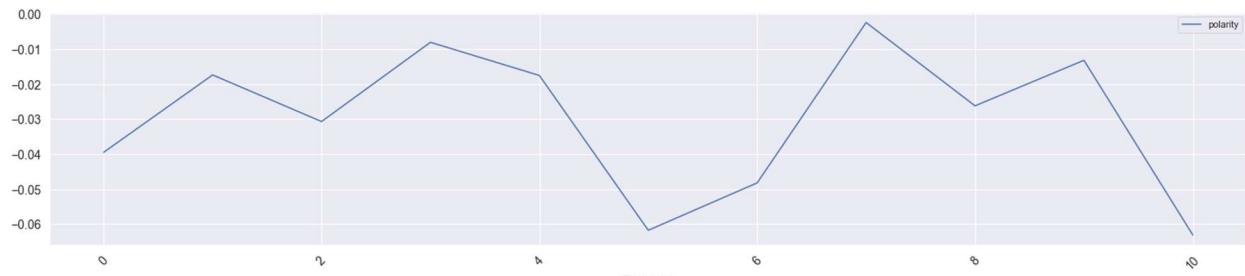


Figure 27. Polarity for Run the Jewels



For Run the Jewels 2:

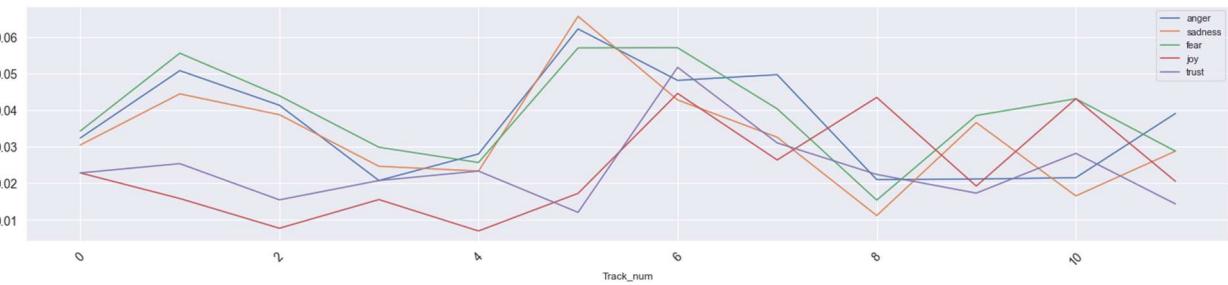


Figure 28. Sentiment for Run the Jewels 2

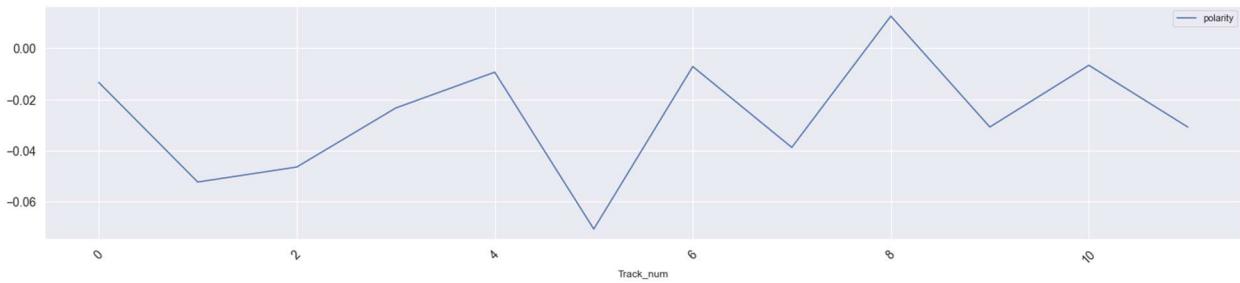


Figure 29. Polarity for Run the Jewels 2

For Run the Jewels 3:

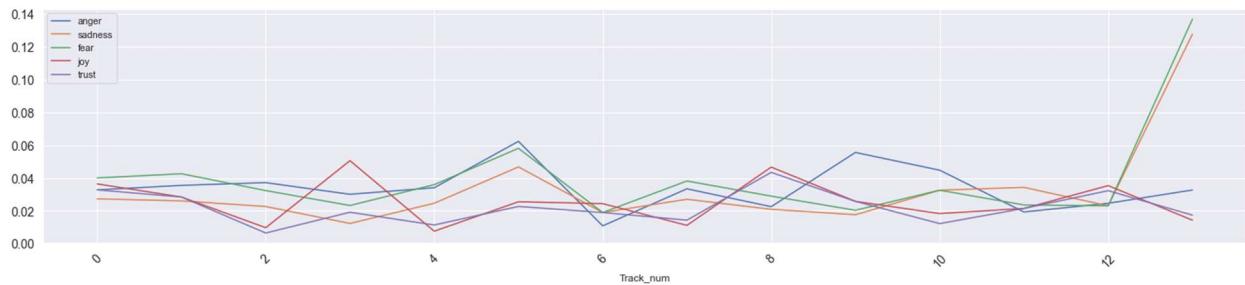


Figure 30. Sentiment for Run the Jewels 3

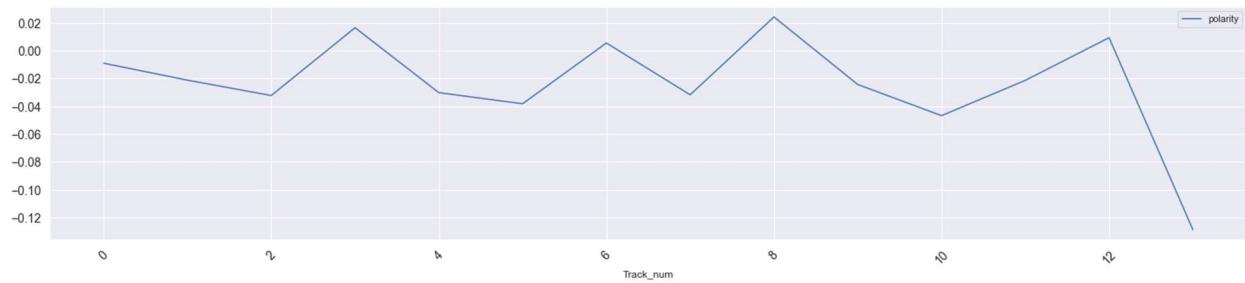


Figure 31. Polarity for Run the Jewels 3



For RTJ4:

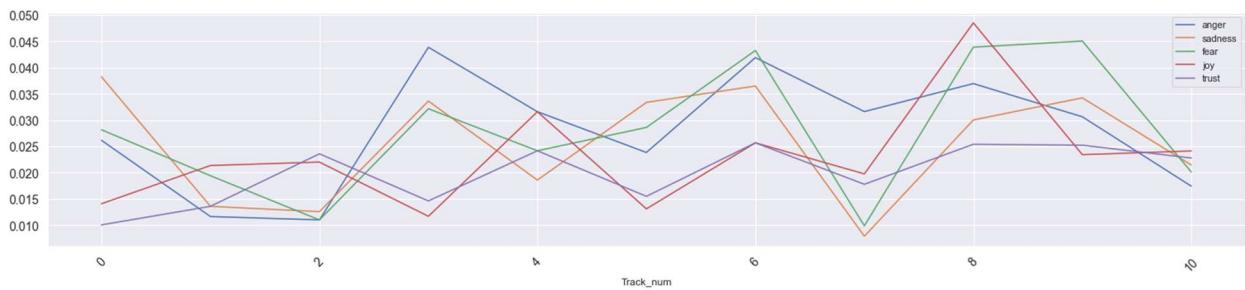


Figure 32. Sentiment for RTJ4

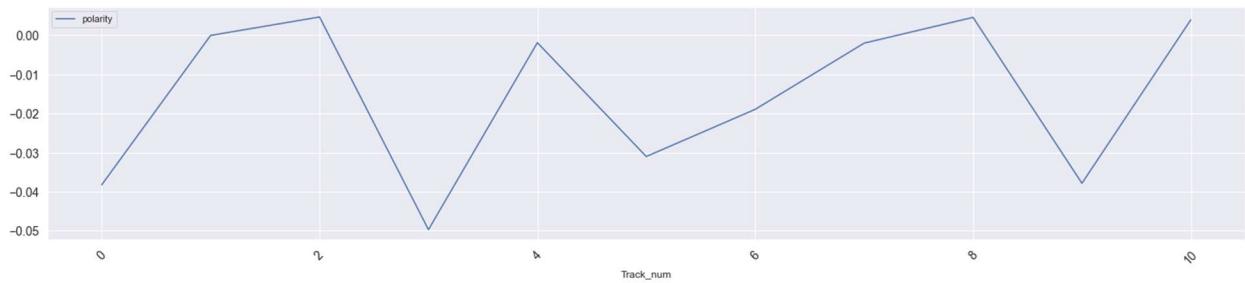


Figure 33. Polarity for RTJ4

For Meow the Jewels:

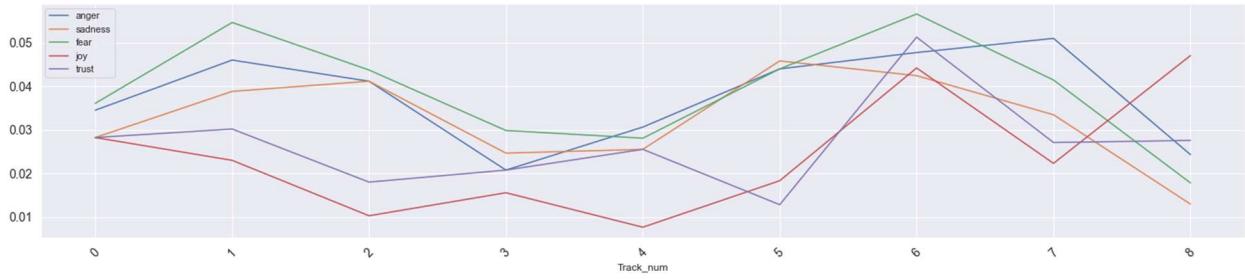


Figure 34. Sentiment for Meow the Jewels

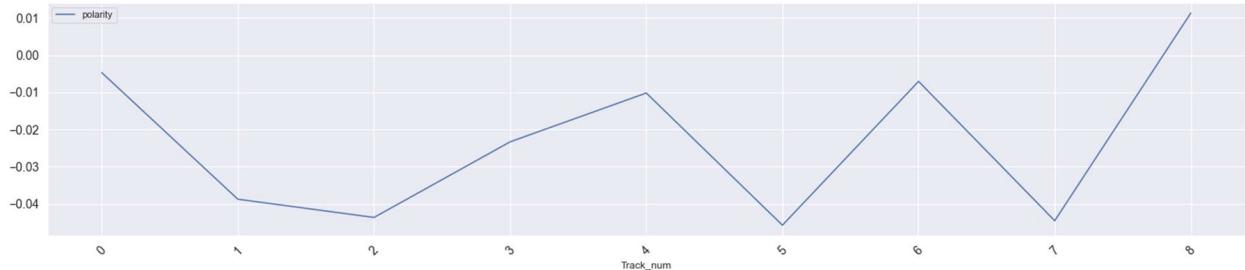


Figure 35. Polarity for Meow the Jewels



To get a sense for what lyrics may contribute to these scores, sampling was performed with negative words highlighted in red, positive words highlight in green, and the entire lyric shaded (red or green) depending on the net sentiment of the lyric (negative or positive). It should be recognized that sentiment is scored on a per-word basis, and that such an approach inherently does not consider context or adjacent words or phrases. As such, it is possible to find a positive scoring lyric that expresses a negative idea and vice versa as the following example demonstrates.

Sentiment	ID	Lyric
-0.2	(2, 10, 49)	got kush for the pain cause the world is dangerous
0.0	(1, 2, 55)	if you come straight from new york you relate
0.0	(3, 13, 60)	devil done got on top of me
0.2	(3, 6, 4)	child soldiers sprayin the chopper
0.2	(2, 7, 84)	cause you get no respect
0.0	(5, 6, 15)	cause i respect the badge and the gun
0.0	(3, 4, 60)	give up that charm or get harmed
-0.1429	(5, 7, 34)	any invader get slayed and stay killed
0.0833	(5, 6, 46)	you know that's the law deal done by the shake of claws
0.6667	(3, 8, 34)	swear to god

Figure 36. Sample Lyrics Identified by Sentiment

Intuitively, one can observe that the two sentences highlighted red are truly negative ideas; however, the third lyric “devil done got on top of me” is scored neutrally as the words “devil” and “top” effectively cancel one another. In reality, this lyric expresses a negative sentiment however the unsupervised algorithm employed in this study cannot achieve that conclusion as-written.

Likewise, the lyric “cause I respect the badge and the gun” is scored neutrally due to the effects of “respect” and “gun” mutually cancelling; however, here it could be argued that this is a positive sentence expressing reverence for law enforcement (rapper Killer Mike’s father was an Atlanta police officer).

Further, it’s noted that the lyric “child soldiers sprayin the chopper” cannot reasonably be expected to be a positive lyric; yet, the EmoLex library views the word “child” as positive, and as such the entire lyric is labelled accordingly.

Across the entire discography, this list includes the highest-scoring lyrics based on positive sentiment:



Out[1353]:													
Album_num	Track_num	Lyric_num	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	polarity	lyr_str	html_str
3	8	76	0.000	0.000000	0.00	0.000000	0.000000	0.00	0.000	1.000000	1.000000	worthy worthy worthy worthy	 worthy <span class=...
			34	0.000	0.333333	0.00	0.333333	0.333333	0.00	0.000	0.666667	0.666667	swear to god
2	10	52	0.000	0.500000	0.00	0.000000	0.500000	0.00	0.500	0.500000	0.500000	angel dust	 angel <span class=...
			30	0.000	0.000000	0.00	0.000000	0.500000	0.00	0.000	0.500000	0.500000	but i promise promise
5	47	1	0.250	0.000000	0.25	0.250000	0.500000	0.25	0.000	0.500000	0.500000	i promise im honest	 i <span class='se...
			0.000	0.500000	0.00	0.000000	0.500000	0.00	0.500	0.500000	0.500000	i hope i hope	 i <span class='se...
2	10	51	0.000	0.500000	0.00	0.000000	0.500000	0.00	0.500	0.500000	0.500000	angel dust	 angel <span class=...
			52	0.000	0.200000	0.00	0.200000	0.400000	0.20	0.200	0.400000	0.400000	true romance in one lapdance
3	12	31	0.000	0.125000	0.00	0.000000	0.375000	0.00	0.125	0.250000	0.375000	just know that i love you good friend	 just <span class=...

Figure 37. Highest Scoring Polarity Lyrics in Corpus

Conversely, this list represents the most negative lyrics based on sentiment. It is recognized that for both the positive and negative case, scores are based on the average sentiment per word per lyric. This is especially evident in the negative list, as single-word negative lyrics are evaluated as having a polarity of -1 for the entire lyric (-1 polarity x 1 word / 1 word).

Out[1354]:													
Album_num	Track_num	Lyric_num	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	polarity	lyr_str	html_str
4	3	35	1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	-1.0	mad	 mad
			36	1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	-1.0	mad
1	5	16	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	-1.0	beast	 beast
			8	27	0.5	0.0	1.0	0.5	0.0	0.5	0.0	-1.0	bad boy
5	0	3	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	-1.0	da*n	 da*n
			34	1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	-1.0	mad
1	10	41	1.0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	-1.0	bang	 bang
			36	0.6	0.0	0.4	0.4	0.0	0.6	0.0	0.0	-0.8	lie cheat steal kill win
2	5	69	0.6	0.0	0.4	0.4	0.0	0.6	0.0	0.0	-0.8	lie cheat steal kill win	 lie <span class=...

Figure 38. Lowest Scoring Polarity Lyrics in Corpus



Next, VADER (Valence Aware Dictionary for Sentiment Reasoning) was considered for each album and the collective corpus. The following represents the results for the collective corpus. It is noted that the Jupyter Notebook contains breakdowns for each of the albums for both the VADER modelling and the above-given sentiment graphs and tables. As can be observed, negative sentiment is consistently larger than positive sentiment throughout the discography, with all albums having a net-negative sentiment.

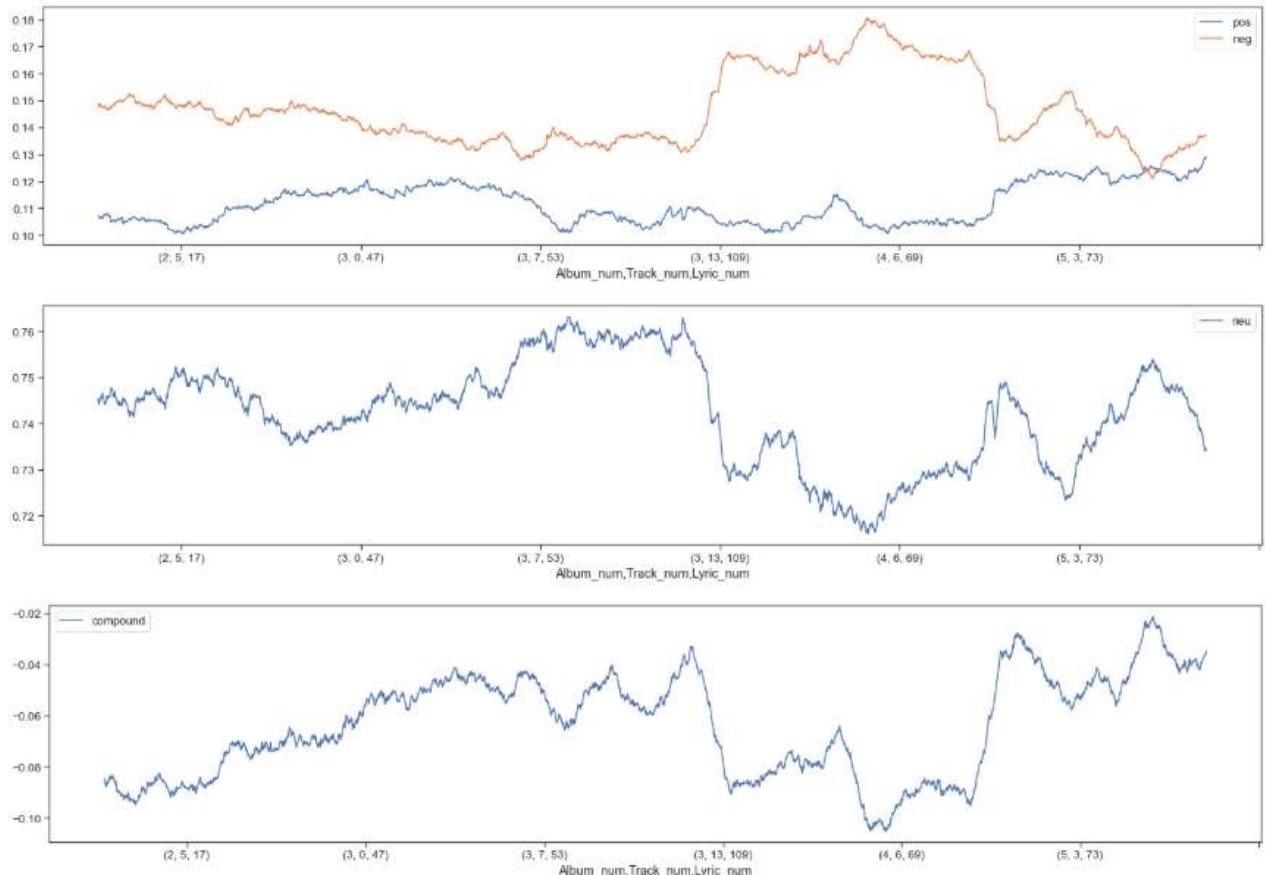


Figure 39. Corpus VADER Plots

To further support the idea that the Run the Jewels discography can be described as “negative”, a heat map was produced based on the EmoLex scoring used to produce the previous figures.



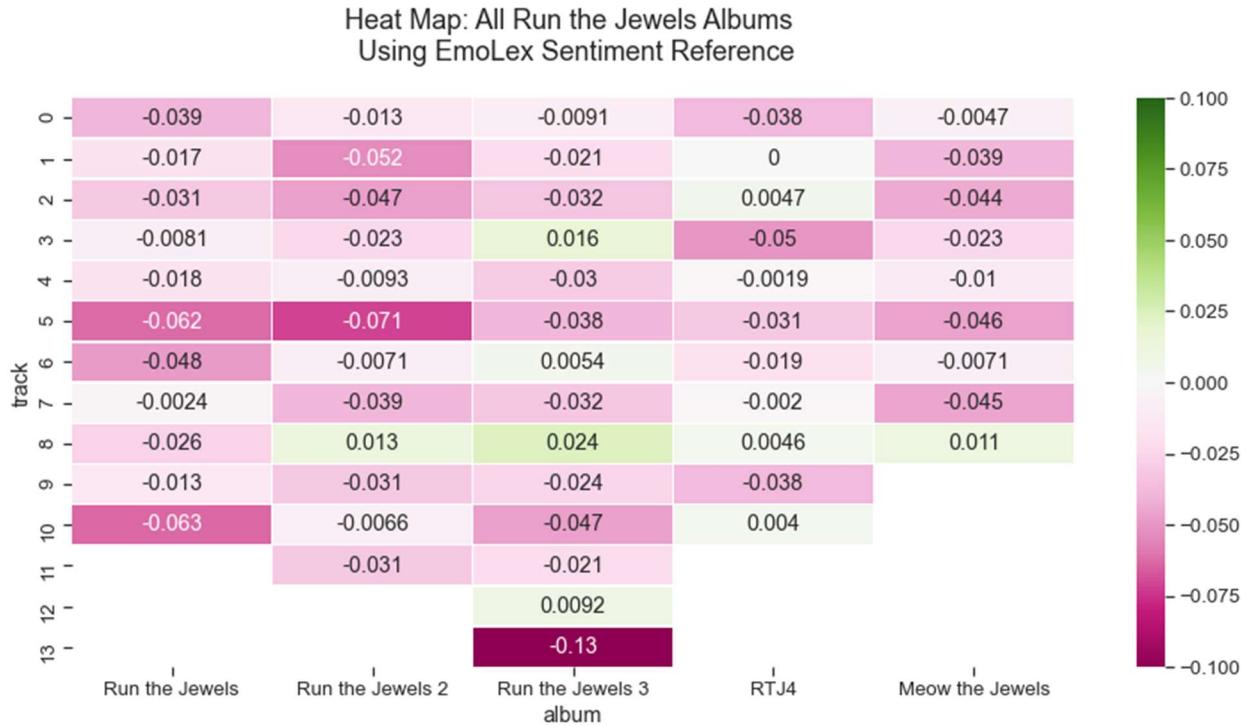


Figure 40. Heat Map of Song Sentiment by Album

This heat map is an average over each track, where the lyrics per song are taken holistically. To break this down more granularly, the following heat map is created. This map shows each album by track, plotted against the lyric number. The result gives an idea for how sentiment fluctuates as each song progresses.

Heat Map of All Songs in Corpus
by Album Number + Track Number vs. Lyric Number
Using EmoLex Sentiment Reference

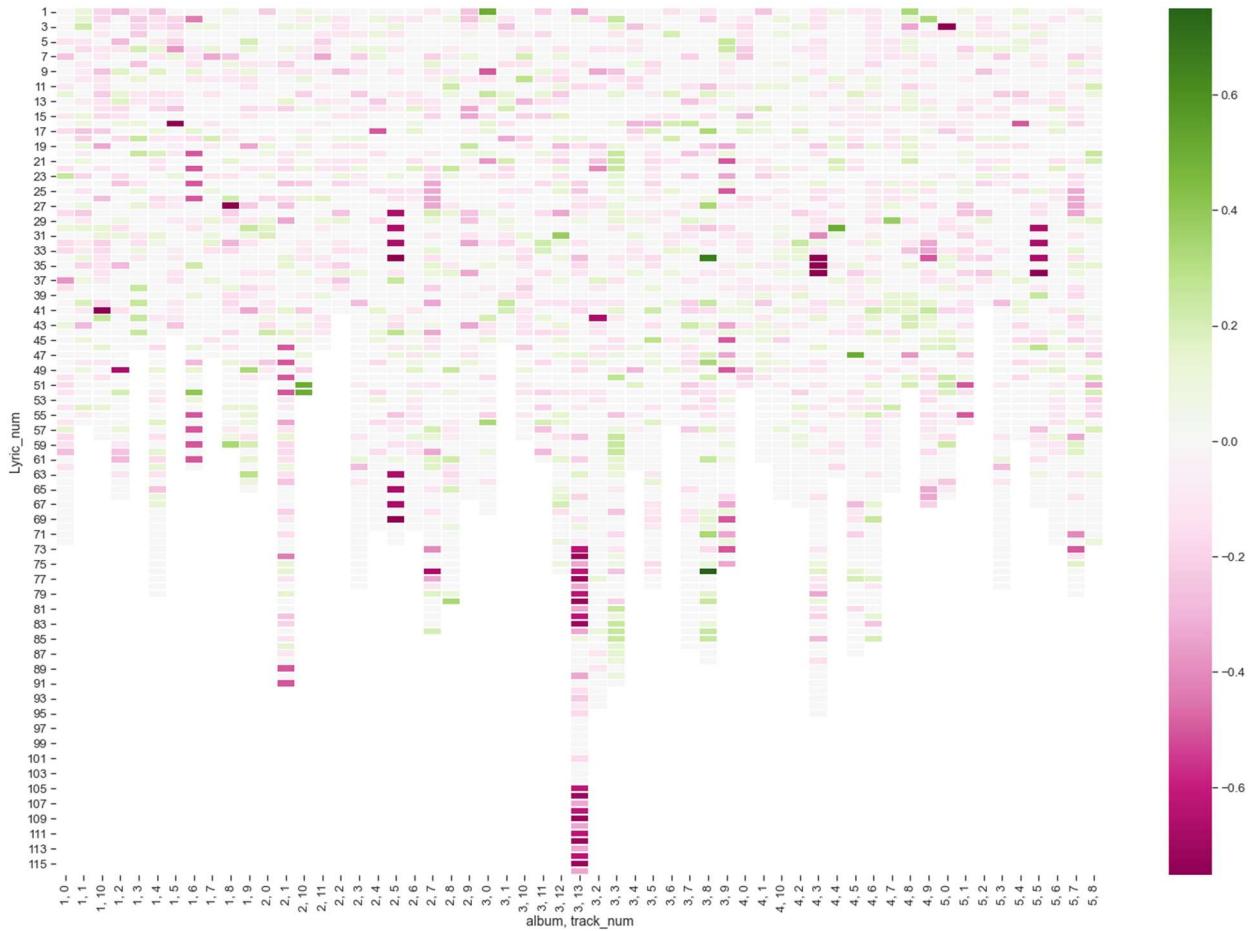


Figure 41. Heat Map of Lyric Sentiment by Album + Track

In general, we observe that the peak negative sentiment in several songs occurs about midway through the song and/or near the end of the song. Very few exceptions, such as the song at index (3, 8) → corresponding to *Run the Jewels 3*, track 9 (the song “2100”) or the song at index (2,8) → corresponding to *Run the Jewels 2*, track 9 (the song “Love Again (Akinyele Back)”), end on positive sentiment.

For reference, “Love Again (Akinyele Back)” ends with the lines “I’ve got this fool in love again (love again) / In love again”, while “2100” ends with the lines “Hoping just to see a beam of the light / (Beam of the light, beam of the light) / Save my swollen heart / Bring me home from the dark / Take me up, take me up, take me up / Take me up, take me up, take me up”. Each of these songs score very positively due to words like “love”, “heart”, and “beam”. Curiously, words like “light”, “up”, and “save” are considered as having neutral sentiments.

To further investigate the Run the Jewels song sentiments, a subset of the songs was selected, including “Legend Has It”, “Call Ticketron”, “2100”, and “A Report to the Shareholders / Kill Your Masters”. These songs were selected, as “Legend Has It” is the groups most mainstream



successful track (achieving RIAA Gold status), while “Call Ticketron” and “2100” have the two most-positive sentiment scores. “A Report to the Shareholders / Kill Your Masters” has the most negative sentiment score of any Run the Jewels track.

For “Legend Has It”, the sentiment score through the song is per:

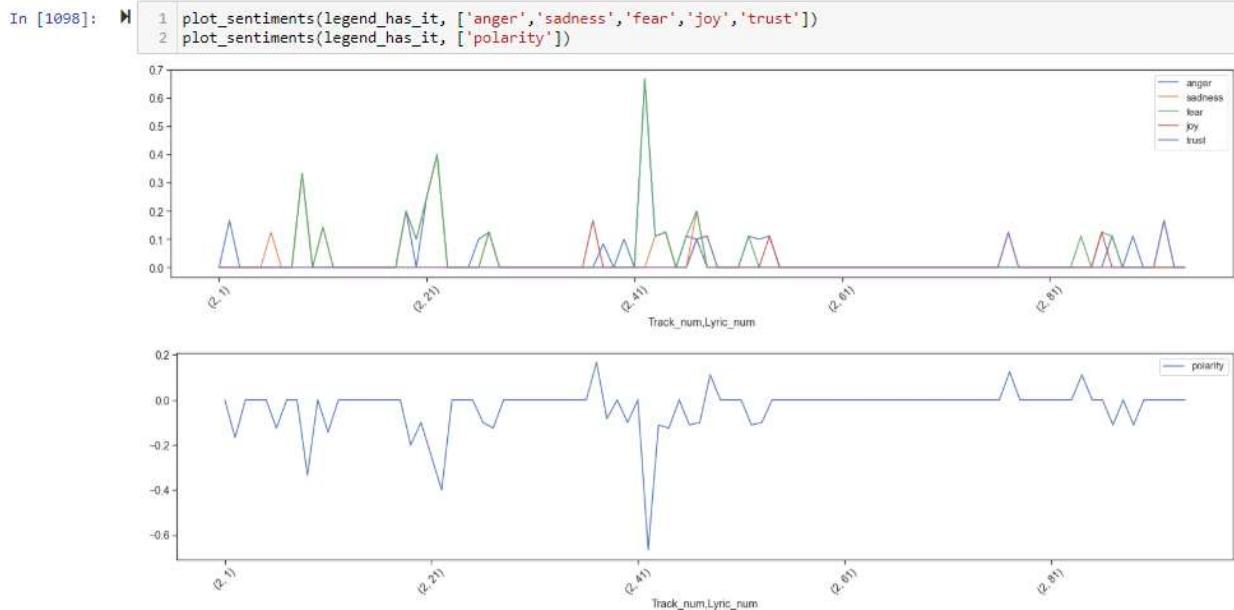


Figure 42. Lyric Sentiment and Polarity for "Legend Has It"

Lyric 42 has the most negative spike, which upon investigation is determined to be the following:

```
In [1099]: 1 legend_has_it.iloc[41]
Out[1099]: anger          0.666667
anticipation    0.000000
disgust          0.666667
fear             0.666667
joy              0.000000
sadness          0.000000
surprise         0.000000
trust             0.000000
polarity        -0.666667
Name: (2, 42), dtype: float64

In [1100]: 1 RTJ_cens.loc[(3,2,42),"Lyrics"].head(1)
Out[1100]: Album_num  Track_num  Lyric_num
3           2          42      villainous treacherous things
Name: Lyrics, dtype: object
```

Figure 43. Negative Lyrics in "Legend Has It"

The lyric “villainous treacherous things” scores 2 out of 3 words as negative (“things” is neutral) for a net sentiment of -2/3.



The next song considered was “Call Ticketron”, whose net sentiment using EmoLex was 0.016. The sentiment plots for this song as per:

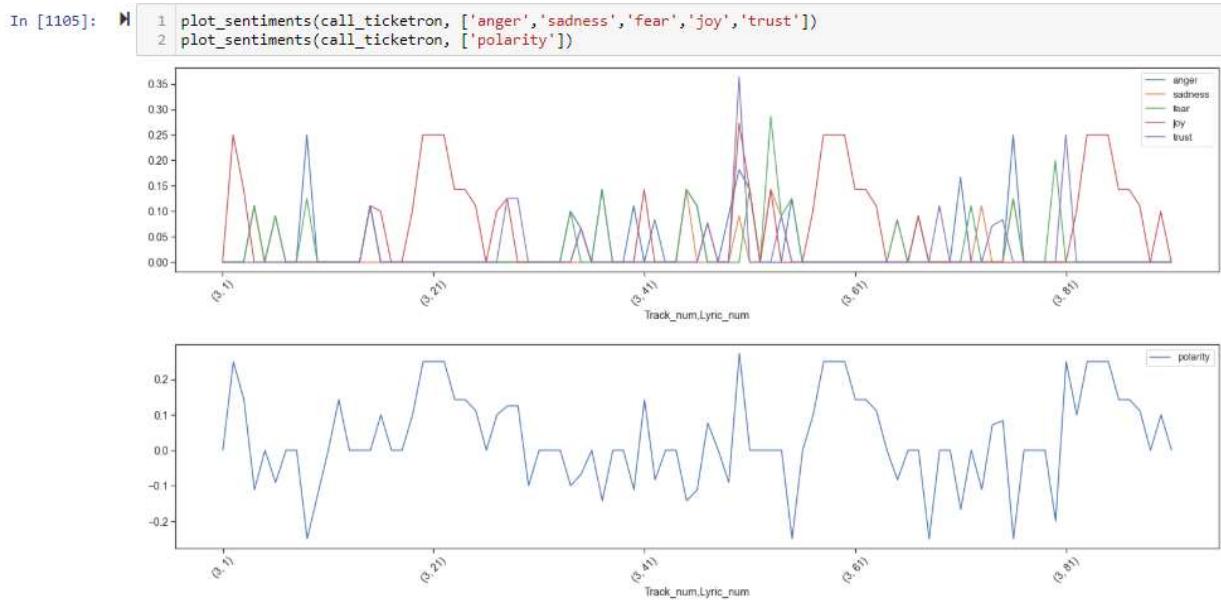


Figure 44. Lyric Sentiment and Polarity for “Call Ticketron”

This song appears to be more complicated than “Legend Has It”, with greater numbers of sentiment spikes throughout its lyrics. However, there is a pattern of three positive lyrics which appear to recur in the song, most likely a repeating chorus. These appear as prominent “joy” sentiments around lyrics 20, 58, and 83. As can be observed in the Jupyter Notebook, these patterns are equivalent; here, only the lyrics around line 20 will be considered.

In [1106]:	call_ticketron.iloc[20:25] # We want to focus on 20 through 25									
Out[1106]:										
Track_num	Lyric_num	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	polarity
3	21	0.0	0.0	0.0	0.0	0.250000	0.0	0.0	0.0	0.250000
	22	0.0	0.0	0.0	0.0	0.250000	0.0	0.0	0.0	0.250000
	23	0.0	0.0	0.0	0.0	0.142857	0.0	0.0	0.0	0.142857
	24	0.0	0.0	0.0	0.0	0.142857	0.0	0.0	0.0	0.142857
	25	0.0	0.0	0.0	0.0	0.111111	0.0	0.0	0.0	0.111111

Figure 45. Typical Positive Signature in “Call Ticketron”

The corresponding lyrics are per the following, found to be a repetition of the proclamation that “Run the Jewels” are “live from the Garden”. The EmoLex dictionary scores “garden” as +1 in the joy and corresponding net positive sentiment categories. Had this lyric been referencing some sort of outdoor area typical of plant growth or other such ideas, then the positive sentiment would



be reasonable. However, in this context the “Garden” is a reference to New York’s Madison Square Garden (home to the NBA’s New York Knicks). While in the context of the song (and corresponding music video), the idea is certainly positive for the band (as in: they were able to perform at such a highly revered venue), it is not exactly the same definition of “garden” as the EmoLex document anticipated. Nevertheless, these lyrics are scored positively, leading to “Call Ticketron” to be among the most positive songs in the discography.

```
In [1109]: 1 RTJ_cens.loc[(3,3,20),"Lyrics"].head(1)
Out[1109]: Album_num Track_num Lyric_num
            3           3          20      live from the garden
Name: Lyrics, dtype: object

In [1110]: 1 RTJ_cens.loc[(3,3,21),"Lyrics"].head(1)
Out[1110]: Album_num Track_num Lyric_num
            3           3          21      live from the garden
Name: Lyrics, dtype: object

In [1111]: 1 RTJ_cens.loc[(3,3,22),"Lyrics"].head(1)
Out[1111]: Album_num Track_num Lyric_num
            3           3          22      live from the garden
Name: Lyrics, dtype: object

In [1112]: 1 RTJ_cens.loc[(3,3,23),"Lyrics"].head(1)
Out[1112]: Album_num Track_num Lyric_num
            3           3          23      run the jewels live at the garden
Name: Lyrics, dtype: object

In [1113]: 1 RTJ_cens.loc[(3,3,24),"Lyrics"].head(1)
Out[1113]: Album_num Track_num Lyric_num
            3           3          24      run the jewels live from the garden
Name: Lyrics, dtype: object

In [1114]: 1 RTJ_cens.loc[(3,3,25),"Lyrics"].head(1)
Out[1114]: Album_num Track_num Lyric_num
            3           3          25      run run run the jewels live from the garden
Name: Lyrics, dtype: object
```

Figure 46. Positive Lyrics in “Call Ticketron”

The next song considered is “2100”, which ranks according to the EmoLex sentiment reference as the most-positive song. This song has already been referenced earlier in this document. Here, the sentiment is plotted per the following (noting the large spikes in trust at lines 34 and 76):



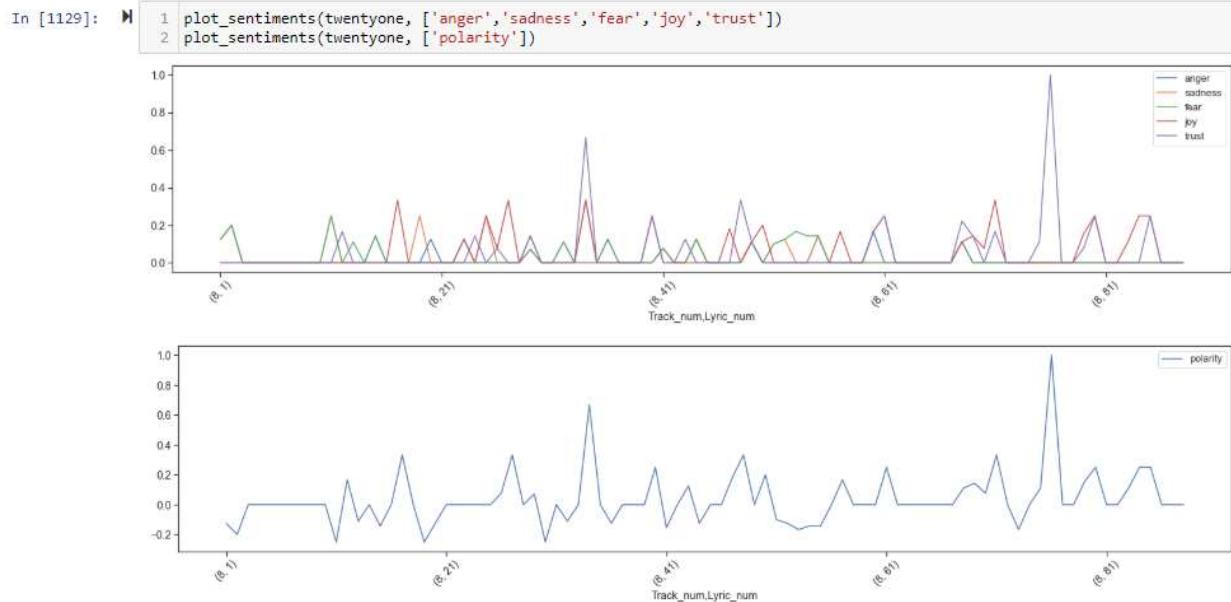


Figure 47. Lyric Sentiment and Polarity for "2100"

These sentiments are scored per the following:

In [1130]:	1	twentyone.iloc[33]
Out[1130]:	anger	0.000000
	anticipation	0.333333
	disgust	0.000000
	fear	0.333333
	joy	0.333333
	sadness	0.000000
	surprise	0.000000
	trust	0.666667
	polarity	0.666667
	Name: (8, 34), dtype: float64	
In [1131]:	1	twentyone.iloc[75]
Out[1131]:	anger	0.0
	anticipation	0.0
	disgust	0.0
	fear	0.0
	joy	0.0
	sadness	0.0
	surprise	0.0
	trust	1.0
	polarity	1.0
	Name: (8, 76), dtype: float64	

Figure 48. Scoring of Example Positive Lyrics in "2100"



Where the actual lyrics are:

```
In [1132]: 1 RTJ_cens.loc[(3,8,34),"Lyrics"].head(1)
Out[1132]: Album_num  Track_num  Lyric_num
            3           8          34      swear to god
            Name: Lyrics, dtype: object

In [1133]: 1 RTJ_cens.loc[(3,8,76),"Lyrics"].head(1)
Out[1133]: Album_num  Track_num  Lyric_num
            3           8          76      worthy worthy worthy worthy worthy
            Name: Lyrics, dtype: object
```

Figure 49. Positive Lyrics in "2100"

At lyric 34, the 3-word line “Swear to God” nets the following: +1/3 trust (“swear”), +1/3 anticipation (“God”), -1/3 fear (“God”), +1/3 joy (“God”), +1/3 trust (“God”), for a net sentiment of 2/3 (or approximately 0.6667).

At lyric 76, the word “worthy” is used 5 times with a +1 trust sentiment (for an average line sentiment of $5 \times 1 / 5 = 1$). The repetition of the word “worthy” evokes ideas like the sequence of the term “holy” in Allen Ginsberg’s poem “Howl”. In general, this song provides a hopeful message in general, despite the occasional negative sentiment that is evidently common to Run the Jewels songs.

Finally, “A Report to the Shareholders / Kill Your Masters” is considered. In both the Bing Liu sentiment study (available in the Jupyter Notebook) and the EmoLex study, this song ranked as *the most negative*. As referenced earlier, this song uses the word “kill” an impressive 93 times. This number is inflated particularly due to sets of lyrics that occur near the song’s end. This can be observed in the following plots:



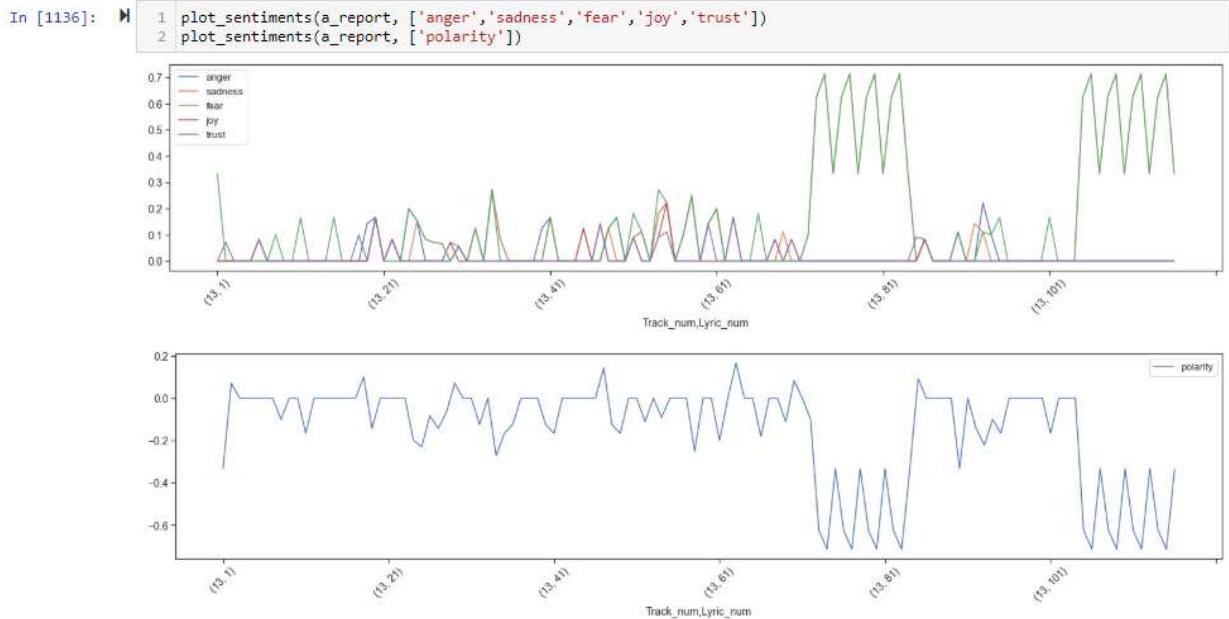


Figure 50. Lyric Sentiment and Polarity for "A Report to the Shareholders / Kill Your Masters"

This behavior aligns with the corpus heat map, with the song (and its corresponding negative lyrics) emphasized for reference:

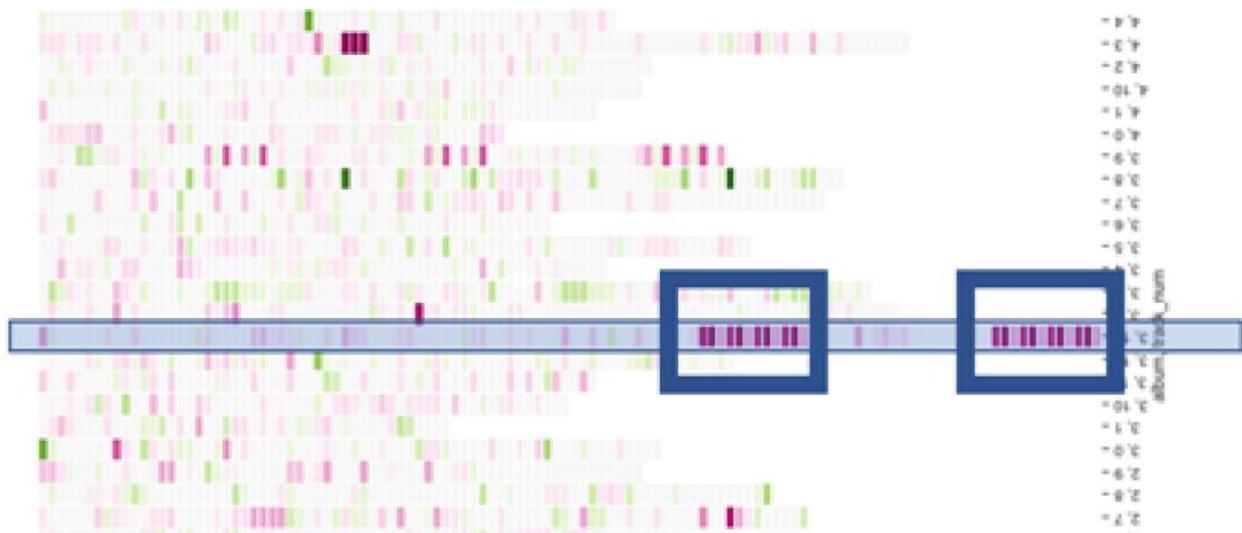


Figure 51. Detail from Heat Map

Those regions (spikes in fear according to the EmoLex plot) represent lyrics 74 through 84, and 106 to 116. These lyrics are equivalent, so only one set will be included below.



```

In [1141]: In [1147]: 
Out[1141]: Out[1147]: 
In [1142]: In [1148]: 
Out[1142]: Out[1148]: 
In [1143]: In [1149]: 
Out[1143]: Out[1149]: 
In [1144]: In [1150]: 
Out[1144]: Out[1150]: 
In [1145]: In [1151]: 
Out[1145]: Out[1151]:

```

	Album_num	Track_num	Lyric_num	Lyrics
In [1141]	3	13	76	kill your kill kill kill your kill
Out[1141]	3	13	76	Name: Lyrics, dtype: object
In [1142]	3	13	76	kill your masters
Out[1142]	3	13	76	Name: Lyrics, dtype: object
In [1143]	3	13	76	kill your kill kill kill your kill
Out[1143]	3	13	76	Name: Lyrics, dtype: object
In [1144]	3	13	77	kill your kill kill kill your kill
Out[1144]	3	13	77	Name: Lyrics, dtype: object
In [1145]	3	13	78	kill your masters kill your masters
Out[1145]	3	13	78	Name: Lyrics, dtype: object
In [1146]	3	13	79	kill your kill kill kill your kill
Out[1146]	3	13	79	Name: Lyrics, dtype: object
In [1147]	3	13	80	kill your kill kill kill your kill
Out[1147]	3	13	80	Name: Lyrics, dtype: object
In [1148]	3	13	81	kill your masters
Out[1148]	3	13	81	Name: Lyrics, dtype: object
In [1149]	3	13	82	kill your kill kill your kill your kill
Out[1149]	3	13	82	Name: Lyrics, dtype: object
In [1150]	3	13	83	kill your kill kill kill your kill
Out[1150]	3	13	83	Name: Lyrics, dtype: object
In [1151]	3	13	84	kill your masters kill your masters
Out[1151]	3	13	84	Name: Lyrics, dtype: object

Figure 52. Negative Lyrics in "A Report to the Shareholders / Kill Your Masters"

It is clear that the sentiment of killing one's masters is repeated here, where the word kill has a -1 sentiment. With so many repeated negative lines, it is understandable how this track became the most negative in the Run the Jewels discography. It is noted, however, that singer Killer Mike on a 2020 interview on *Skip and Shannon: Undisputed* clarified the sentiment to kill one's masters, suggesting that he evokes the notion of standing up against an oppressive ruling socio-economic class, such as corporations or politicians. Said interview is provided within the Jupyter Notebook supplementing this document, for reference.

N-GRAMS AND RANDOM SONGS

As an exercise with the data set, an n-gram model of the corpus was created based on the tokenized terms and corresponding count of occurrence in the Run the Jewels discography. This n-gram model was applied to unigrams, bigrams, and trigrams, the top examples of which are per the following (note that stop-words are included). It is also recognized that probabilities (as taken per the number of specific n-gram occurrences over the set of all equivalent n-grams within the corpus) is also provided:

```

In [1167]: In [1167]: 
Out[1167]: 

```

	n	p
w0		
the	1695	0.051385
a	999	0.030286
i	830	0.025182
and	775	0.023495
you	689	0.020281

Figure 53. Unigram Count and Probability



```
In [1168]: 1 m2.sort_values('p', ascending=False).head()
Out[1168]:
      n      p
w0   w1
in   the  143  0.004335
run  them  125  0.003789
to   the  105  0.003183
of   the   91  0.002759
for  the   90  0.002728
```

Figure 54. Bigram Count and Probability

```
In [1169]: 1 m3.sort_values('p', ascending=False).head(15)
Out[1169]:
      n      p
w0   w1   w2
them run  them  82  0.002486
run  them   run  82  0.002486
it   <s>  <s>  77  0.002334
you  <s>  <s>  53  0.001607
go   go   go  51  0.001546
sh*t <s>  <s>  50  0.001516
me   <s>  <s>  44  0.001334
run  the  jewels 43  0.001304
bi**h <s>  <s>  40  0.001213
kill your kill  40  0.001213
up   <s>  <s>  33  0.001000
mouth all   day  32  0.000970
again <s>  <s>  31  0.000940
ayy   <s>  <s>  30  0.000909
hey   hey  hey  30  0.000909
```

Figure 55. Trigram Count and Probability

These results can be rearranged to provide a trigram matrix whereby rows of bigrams are referenced against columnar unigrams to give the probability of the corresponding trigram:



In [1173]: 1 m3m

Out[1173]:

w0	w1	w2	03	1	2	22	2pac	3	36	45	5th	6	...	zags	zaprunder	zebra	zig	zip	zombies	zone	zoning	zoomin	zulu		
w0	w1	03	but	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	<s>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	and	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	ba***d	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	<s>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
...
zombies	folded	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
zone	hope	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
zoning	district	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
zoomin	<s>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
zulu	mansa	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

17547 rows × 4345 columns

Figure 56. Trigram Dataframe

The following function, adapted from code written by UVA's Raf Alvarado and appearing as follows, randomly generates song lyrics using the n-gram probabilities as found above. The following are examples of such "songs", recognizing that these are ultimately weighted concatenations of lyric fragments within the corpus. It is recognized that missing from these examples will be any sense of rhyme, syllabic structuring, or theming. Any occasions to the contrary are purely coincidence.

In [1539]: 1 generate_text()

```

drinking and partying with fast women with his mom and auntie
which must be a fool for the win so we stumble blind through depths of the opera
the summertime choke ayy ayy ayy ayy
talkin middle of the kill yes it excites me
wheel one black one white we shoot to kill
return to us victorious smoke pounds of the sirens
cutter spit
tryin to feed tummies
know it
run them run them run them run them run them jewels fast run them run them run them run them jewels fast run them
run them
when we were finished
our bl***y noggins off
the ones whose body hung from a shriek to whisper i cant behave
devils they do the deed and then dash
halfwit vision
back see
know a few people pray for my demise yall
run them run them run them run them run them run them
scrumptious theyll put your fu****g jazz hands back in

```

Figure 57. Randomly Generated Run the Jewels-inspired Song, #1



```
In [1577]: 1 generate_text()
slapper
sleep depriver
i stay holy and high high
for a wish for you all night
for data
like mellow out man just relax
and now we shine just like eddie
michelin star im on mine i be miserable
of the phone that shakes and rattles his bones
invader get slayed and stay killed
of that pressure pressure
life that im livin
a thought
run run
mistake to think about it
meant much
highest of hopes
for ritalin
live from the pride in the physical
came out alright
wont let you be bantering
killed by his property
im missing my girl got my style is gone im a nut punch wizard
lobotomy toxicity
lights of fu*kery stuck in a fist and gun position
mike get along
dog
and a sh*t not at all
game where the cops choke out a cloud of reefer
```

Figure 58. Randomly Generated Run the Jewels-inspired Song, #2

```
In [1605]: 1 generate_text()
devotion
mouth all day day
hate that we cant end it
is gone im a mixture of mfg and the busters of locks locks
and imma put a mirror on the kingdoms that killed the dreams of the gas mike twisted
ask why cause the world will slice it up with frizzight like mizzike is you
or uncle johnny
of their mouths
not to slam into traffic
i prayed that god was her lover
im a living on limited time
tellin her to tell me how could i not show it
go****n screen
the coast scream yall got another planet on stash
that sh*t thats right
yankee and the shriek of the villains is here
cheat steal kill win win
im selfinvented
yourself
dont lock me up in my coliseum
you say that you never get better then you can sip my bi****s brew
stopped talkin
my son time to go after the fact that i would
```

Figure 59. Randomly Generated Run the Jewels-inspired Song, #3

This topic can be explored further by re-running any of the Jupyter Notebook cells structured for generating random songs. From this exercise, it was observed that most randomly generated Run the Jewels-inspired songs contain one or more censored words, as well as violent and/or sexual lyric content.



PCA

Principal Component Analysis was performed on the corpus at song level. This method initially required the calculation of the covariance matrix as derived from the previously calculated TFIDF values. The results of such calculations are per the following:

term_id	0	1	2	3	4	5	6	7	8	9	...	4832	4833
term_id	0	1.038035e-04	-0.000004	-1.040504e-06	-8.053528e-07	-1.473132e-06	-0.000003	-0.000004	-0.000004	-1.845597e-06	-	-1.140662e-06	-1.040504e-06
1	-3.958782e-06	0.000153	-0.000008	-2.228630e-06	-1.724955e-06	-3.165266e-06	-0.000007	-0.000008	-0.000008	-3.955182e-06	-	-2.443135e-06	-2.228630e-06
2	-1.591948e-05	-0.000008	0.000130	-2.022116e-06	-1.565123e-06	-2.862887e-06	-0.000008	-0.000008	-0.000007	-3.588579e-06	-	-2.215745e-06	-2.022116e-06
3	-1.040504e-05	-0.000002	-0.000002	3.280257e-06	-4.533801e-07	-8.299121e-07	-0.000002	-0.000002	-0.000002	-1.039557e-06	-	-5.421397e-07	3.280257e-06
4	-8.053526e-07	-0.000002	-0.000002	-4.533801e-07	1.965137e-05	-4.418899e-07	-0.000001	-0.000002	-0.000002	-8.048202e-07	-	-4.970179e-07	-4.533801e-07

Figure 60. Covariance Matrix for Song-Level Terms

term_id	0	1	2	3	4	5	6	7	8	9
term_id	0	0.000104	-0.000004	-0.000001	-0.000001	0.000001	-0.000002	-0.000004	-0.000004	-0.000004
1	-0.000004	0.000153	-0.000008	-0.000002	-0.000003	-0.000007	-0.000008	-0.000008	-0.000004	-
2	-0.000004	-0.000008	0.000130	-0.000002	-0.000002	-0.000003	-0.000008	-0.000008	-0.000007	-0.000004
3	-0.000001	-0.000002	-0.000002	0.000033	-0.000005	-0.000001	-0.000002	-0.000002	-0.000002	-0.000001
4	-0.000001	-0.000002	-0.000002	-0.000001	0.000020	-0.000001	-0.000001	-0.000002	-0.000002	-0.000001

Figure 61. Subset of Covariance Matrix for Song-Level Terms

Using these results, a Document Composition Matrix (DCM) can be calculated, whereby each of ten principal components (P0 through P9) are considered (however only P0 through P3 will be used).



	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	title	label
title												
Oh My Darling Don't Meow	0.151961	-0.217862	-0.256062	0.138546	0.065932	-0.735793	0.330232	0.272652	0.003600	-0.062663	Darling Don't Meow	Meow the Jewels
ooh la la	0.036586	-0.014093	-0.019262	0.006812	0.019254	-0.004218	0.018928	-0.033432	-0.006225	0.027861	ooh la la	RTJ4
Job Well Done	0.013550	-0.029087	-0.017607	-0.016739	-0.005834	-0.072310	-0.008596	-0.093108	-0.043295	-0.082006	Job Well Done	Run the Jewels
Oh Mama	-0.026116	-0.005338	-0.017738	-0.058447	-0.032038	0.017269	-0.009678	-0.092730	-0.085713	-0.007751	Oh Mama	Run the Jewels 3
All My Life	-0.495195	0.449995	-0.099473	0.187493	0.573440	-0.057447	-0.156900	0.277512	-0.011530	-0.008558	All My Life	Run the Jewels 2
pulling the pin	-0.008376	0.005719	-0.024621	0.002433	-0.023317	0.002087	0.014389	-0.082255	-0.045679	-0.026238	pulling the pin	RTJ4
Early	-0.589731	0.055624	-0.212832	-0.231664	-0.395467	0.174600	0.486998	0.223936	0.023917	-0.005802	Early	Run the Jewels 2
A Christmas Fu***g Miracle	-0.025132	-0.050015	-0.013662	0.016576	0.008549	-0.018435	-0.024260	-0.040924	-0.005423	-0.073975	A Christmas Fu***g Miracle	Run the Jewels

Figure 62. DCM for Song-Level Terms

The resultant plots of P0 vs. P1, P1 vs. P2, and P2 vs. P3 are provided below.

In [636]: 1 vis_pcs_s(DCM_track, 0, 1)

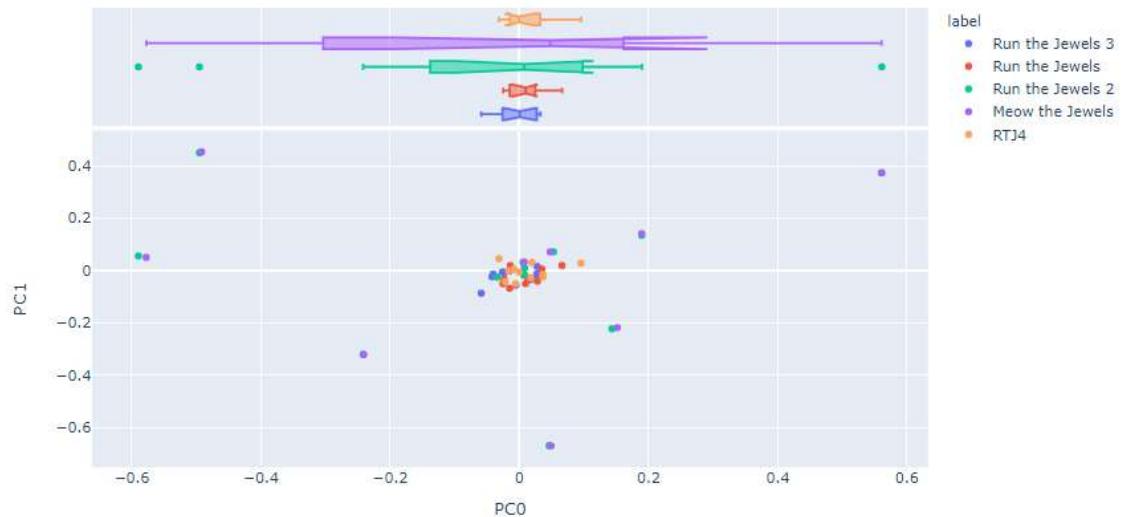


Figure 63. P0 vs. P1 by Songs



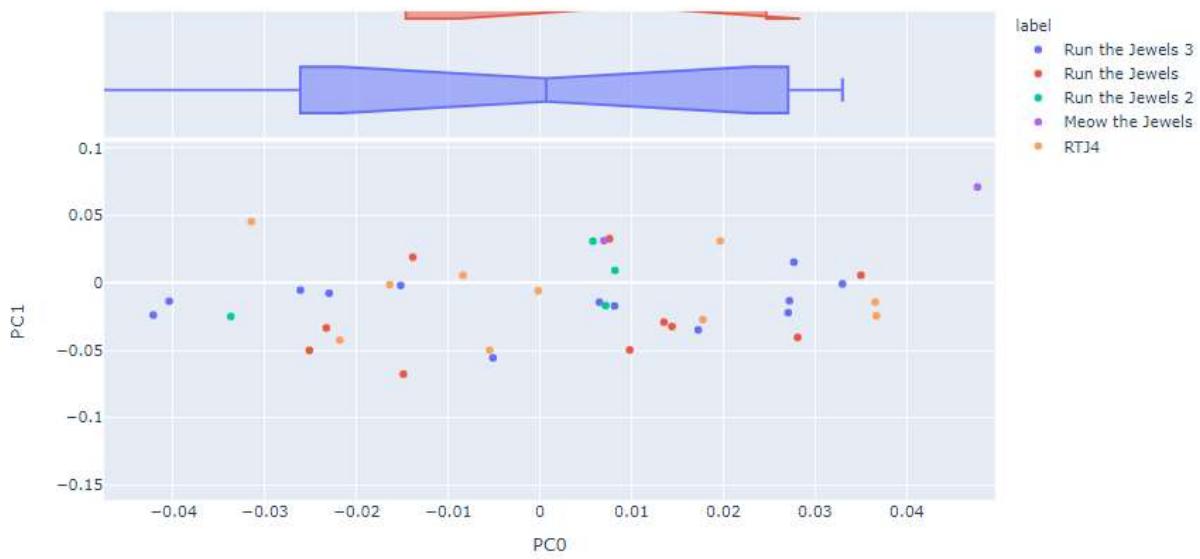


Figure 64. Central Detail of P0 vs. P1 by Songs

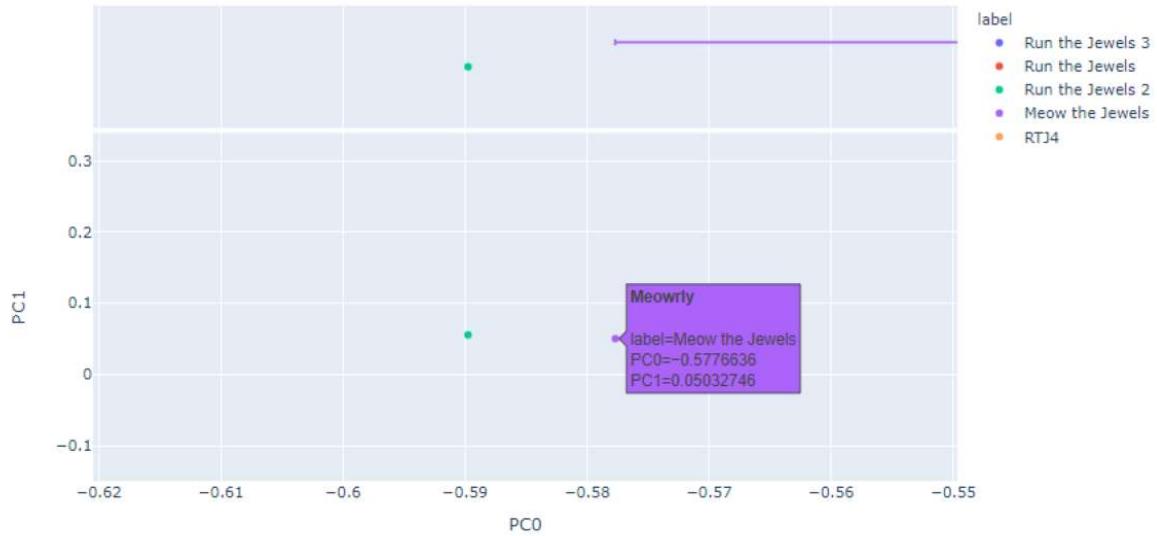


Figure 65. Other Detail of P0 vs. P1 by Songs



```
In [637]: 1 vis_pcs_s(DCM_track, 1, 2)
```

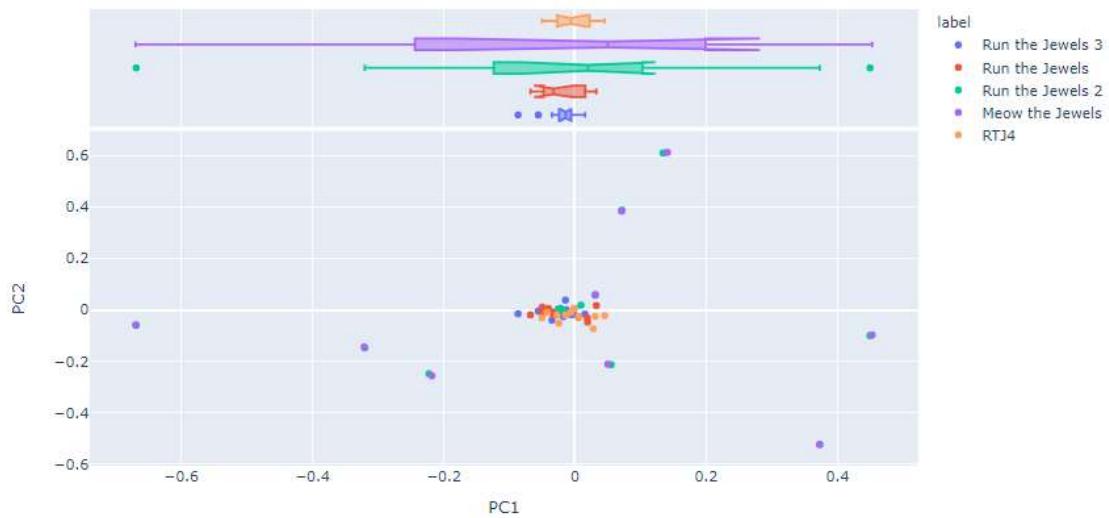


Figure 66. P_1 vs. P_2 by Songs

```
In [638]: 1 vis_pcs_s(DCM_track, 2, 3)
```

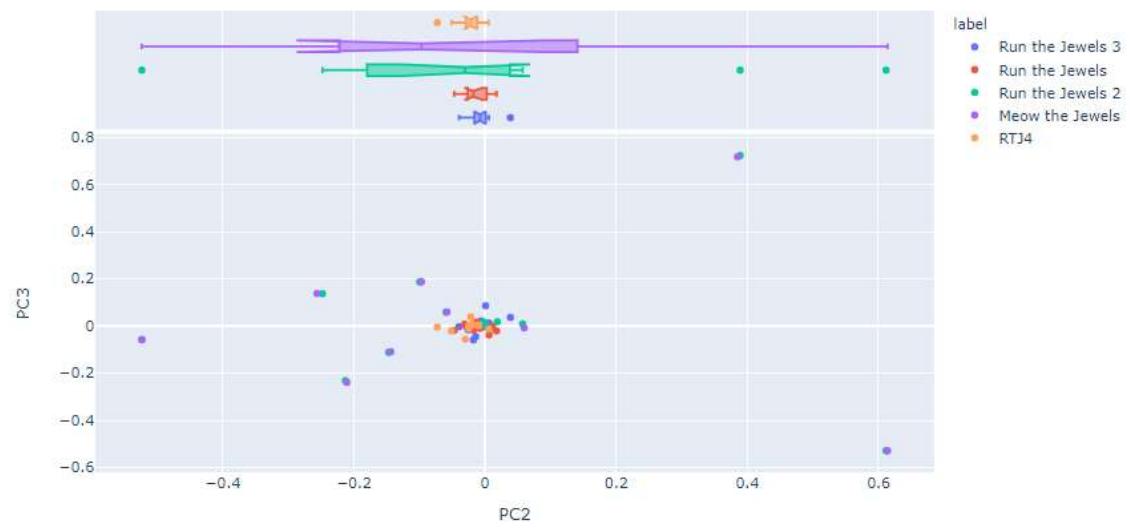


Figure 67. P_2 vs. P_3 by Songs



From PCA, it can be seen that *Meow the Jewels* on all three charts has the widest spread, followed by *Run the Jewels 2*, with the remaining three albums all reasonably tightly spread, centered near the origin coordinate. As observed, there is some slight spread around the origin in the range +/- 0.05 in both the P0 and P1 directions for the first of the shown PCA graphs.

It's also recognized that in all instances of a *Meow the Jewels* track, a corresponding *Run the Jewels 2* track was tightly clustered nearby for each of the PCA graphs. This result is informative, as it suggests that Run the Jewels is lyrically consistent across their discography. It is also observable that *Meow the Jewels* and *Run the Jewels 2* appear to have the widest ranges, implying that the songs on these albums are not only dissimilar from the bulk of the Run the Jewels corpus, but that these songs are also mutually dissimilar against tracks on the same such albums. That result appears to be supported by the previously-observed clustering seen in the dendrograms provided earlier in this document.

LDA

Latent Dirichlet Allocation calculations were performed on the Run the Jewels corpus, the details of which are provided in the corresponding Jupyter Notebook that accompanies this document. The OHCO model was defined as ["Album", "Track", "Lyric"] in the first set of LDA calculations, then generalized to just ["Album", "Track"]. Results for each are provided herein.

To perform these calculations, the previously-determined tokens (stripped of case and punctuation, with stop-words removed) were vectorized using Python's **Scikit-learn** package. This package was also used to perform LDA directly upon passing the relevant tokens from the corpus, grouped according to the respective OHCO definition as mentioned.

For the lyric-level OHCO model, the following is a sample of the resulting topic set as determined by the LDA calculations.

NOTE: Only words classified with a part of speech as "noun" were considered, however "nouns" in this context mean simply what nltk assigned as the part of speech. This is important, as some words that appear will not necessarily be something that would conventionally be considered a "noun", yet it was beyond the scope of this project to attempt to correct every such discovered instance. A future study would attempt to improve the part of speech classification assigned to the tokens of the corpus.

It's also noted that proper nouns were kept in the corpus for LDA as references to Killer Mike and El-P (or his birth name, Jaime) occur throughout the discography. Also referenced are relevant geographic locations, including Atlanta, Georgia and Brooklyn, New York (corresponding to areas associated with Killer Mike and El-P, respectively). For these reasons while also recalling that the corpus is limited to only music by Run the Jewels, such categories of words were not omitted.



term_str	0	1	2	3	4	5	6	7	8	9
topic_id										
0	life	pick	crown	night	wild	man	peace	time	feelin	control
1	love	trash	ashes	straight	fu*k	arrogance	fact	york	violence	dummy
2	kill	mean	walk	way	bop	lean	watch	power	speak	store
3	fu*k	di*k	elip	eat	law	told	word	pimp	question	folks
4	future	bi**h	work	billi	momma	fear	ran	ways	pound	cop
5	day	di*k	mouth	he'l	sea	legs	right	soul	fight	half
6	brave	yankee	fund	spit	gettin	jaime	perfect	space	right	spot
7	raw	ho	da*n	sh*t	catch	blood	tv	mic	lane	ni**a
8	smoke	room	mad	fu*k	marijuana	blow	shop	horrors	truths	death
9	want	ones	worthy	wife	talk	pop	riches	rags	fit	fu*k
10	cl*t	mouth	day	want	crooked	suck	check	lies	line	di*k
11	die	girl	bi**h	thing	pray	rare	demise	a*s	brain	people
12	beat	son	top	team	tag	sh*t	jewel	runners	half	bit
13	boy	kids	fu*k	man	sight	shoot	hear	style	try	smoke
14	move	want	water	land	begins	use	stops	pair	drugs	earth
15	mama	fire	told	dope	mind	lights	crack	yes	right	dance
16	everything	trip	pay	cash	jewels	baby	murder	knee	mom	runnin
17	look	masters	slave	posin	dollar	lion	face	master	world	food
18	call	woke	smoke	stop	beat	air	land	man	stuck	paid
19	promise	think	year	sh**s	chain	act	face	rules	lit	walk
20	gold	rtj	wish	smoke	crowd	heart	crew	screamin	neck	man
21	everybody	pistol	bi**h	fu*k	doin	fist	stay	throw	place	nothing
22	walk	hit	fool	bi**h	hoes	bruh	wear	window	room	gold
23	love	meant	heard	hate	ghosts	friend	child	light	locks	boy
24	speak	album	dog	cat	click	voodoo	thrill	holders	panty	pansy
25	time	man	fu*k	reefer	hope	pill	gun	dope	pow	knife
26	trust	sound	oodles	queen	list	dumb	step	matter	fact	reason
27	garden	truth	eyes	jewels	grenade	times	charlatans	marijuana	prize	couple
28	witness	sh*t	right	suck	coffin	chillin	life	mo*****r	pain	number
29	rise	gun	face	girl	stare	grin	greet	town	friends	paul
30	right	need	pain	stop	touch	jungle	brothers	sorry	preacher	books
31	walkin	snow	men	bunches	help	hold	today	bi**h	life	mike
32	ni**s	mike	anybody	talkin	hearts	fill	tryna	car	middle	step
33	cold	go***n	highway	mo*****r	fu*kboy	sh*t	gun	dollar	stay	evidence
34	man	talk	words	beat	right	home	freedom	side	bi**h	backflip
35	punch	sh*t	doin	mike	chance	everybodys	ask	bout	word	waves
36	sh*t	bi**h	order	rapper	pop	bar	spot	spit	baby	case
37	hope	boys	house	fame	smart	round	stay	cheri	job	killer
38	win	lie	kill	steal	cheat	devil	kid	head	heart	whiskey
39	jewels	fast	a*s	way	shorty	fake	tell	he'l	tongue	hate
40	sh*t	fu*k	baby	bi***s	people	boo	tin	rap	minds	rappers
41	meow	feel	man	think	livin	home	light	tell	ni**a	jesus
42	way	bi**h	kush	fu*k	past	pass	luck	world	smoke	rob
43	think	thieves	afraid	drop	dark	hang	days	trouble	god	tell
44	cry	respect	fu*k	angel	dust	insane	anchor	driven	men	guys
45	fu**n	tell	fu*k	talk	show	tusslin	class	baby	ni**a	crew
46	living	right	mama	mo*****s	guess	villain	left	god	bring	bang
47	god	left	bumaye	tell	play	fu*k	michael	swear	baby	truth
48	money	ride	mine	crazy	fat	a*s	talk	fact	cops	bit
49	guy	forget	sh*t	bi**h	born	hope	carry	years	mike	vroom

Figure 68. Top 50 Topic Sets for Lyric-Level Model



With weights assigned to these topic sets (of which there were 50 in total), the following plot was produced, showing the ranking of the topics by the summed weight values:

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x2216276a4e0>



Figure 69. Weighted Topics for Lyric-Level Model

Assigning these topics to the five albums yields the following, wherein the top topic sets per album are:

Run the Jewels: sh*t bi**h order rapper pop bar spot spit baby case

Run the Jewels 2: jewels fast a*s way shorty fake tell he*I tongue hate

Run the Jewels 3: garden truth eyes jewels grenade times charlatans marijuana prize couple

RTJ4: look masters slave posin dollar lion face master world food

Meow the Jewels: jewels fast a*s way shorty fake tell he*I tongue hate



Out[26]:

Album_num	1	2	3	4	5	topterms
topic_id						
36	0.032950	0.020180	0.028519	0.024888	0.022564	sh*t bi**h order rapper pop bar spot spit baby case
5	0.029687	0.030226	0.019551	0.020393	0.027907	day di*k mouth he*I sea legs right soul fight half
34	0.027417	0.017467	0.021425	0.024384	0.019102	man talk words beat right home freedom side bi**h backflip
28	0.026598	0.014889	0.015217	0.014867	0.011367	witness sh*t right suck coffin chillin life mo*****r pain number
27	0.026552	0.011143	0.037963	0.028103	0.011490	garden truth eyes jewels grenade times charlatans marijuana prize couple
1	0.026028	0.023288	0.017687	0.020485	0.020998	love trash ashes straight fu*k arrogance fact york violence dummy
42	0.025996	0.031348	0.018430	0.019966	0.029976	way bi**h kush fu*k past pass luck world smoke rob
25	0.025309	0.026459	0.016254	0.015311	0.030539	time man fu*k reefer hope pill gun dope pow knife
11	0.024410	0.018862	0.027225	0.020263	0.018000	die girl bi**h thing pray rare demise a*s brain people
3	0.024386	0.018803	0.019367	0.019407	0.018726	fu*k di*k elp eat law told word pimp question folks
40	0.023863	0.011273	0.016349	0.023624	0.011756	sh*t fu*k baby bi***s people boo tin rap minds rappers
39	0.023699	0.034210	0.013730	0.019210	0.038967	jewels fast a*s way shortly fake tell he*I tongue hate
43	0.023404	0.015319	0.022792	0.021874	0.015541	think thieves afraid drop dark hang days trouble god tell
31	0.022806	0.017333	0.013950	0.021342	0.018653	walkin snow men bunches help hold today bi**h life mike
7	0.022771	0.014837	0.015665	0.015083	0.013632	raw ho da*n sh*t catch blood tv mic lane ni**a
8	0.022617	0.018983	0.015252	0.015216	0.018238	smoke room mad fu*k marijuana blow shop horrors truths death
21	0.021781	0.025800	0.026647	0.020298	0.026329	everybody pistol bi**h fu*k doin fist stay throw place nothing
35	0.021330	0.021825	0.015307	0.019508	0.021816	punch sh*t doin mike chance everybodys ask bout word waves
16	0.021230	0.019443	0.016412	0.016433	0.018666	everything trip pay cash jewels baby murder knee mom runnin
13	0.020547	0.020478	0.019098	0.021242	0.022381	boy kids fu*k man sight shoot hear style try smoke
44	0.020515	0.024465	0.017297	0.017565	0.017020	cry respect fu*k angel dust insane anchor driven men guys
10	0.020464	0.023506	0.015625	0.016092	0.028981	cl**t mouth day want crooked suck check lies line di*k
30	0.020167	0.020856	0.022478	0.019558	0.023027	right need pain stop touch jungle brothers sorry preacher books
6	0.020006	0.018218	0.011284	0.022489	0.023426	brave yankee fund spit gettin jaime perfect space right spot
33	0.019489	0.022323	0.015683	0.023762	0.020942	cold go****n highway mo*****r fu*kboy sh*t gun dollar stay evidence
0	0.019326	0.031255	0.023842	0.021940	0.018083	life pick crown night wild man peace time feelin control
19	0.019275	0.019976	0.017640	0.024060	0.015783	promise think year sh**s chain act face rules lit walk
4	0.019231	0.020467	0.015972	0.013349	0.024492	future bi**h work billi momma fear ran ways pound cop
12	0.019018	0.018020	0.014586	0.022417	0.015726	beat son top team tag sh*t jewel runners half bit
45	0.018889	0.022207	0.019270	0.019037	0.024768	fu***n tell fu*k talk show fusslin class baby ni**a crew
14	0.018149	0.012339	0.019917	0.021557	0.013141	move want water land begins use stops pair drugs earth
49	0.017831	0.011167	0.018372	0.017292	0.008921	guy forget sh*t bi**s born hope carry years mike vroom
47	0.017670	0.016804	0.021677	0.021305	0.018996	god left bumaye tell play fu*k michael swear baby truth
2	0.017287	0.011909	0.032856	0.020225	0.011285	kill mean walk way bop lean watch power speak store
17	0.016900	0.022335	0.021541	0.040976	0.022591	look masters slave posin dollar lion face master world food
9	0.016635	0.020053	0.016683	0.016239	0.017488	want ones worthy wife talk pop riches rags fit fu*k
38	0.016540	0.028339	0.016923	0.016178	0.025015	win lie kill steal cheat devil kid head heart whiskey
24	0.016280	0.020649	0.017114	0.014119	0.027359	speak album dog cat click voodoo thrill holders panty pansy
41	0.016277	0.018043	0.021528	0.021257	0.025693	meow feel man think livin home light tell ni**a jesus
15	0.015751	0.015591	0.025884	0.027048	0.014663	mama fire told dope mind lights crack yes right dance
20	0.015595	0.019621	0.037924	0.020417	0.017946	gold rj wish smoke crowd heart crew screamin neck man
23	0.015584	0.016665	0.016430	0.027631	0.018613	love meant heard hate ghosts friend child light locks boy
32	0.015067	0.019901	0.014739	0.013883	0.022186	ni***s mike anybody talkin hearts fill tryna car middle step
46	0.014833	0.020934	0.024659	0.018524	0.018340	living right mama mo*****s guess villain left god bring bang
22	0.014121	0.022774	0.020221	0.017288	0.020608	walk hit fool bi**h hoes bruh wear window room gold
37	0.013968	0.022079	0.022766	0.016900	0.016634	hope boys house fame smart round stay cheri job killer
48	0.013924	0.016706	0.025652	0.018436	0.015343	money ride mine crazy fat a*s talk fact cops bit
26	0.013204	0.017910	0.017171	0.014115	0.021039	trust sound oodles queen list dumb step matter fact reason
29	0.012834	0.018008	0.016047	0.014430	0.020943	rise gun face girl stare grin greet town friends paul
18	0.011790	0.014711	0.021378	0.020015	0.014297	call woke smoke stop beat air land man stuck paid

Figure 70. Topic Set Score per Album, Lyric-Level Model



The similarities in the albums by topic can be observed not only in the above table but also by way of a scatter plot included in the Jupyter Notebook. Here, the topic scores for *Run the Jewels* is plotted against the corresponding scores for *Run the Jewels 3*. The result shows that the most *Run the Jewels*-ian topics correspond to topic set 36 (sh*t bi**h order rapper pop bar spot spit baby case), while the most *Run the Jewels 3*-ian topics correspond to topic set 27 (garden truth eyes jewels grenade times charlatans marijuana prize couple); however, the next closest set of topics characterizing *Run the Jewels 3* can be observed to be set 20 (gold rtj wish smoke crowd heart crew screamin neck man).

```
In [97]: px.scatter(ALBUMS_cens.reset_index(), 1, 3, hover_name='topterms', text='topic_id')\n      .update_traces(mode='text')
```

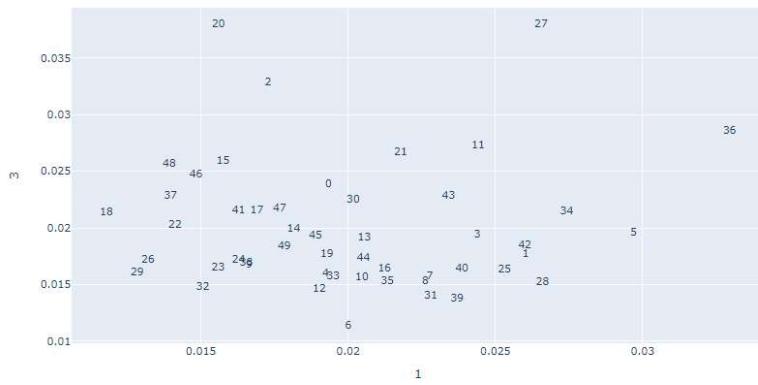


Figure 71. Run the Jewels vs. Run the Jewels 3, Topic Scatterplot, Lyric-Level

Throughout this document, it has been emphasized that *Meow the Jewels* was a remix of *Run the Jewels* 2; as such, one would expect a fairly linear relationship between the two albums with respect to the topic scores. Indeed, this can be observed with the follow plot, observing that topic 39 (jewels fast a*s way shorty fake tell he*I tongue hate) is the most relevant to each album.

```
In [98]: px.scatter(ALBUMS_cens.reset_index(), 2, 5, hover_name='topterms', text='topic_id')\n      .update_traces(mode='text')
```

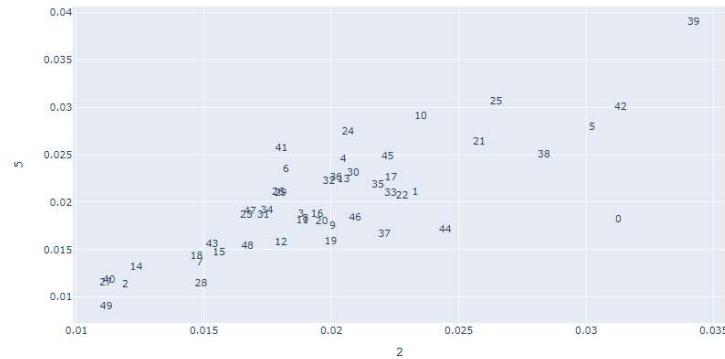


Figure 72. Run the Jewels 2 vs. Meow the Jewels Topic Scatterplot, Lyric-Level



This exercise is repeated but taking the token groupings at the song level instead of the lyric level. Otherwise, the same LDA calculation methods were applied, with the following results observed. Note that the number of topics calculated was reduced to 20 from 50.

term_str	0	1	2	3	4	5	6	7	8	9
topic_id										
0	wild	sh*t	minds	love	bring	born	told	mike	head	lord
1	beat	ni**a	feet	merciless	atlanta	mom	top	guys	town	piece
2	love	day	mouth	bi**h	fu*k	sh*t	man	way	di*k	meow
3	kill	masters	mike	war	world	bi**h	devil	talk	man	sh*t
4	right	need	catch	fat	fu*k	love	chill	smoke	sh*t	work
5	left	forget	truth	guess	love	mama	life	hope	day	friend
6	brave	yankee	couple	people	hope	pray	need	man	right	rtj
7	win	doin	kill	lie	man	steal	cheat	sh*t	everybody	bi**h
8	sh*t	punch	highway	left	beat	note	call	look	class	thrill
9	sh*t	crown	pick	bi**h	money	pistol	tell	think	fist	everybody
10	fast	jewels	fu*k	look	fu***n	bruh	order	rip	kill	book
11	die	garden	jewels	sh*t	eyes	tell	face	smoke	right	fu*k
12	fu*k	sh*t	life	baby	way	man	kill	cry	smoke	wish
13	charlatans	drop	alley	addict	die	brother	runner	matter	crack	mo*****n
14	cold	go***n	walkin	snow	mo*****r	respect	sh*t	locks	jesus	line
15	mama	right	fire	living	fu*k	words	talk	beat	lights	sh*t
16	zulu	forehead	foul	fought	fortyfour	fortunes	fortune	forthback	fort	fork
17	look	slave	fu*k	masters	dollar	money	sh*t	posin	told	smoke
18	gold	bi**h	walk	sh*t	legs	sea	right	girl	heart	love
19	jewels	thieves	rtj	sh*t	air	crowd	today	fu*k	he*l	fu***n

Figure 73. Top 20 Topic Sets for Song-Level Model

The corresponding distribution of weighted sums for these topics appears per the following:

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x2216533eb8>

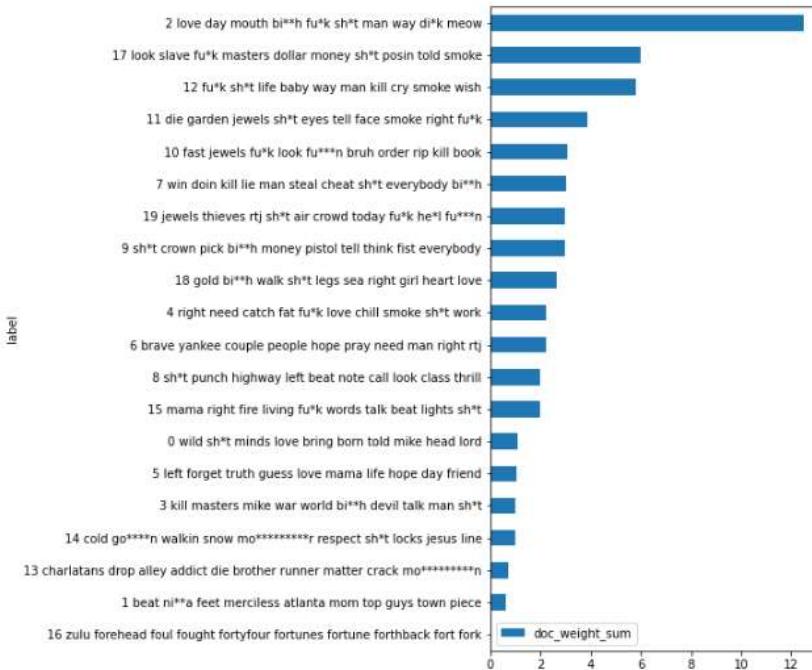


Figure 74. Weighted Topics for Song-Level Model



The top terms per album is given by the following table, where the derived values are:

Run the Jewels: die garden jewels sh*t eyes tell face smoke right fu*k

Run the Jewels 2: love day mouth bi**h fu*k sh*t man way di*k meow

Run the Jewels 3: jewels thieves rtj sh*t air crowd today fu*k he*I fu***n

RTJ4: love day mouth bi**h fu*k sh*t man way di*k meow

Meow the Jewels: love day mouth bi**h fu*k sh*t man way di*k meow

Album_num	1	2	3	4	5	topterms
topic_id						
11	0.168284	0.000255	0.142462	0.000247	0.000251	die garden jewels sh*t eyes tell face smoke right fu*k
4	0.108485	0.000255	0.071038	0.004802	0.000251	right need catch fat fu*k love chill smoke sh*t work
7	0.090790	0.083218	0.000259	0.004345	0.110803	win doin kill lie man steal cheat sh*t everybody bi**h
19	0.090695	0.000255	0.142468	0.000247	0.000251	jewels thieves rtj sh*t air crowd today fu*k he*I fu***n
0	0.090654	0.000255	0.006309	0.000247	0.000251	wild sh*t minds love bring born told mike head lord
9	0.090599	0.083071	0.071390	0.000247	0.000251	sh*t crown pick bi**h money pistol tell think fist everybody
17	0.090535	0.084088	0.071241	0.181234	0.112040	look slave fu*k masters dollar money sh*t posin told smoke
18	0.088282	0.000255	0.120521	0.000247	0.000251	gold bi**h walk sh*t legs sea right girl heart love
2	0.082634	0.414728	0.071341	0.240478	0.331791	love day mouth bi**h fu*k sh*t man way di*k meow
1	0.056169	0.000255	0.000259	0.000247	0.000251	beat ni**a feet merciless atlanta mom top guys town piece
12	0.0155983	0.166061	0.016176	0.130736	0.221345	fu*k sh*t life baby way man kill cry smoke wish
6	0.013126	0.000255	0.071288	0.098485	0.000251	brave yankee couple people hope pray need man right rtj
10	0.009289	0.082268	0.000259	0.089560	0.109602	fast jewels fu*k look fu***n bruh order rip kill book
5	0.002785	0.000255	0.071227	0.000247	0.000251	left forget truth guess love mama life hope day friend
3	0.000280	0.000255	0.071506	0.000247	0.000251	kill masters mike war world bi**h devil talk man sh*t
15	0.000280	0.000255	0.071221	0.090858	0.000251	mama right fire living fu*k words talk beat lights sh*t
8	0.000280	0.083255	0.000259	0.000247	0.110910	sh*t punch highway left beat note call look class thrill
14	0.000280	0.000255	0.000259	0.090842	0.000251	cold go*****n walkin snow mo*****n respect sh*t locks jesus line
13	0.000280	0.000255	0.000259	0.066191	0.000251	charlatans drop alley addict die brother runner matter crack mo*****n
16	0.000280	0.000255	0.000259	0.000247	0.000251	zulu forehead foul fought fortyfour fortunes fortune forthback fort fork

Figure 75. Topic Set Score per Album, Song-Level Model

Typical scatterplots akin to the ones provided above are included here for reference:

```
In [99]: px.scatter(ALBUMS_cens_2.reset_index(), 1, 3, hover_name='topterms', text='topic_id')\
    .update_traces(mode='text')
```

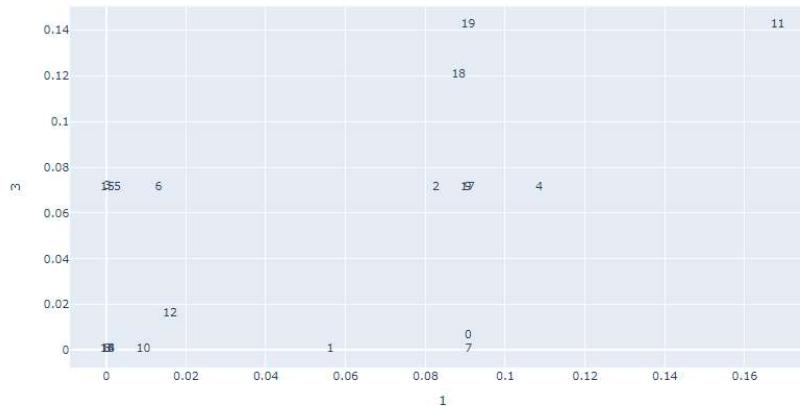


Figure 76. Run the Jewels vs. Run the Jewels 3 Topic Scatterplot, Song-Level



```
In [100]: px.scatter(ALBUMS_cens_2.reset_index(), 2, 5, hover_name='topterms', text='topic_id')\
    .update_traces(mode='text')
```

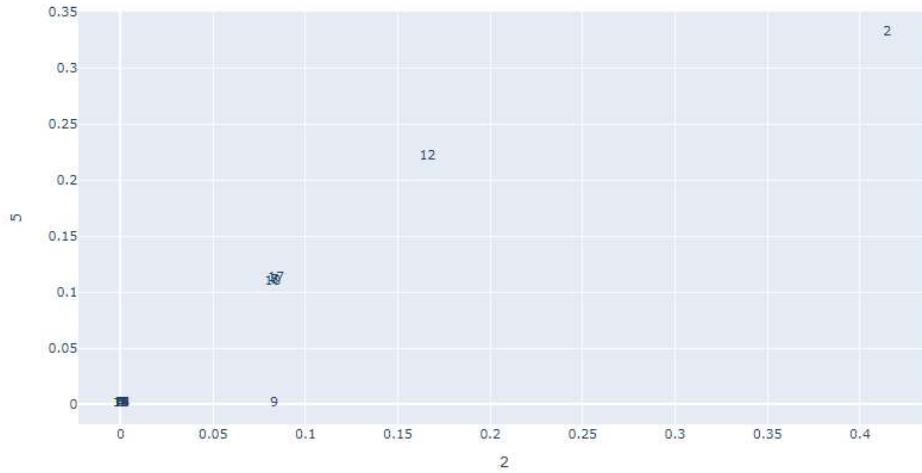


Figure 77. Run the Jewels 2 vs. Meow the Jewels *Topic Scatterplot, Song-Level*

```
In [102]: px.scatter(ALBUMS_cens_2.reset_index(), 2, 4, hover_name='topterms', text='topic_id')\
    .update_traces(mode='text')
```

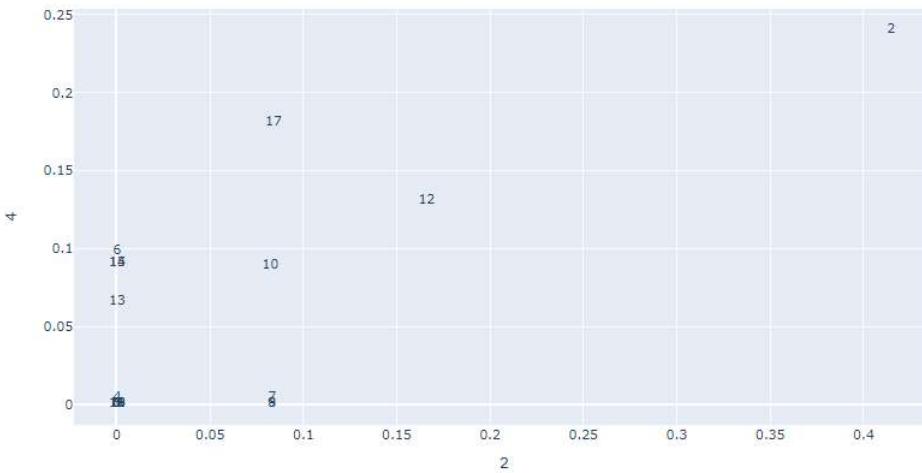


Figure 78. Run the Jewels 2 vs. RTJ4 *Topic Scatterplot, Song-Level*

A comparison of the two OHCO results are summarized below. It is observed that there is very little overlap between the topic lists that were determined through the calculation methods employed to represent each of the respective albums. At best, four of the five albums shared a



single word in common between the two lists, while *RTJ4* has zero words in common between the two topic lists.

One interesting observation is that the term “charlatans” appeared in the lyric-level topic list for *Run the Jewels 3*. That word is exclusively used in the song “yankee and the brave (ep. 4)” off the *RTJ4* album, so further investigation would be needed to understand how it was scored as sufficiently close to the other terms so as to be categorized as a *Run the Jewels 3* topic.

Album	Lyric-Level OHCO Topic List	Song-Level OHCO Topic List
Run the Jewels	sh*t bi**h order rapper pop bar spot spit baby case	die garden jewels sh*t eyes tell face smoke right fu*k
Run the Jewels 2	jewels fast a*s way shorty fake tell he*I tongue hate	love day mouth bi**h fu*k sh*t man way di*k meow
Run the Jewels 3	garden truth eyes jewels grenade times charlatans marijuana prize couple	jewels thieves rtj sh*t air crowd today fu*k he*I fu***n
RTJ4	look masters slave posin dollar lion face master world food	love day mouth bi**h fu*k sh*t man way di*k meow
Meow the Jewels	jewels fast a*s way shorty fake tell he*I tongue hate	love day mouth bi**h fu*k sh*t man way di*k meow

Figure 79. Comparison of Lyric-Level and Song-Level Topic Lists by Album

WORD2VEC

The final exercises performed on the Run the Jewels lyric corpus are **Word2Vec** and **Doc2Vec**, methods to find related words within the corpus by vectorizing the tokenized words. Given the size of the corpus (4,842 unique tokens, of which 191 are categorized as stop-words), with a corpus spanning approximately 33,000 words total, it should not be expected to find much success with these methods, which historically are more reliable on language sets orders of magnitude greater than what was central to this study.

Also impacting both Word2Vec and Doc2Vec is the unusual structure of the lyrics. Rather than conventional text arranged in sentences, paragraphs, and so forth, song lyrics in general are more likely to be arranged as short sequences of words or sentence fragments that do not necessarily obey conventional grammatical structures. Repetition of words or word sequences is also a phenomenon that can be observed in the Run the Jewels corpus (for example, see the previous “kill your masters” case). Taken together, these factors are among the complications which limit the effectiveness of the Word2Vec and Doc2Vec methods.

Nevertheless, such calculations were applied and can be consulted in the supplemental Jupyter Notebook. Notably, tuning each method proved to be a challenge given the wide array of potentially adjustable parameters or hyperparameters. After trial and error along with research done online, the following parameters were used.

```
model_w2v = word2vec.Word2Vec(corpus,
    alpha=0.025, min_alpha=0.025, workers=3, min_count=2,
    window=10, size = AAA, iter=400, sample=0.001, negative=5, hs=0)
```

Figure 80. Word2Vec Parameter Selection

Here, size was taken as 30, 60, 120, and 256. Correspondingly, the following results represent a comparison of what varying the size parameter had on the model and subsequent TSNE plots.

NOTE: Doc2Vec calculations and results can be found in the supplemental Jupyter Notebook.



For size = 30:

```
In [126]: 1 model_w2v.wv.most_similar('mommy') In [128]: 1 model_w2v.wv.most_similar('love') In [130]: 1 model_w2v.wv.most_similar('lamb')
Out[126]: [('stepdad', 0.5886853933334351), ('pro', 0.5815557241439819), ('rich', 0.5520722720011902), ('johnny', 0.5399714112281799), ('victory', 0.5305155515670776), ('figure', 0.514700051223755), ('fill', 0.4833418369293213), ('times', 0.4809528589248657), ('daddy', 0.476850688457489), ('uncle', 0.4759614169597626)]
In [127]: 1 model_w2v.wv.most_similar('gun') In [129]: 1 model_w2v.wv.most_similar('weed') In [131]: 1 model_w2v.wv.most_similar('police')
Out[127]: [('badge', 0.594937801361084), ('thumbs', 0.5620632767677307), ('thrown', 0.53519606590271), ('molotov', 0.5061237812042236), ('flows', 0.499906063079834), ('flare', 0.4975280463695526), ('egyptians', 0.49656495451927185), ('chickens', 0.49505969882011414), ('somewhere', 0.48651212453842163), ('mother', 0.48288899660110474)]
Out[128]: [('stay', 0.5622631311416626), ('walls', 0.535248875617981), ('hurts', 0.5050921440124512), ('had', 0.4934596173576355), ('bang', 0.49177461862564087), ('still', 0.48953473567962646), ('baby', 0.48887333273887634), ('first', 0.48461171984672546), ('missin', 0.4775206744670868), ('pardon', 0.47092944383621216)
Out[130]: [('venison', 0.9784896373748779), ('hennessy', 0.9658875465393066), ('innocence', 0.851526141166687), ('kitchenette', 0.7524877190589905), ('michael', 0.6452502608299255), ('drank', 0.6381266713142395), ('ate', 0.6282105445861816), ('seizing', 0.580148458480835), ('puma', 0.5772007703781128), ('pleas', 0.55009553793525696)]
Out[131]: [('unite', 0.9580095410346985), ('jail', 0.8201912641525269), ('cos', 0.6741766335580017), ('choose', 0.621841311454773), ('win', 0.5763965845108032), ('pull', 0.5692388415336609), ('mission', 0.5478569269180298), ('none', 0.5293303728103638), ('come', 0.512002825736995), ('peered', 0.49936771392822266)]
```

Figure 81. Most Similar Set, Size = 30

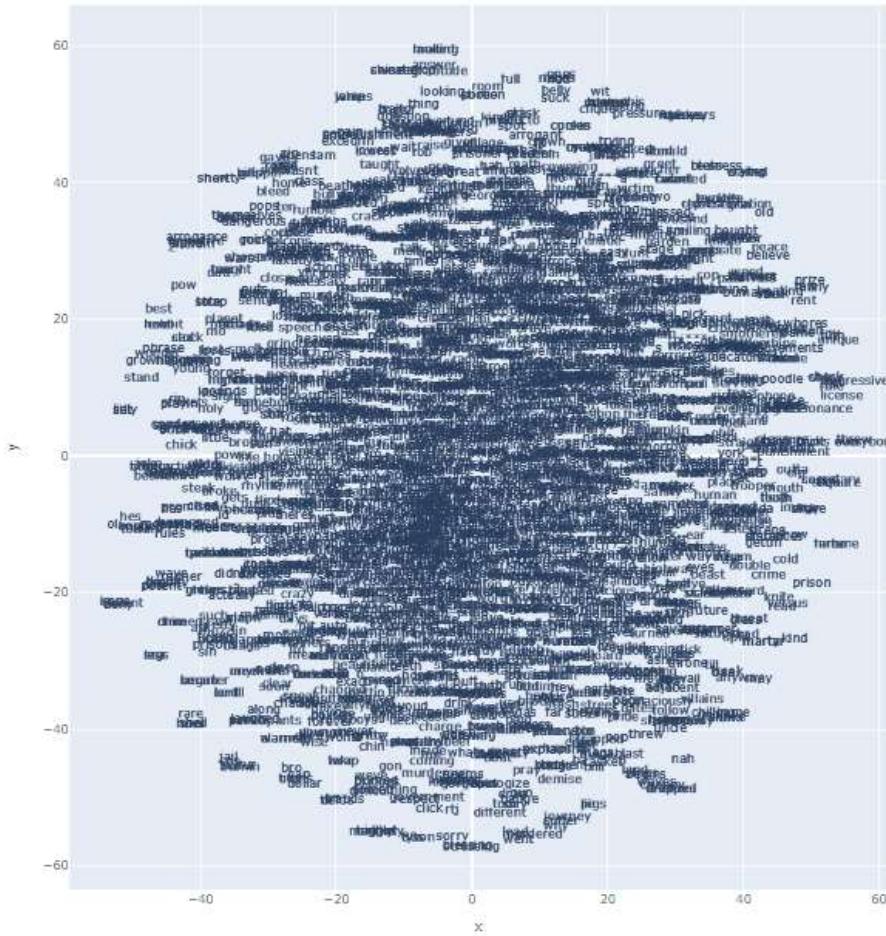


Figure 82. TSNE Plot, Size = 30



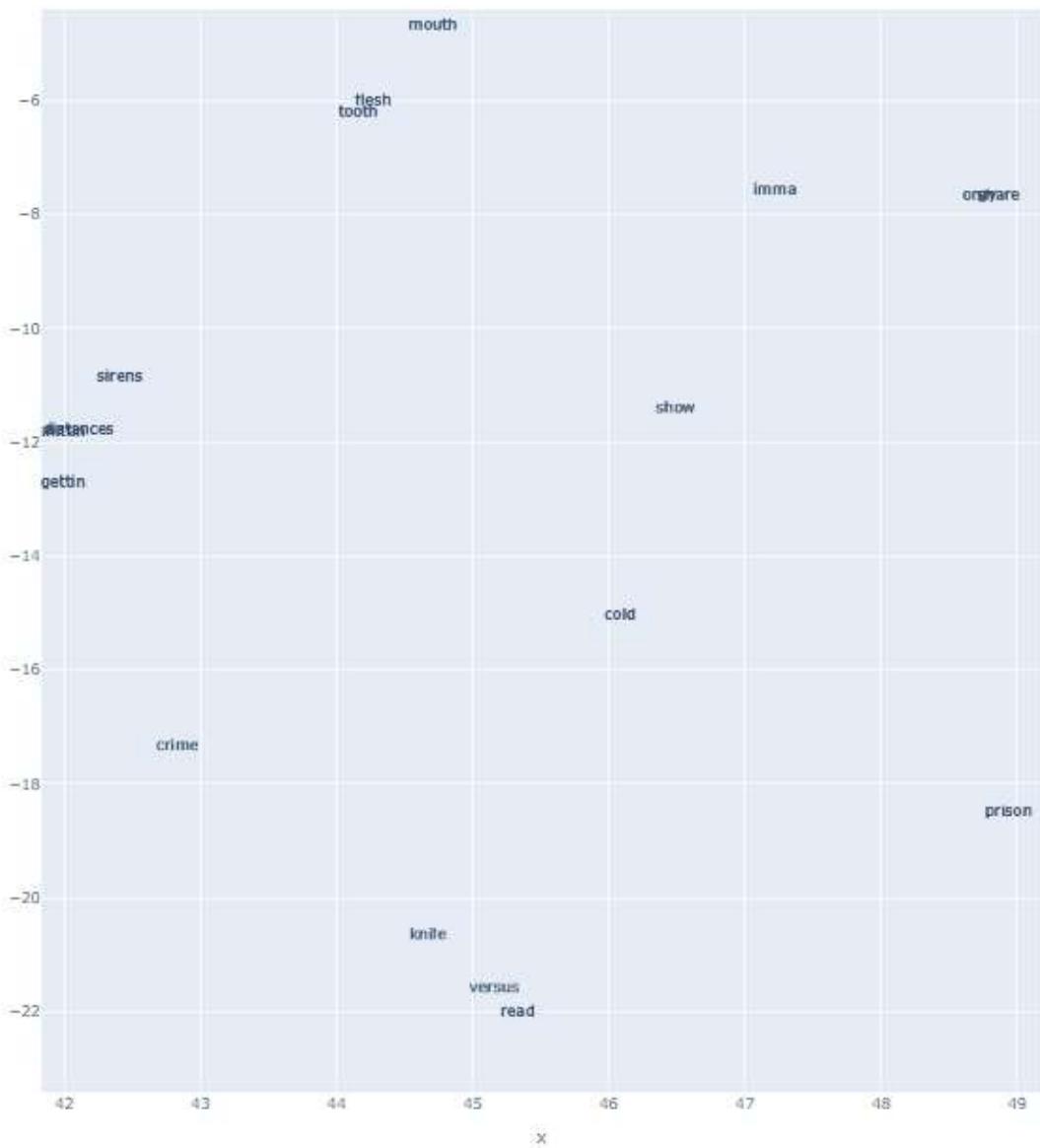


Figure 83. TSNE Plot Detail, Size = 30

At size = 30, the most similar words list appears plausible for the top results per word. “Mommy” and “stepdad” do have familial connections, for example. “Gun” and “badge” as well seem to go together when considering the association with law enforcement. The TSNE plot appears densest at its center, where separation of like-word hasn’t developed, however some clustering has begun to appear near the edges. In the plot detail above, it can be observed that words like “crime”, “prison”, “knife”, and “versus” are in near-proximity. Likewise, anatomical terms like “mouth”, “tooth”, and “flesh” have formed a small cluster. With a larger corpus, this size parameter appears promising.



For size = 60:

```
In [141]: 1 model_w2v.wv.most_similar('mommy') In [143]: 1 model_w2v.wv.most_similar('love') In [145]: 1 model_w2v.wv.most_similar('lamb')
2
Out[141]: [('stepdad', 0.5933989882469177), ('uncle', 0.4649159618271454), ('product', 0.44378453493118286), ('johnny', 0.43532079458236694), ('grip', 0.43168730608357086), ('rescue', 0.4145405883557434), ('pro', 0.4144258499145508), ('fortune', 0.4023471474647522), ('chance', 0.39048489928245544), ('dollar', 0.3828037679195404)]
```

```
In [142]: 1 model_w2v.wv.most_similar('gun') In [144]: 1 model_w2v.wv.most_similar('weed') In [146]: 1 model_w2v.wv.most_similar('police')
2
Out[142]: [('note', 0.4799622595310211), ('road', 0.43824678659439087), ('scarier', 0.42385220527648926), ('after', 0.42133239553604126), ('sharp', 0.41894054412841797), ('dad', 0.402624249458313), ('ballerina', 0.3991867898947144), ('meet', 0.389358252286911), ('speed', 0.3847653567790985), ('knife', 0.38455730676651)]
```

```
Out[143]: [('nothing', 0.40780192613601685), ('baby', 0.40010765194892883), ('lotta', 0.39024412631988525), ('be', 0.38870757818222046), ('dump', 0.3881239891052246), ('walls', 0.3740663528442383), ('heart', 0.37074825167655945), ('astonishin', 0.3561950922512329), ('*s', 0.3535394072532654), ('pay', 0.3447851538651824)]
```

```
Out[144]: [('farmers', 0.7597537040710449), ('harvest', 0.6258499622344971), ('starless', 0.5645645260810852), ('bleed', 0.4281286597251892), ('shyne', 0.41607469320929724), ('blind', 0.399428129196167), ('single', 0.398187483174896), ('hit', 0.3965588940849304), ('lock', 0.3902689814567566), ('moon', 0.3868521749973297)]
```

```
Out[145]: [('venison', 0.9684486389160156), ('hennessy', 0.952221155166626), ('innocence', 0.7666413187980652), ('kitchenette', 0.6972540616989136), ('drank', 0.5661804088000946), ('ate', 0.557312084602356), ('seizing', 0.5119719505310059), ('pump', 0.49386001089510193), ('pleas', 0.4806492328643799), ('facing', 0.47093677520751953)]
```

Figure 84. Most Similar Set, Size = 60

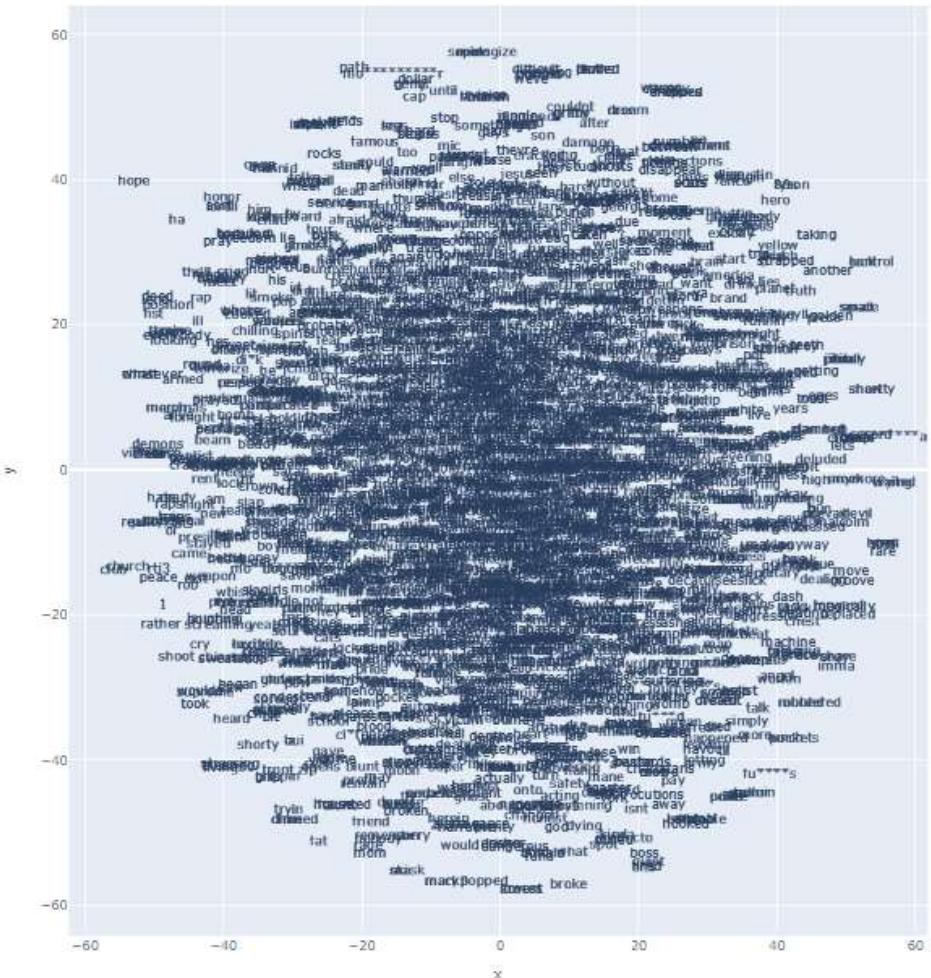


Figure 85. TSNE Plot, Size = 60



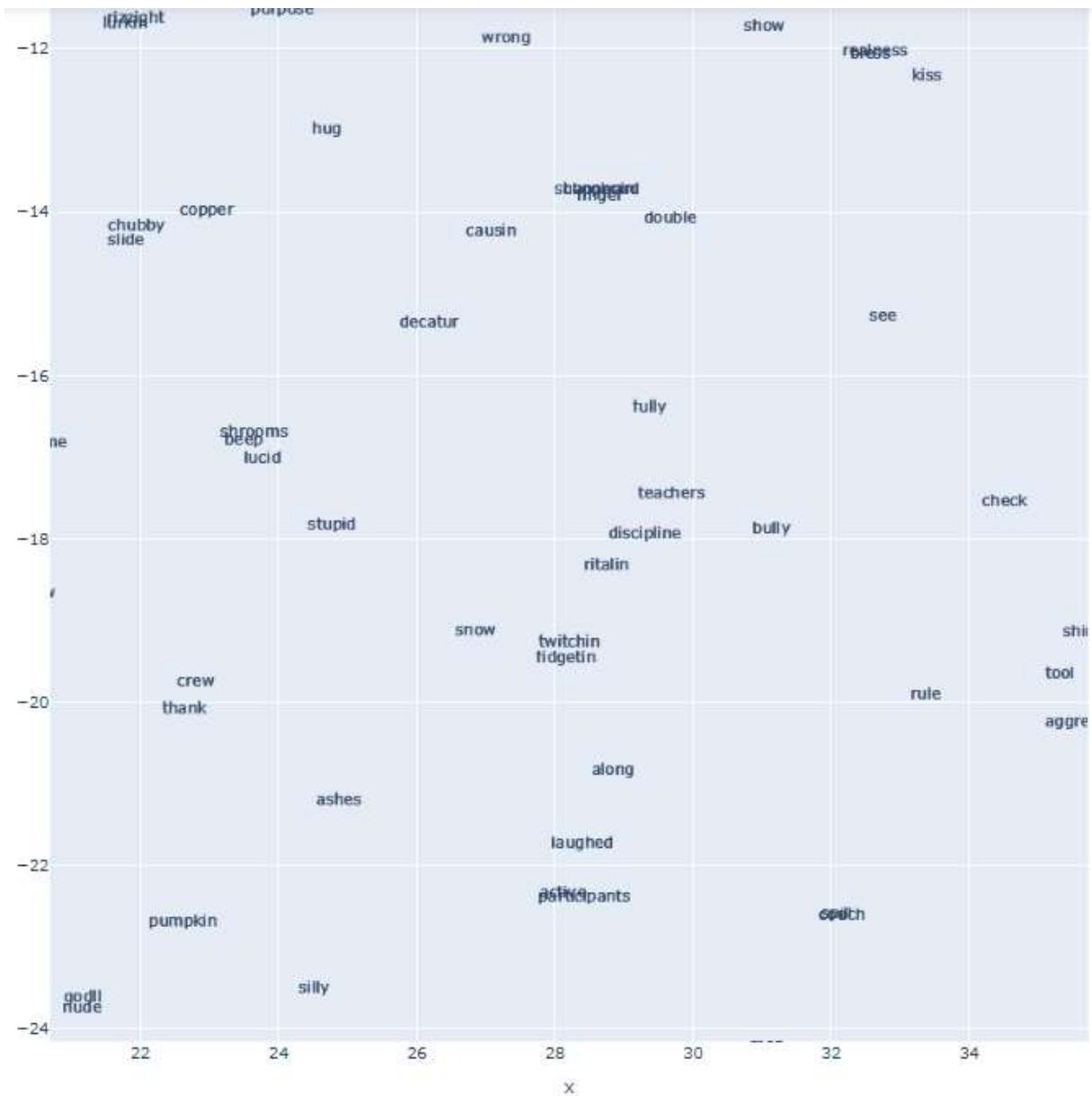


Figure 86. TSNE Plot Detail, Size = 60

At size = 60, the most similar words list appears mostly plausible for the top results per word. “Mommy” and “stepdad” do have familial connections, for example, and moving down that the next word encountered is “uncle”. However, “badge” is not clustered with “gun”; instead the word “note” became the most similar next term which is somewhat nonsensical. It is somewhat surprising that “jail” is the second closest word to “police”, behind the word “unite”. Like the size = 30 case, the TSNE plot appears densest at its center, where separation of like-word hasn’t developed, however some clustering has begun to appear along the edges. In the plot detail above, it can be observed that words like “teachers”, “bully”, “discipline”, and “bully” are in near-proximity. However, many of the other words appear somewhat random, implying that a larger corpus may be needed to develop a more effective Word2Vec model.

For size = 120:

```
In [153]: 1 model_w2v.wv.most_similar('mommy')
2
Out[153]: [('stepdad', 0.5228266716003418),
('rescue', 0.41732707619667053),
('uncle', 0.41275477409362793),
('johnny', 0.38712793588638306),
('daddy', 0.3578757345676422),
('dollar', 0.3493652939796446),
('wheres', 0.34153568744659424),
('pro', 0.3269645571708679),
('cents', 0.32323259618282318),
('single', 0.32058340311050415)]
```

```
In [154]: 1 model_w2v.wv.most_similar('gun')
2
3
Out[154]: [('knife', 0.46675926446914673),
('ruin', 0.3324507176876068),
('apologize', 0.33214253187179565),
('drag', 0.324108749628067),
('badge', 0.2923866808414459),
('shriek', 0.2915780544281006),
('pound', 0.28359222412109375),
('professor', 0.2803645730018616),
('note', 0.2713831067085266),
('ski', 0.27052101429881775)]
```

```
In [155]: 1 model_w2v.wv.most_similar('love')
2
3
Out[155]: [('pay', 0.29629337787628174),
('dump', 0.2950832979583274),
('astonishin', 0.28914088010787964),
('honey', 0.26934799551963806),
('spit', 0.26712986626895667),
('apart', 0.26695653796195984),
('threshold', 0.2585100034332275),
('lotta', 0.255927413366251),
('livid', 0.25437456369400024),
('water', 0.25425654649734497)]
```

```
In [156]: 1 model_w2v.wv.most_similar('weed')
2
3
Out[156]: [('farmers', 0.76734585751297),
('harvest', 0.6134743696490723),
('starrless', 0.5249470472335615),
('sweatpants', 0.4134628746318817),
('funerals', 0.3990219235420227),
('lunch', 0.38860711455245154),
('cop', 0.3658129705657959),
('ja', 0.3620165268448334),
('bleed', 0.3431547284126282),
('blind', 0.33992806077000348)]
```

```
In [157]: 1 model_w2v.wv.most_similar('lamb')
2
3
Out[157]: [('venison', 0.9613302946090698),
('hennessy', 0.9452065229415894),
('innocence', 0.7695332765579224),
('kitchenette', 0.6732368469288281),
('ate', 0.6169108152389526),
('drank', 0.5452289581298828),
('pleas', 0.4169512093067169),
('beef', 0.39763352274894714),
('seizing', 0.3790878653526306),
('facing', 0.37885862588882446)]
```

```
In [158]: 1 model_w2v.wv.most_similar('police')
2
3
Out[158]: [('unite', 0.9518290162086487),
('jail', 0.6963355541229248),
('blooders', 0.6187407970428467),
('cos', 0.573973536491394),
('crippers', 0.4937674403190613),
('brothers', 0.44398534297943115),
('denied', 0.36211462176245544),
('thuggers', 0.33118953043899536),
('co*k', 0.31879866123199463),
('over', 0.3178580350494385)]
```

Figure 87. Most Similar Set, Size = 120

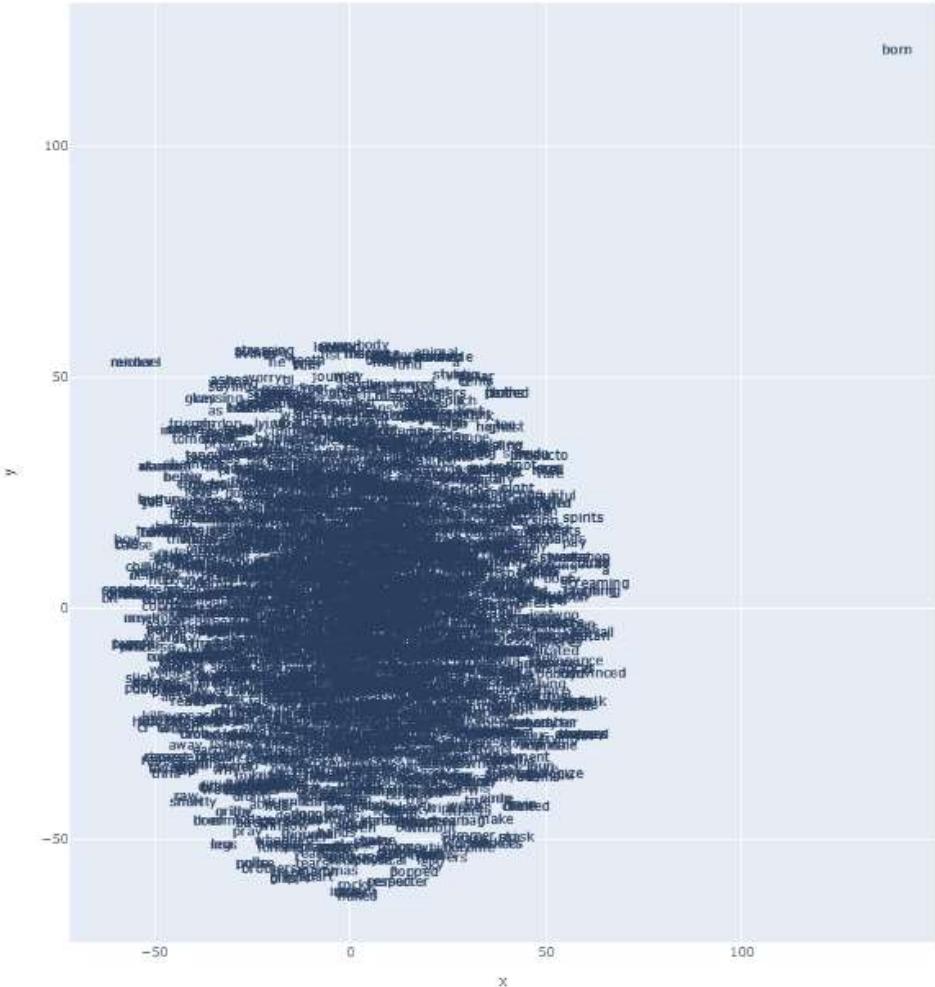


Figure 88. TSNE Plot, Size = 120



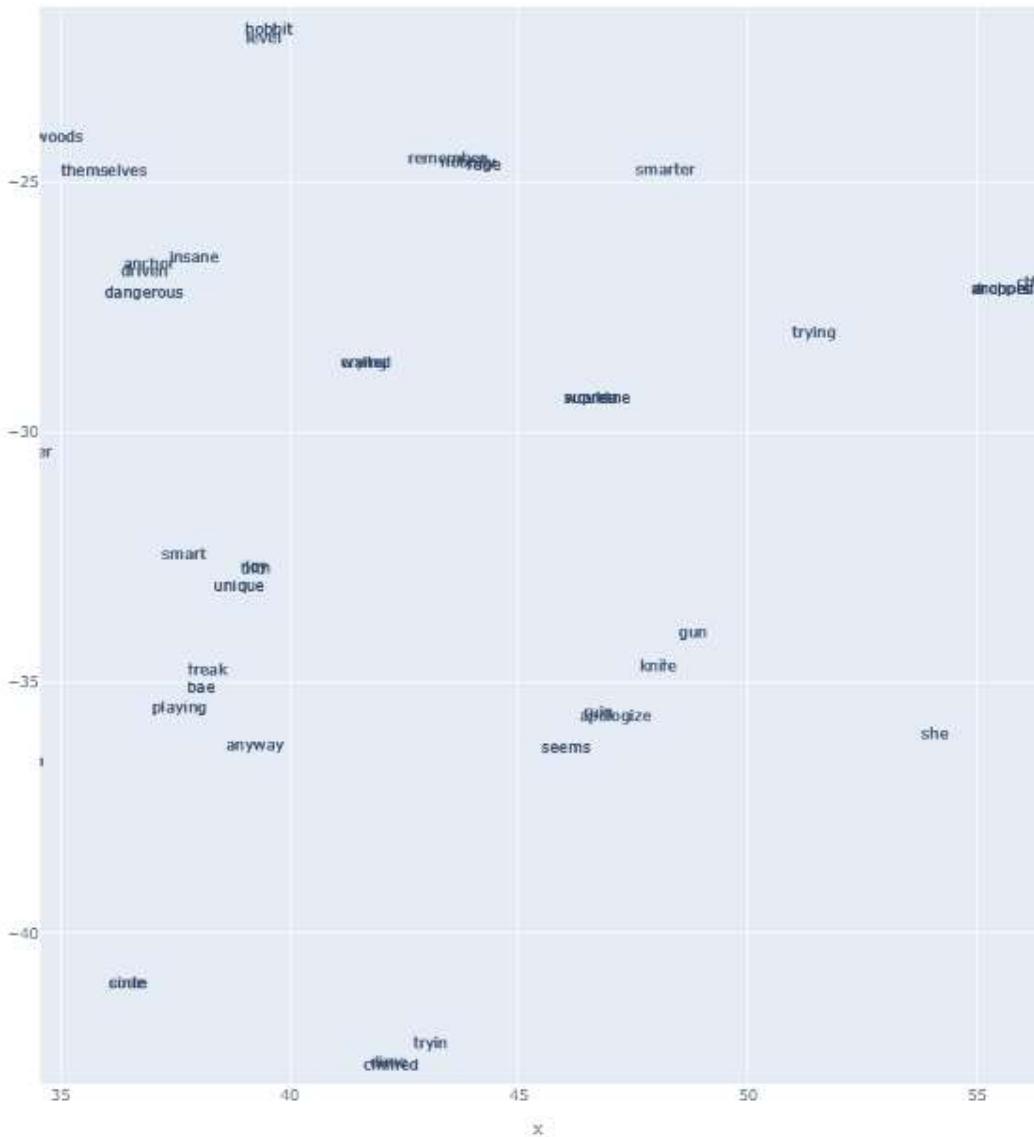


Figure 89. TSNE Plot Detail, Size = 120

At size = 120, the most similar words list appears mostly plausible for the top results per word. “Mommy” and “stepdad” once again appear together, for example. Another weapon term, “knife” appears as the most similar to the word “gun”. As has been observed in this and the prior two models, “venison” appears as the next closest term to “lamb”, which makes sense as an edible meat product. Again, “jail” is the second closest word to “police”, behind the word “unite”. The size = 120 TSNE plot appears very dense, with less separation along its perimeter; there are also escaped terms like “born”. In the plot detail above, it can be observed that words “gun” and “knife” are in near-proximity, supporting the most-similar word list. Also observed is the clustering of terms like “insane” and “dangerous”.

However, given the density of the primary cluster for this corpus, size = 120 is probably too large.



For size = 256:

```
In [165]: 1 model_w2v.wv.most_similar('mommy')
Out[165]: [('stepdad', 0.5098909139633179),
 ('uncle', 0.419393181008423),
 ('rescue', 0.4084412245750427),
 ('johnny', 0.3930738568305869),
 ('wheres', 0.368769122383974),
 ('dollar', 0.322119414682087),
 ('cents', 0.320900643369385),
 ('product', 0.31177412509918213),
 ('single', 0.311015307903288),
 ('plan', 0.305476158573456)]
```

```
In [167]: 1 model_w2v.wv.most_similar('love')
Out[167]: [('bleed', 0.23184943199157715),
 ('him', 0.219655364781518),
 ('parents', 0.2140382105146866),
 ('livid', 0.209988665248773),
 ('water', 0.2089268416163085),
 ('dump', 0.20864298977034),
 ('pay', 0.2064499408006668),
 ('astonishin', 0.20428212991142273),
 ('lotta', 0.2039965391159637),
 ('honey', 0.2025773227148132)]
```

```
In [168]: 1 model_w2v.wv.most_similar('gun')
Out[168]: [('knife', 0.41269195079803467),
 ('apologize', 0.313580723924652),
 ('drag', 0.3098662156851593),
 ('note', 0.3059545159339905),
 ('run', 0.30384378271202905),
 ('shriek', 0.302570641040882),
 ('company', 0.278586564385669),
 ('badge', 0.2768773110848236),
 ('somethin', 0.2740913038445126),
 ('clouds', 0.2677767276763916)]
```

```
In [169]: 1 model_w2v.wv.most_similar('lamb')
Out[169]: [('venison', 0.9673151969909668),
 ('hennessy', 0.9519361853599548),
 ('innocence', 0.775259487414941),
 ('kitchenette', 0.6389948785165265),
 ('ate', 0.615019017167371),
 ('drank', 0.522568147392273),
 ('pleas', 0.41396987480821904),
 ('facing', 0.405863539241791),
 ('beef', 0.40400147438049316),
 ('case', 0.3865012526512146)]
```

```
In [170]: 1 model_w2v.wv.most_similar('police')
Out[170]: [('unite', 0.9600796699523926),
 ('jail', 0.725926756858257),
 ('blooders', 0.6166865364577026),
 ('cos', 0.6005105927290839),
 ('crippers', 0.465949141127014),
 ('brothers', 0.4296146621240845),
 ('boom', 0.3195308818598165),
 ('thuggers', 0.3045167326927165),
 ('win', 0.292457223985637),
 ('nudge', 0.28927338123321533)]
```

Figure 90. Most Similar Set, Size = 256

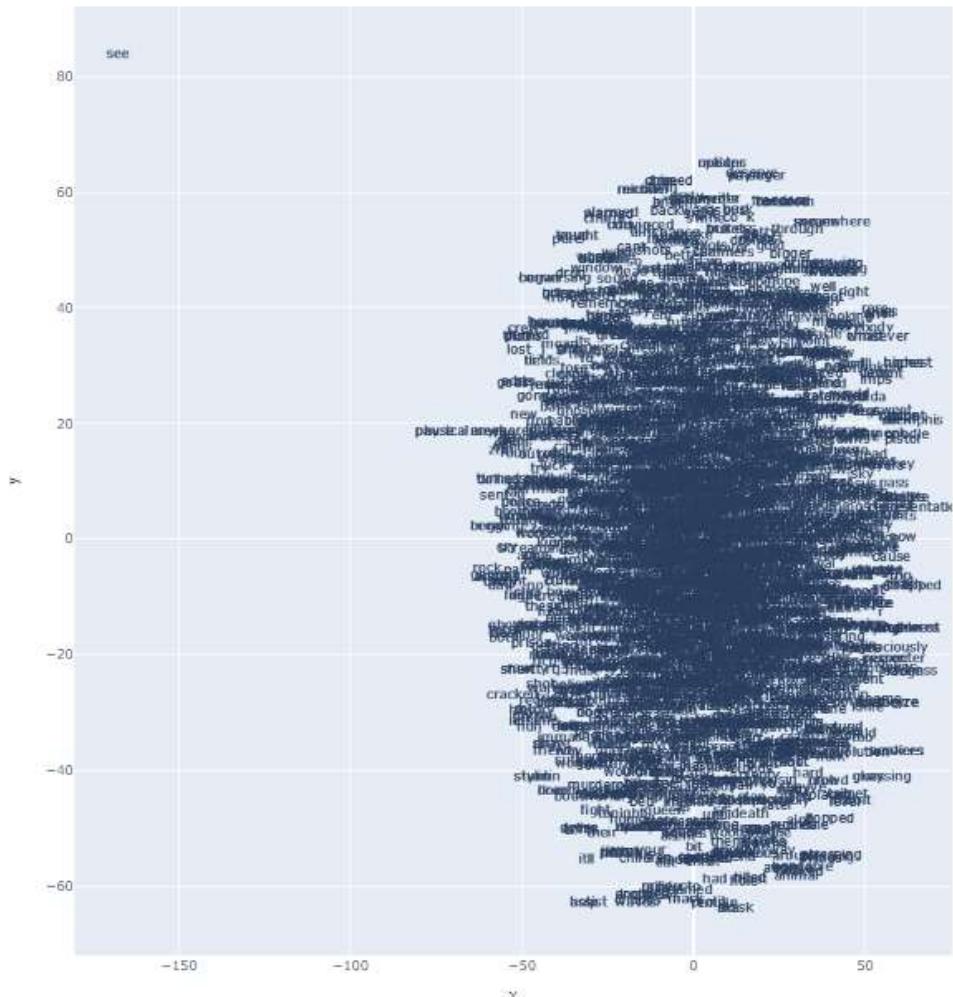


Figure 91. TSNE Plot, Size = 256



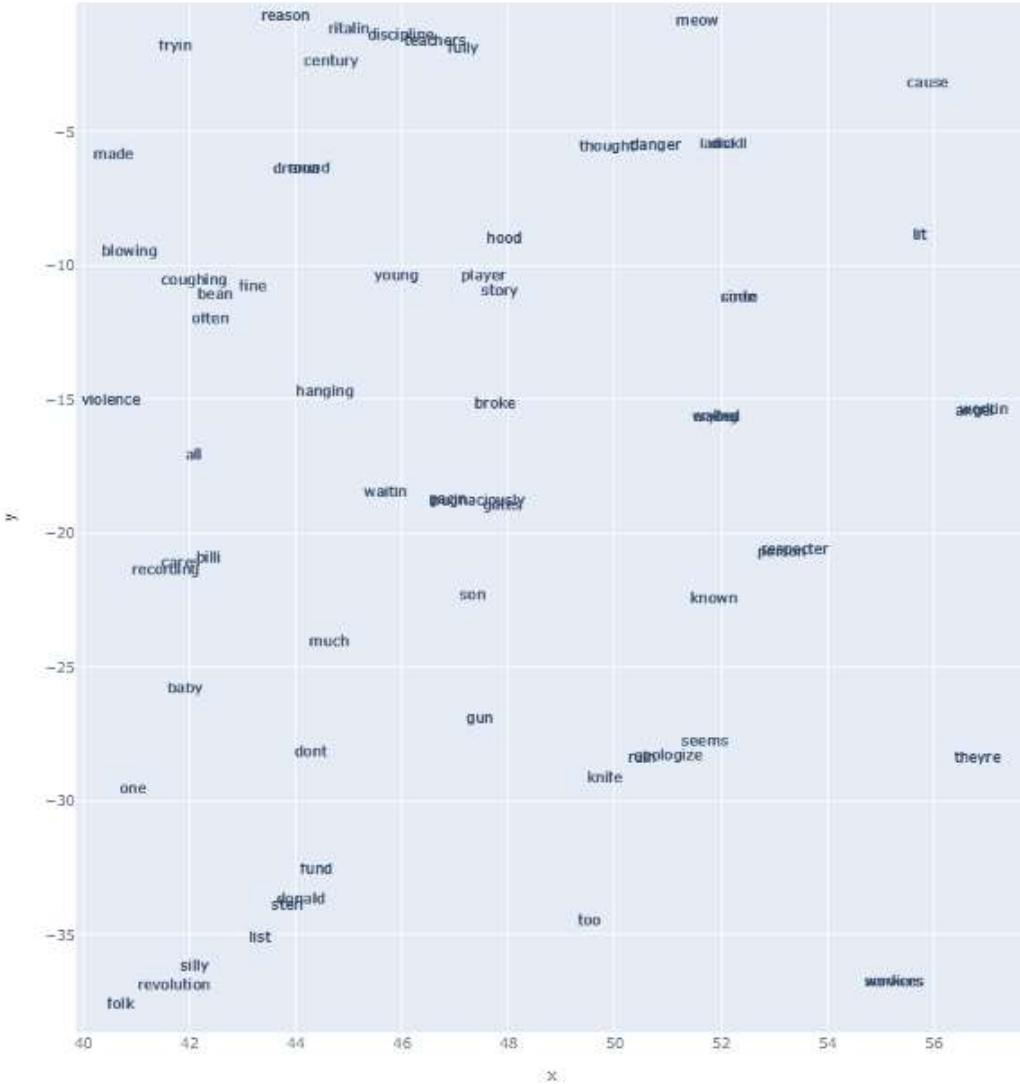


Figure 92. TSNE Plot Detail, Size = 256

At size = 256, the most similar words list appears mostly plausible for the top results per word. “Mommy” and “stepdad” yet again appear together, for example. Another weapon term, “knife” appears as the most similar to the word “gun”. As has been observed in this and the prior models, “venison” appears as the next closest term to “lamb”, which makes sense as an edible meat product. Interestingly, the next closest word to “love” is “bleed”, which maybe suggests the idea that love is painful. The size = 256 TSNE plot appears especially dense and oblong, with little separation along its perimeter; there are also escaped terms like “see”. In the plot detail above, it can be observed that words “gun” and “knife” are in near-proximity, supporting the most-similar word list. Also observed is the clustering of terms like “ritalin”, “teacher”, and “discipline”.

Like the size = 120 model, size = 256 is probably too large for this corpus to create meaningful clustering.



CONCLUSION

The East Coast rap duo Run the Jewels has proven to be a successful musical act since their debut nearly a decade ago. Lyrically different from more mainstream performers in the genre, Run the Jewels is known for challenging more political and social themes than conventional rap music that is perhaps more focused on consumerism and wealth. However, the band still maintains itself partly in the genre, with lyrical references to sexual themes and recreational narcotic usage; yet as observed in this study, words associated with violence like “kill” do have a prevalence in their discography.

Such topics and the word choices associated therewith heralded the linguistic negativity of the Run the Jewels songs, sentiment-wise. Not only was the entire corpus found to have a negative polarity, but according to the EmoLex sentiment library, only 9 of the 57 songs studied had a net-positive sentiment. As observed in the heat mapping of the songs by lyrics and in the individual sentiment plots for select songs, much of the group’s music begins relatively neutral before transitioning to negative language midway through, only to return to relatively neutral before finally ending with a heavily negative sentiment.

This phenomenon can be seen in songs like “No Come Down” which ends with the repeating of the phrase “I get so high, I close my eyes / Like I may die, die”, the song “Like, Cheat, Steal” which near its end concludes by repeating the phrase “Lie, cheat, steal, kill, win, win”, and “Paw Due Respect” that includes lyrics like “Can’t relate to your first world struggles”, “IED’s will leave bloody puddles”, “Case of malaria, sh*t got scarier”, and “Got left with a gun and a pitbull terrier”. While there is optimism in some of their songs, like in the track “2100”, much of the corpus leverages contrasting negative imagery and sentiment, perhaps reflective of the band’s frustrations with topics like racial inequality and injustice, as well as greed and corruption in politics and the corporate world.

This last point is a driving factor behind the repetition of the word “kill” in “A Report to the Shareholders / Kill Your Masters”. The so-called “masters” in the context of this song (and in the Run the Jewels mantra, in general) are representative of the societal structures constraining citizens reliant on those systems through policies and prices. Hence, calls to rise up against such institutions are evident in the Run the Jewels discography, even if the words used evoke radical, violent imagery at times.

Through the clustering and vectorization methods of this study, it could be concluded that *Run the Jewels 2* is a very similar album to *Meow the Jewels*, which is an unsurprising result. *Meow the Jewels* is a cat-themed remix of *Run the Jewels 2*, and would be expected to have significant similarity. It is also observed that *Run the Jewels 3* and *RTJ4* are mutually similar, in contrast to *Run the Jewels*, the band’s earlier self-titled debut album. This demonstrates some evolution from their earliest music to their most recent recordings, the latter two albums of which have achieved more mainstream success on the *Billboard* charts. However, from principal component analysis, it can be observed that all albums are still closely clustered, implying that while there have been some lyrical shifts in the band’s albums over time, their language and themes are still consistent with their earliest music.



It's noted that the discography selected for this study was limited, so training a natural language processing algorithm against the corpus had correspondingly limited success. Rather than typical bodies of work like novels, papers, articles, and so forth, song lyrics are much more fragmented and unconventional due to artifacts like the usage of repetition and fluid grammar. That is to say: training a machine to learn language solely by feeding it Run the Jewels lyrics is not a recipe for much success.

However, with additional bodies to include in the corpus such as future recordings, the machine learning model in this study would be expected to improve. Other opportunities for improvement could be to adjust the nltk assignment of parts of speech to tokens (which defaulted to setting many words as "nouns"), and to fine tune the parameters and hyperparameters used in methods like Word2Vec.

Yet this study was considered successful in demonstrating techniques for censorship and in finding sentiment based on the EmoLex library (in the supplementary Jupyter Notebook, the Bing Liu positive / negative word lists were also used with mostly similar results) and how such sentiment transitions through songs and through albums. Topic themes were also derived to characterize the albums within the discography, using words that appear plausible particularly given the generated word clouds. To build upon this study, other avenues may include incorporating the discographies of other music acts (perhaps those of similar geography, Atlanta-based acts such as Migos or Lil Yachty or New York-based acts such as the Wu-Tang Clan or Cardi B, obviously among many, many others).

Understanding what shapes popular music such as the songs of Run the Jewels not only can provide insights into the culture of our society, but may also help in the shaping of future music based on such past experiences. This may be especially relevant if Run the Jewels were to want to evolve their music through an awareness of their past tendencies, or if another artist wanted to emulate themselves on the Run the Jewels model. And while human experience and creativity is unparalleled, understanding the Run the Jewels model or equivalent can certainly help train algorithms to be leveraged for generating themed lyrics, rudimentary examples of which were given in this study.

