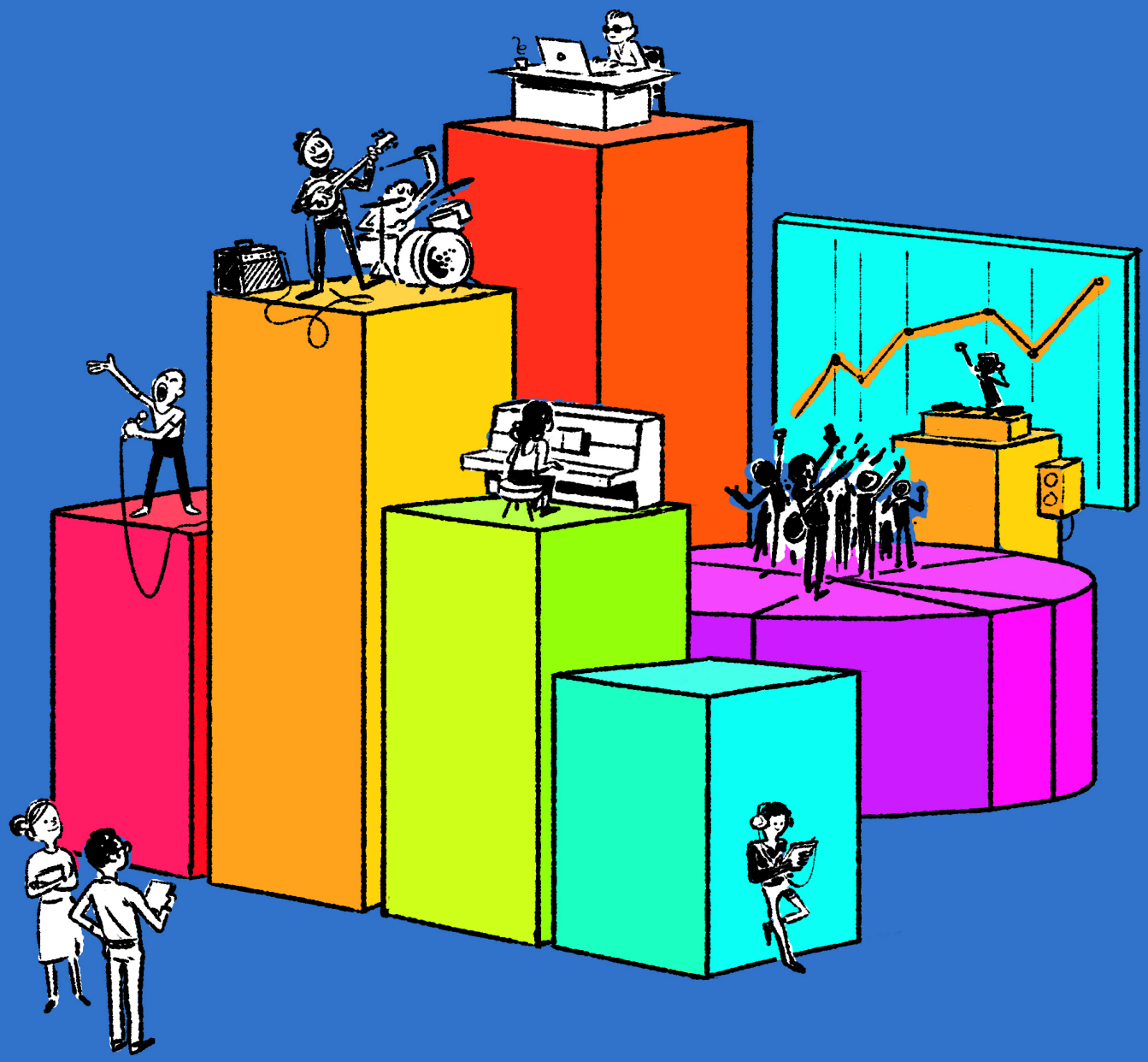
#MusicTech

Experimenting with Data Science and Recommender Systems



Alexandre Passant, Ph.D.

#MusicTech

Experimenting with Data Science and Recommender Systems

Alexandre Passant

This book is for sale at http://leanpub.com/musictech

This version was published on 2015-05-28

Tweet This Book!

Please help Alexandre Passant by spreading the word about this book on Twitter!

The suggested tweet for this book is:

I just bought the #MusicTech book by @apassant

The suggested hashtag for this book is #MusicTechBook.

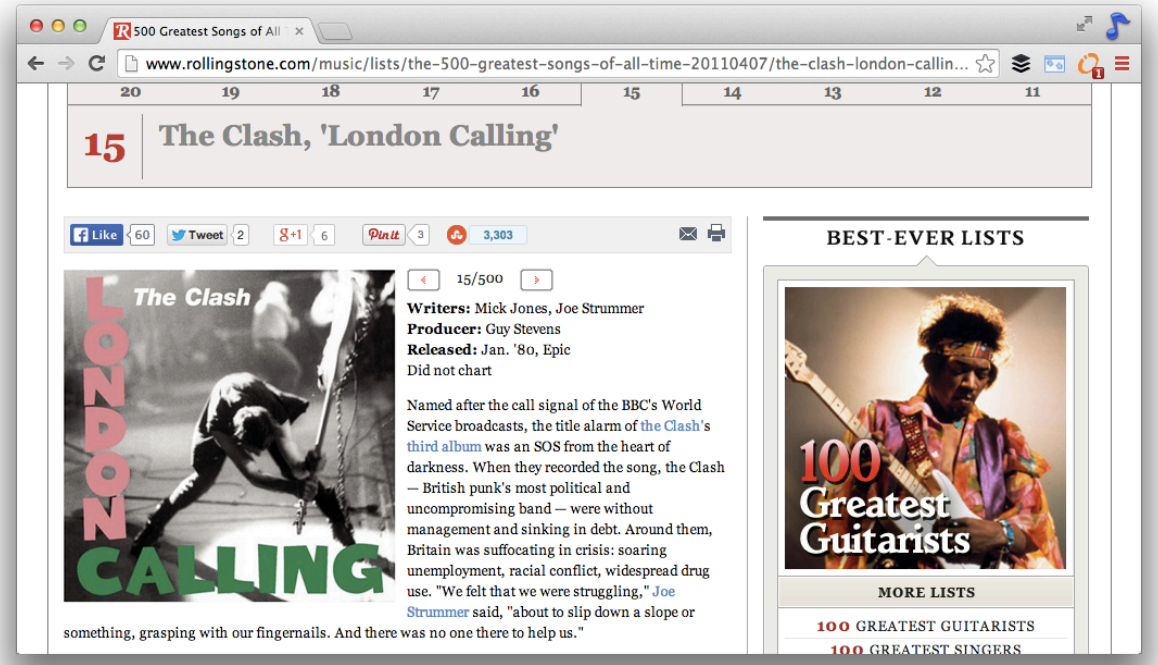
Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#MusicTechBook

Contents

Sex and drugs and Rock'n'roll: Analytics of the Rolling Stone's 500 greatest songs of all time

A few years ago, the Rolling Stone magazine published an update to its 500 Greatest Songs of All Time[1]. While the related Wikipedia entry[2] contains lots of interesting statistics (shortest and longest songs, popular decades, covers, etc.), I wondered how new insights could be gathered using external APIs. In this chapter, I will describe how I mined the songs lyrics from this top 500 to extract various patterns, before focusing on their tempo and loudness in the next chapter.



***London Calling*, one of 5 songs from The Clash in the Rolling Stone's 500 greatest songs of all time**

[1] http://www.rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407

[2] http://en.wikipedia.org/wiki/Rolling_Stone's_500_Greatest_Songs_of_All_Time

Come together

In order to run this first experiment, I used the following pipeline to prepare the data. If you want to skip the technicalities of the experiment, you can directly jump to the insights section):

1. First, extract titles, artists and reviews of all songs from the Rolling Stone website, using BeautifulSoup[3], starting from the 500th one, *Shop Around*[4], to the 1st one, *Like a Rolling Stone*[5];
2. Then, get lyrics of each songs via the Lyrics'n'Music API[6], powered by LyricFind[7];
3. Finally, run some Natural Language Processing tasks on the corpus using NLTK[8] to pre-process the data before analyzing it.

> ## Natural Language Processing
>
> Natural Language Processing](http://en.wikipedia.org/wiki/Natural_language_processing) (NLP) is a field of Computer Science – including concepts from Artificial Intelligence, Information Retrieval, and Linguistics – that focuses on processing, and eventually understanding, text in order to automate certain tasks.
>
> It has a wide range of implications, such as sentiment analysis, knowledge extraction, automatic translation, etc. A multitude of toolkits, including open-source ones, are available to run some of the core NLP tasks, such as the aforementioned Python's NTLK (Natural Language ToolKit).

The NLP pre-processing focused on:

1. Tokenizing, *i.e.* splitting lyrics into words;
2. Stemming, *i.e.* extracting the root of each word, so that "love", "loved" and "loving" all map to "love"; and
3. Extracting n-grams, *i.e.* finding sequences of n consecutive words.

NLTK offers different tokenizers, and I used `PunktWordTokenizer`, which gave better results than the default `word_tokenize`. As most lyrics are in English, it makes sense to use one already trained for it (as is Punkt). Stemming was powered by the Snowball algorithm[9]. Here is a quick overview of how they all work together.

[3] http://www.crummy.com/software/BeautifulSoup/

[4] http://www.rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/smokey-robinson-and-the-miracles-shop-around-20110526

[5] http://www.rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/bob-dylan-like-a-rolling-stone-20110516

[6] http://www.lyricsnmusic.com/api

[7] http://lyricfind.com

[8] http://www.nltk.org/

[9] http://snowball.tartarus.org/texts/introduction.html

```
from nltk.tokenize.punkt
import PunktWordTokenizer
from nltk.stem.snowball import SnowballStemmer

elvis = """
Here we go again
Asking where I've been
You can't see these tears are real
I'm crying (Yes I'm crying)
"""

sb = SnowballStemmer('english')
pk = PunktWordTokenizer()
print [sb.stem(w) for w in pk.word_tokenize(elvis)]
```

Which lead to:

```
['here', 'we', 'go', 'again', 'ask', 'where', 'i', "'ve", 'been', 'you', 'can', \
"'t", 'see', 'these', 'tear', 'are', 'real', 'i', "'m", 'cri', '(', 'ye', 'i', "\
'm", 'cri', ')']
```

You might notice a few glitches: "me" is stemmed to "m", and "crying" to "cri" and not to "cry" – as one could expect. Yet, "cried", "cry", "cries" all relate to the same root with Snowball, which is OK to group words together. However, no stemming algorithm worked perfectly. Snowball identified different roots for "love" and "lover", while the Lancaster algorithm[10] matched both to "lov", but fails for the previous cry example.

```
>>> from nltk.stem.snowball import SnowballStemmer
>>> from nltk.stem.lancaster import LancasterStemmer
>>>
>>> sb = SnowballStemmer('english')
>>> lc = LancasterStemmer()
>>>
>>> cry = ['cry', 'crying', 'cries', 'cried']
>>> [lc.stem(w) for w in cry]
['cry', 'cry', 'cri', 'cri']
>>> [sb.stem(w) for w in cry]
[u'cri', u'cri', u'cri', u'cri']
>>>
>>> love = ['love', 'loves', 'loving', 'loved', 'lover']
```

[10] http://www.comp.lancs.ac.uk/computing/research/stemming/

```
>>> [lc.stem(w) for w in love ]
['lov', 'lov', 'lov', 'lov', 'lov']
>>> [sb.stem(w) for w in love ]
[u'love', u'love', u'love', u'love', u'lover']
```

That being said, on the full corpus of 500 songs lyrics, the top 10 stemmed words were the same whatever the algorithm was (albeit with a different count and different syntaxes), so I have decided to focus on the Snowball extraction. But enough talking about NLP, let's focus on the analysis!

Baby love

So, what kind of lyrics can we expect from a rock'n'roll compilation like this one? Well, surprising or not, the most popular term in the top 500 is "love". It appears 1057 times in 219 songs (43.8%), and other popular terms include:

Word	# of times	# of songs (/500)
"love"	1057	219
"I'm"	1000	242
"oh"	847	180
"know"	779	271
"baby"	746	163
"got"	702	182
"yeah"	656	155

One could probably write a song with "Oh yeah baby I got you, yeah I'm in love with you, yeah!", and easily fits here (well, look at that opening line from the Wu Tang[11]). Sorting results by song appearances only, "like" is also among the top words, included in 194 tracks.

I wanna be anarchy

If we now look at the top 5 3-grams (*i.e.* groups of three words), we still have a general "you-and-me" feeling occurring in those songs:

3-gram	# of songs (/500)
"I want to"	38
"I don't know"	35
"I love you"	26
"You know I"	22
"You want to"	21

[11] http://rapgenius.com/Drake-wu-tang-forever-lyrics

Those 3-grams are then followed by other "(do not) want" combinations, and once again, most of the want-list is love-related. While The Beatles want to hold her hand[12], the Drifters prefer to know if she loved them[13], and Foreigner simply want to know what love is[14]. Yet, others, such as the Stooges, prefer to be your dog[15], while Otis Redding just want to be free[16].

There was no real pattern on the 4-grams and 5-grams, besides that Blondie[17], Jimmy Hendrix[18] and 7 others "don't know why", and that the B-52's[19], Bob Dylan[20] and Jay-Z[21] have something to do on "the other side of the road".

Hotel California

I was expecting a lot of tracks to fall into the sex, drugs and rock-n-roll stereotype. Yet, that was not really the case. Only 13 songs contain the word "sex", 5 "drug", and 4 "rock'n'roll"; no songs contain all three. Looking in more detail into the drug-theme, and querying Freebase to find a list of abused substances and their aliases, I found 7 references to "cocaine" and 4 for "heroin" – including some in the eponym song[22], while "grass" and "pot" appear a few times, even though it would require more analysis to see in which context they are used.

Of course, a simple word-based analysis like this one cannot capture the full meaning of a song. Thus, we miss classic anthems on the drug-related theme like the awesome *Comfortably numb*[23] by Pink Floyd, or *White Rabbit*[24] by Jefferson Airplane.

[12] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-beatles-i-want-to-hold-your-hand-20110517

[13] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-drifters-there-goes-my-baby-20110526

[14] //rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/foreigner-i-want-to-know-what-love-is-20110526

[15] http://www.rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-stooges-i-wanna-be-your-dog-20110526

[16] //rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/otis-redding-ive-been-loving-you-too-long-to-stop-now-20110526

[17] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/blondie-call-me-20110526

[18] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-jimi-hendrix-experience-purple-haze-20110517

[19] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-b-52s-love-shack-20110527

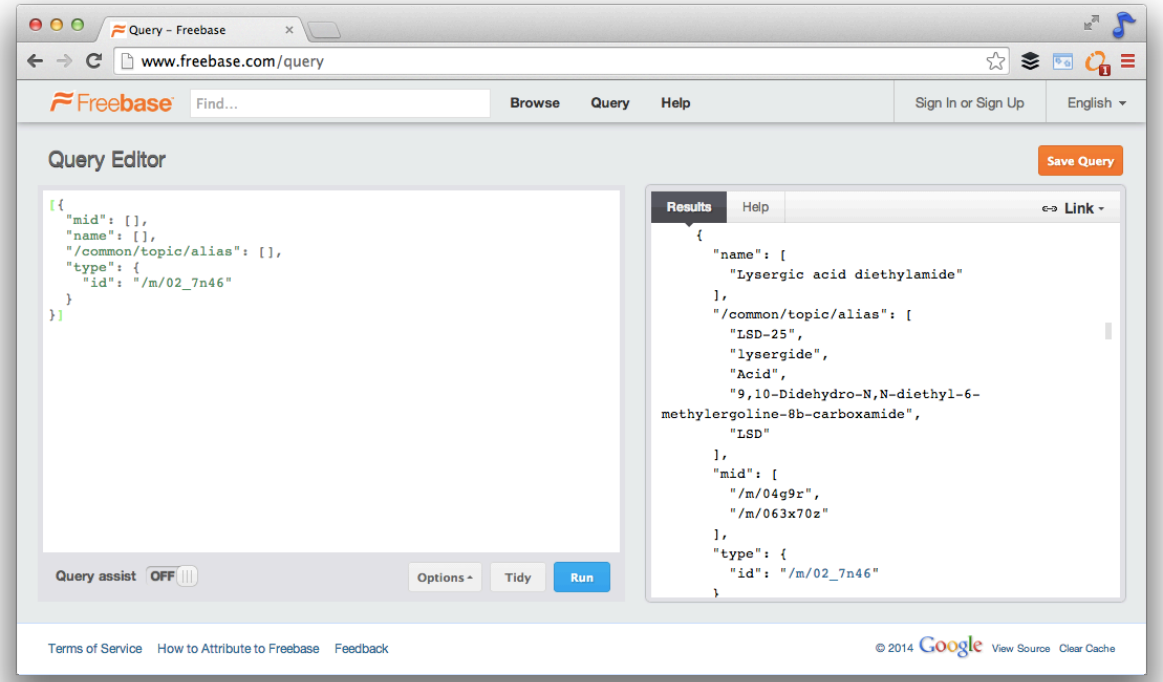
[20] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/bob-dylan-tangled-up-in-blue-20110525

[21] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/jay-z-99-problems-20110527

[22] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-velvet-underground-heroin-20110526

[23] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/pink-floyd-comfortably-numb-20110526

[24] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/jefferson-airplane-white-rabbit-20110526



Querying Freebase to find drug aliases

Interestingly, the reviews accompanying each song often include background stories and contain many drug references: "heroin" is mentioned 11 times, "acid" 3, "alcohol" 3, and "cocaine" twice.

Good vibrations

Finally, I used AlchemyAPI[25], an API for topic extraction and sentiment analysis, to identify the mood of each song based on its lyrics. Here are the most negative songs from the list.

Track	Artist	Sentiment rating
Ain't It a Shame[26]	Fats Domino	-0.71
Why Do Fools Fall In Love[27]	Frankie Lymon and The Teenagers	-0.56
The Girl Can't Help It[28]	Little Richard	-0.54
Monkey Gone to Heaven[29]	The Pixies	-0.54
I Can't Make You Love Me[30]	Bonnie Raitt	-0.51

[25] http://www.alchemyapi.com/

[26] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/fats-domino-aint-it-a-shame-20110526

[27] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/frankie-lymon-and-the-teenagers-why-do-fools-fall-in-love-20110526

[28] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/little-richard-the-girl-cant-help-it-20110526

[29] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/pixies-monkey-gone-to-heaven-20110526

[30] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/bonnie-raitt-i-cant-make-you-love-me-20110526

And the most positive ones:

Track	Artist	Sentiment rating
Can't Buy Me Love[31]	The Beatles	0.67
Everyday[32]	Buddy Holly and the Crickets	0.63
All Shook Up[33]	Elvis Presley	0.59
Love and Happiness[34]	Al Green	0.58
Miss You[35]	The Rolling Stones	0.58

For both, there seems to be a clear bias towards the words used in the song (*e.g.* "shame" or "love"), rather than extracting sentiments from the proper songs' meaning – where an analysis based on data from SongMeanings[36] or Songfacts[37] would make more sense.

Overall, even though lyrics can say a lot about a song, let's move on to analyzing the acoustic properties of these top 500 songs.

[31] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-beatles-cant-buy-me-love-20110526
[32] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/buddy-holly-and-the-crickets-everyday-20110527
[33] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/elvis-presley-all-shook-up-20110526
[34] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/al-green-love-and-happiness-20110526
[35] http://rollingstone.com/music/lists/the-500-greatest-songs-of-all-time-20110407/the-rolling-stones-miss-you-20110526
[36] http://songmeanings.com/
[37] http://songfacts.com

How YouTube music is shared on Twitter

So far, I mostly used "static" data-sources in my experiments. This chapter focuses on dynamic ones, and analyzes the use of Twitter for sharing music, especially YouTube music videos.

As in the previous chapters, before describing these insights, here is a quick description of the data aggregation process – which combines many of the building blocks presented so far:

1. Mining Tweets (from the public stream) containing links to YouTube videos;
2. Identifying which artist(s) and song(s) the videos are about, via Freebase;
3. Loading the Tweets, with data from the previous step, into BigQuery; and finally
4. Running the analytics and the discovery algorithms.

The pipeline uses the stream APIs of both Twitter[38] and BigQuery[39] to retrieve and save the data. In-between, a middleware parses the Tweets and calls the YouTube API[40] and Freebase's Knowledge Graph[41] to extract additional data for each video. Using two streams APIs (in and out) ensures a low latency between the time a Tweet is published and when it appears in the database, enabling the analytics process to work in almost real-time over the Twitter firehose.

After running the pipeline for three days, more than 1.2M Tweets were aggregated, for a total or 516,056 distinct videos, of which 345,410 have been linked to Freebase entities.

Interestingly, within the full dataset, 28.2% of the videos are in the Music category, followed by People and Blog[42] (19.5%) and Entertainment[43] (14.4). This is actually not surprising, considering how people use YouTube as an online music platform.

Freebase and YouTube: A Perfect Combo

Since its update to Version 3, the YouTube API[a] has been fully integrated with Freebase. This way, videos can be queried using Freebase IDs (*i.e.* entities IDs) rather than keywords, avoiding issues such as retrieving the Frank Ocean song *Bad Religion* when looking for videos from the punk-rock band of the same name. In addition, video meta-data from the API includes Freebase IDs as well, so that artist and songs IDs are included in the API results – rather than simple text strings to

[38] https://dev.twitter.com/streaming/overview
[39] https://cloud.google.com/bigquery/streaming-data-into-bigquery
[40] https://developers.google.com/youtube/
[41] https://developers.google.com/freebase/
[42] https://www.youtube.com/people
[43] https://www.youtube.com/entertainment

identify an artist or song name.

However, the YouTube-Freebae mappings are not always perfect, especially for covers songs or featured artist. As an add-on to the original API, I have built youplay (available on PyPI[b] and github[c]), which uses various heuristics to extract additional artist and song information from YouTube videos.

```
> import youplay
>
> (artists, tracks) = youplay.extract('0UjsXo9l6I8')
> artists = ', '.join([artist.name for artist in artists])
> print '%s - %s' %(artists, tracks[0].name)
Jay-Z, Alicia Keys - Empire State of Mind
```

[a] https://developers.google.com/youtube/v3/
[b] https://pypi.python.org/pypi/youplay
[c] https://github.com/apassant/youplay

Popular videos: Super fans or spammers?

Using this dataset, the first query of this experiment – using solely BigQuery SQL capabilities – identifies the most popular videos on Twitter, as well as their corresponding YouTube views. Note that the number of views was gathered from the YouTube API at the time the last Tweet about the video was posted.

```
SELECT
  tweet_youtube.youtube_id, tweet_youtube.youtube_title,
  COUNT(tweet_id) as num_tweets,
  MAX(tweet_youtube.youtube_views) as num_views
FROM
  [Twitter.TwitterStream]
WHERE
  tweet_youtube.youtube_category_id = 10
GROUP EACH BY 1, 2
ORDER BY 3 DESC
```

Row	tweet_youtube_youtube_id	tweet_youtube_youtube_title	num_tweets	num_views
1	sH4QIlpcEMU	5 Seconds of Summer- Intro & End Up Here- Inglewood, CA- November 15, 2014	3820	33942
2	R_DX64EwH9M	GOT7 "하지하지마(Stop stop it)" M/V	3627	1821510
3	1ZRb1we80kM	GD X TAEYANG - GOOD BOY M/V	3397	794296
4	2zwTaT0JeOY	HI SUHYUN - '나는 달라(I'M DIFFERENT)' (ft.BOBBY) M/V	3042	1870955
5	5GFtYgFRkx4	One Direction - Night Changes (Acoustic)	2061	1072946
6	8fWL3lGjgpc	Cala a Boca e Me Beija - Mc Felix	1501	16561
7	RrBq86xw9XE	GD X TAEYANG- 'GOOD BOY' TEASER SPOT	1432	207293
8	SyfMj5lbgcM	Cool Tools - Duality (The X Fucktor) animated funny musical video - Youtube 2014	1292	4499
9	747lZF7RHXI	三代目 J Soul Brothers / O.R.I.O.N.	1244	647038
10	soQ7GezfNl4	One Direction - Night Changes (4 days to go)	1017	1440315
11	3Wz-KiqFXGc	'Danger (Mo-Blue-Mix) ft. THANH' MV	971	51085

Most popular videos in the dataset

For some tracks, I was intrigued by the ratio between the number of Tweets and their views. As you can see, some videos have lots of YouTube views compared to the number of Tweets mentioning them – for instance, the 6th result in the previous list. Digging further, I ran a second SQL query to measure the number of Tweets per user, for any video: as depicted below, we can clearly observe that some videos are self-promoted, or should I say spammed, on Twitter.

```
SELECT
  tweet_youtube.youtube_id, tweet_youtube.youtube_title,
  COUNT(DISTINCT(tweet_user_id)) as num_users,
  COUNT(tweet_youtube.youtube_id) as num_tweets,
  MAX(tweet_youtube.youtube_views) as num_views,
  CAST(COUNT(DISTINCT(tweet_user_id)) as float)/CAST(COUNT(tweet_youtube.youtube\
_id) as float) as ratio
FROM
  [Twitter.TwitterStream]
WHERE
  tweet_youtube.youtube_category_id = 10
GROUP EACH BY 1, 2
ORDER BY 6 ASC
```

Row	tweet_youtube_youtube_id	tweet_youtube_youtube_title	num_users	num_tweets	num_views	ratio
1	mIOy3CTe4eE	Tre Sixty "500+" (24 Mins, 958 Bars, No Cursing, No Hook)	1	155	442	0.0064516129032258064
2	bGEiusNVboU	#OGLK - Season Opener (Official Video)	1	153	301	0.006535947712418301
3	biZjBqJu0bI	'TM' Full Album (2014) - TheMisanthropists	1	93	418	0.010752688172043012
4	it8R38JmaVU	Richie Nuzz – My Crib (Kromatiks Remix) (Official)	2	182	10450	0.01098901098901099
5	W3VlaQFiSFs	Humam - Almni | علمني - همام	1	64	14380	0.015625

Number of Tweets versus views on YouTube

To measure the "real" popularity of videos, I updated the first query to consider only one Tweet per video per user – an easy way to find top tracks based on the number of unique users who tweeted them.

Row	tweet_youtube_youtube_id	tweet_youtube_youtube_title	num_tweets	
1	sH4QIIpcEMU	5 Seconds of Summer- Intro & End Up Here- Inglewood, CA- November 15, 2014	3798	
2	1ZRb1we80kM	GD X TAEYANG - GOOD BOY M/V	3255	
3	R_DX64EwH9M	GOT7 "하지하지마(Stop stop it)" M/V	3178	
4	2zwTaT0JeOY	HI SUHYUN - '나는 달라(I'M DIFFERENT)' (ft.BOBBY) M/V	2934	
5	5GFtYgFRkx4	One Direction - Night Changes (Acoustic)	2028	
6	RrBq86xw9XE	GD X TAEYANG- 'GOOD BOY' TEASER SPOT	1400	
7	747IZF7RHXI	三代目 J Soul Brothers / O.R.I.O.N.	1189	
8	soQ7GezfNI4	One Direction - Night Changes (4 days to go)	967	
9	3Wz-KiqFXGc	'Danger (Mo-Blue-Mix) ft. THANH' MV	925	
10	7x4_JSxvkJA	One Direction - Night Changes (2 days to go)	863	

Popular videos (one Tweet per user)

Entities: Better than tags

By linking videos to entities via Freebase, rather than doing a simple tag or keyword-based extraction, much more meaning can be derived from Tweets. Since every entity has a type (artist, genre, etc.), additional filtering can be done. For instance, the previous query can be updated to find the top artists (*i.e.* entities having a type "music artist"), rather than the top tracks.

Row	tweet_youtube_youtube_topic_topic_id	tweet_youtube_youtube_topic_topic_name	num_tweets	
1	/m/0fqqnpw	One Direction	4390	
2	/m/0kvf4_q	5 Seconds of Summer	4365	
3	/m/0_1c8x0	Got7	3025	
4	/m/0dl567	Taylor Swift	1765	
5	/m/06w2sn5	Justin Bieber	1150	
6	/m/0g9sr1k	Ed Sheeran	1077	
7	/m/0ndsknr	Sandaime J Soul Brothers	943	
8	/m/01vsgrn	Eminem	825	
9	/m/09gkdy4	Ariana Grande	792	
10	/m/0123r4	Band Aid	712	

Popular artists

Going further, the following query identifies the most popular music genres in the dataset. Once again, this is possible thanks to the initial mappings between Tweets, videos and entities.

```
SELECT
  tweet_youtube.youtube_relevant_topic.topic_id, tweet_youtube.youtube_relevant_\
topic.topic_name,
  COUNT(DISTINCT tweet_youtube.youtube_id) as num_videos
FROM
  [Twitter.TwitterStream]
WHERE
  tweet_youtube.youtube_relevant_topic.topic_type = '/music/genre'
GROUP EACH BY 1, 2
ORDER BY 3 DESC
```

Row	tweet_youtube_youtube_relevant_topic_topic_id	tweet_youtube_youtube_relevant_topic_topic_name	num_videos
1	/m/064t9	Pop music	8958
2	/m/09qxq7	Acoustic music	7059
3	/m/0glt670	Hip hop music	6411
4	/m/06by7	Rock music	6348
5	/m/016clz	Alternative rock	4544
6	/m/03lty	Heavy metal	3210
7	/m/0l14gg	Soundtrack	2529
8	/m/025sc50	Contemporary R&B	2103
9	/m/06j6l	Rhythm and blues	1851
10	/m/05bt6j	Pop rock	1775

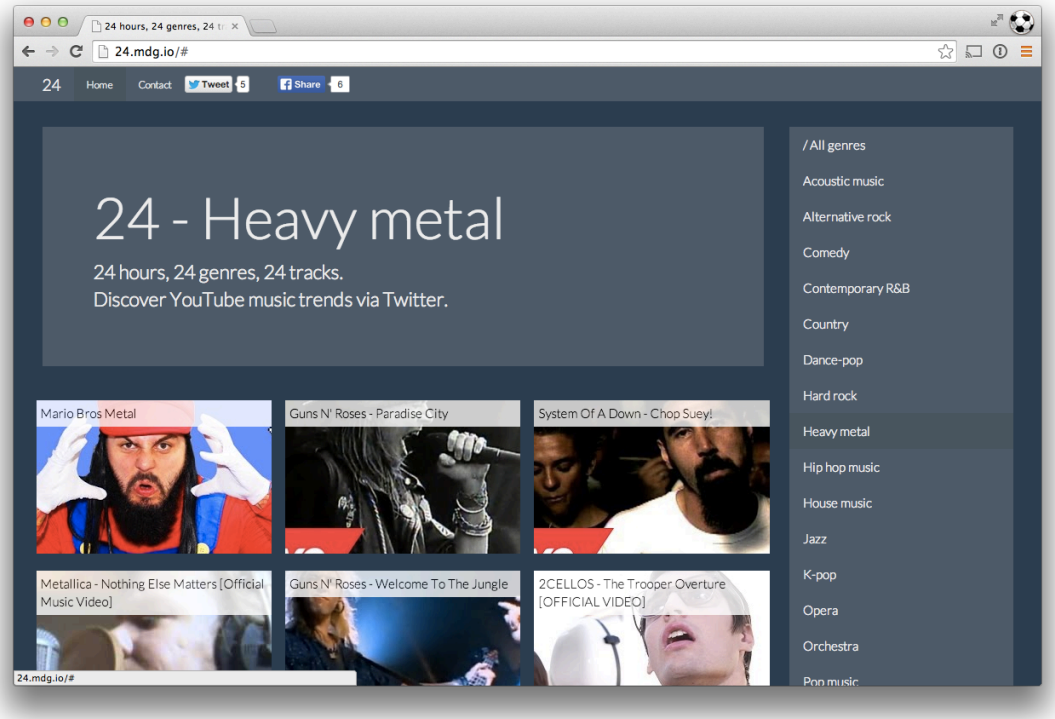
Top-10 genres in the dataset

Using the relations between Tweets, videos, and Freebase entities, opens the door to a wide range of analytics capabilities. So let's extend the approach to build genre-specific rankings. As an example, below is the list of top 10 Heavy-metal tracks.

Row	tweet_youtube_youtube_id	tweet_youtube_youtube_title	num_tweets	
1	CSvFpBOe8eY	System Of A Down - Chop Suey!	82	
2	Y7VL_PTCLGU	Therapy?-Diane	62	
3	Tj75Arhq5ho	Metallica - Nothing Else Matters [Official Music Video]	42	
4	eVH1Y15omgE	2CELLOS - The Trooper Overture [OFFICIAL VIDEO]	39	
5	uT3SBzmDxGk	2CELLOS - Thunderstruck [OFFICIAL VIDEO]	35	
6	Ckom3gf57Yw	Metallica - The Unforgiven (Video)	33	
7	v2AC41dglnM	AC/DC - Thunderstruck	30	
8	DelhLppPSxY	Avenged Sevenfold - Hail To The King [Official Music Video]	30	
9	zUzd9KylDrM	System Of A Down - B.Y.O.B.	30	
10	WlKqgE4BwAY	BABYMETAL - ギミチョコ！！- Gimme chocolate!! - Live Music Video	28	

Top Heavy-metal videos

24 hours, 24 genres, 24 tracks



24 top Heavy-metal tracks

Using the approach described in this chapter, I built 24[a], a proof of concept for Twitter-based music discovery. Its idea is simple: identify the top 24 tracks of the top 24 genres played via YouTube

during the last 24 hours on Twitter.

While the original Twitter music app eventually failed[b], this hack showcases the huge dataset that Twitter has from a music discovery standpoint, and the opportunities around it, such as the recent introduction of audio cards[c].

[a] http://24.mdg.io
[b] https://twitter.com/TwitterMusic/statuses/447136704462209025
[c] https://blog.twitter.com/2014/introducing-a-new-audio-experience-on-twitter

Twitter, personalization and music

Extracting meta-data from links[44] can bring lots of valuable content to social-media content. As we have just seen, even when a Tweet contains a single YouTube link, mining data from this link can tell a lot about the content, and eventually the user who shared it. With enough data, users could be clustered by music tastes, and for instance receive recommendations of artists to follow, videos to watch, or music to buy[45] based on this data mined from external sources.

Of course, recommendations could be done on any topic using this approach: music is just a use case there. Thus, such a data-mining approach to enable personalization becomes even more relevant in the context of the Twitter feed update[46] and how to make the stream more personal, especially when on-boarding users.

From a pure music standpoint, bands and advertising managers who want to promote their music on Twitter could use those data mined signals to target specific users in advertising campaigns. As music marketing is a constant struggle for artists and managers, the approach could enrich traditional advertisement strategies on Twitter.

[44] http://www.slideshare.net/sheilakinsella/sheila-kinsella-phd-defense-10115432
[45] https://blog.twitter.com/2014/testing-a-way-for-you-to-make-purchases-on-twitter
[46] https://blog.twitter.com/2014/coming-soon-to-twitter