Search System for Billboard Songs

ISLA CASSAMO, FEUP, Mozambique
MARIA BAIA, FEUP, Portugal
RICARDO NUNES, FEUP, Portugal

In this project, the data preparation, retrieval and evaluation phases of a search engine for the Billboard songs was implemented. The primary dataset containing the song names and rank was fetched from the data.world site. The rest of the data has fetch using Genius API. All song and artist names were normalized and duplicates and missing values were removed. Some key statistics were drawn from the data, such as the best musics and artists based on time spent on the Billboard and the frequency of the words love and hate. Two systems were developed for the task of document retrieval, a simple one without boosting and a second with boosting. The information needs were expressed in queries and the results evaluated using the metrics Precision at 10 and Average Precision. Finally, the Mean Average Precision was calculated using all the results.

Additional Key Words and Phrases: datasets, data gathering, data cleaning, data analysis

ACM Reference Format:
Isla Cassamo, Maria Baia, and Ricardo Nunes. 2021. Search System for Billboard Songs. *ACM Trans. Graph.* 1, 1, Article 1 (November 2021), 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Music is the art of arranging sounds in time through the element of melody, harmony, rhythm and timber, and has been with humanity since the beginning of times [3]. With the popularization of the internet since 1998, music stopped being distributed through the typical physical means, such as CD and cassettes, and assumed its place on the internet, allowing the easy globalization of songs.

For having numerous stands, themes, origins and consequently being targeted by diverging audiences, institutions such as Spotify [7] and Apple Music [2] started collecting information on anything about music, such as lyrics, artist articles, writers notes, samples and placing it on diverse platforms across the web, to provide easy access to such desirable information.

This project then aims to centralize this information by collecting data already existent on the internet, with particular focus on the most popular songs, so as to allow those that procure information about a song to do so in just one click in a platform with data that was refined and analysed.

Authors' addresses: Isla Cassamo, FEUP, Maputo, Mozambique; Maria Baia, FEUP, Porto, Portugal, up201704951@g.uporto.pt; Ricardo Nunes, FEUP, Porto, Portugal, up201706860@g.uporto.pt.

0730-0301/2021/11-ART1 $15.00
https://doi.org/10.1145/1122445.1122456

2 DATA GATHERING AND PREPROCESSING

All the following subsections represent the several steps taken in the overall process of data gathering and preprocessing. This can be seen in a more summarised way in Figure 3.

2.1 Data collection

As music has been present in the life of humanity for so long, there is a lot of data available on the topic. Therefore, before finding the appropriate datasets, it was necessary to choose a feasible scope that would represent the industry well, while being relevant to the society at hand and with enough documentation to still be aligned with the project's objectives. For this reason, we choose to restrict our search to the Billboard dataset [1]. This dataset proved adequate as for a reasonable size of data it portrayed the most representative sample that aligned with the reality of the industry.

The second step was to merge this primary data, that contained no more than the song name, leading artist, the release date and Billboard ratings with other sources of information available on the internet. In this project, the Genius API [4] and its Python extension lyricsGenius [5], allowed access to data from a reputable company that creates data of the musical scope. Despite not allowing direct access to its datasets, one could use Python and crawling methods in order to scour the API and retrieve information about the primary data and arranging it in CSV files one by one.

2.2 Data storage

The process of obtaining data from the Billboard dataset put together a great quantity of primary data that was compiled into a single CSV file. Using Python, the next step was to crawl the web using the genius API and the previous obtained songs, which required careful calibration in order to create three new files that contained all relevant information about the artists, albums and songs.

The decision to compile the data in CSV files was due to the necessity to store data in a structural manner, that would be supported by Python and that could be easily navigated through.

2.3 Data cleaning and refinement

Since the data was collected through trustworthy means (Genius API [4] and data.world [1]), it was mostly clean and complete. Despite this, the names of some key columns needed for merging the data were different from one dataset to another, e.g., the artist and song name. This required the standardization of all column's names.

The different sources of data lead to the appearance of some missing values. This values were rare and represented a small fraction of overall data. For this reasons, they were discarded.

2.4 Data analysis

In this section, some statistical analyses methods were used on the data to get a better understanding of it.

Table 1. Top 10 artists on the BillBoard

Artist Name	Nº of songs
Glee Cast	183
Taylor Swift	67
Drake	63
Elton John	59
Madonna	57
Tim McGraw	50
Rod Stewart	48
Kenny Chesney	46
Chicago	44
Billy Joel	43

Table 2. Top 10 albums on the Billboard

Album Name	Nº of songs
Ulysses	46
Finnegans Wake	28
Glee: The Music, The Complete Season Two	28
Glee: The Music - The Complete Season Three	23
More Life	21
beerbongs & bentleys	17
The Big Book of Song Lists	17
Starboy	16
Views	16
Purpose (Deluxe)	15

First, the search for the top 10 artists, albums and musics of the Billboard was made to know which ones were the most popular (Table 1, Table 2 and Table 7, respectively. Table 7 is located on the Annexes section, due to its large size).

The average length of all lyrics is around 5,875 characters. The word *love* is present in 8,450 music lyrics, while the word *hate* is only present in 1,763.

We wanted to see the evolution of number of songs in-and-out of the Billboard through the years, so we constructed the line plot in Figure 2. The lack of songs of 1970 can be explained by the data collection starting mid year.

3 CONCEPTUAL MODEL

The conceptual model follows as shows in Figure 1.

The central class of the domain of the application is the song. It has a *name*, *artist name*, *peak position*, *last position* obtained, *number of weeks* on the Billboard, *rank* and *lyrics*.

This model also makes use of the notion of artist. An artist is the creator of a song and can be a single artist or an collaboration between several. All artists have a *name*, a *description* (corresponding to his/her biography), a list of *aliases*.

Another concept identified is the album. An album represents an agglomeration of songs and is created by a single artist. An album is characterized by a short *name*, a *full title*, a *release date* and a *description*.

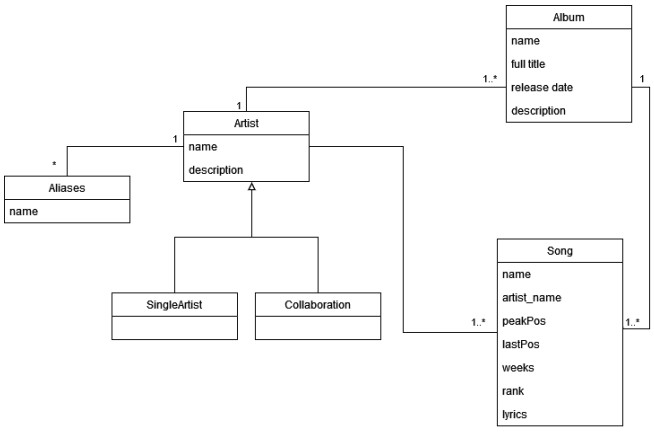

Fig. 1. Conceptual model

4 SEARCH TOPICS

The data search engine will consult a set of songs, originated between the years 1970 and 2018, their lyrics, albums they belong to, and the artists who produced them. For a more detailed search, you'll be able to filter the results based on year, artist, among other filter options. The engine will have the capacity to:

- Search musics where the primary theme is friendship
- Search for debut albums of artists in 2000
- Search for New York born artists with albums in the billboard 200 record chart
- Search for albums that won the Grammy Award for Best R&B Album since the '90s

5 INFORMATION RETRIEVAL TASK

Information Retrieval can be defined, in this context, as the obtaining of relevant documents from a collection, driven by a need for information. In the context of this paper, the collection of documents is all information collected about artists and their musics and albums.

To evaluate the pertinence of the obtained results, two systems were developed: System A, the base system, designed to match on pre defined fields, and System B, similar to System A but with boosting of relevant fields. We will analyse the the top 10 results for each information need using the metrics Precision at 10 and Average Precision. The Recall metric is not used, since the document collection is quite large and impractical to get the number of relevant documents contained on it. In the end, the Mean Average Precision will be calculated over all queries.

Apache Solr [6] was the chosen tool for the retrieval and evaluation of the documents, since it has a intuitive interface and an extensive documentation. Moreover, it is relatively simple to use. The usage of *eDisMax* is justified because it is design to handle simple phrases, with the ability to add boosting to wanted fields.

5.1 Collections and Documents

A single core *music* was created to hold all documents. In future work, as an enhancement, multiple cores will be added. In addition

to the core, three types of documents were created: *songs*, *artists* and *albums*. This documents types follow closely the UML presented (Fig 1). The following pseudo-code (Listing 1, 2 and 3) represents a simplified version of the schemes. The structure of the pseudo-code is as follows: in the first line, a comma separated list of fields, followed by their attributes. The types *descriptiveText* and *artistName* were defined by us (explained in section 5.2). The types *pdate* and *pint* are defined by Solr itself. The field *artist_name* is present in all document types, and will serve has a "bridge" between the document types.

```
song_name, lyrics
    type: descriptiveText
    indexed: true

artist_name
    type: artistName
    indexed: true

music_date
    type: pdate

peakPos, lastPos, weeks, music_rank
    type: pint
```

Listing 1: Simplified version of song schema

```
album_name, full_title, album_description
    type: descriptiveText
    indexed: true

artist_name:
    type: artistName
    indexed: true

release_date
    type: pdate
    indexed: true
```

Listing 2: Simplified version of album schema

```
artist_description
    type: descriptiveText
    indexed: true

artist_name
    type: artistName
    indexed: true
```

Listing 3: Simplified version of artist schema

5.2 Indexing Process

For the indexing process the main goal was to index the information of the dataset. We start by deciding which fields were needed and how to split their data in order to meet the retrieval requirements.

We identified non complex fields that return a single value: *music_rank*, *weeks*, *lastPos* and *peakPeek*. Although there fields contain relevant information to evaluating a song according to the Billboard parameters, they do not follow the search text that is intended. Taking that into account, this fields were stored with the default Solr field type *pint* and set to false the indexing property.

Next, for the fields that represent dates, we stored them with the default Solr field type *pdate* and with the indexing properties set to true.

For the remaining fields, although we could use the default Solr field type for textual fields, since they represent textual data, we decided to define two different field types that describe analyzers, which are pipelines that take a field's text value as inputs, and produce a token stream as output. The pipeline is composed by a tokenizer, which breaks a field data into tokens, following a given subset of given filters. The filters examine a stream of tokens, changing it depending on the type of filter used. As mention in the previous sections, we generated two different field types, so we could be able to customize them according to our need: *descriptiveText* and *artistName*.

Both field types use a standard tokenizer, that splits the text field into tokens, according to the white-spaces and punctuation characters and removing them. However, a dot that's not followed by white-space is considered part of a token. For the filter, we use the ASCII folding filter, which converts alphabetic, numeric, and symbolic Unicode characters that are not in the Basic Latin Unicode block into their ASCII equivalents, if one exist, and we set to true the preserveOriginal property, so we could keep the original tokens. We also use the Lower case filter, which converts tokens with uppercase letters into an equivalent token in lowercase letters.

For the *descriptiveText* field we use a filter to discard irrelevant words for our search, specified in the *stopwords.txt* file, that is included in the Solr *conf* directory, which includes typical English language text that can be ignored, since that are considered irrelevant for information retrieval. We also use a stemming filter which applies the Porter Stemming Algorithm, to reduce inflected words to their word base.

For the *artistName* field we use and filter that implements the Beider-Morse Phonetic Matching algorithm, which identifies similarity between different names, even if they are spelled differently or in different languages.

5.3 Retrieval Process

The query and it's parameters and the results are specified for each information need. All information needs are described in Section 4

5.3.1 Information Need N.1.

This information need represents all musics where the central theme is friendship. The built query to retrieve the relevant documents is:

```
friendship
```

The chosen parameters were:

```
System A:
    qf: song_name lyrics

System B:
    qf: song_name^10 lyrics^2
    pf: song_name
```

Songs that contain the word "friendship" in the title are more prone to be about friendship, hence the boosting on the field.

The results of both systems can be seen in Table 3

	System A		System B	
Rank	Song	R	Song	R
1	Reap The Wild Wind	Y	Headlines (...)	Y
2	Headlines (...)	Y	Reap The Wild Wind	Y
3	Freedom Comes, Freedom Goes	N	Freedom Comes, Freedom Goes's	N
4	Could This Be Love	N	Could This Be Love	N
5	Wannabe	Y	Wannabe	Y
6	Buddy	Y	Buddy	Y
7	I Don't Need You	N	I Don't Need You	N
8	I Don't Need You	N	I Don't Need You	N
9	Mirror Man	Y	Mirror Man	Y
10	Simple Kind Of Life	N	Simple Kind Of Life	N
P@10	0.5		0.5	
AP	0.656		0.656	

Table 3. P@10 and AP results

Note: The R in this table and in the following ones stands for *Document Relevance*. Moreover, the symbol (...), present in some titles, means that the name was truncated.

The result from both systems are identical, since this is a simple query and only on song had the word friendship in the title.

5.3.2 *Information Need N.2.*

With this information need, the systems should return any artist's debut album from 2000.

The chosen query for this effect was:

```
("first album" OR "debut album") AND 2000
```

The expressions "first album" and "debut album" are synonymous, hence the parenthesis. The double-quotes serves the propose of In conjunction with the query, the following parameters were picked:

```
System A:
    qf: release_date album_description

System B:
    qf: release_date^10 album_description^2
    pf: album_description
    ps: 3
```

The fields chosen for the search were *release_date* and *album_description*. For System B, a greater boost was given to the field release_date, since it's imperative for the year to be 2000.

The results are presented in Table 4

As it can be seen from the results, the P@10 and AP are relatively low. This can be explained by the complexity of the query, and because the word "first" appears in a lot of descriptions with other contexts.

5.3.3 *Information Need N.3.*

This information need retrieves all New York born artists with albums in the billboard 200 record chart. The following query was the chosen to represents how the search:

	System A		System B	
Rank	Album	R	Album	R
1	Madonna by Madonna	N	Madonna by Madonna	N
2	Parachutes by Coldplay	Y	Parachutes by Coldplay	Y
3	Who Is Jill Scott? by Jill Scott	Y	Who Is Jill Scott? by Jill Scott	Y
4	Return of Saturn by No Doubt	Y	Return of Saturn by No Doubt	Y
5	2Ge+her: Again by 2gether	N	Best Of by 50 Cent	N
6	Everyday by Dave Matthews Band	N	Operation Stackola by Luniz	N
7	Dance with Me by Debelah Morgan	N	Come Away With Me by Norah Jones	N
8	Best Of by 50 Cent	N	Mad Season by Matchbox Twenty	N
9	Human Clay by Creed	N	Beware of Dog by Bow Wow	Y
10	Operation Stackola by Luniz	N	Tical 2000: Judgement Day by Method Man	N
P@10	0.3		0.4	
AP	0.445		0.467	

Table 4. P@10 and AP results

```
q: billboard 200" AND "New York"
fq: !(artist:"featuring")
```

The expression "billboard 200" intends to get the 200 most requested albums and the "New York" expression is to indicate the required city or state where the artist was born. The expression "!(artist:"featuring")" it was used to prevent albums that were made together with more than one artist from not appearing.

The chosen parameters were:

```
System A:
    qf: artist_description album_description

System B:
    qf: artist_description^10 album_description^5
```

For the qf (query fields) parameter we use the *artist_description* and the *album_description*, which would be the indicated fields to look for the expressions used in the query. For system B, it was set a greater weight to the *artist_description*, since it can be the field with the most information, not only about the birthplace, but also about the references of the artist albums.

The results of both systems can be seen in Table 5

5.3.4 *Information Need N.4.*

For this information need we intend to retrieve the albums that won the Grammy Award for Best R&B Album since the '90s. To do that, the following query was used:

```
 q: "grammy" AND "r&b" AND ("win" OR "won")
fq: release_date:[1990-01-01T00:00:00Z TO *]
```

	System A		System B	
Rank	Artist	R	Artist	R
1	G Unit	N	G Unit	N
2	Frank Mills	N	Frank Mills	N
3	Double or Nothing	N	Sharissa	Y
4	Dogg Food	N	N.O.R.E.	Y
5	Sharissa	Y	Post Malone	Y
6	Get Rich or Die Tryin'd	N	Gladys Knight And The Pips	N
7	N.O.R.E.	Y	Automatic Man	N
8	52nd Street	N	A Tribe Called Quest	Y
9	Chapter V	N	AJR	Y
10	Post Malone	Y	Bow Down	N
P@10	0.3		0.5	
AP	0.142		0.341	

Table 5. P@10 and AP results

With this query we intend that the expressions "grammy" and "r&b" must be contained in the search, and the expression "win" or "won", which represent synonyms. The expression "release_date:[1990-01-01T00:00:00Z TO *]" is used to indicate all documents that have the *data_release* field data greater than the indicated data.

The chosen parameters were:

```
System A:
    qf: album_description

System B:
    qf: album_description^2
```

The results of both systems can be seen in Table **??**

	System A		System B	
Rank	Albums	R	Albums	R
1	Get Lifted	Y	Get Lifted	Y
2	As I Am	Y	As I Am	Y
3	Songs in A Minor"	Y	Songs in A Minor"	Y
4	Toni Braxton	N	Toni Braxton	N
5	Words	N	Words	N
6	Power of Love	N	Power of Love	N
7	The Diary of Alicia Keys	N	The Diary of Alicia Keys	N
8	The Sound	N	The Sound	N
9	Secrets	N	Secrets	N
10	G I R L	N	G I R L	N
P@10	0.3		0.3	
AP	0.623		0.623	

Table 6. P@10 and AP results

5.4 Global Evaluation

Considering all results from all metrics, it can be concluded that the overall metric values were bellow of what was expected. This can be explained by the following reasons: on one hand, some queries were complex. On the other hand, some queries were very specific, and it's possible that there were not enough documents in the collection to satisfy the information need.

Finally, the calculated Mean Average Precision was 0.4665 for System A and 0.52175 for System B. System B outperformed system A, due to the boosting.

6 CONCLUSION

This project required the procure of appropriate datasets, determining the tools needed to explore them and make use of those tools to the desired extent. After analysis on the primal dataset, it proved imperial to understand how to cross data from other sources in order to enrich the initial dataset chosen. One can although conclude that the final result table included a large dataset of music with complete and coherent data throughout, which can easily be queried to obtain sets of information needed to produce a useful and effective engine for the future platform.

REFERENCES

[1] Michael Tauberg. *Billboard Hot-100 songs from 1970-2017*. 2018. URL: https://data.world/typhon/billboard-hot-100-songs-from-1970-2017 (visited on 10/29/2021).
[2] *Apple Music*. 2021. URL: https://music.apple.com/us/browse (visited on 12/02/2021).
[3] Gordon Epperson. *music*. 2021. URL: https://www.britannica.com/art/music (visited on 11/18/2021).
[4] Genius. *Genius API*. 2021. URL: https://docs.genius.com/ (visited on 11/05/2021).
[5] John W. Miller. *LyricsGenius: a Python client for the Genius.com API*. 2021. URL: https://pypi.org/project/lyricsgenius/ (visited on 11/05/2021).
[6] *Solr Apache*. 2021. URL: https://solr.apache.org/ (visited on 12/14/2021).
[7] *Spotify*. 2021. URL: https://www.spotify.com/pt-en/ (visited on 12/02/2021).

ANNEXES

Table 7. Top 10 Billboard musics

Song Name	Artist Name	N° of weeks
Radioactive	Imagine Dragons	87
Sail	AWOLNATION	79
I'm Yours	Jason Mraz	76
How Do I Live	LeAnn Rimes	69
Counting Stars	OneRepublic	68
Party Rock Anthem & LMFAO Featuring Lauren Bennett	GoonRock	68
Rolling In The Deep	Adele	65
Foolish Games/You Were Meant For Me	Jewel	65
Before He Cheats	Carrie Underwood	64
Ho Hey	The Lumineers	62

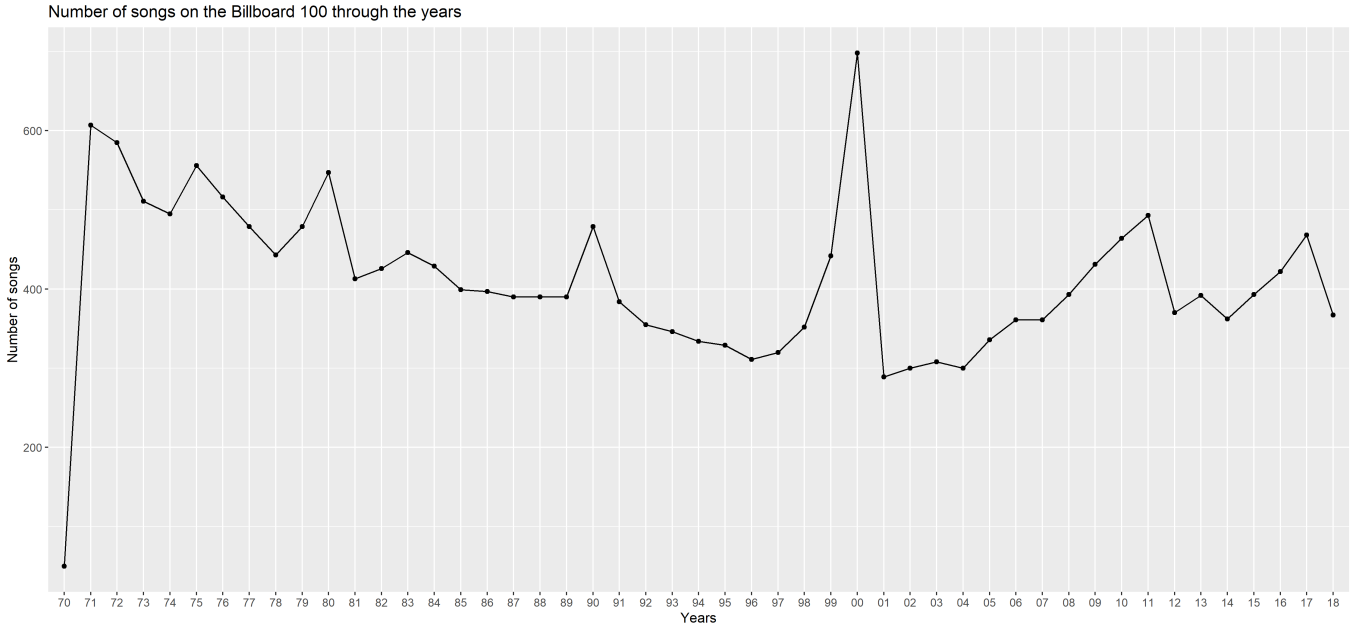

Fig. 2. Evolution of the number of musics passing through Billboard over the years

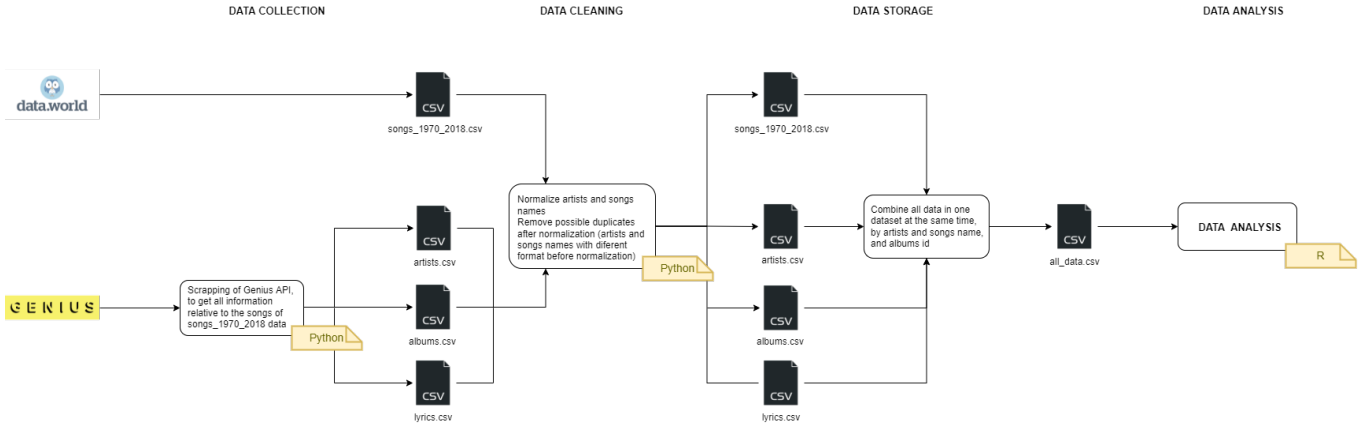

Fig. 3. Pipeline diagram representing the various phases of data preparation.