

Personalized Search Engine for Microblogs

Information Retrieval
2021-2-F1801Q110
Final Project

Claudio Rota 816050
Simone Paolo Mottadelli 820786
Stefano Zuccarella 816482

Contents

1	Introduction	1
2	Dataset	2
3	Search engine	4
4	Demonstration plan	8
4.1	Textual search on a specific field using keywords	8
4.2	Textual search on a combination of fields	10
4.3	Rank the tweets taking into account the user profile	12
4.4	Expand the search adding synonyms of words in the query	13
4.5	Additional functionalities	14

1 Introduction

The goal of this project is to develop a personalized search engine for tweets using Lucene.

The document corpus has been generated considering posts published by 6 different users having 2 different topical interests, that is, sport and space missions.

Every tweet in the repository has been processed to extract the author, the publication date and time, the textual content as well as various social media items, such as emojis, hashtags and tags. To accomplish this task, several preprocessing steps have been applied.

The search engine provides the possibility to perform both traditional and personalized searches. In the former case, the system executes basic searches using keywords specified on a single field or on multiple fields. In the latter case, the system executes the query in the same way as in the traditional search, but it also performs a re-ranking of the initial results by taking into account the similarity between each retrieved document and a specific user model, which formally represents the user topical interests.

Moreover, the system can also operate a query expansion in order to consider also synonyms of the query terms.

Finally, users can interact with the search engine through a simple GUI.

2 Dataset

The dataset has been created using the official Twitter API and consists of 6039 tweets posted from 31/08/2020 to 16/12/2020 by 6 different users: 3 of them are the main American sport leagues (NBA, NHL and MLB) while the others are space agencies (NASA, ESA and UK Space Agency). These users are very active and have produced many tweets during this period. Intuitively, NASA, ESA and UK Space Agency usually post contents related to rocket launches, space discoveries, news about astronauts and so on. Instead, NBA, NHL and MLB mainly post tweets about players, sport matches, match results and other contents related to sport.

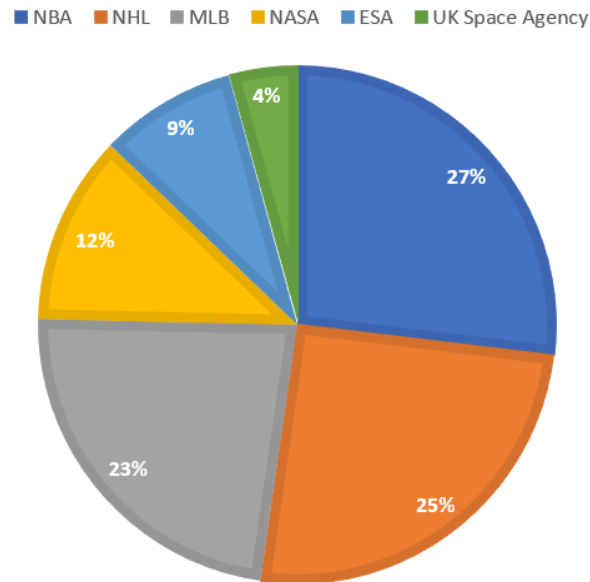


Figure 1: Tweet distribution

In particular, Figure 1 shows the tweet distribution with respect to each user. As it is possible to notice, American sport leagues are more active on Twitter than space agencies.

The original dataset is structured in JSON format and contains the textual

content of tweets as well as other information, as shown in Listing 1. For this project, just the publication date, the author, the user mentions, the hashtags and the tweet text have been considered. In particular, the textual content of the tweets may consist of plain text and other social media items, that is, hashtags, tags, emojis and urls.

The average number of words contained in each tweet is about 25. More in detail, Figure 2 shows the average number of words per tweet posted by each user. It emerges that tweets posted by space agencies contain more words than the ones posted by American sport leagues.

```
{ [
  "created_at": "Sat Oct 24 19:09:00 +0000 2020",
  "id": 1320079864326619137,
  "full_text": "Every fifth inning, all #WorldSeries long. Keep your
    eyes on the @TMobile feed later tonight! https://t.co/ag4pt2XJYb",
  ...,
  "entities": {
    "hashtags": [{
      "text": "WorldSeries",
      "indices": [24, 36]
    }],
    "user_mentions": [{
      "screen_name": "TMobile",
      "name": "T-Mobile",
      "id": 17338082,
      "indices": [71, 79]
    }],
    ... },
  ... ],
  ... }
```

Listing 1: Example of a tweet in the dataset

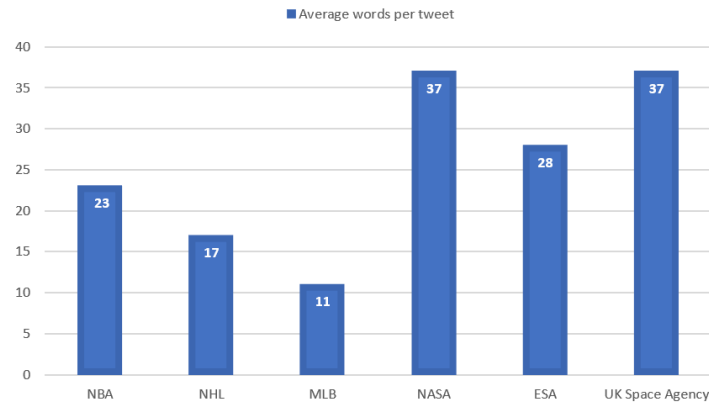


Figure 2: Average number of words per tweet

3 Search engine

The project has been developed in Java using Lucene. First, the indexing component has been implemented, followed by the definition of the user model and finally by the matching component.

The dictionary has been extracted using the indexing component and stored in the system through an inverted file structure.

For each tweet, a document has been created with the following fields:

- *created_at*: the publication date and time of the tweet;
- *screen_name*: the author of the tweet;
- *full_text*: the textual content of the tweet;
- *hashtags*: the hashtags of the tweet;
- *citations*: the user tags in the tweet.

To properly preprocess the textual content of the tweets, i.e., the *full_text* field, a custom analyzer has been defined, consisting of the following steps:

- *Tokenization preparation*: white spaces are added before and after every emoji, so that in the tokenization phase they are considered

as single tokens. This step is fundamental to correctly detect emojis because users usually insert multiple emojis without separating them with white spaces;

- *White space tokenization*: the character sequence is split into a token sequence by using white spaces as token delimiters;
- *Punctuation removal*: the punctuation marks have been removed at the beginning and at the end of the tokens. Note that, in order to correctly handle hashtags and tags, the characters “@” and “#” have not been considered as punctuation;
- *URL removal*: the URL tokens have been removed because they are considered meaningless indices;
- *Lower case conversion*: the tokens have been converted to lower case.

An example of the preprocessing phase is shown in Figure 3.

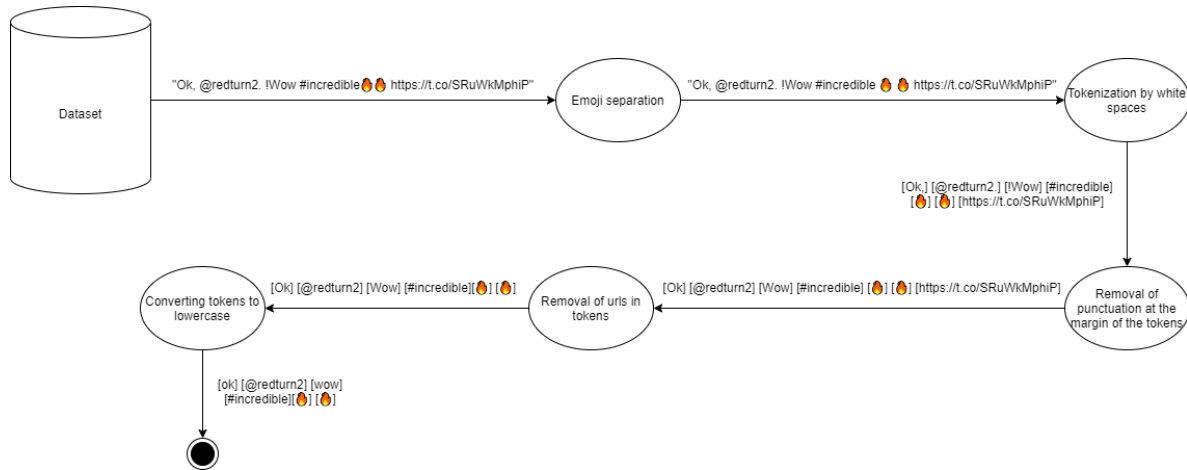


Figure 3: Preprocessing steps applied to a tweet text

The reason why a custom analyzer has been defined is that the available Lucene analyzers did not allow to properly handle social media items. In addition, stop words have not been removed because it is undesirable to

receive empty results in response to a query.

The Extended Boolean Model has been used in order to represent documents as vectors without losing the possibility to submit boolean queries. The user model has been created by taking into account the top 100 most frequent words written by the user in the tweets. Figure 4 shows the workflow that allows to generate the user profile.

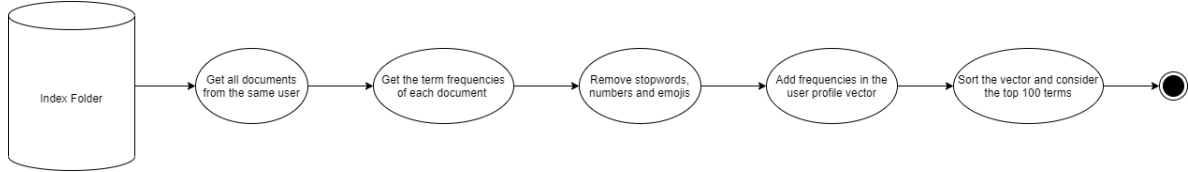


Figure 4: User model creation

The user model has been represented by means of a vector, whose entries are the term frequencies of the corresponding words. In the same way, to compare the user model with a document, the document has been represented as a vector, considering only the words in the user model instead of all the words in the document. Finally, to compute the similarity score, the cosine similarity has been used.

As regards the computation of the score of a document with respect to a query, the BM25 similarity has been used. When the user model is considered, the document score is computed using the re-ranking approach. In this case, the final score is computed as follows:

$$FinalScore = (1 - \alpha) \times Sim(q, d) + \alpha \times Sim(d, u) \quad (1)$$

The α parameter balances the impact of each similarity score: if $\alpha = 0$ then just the similarity between the document and the query is considered, while if $\alpha = 1$ then just the similarity between the document and the user model is used.

The workflow shown in Figure 5 describes the steps to produce the results in response to a query.

In order to perform searches considering also synonyms, the query is ex-

panded with additional words corresponding to the synonyms of the original query keywords. These synonyms have been taken from WordNet, a very popular semantic-lexical database.

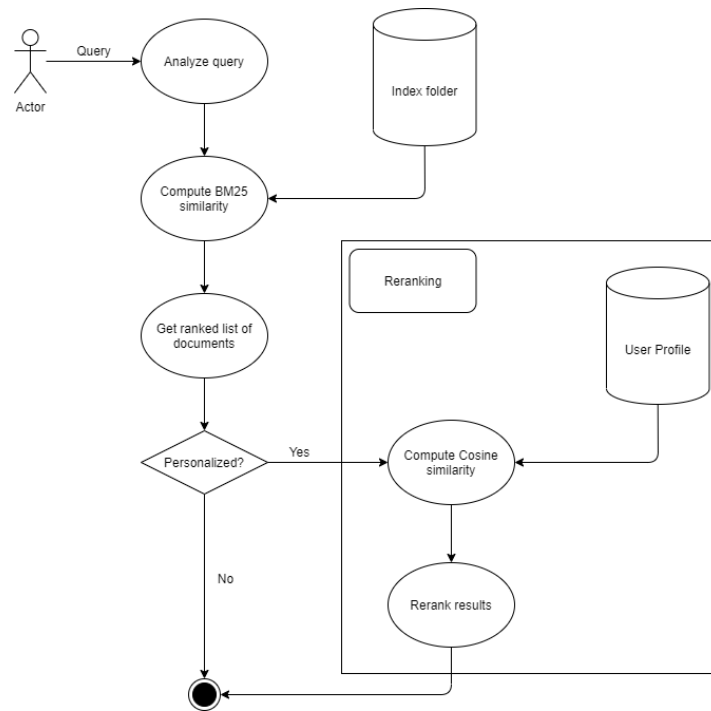


Figure 5: Steps to generate query results

4 Demonstration plan

The implemented search engine runs on a Tomcat server and can be accessed using any browser at `http://localhost:8080`. It can be started by executing its .jar file.

The system is provided with a very simple yet effective GUI, which users can interact with. The GUI is shown in Figure 6.

Search Engine for Twitter posts



Select the field on which you want to search:

Text

Type your query here:

Flag this checkbox if you want to use synonyms: ☐

Type the weight for the user profile: 0.42

Search

Figure 6: GUI provided by the search engine

4.1 Textual search on a specific field using keywords

In order to process basic searches using keywords on a specific field, users must select the field from the menu on which they want to perform the search. For example, if users want to find all the tweets containing both the words “game” and “final” in their text, they have to select the *text* option and write *game AND final*. The search engine applies the same preprocessing steps applied by the indexing component and computes the BM25 similarity between the query and all the documents satisfying the query. The top 10 documents are then returned to the users. Figure 7 shows the results of the aforementioned query. As expected, all the returned documents contain both the query terms .

No personalization			
Score	Created at	Author	Text
1,00	2020/09/18 02:03	NHL	Game 6, Eastern Conference Final. Who's taking it? #GoBolts or #Isles https://t.co/n0iJEkmxF0
0,98	2020/09/19 19:35	NHL	🔊 EXTRA EXTRA Please enjoy this virtual #StanleyCup Final game program! https://t.co/TIFsOfAcGV
0,94	2020/09/20 01:10	NHL	Game 6 of the Eastern Conference Final was a battle and it's the reason why the @TBLightning are here. Watch Game 1 of the #StanleyCup Final at 7:50 ET on @NHLonNBCSports, @Sportsnet and @TVASports. https://t.co/rHcqGUtXC3x
0,93	2020/10/03 03:05	NHL	The best mic'd up sound from Game 6 of the #StanleyCup Final RIGHT HERE. https://t.co/prPMUGQd69
0,91	2020/10/02 02:06	NHL	From the 5OT game to the #StanleyCup Final, this postseason was one we'll never forget. https://t.co/2DTQjpXdaT
0,91	2020/09/22 15:36	NHL	The @TBLightning hang on to win Game 2, evening the #StanleyCup Final at 1-1. https://t.co/A2DERq5P1g
0,91	2020/09/20 13:09	NHL	ICYMI: @madisonbeer sang the U.S. National Anthem prior to Game 1 of the #StanleyCup Final. https://t.co/AnSZOvgwrs
0,91	2020/09/19 18:46	NHL	All aboard! The #StanleyCup Final begins tonight. Watch every game on @NHLonNBCSports, @Sportsnet and @TVASports. https://t.co/Fdbl16AeB8
0,91	2020/09/15 15:14	NHL	Game. Set. #StanleyCup Final. Take a bow, Denis Gurianov! 🏒🏒 https://t.co/7I7sqROdqQ
0,91	2020/09/14 15:35	NHL	The @TBLightning take Game 4 and are now one win away from the #StanleyCup Final! https://t.co/hSCiuxsDhJ

Figure 7: Results obtained in response to the query of the first use case

4.2 Textual search on a combination of fields

To process basic searches on multiple fields, users must select the *custom* option from the menu and type their query on the search bar. For example, to find all the documents containing the “#KiaTipOff20” hashtag, not citing “@celtics”, containing either “point” or “rebound” and published from 05/12/2020 at 00:00 to 16/12/2020 at 23:59, users must type the following query: *hashtags:#KiaTipOff20 AND created_at:[“2020/12/05 00:00” TO “2020/12/16 23:59”] AND NOT citations:@celtics AND (full_text:point OR full_text:rebound)*. The search engine parses and evaluates the query considering different fields and query types (i.e., range queries). Each field is treated in the same way it was treated by the indexing component. The matching mechanism is the same as the one described in Subsection 4.1. Figure 8 shows the results of the custom query.


No personalization			
Score	Created at	Author	Text
1,00	2020/12/15 22:00	NBA	We flashback to @BenSimmons25's 18-point, 10-rebound double-double in his debut with the @sixers in 2017! The 2020-21 NBA season tips off Christmas Week with games beginning Tuesday, December 22nd. #KiaTipOff20 https://t.co/8BIfWOwzg9
0,90	2020/12/05 23:30	NBA	Rebound. Defend. Knock down open shots. Rookie Kawhi kept it simple. The 2020-21 NBA Season Starts Christmas Week with Games Beginning Tuesday, December 22. #KiaTipOff20 https://t.co/m8AWWlr5Hj
0,57	2020/12/14 22:40	NBA	 @Zionwilliamson drops 17 straight points, 22 total in his unforgettable debut! The 2020-21 NBA season tips off Christmas Week with games beginning Tuesday, December 22nd. #KiaTipOff20 https://t.co/GLOBiDCdwp
0,57	2020/12/14 21:50	NBA	Look back at @JoelEmbiid's 20-point NBA debut with the @sixers in 2016! The 2020-21 NBA season tips off Christmas Week with games beginning Tuesday, December 22nd. #KiaTipOff20 https://t.co/87RWtTsoSA

Figure 8: Results obtained in response to the query of the second use case

4.3 Rank the tweets taking into account the user profile

When the user profile is taken into account, the search engine works in the same way as the previous use cases, but it also performs a re-ranking procedure that modifies the initial ranked list of documents in such a way that also the similarity score computed between the documents and the user models is considered.

In particular, the search engine first computes the BM25 similarity between the query and the documents, producing an initial ranked list of documents. Then, each document of this list is compared with the user model using the cosine similarity. The scores of these similarities are then linearly combined to compute the final score and, consequently, the final ranked list of documents.

The similarity between a user model and a document is computed as follows: first, the vector representation of the document is generated by considering just the frequencies of the terms that also appear in the user model, then the cosine similarity between the document and the user model vectors is computed.

Since the cosine similarity spans the $[0, 1]$ interval, the BM25 similarity score of each document has been normalized by dividing it by the highest BM25 similarity score obtained. Then, the final score is obtained by linearly combining the normalized BM25 similarity score with the cosine similarity, where the importance of both the similarities is controlled by a parameter α that takes values in $[0, 1]$, according to Equation 1. Figure 9 shows the results obtained by submitting the query *play OR kid* considering the *full_text* field. As shown, depending on the user, the ranked list of documents is different. The results in the example have been obtained using $\alpha = 0.42$. If $\alpha = 0$, as already mentioned in Section 3, the ranked list is the same for all the users, while if $\alpha = 1$, then, since the query keywords are more related to American sport leagues than spatial agencies, the scores assigned to the retrieved documents is much higher for NBA, MLB and NHL.

No personalization				User profile: MLB				User profile: esa			
Score	Created at	Author	Text	Score	Created at	Author	Text	Score	Created at	Author	Text
1,00	2020/11/30 01:39	MLB	Who was your favorite player when you were a kid?	0,60	2020/11/30 01:39	MLB	Who was your favorite player when you were a kid?	0,58	2020/11/30 01:39	MLB	Who was your favorite player when you were a kid?
0,52	2020/10/07 19:12	NBA	“Just a kid from Akron” 🍌 LeBron James is featured on his first @wheaties box cover! https://t.co/muK7cPPYgj	0,49	2020/09/15 16:50	MLB	“This kid is a mini me with better stuff.” - @45PedroMartinez on Sixto Sánchez 🍷 (MLB x @Woodbridge_Wine) https://t.co/AIqwZqENHw	0,36	2020/10/07 19:12	NBA	“Just a kid from Akron” 🍌 LeBron James is featured on his first @wheaties box cover! https://t.co/muK7cPPYgj
0,51	2020/09/15 16:50	MLB	“This kid is a mini me with better stuff.” - @45PedroMartinez on Sixto Sánchez 🍷 (MLB x @Woodbridge_Wine) https://t.co/AIqwZqENHw	0,41	2020/11/12 03:30	NBA	NBA Draft prospect @niccolomannion discusses being around the game as a kid while his father played professionally. 2020 #NBADraft: Wednesday, November 18 8:00pm/et, ESPN https://t.co/8fQuXB3Y6c	0,36	2020/12/12 17:25	MLB	Tell us your favorite player without telling us your favorite player.

Figure 9: Results obtained in response to the query of the third use case

4.4 Expand the search adding synonyms of words in the query

In this scenario, users specify a query and flag the option to add synonyms and the system processes the query terms exactly as it is done in the other use cases, but it also expands the query by adding, for each keyword, its synonyms, according to WordNet.

For example, if users specify the query *full_text:objective*, then the search engine expands the query with the keywords “accusative”, “aim”, “documentary”, “nosubjective”, “object” and “target”. Figure 10 shows the results of the query taking into account also the synonyms. As shown, the first document does not contain the keyword “objective”, but it contains the word “target”. Furthermore, the same situation happens for the second and the third document, which contain “documentary” and “target”, respectively.

No personalization			
Score	Created at	Author	Text
1.00	2020/12/15 19:35	NHL	Target acquired. 🇨🇦 Eric Staal was shooting heaters this time last year. https://t.co/aig3NSlyUj
0.87	2020/11/28 23:57	esa	.@2020spacebeyond, the documentary about @astro_luca's #MissionBeyond, is finally out. Preview on @SKYArte 28 November and streaming on @NowTVIt. https://t.co/nFDjjn0eSe
0.78	2020/09/24 18:25	NASA	Today's scheduled launch of a @BlueOrigin suborbital rocket with @NASA_Technology payloads has been scrubbed, with a new target date forthcoming. About the flight test: https://t.co/9waZgoWcOC https://t.co/4Czy9tXWkQ

Figure 10: Results obtained in response to the query of the forth use case

4.5 Additional functionalities

An additional functionality provided by the system is the possibility to submit range queries considering the *created_at* field, allowing to find tweets that have been published in a given period. In order to do this, users have to type a query like: *[“2020/11/30 00:00” TO “2020/12/02 23:59”]*. Note that this field has been created just to handle range queries.

The system is also capable of managing phrase queries. In this way, users can search for tweets that contain a specific term sequence just by surrounding their queries with double quotes. For example, *full_text:“Who was your favorite player”*.

Finally, it is possible to use social media items like emojis inside the queries because they have been indexed.