Personalized Search Engine for microblog

Information Retrieval project

Christian Bernasconi - 816423

Marco Ripamonti - 806785

Round 25/02/2021

Project Overview

- Search Engine for microblog: Elasticsearch
  - Tweets
  - Users

- Search modalities
  - Basic
  - Ranked by popularity
  - Personalised by user

- Demo application

Dataset Creation - Tweets

Tweepy has been used to collect 7200 tweets with the following fields:

- Content
- Username
- Hashtags
- Retweet count
- Favorite count

Politics

Science

Joe Biden

Neil deGrasse Tyson

Bernie Sanders

Alexandria Ocasio-Cortez

Brian Greene

Dr. Michio Kaku

Preprocessing - Issues

- Hashtags
- Mentions
- Emojis

Example:

#hashtag

“ This is a tweet example which is related to politicians arguing about Covid-19 with an #hashtag and a @mention! 🤝 “

Preprocessing - Steps Analyzer

1. **Chars filtering:**
   a. ’#’ replacement with ’ symbolhash_ ’
   b. ’@’ replacement with ’ symbolat_ ’
   c. strip HTML entities and replacement with decoded values

Preprocessing - Steps Analyzer

1. **Chars filtering:**
   a. ’#’ replacement with ’ symbolhash_ ’
   b. ’@’ replacement with ’ symbolat_ ’
   c. strip HTML entities and replacement with decoded values
2. **Tokenization:** standard tokenizer [1]

[1] based on the Unicode Text Segmentation algorithm, as specified in Unicode Standard Annex

Preprocessing - Steps Analyzer

1. **Chars filtering:**
   a. ’#’ replacement with ’ symbolhash_ ’
   b. ’@’ replacement with ’ symbolat_ ’
   c. strip HTML entities and replacement with decoded values
2. **Tokenization:** standard tokenizer
3. **Tokens filtering:**
   a. ASCII folding
   b. Lowercase
   c. Stemming of possessive forms
   d. Removal of special character for emojis
   e. Synonyms expansion of english emojis
   f. Removal of english stopwords
   g. Expand hashtags and mentions with relative texts

Preprocessing - Steps Analyzer

1. **Chars filtering:**
   a. ’#’ replacement with ’ symbolhash_ ’
   b. ’@’ replacement with ’ symbolat_ ’
   c. strip HTML entities and replacement with decoded values
2. **Tokenization:** standard tokenizer
3. **Tokens filtering:**
   a. ASCII folding
   b. Lowercase
   c. Stemming of possessive forms
   d. Removal of special character for emojis
   e. **Synonyms expansion of english emojis**
   f. Removal of english stopwords
   g. Expand hashtags and mentions with relative texts

Preprocessing - Emojis expansion

Examples:

😂 → 😂, face, face with tears of joy, joy, laugh, tear

👍 → 👍, +1, hand, thumb, thumbs up, up

🧐 → 🧐, face with monocle, stuffy

[2] https://github.com/jolicode/emoji-search

Preprocessing - Steps Analyzer

Original tweet

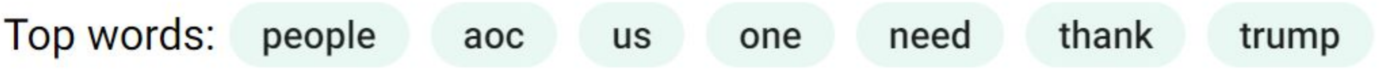
“ This is a tweet example with an #hashtag and a @mention! 🤝 *“*

Processed tweet

“tweet example symbolhash_hashtag hashtag symbolat_mention mention 🤝 *agreement hand handshake meeting shake”*

User profile

A user profile has been automatically extracted from tweets of each user with the following fields:

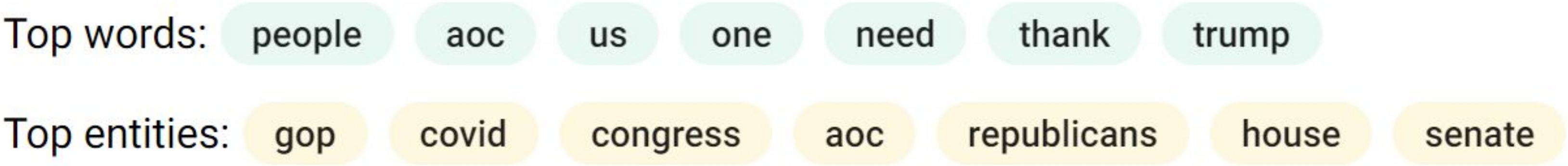


Top words: people aoc us one need thank trump

- Computed by words frequencies
- Top 7 most frequent words

User profile

A user profile has been automatically extracted from tweets of each user with the following fields:



Top words: people aoc us one need thank trump

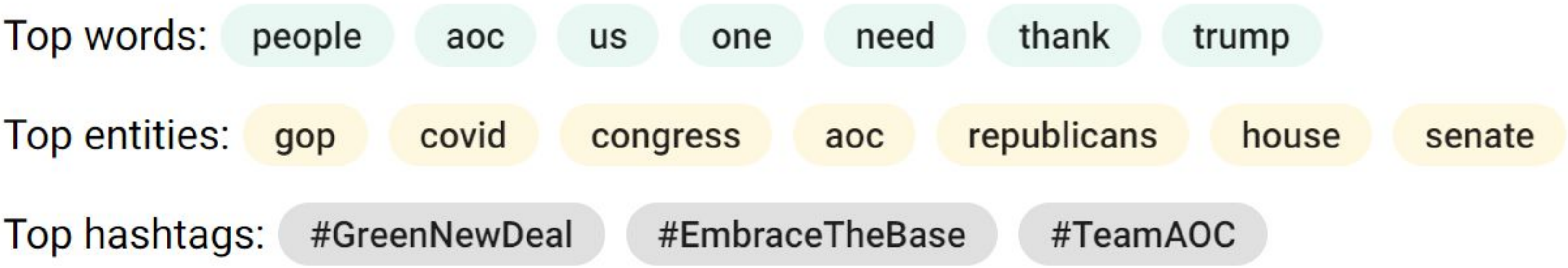
Top entities: gop covid congress aoc republicans house senate

- Extracted with spaCy’s NER [2]
- Text preprocessed removing ‘#’, ‘@’, links and ‘RT’
- Top 7 most frequent entities

[3] https://spacy.io/usage/linguistic-features#named-entities

User profile

A user profile has been automatically extracted from tweets of each user with the following fields:



Top words: people aoc us one need thank trump

Top entities: gop covid congress aoc republicans house senate

Top hashtags: #GreenNewDeal #EmbraceTheBase #TeamAOC

- Computed by hashtags frequencies
- Top 3 most frequent entities

Search Engine and Indexing

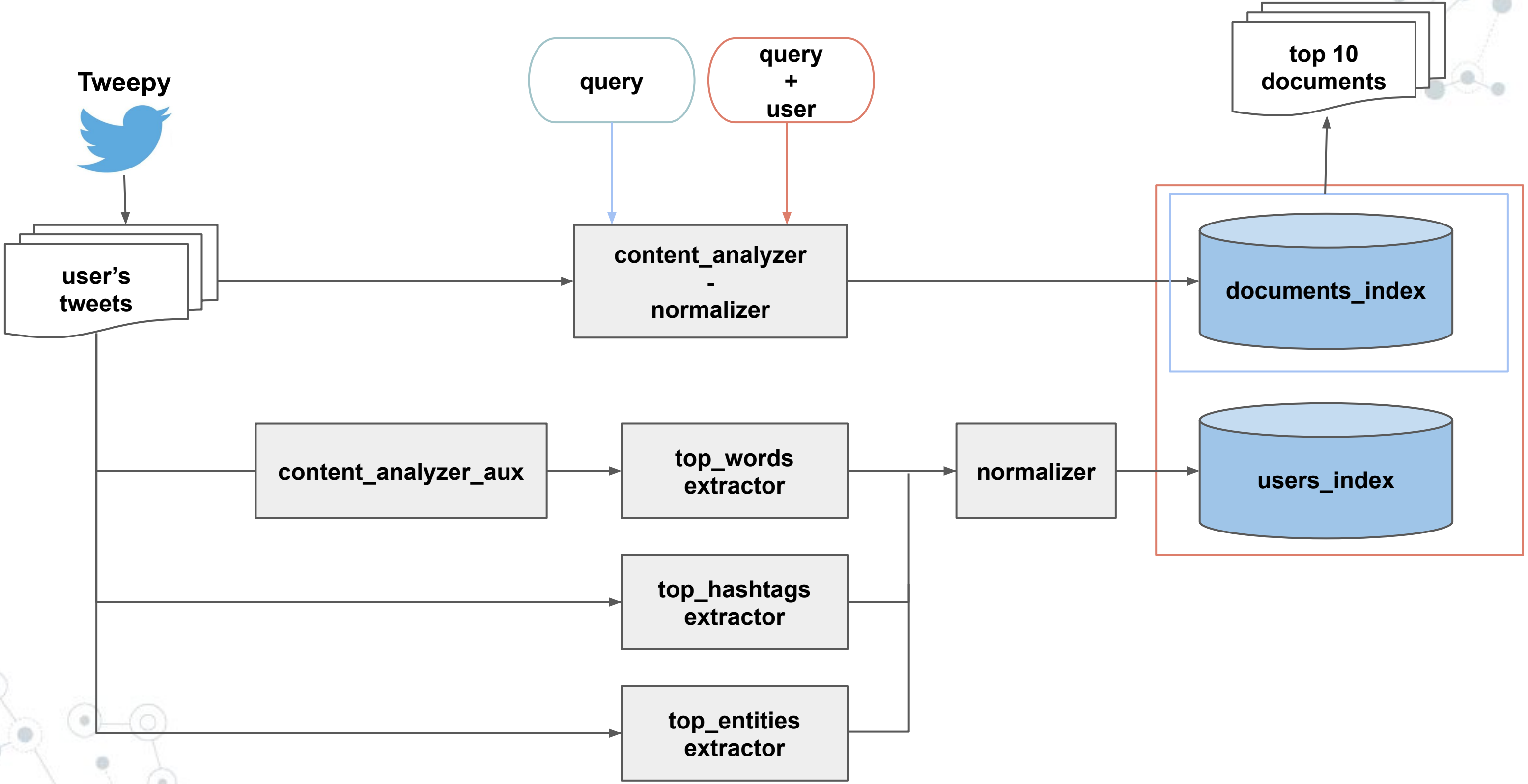
The **model** on which the search engine relies on is **BM25** which is the vector space model used by default by Elasticsearch.

Documents index:

- username: *keyword*
- content: *text*
- hashtags: *keyword*
- favorite_count: *long*
- retweet_count: *long*
- favorite_count_rf: *rank_feature*
- retweet_count_rf: *rank_feature*

$$\log_2(x + 1) + 0.001$$

System pipeline



Search Modalities

- **Basic search:**
  textual fuzzy search on tweets *content*

Search Modalities

- **Basic search:**
  textual fuzzy search on tweets *content*
- **Search by popularity:**
  textual search on tweets *content* favoring popular tweets (i.e.: retweets, favorites) giving more importance to the number of retweets

Search Modalities

- **Basic search:**
  textual fuzzy search on tweets *content*
- **Search by popularity:**
  textual search on tweets *content* favoring popular tweets (i.e.: retweets, favorites) giving more importance to the number of retweets
- **Search by users top words preference:**
  Textual search on tweets *content* with a preprocessing strategy of personalization (i.e.: expanding queries with top words and top entities)

Search Modalities

- **Basic search:**
  textual fuzzy search on tweets *content*
- **Search by popularity:**
  textual search on tweets *content* favoring popular tweets (i.e.: retweets, favorites) giving more importance to the number of retweets
- **Search by users top words preference:**
  Textual search on tweets *content* with a preprocessing strategy of personalization (i.e.: expanding queries with top words and top entities)
- **Search by users top hashtags preference:**
  Textual search on tweets *content* with a preprocessing strategy of personalization (i.e.: taking into account tweets *hashtags* with different boost levels according to the rank)

Search Modalities

- **Basic search:**
  textual fuzzy search on tweets *content*
- **Search by popularity:**
  textual search on tweets *content* favoring popular tweets (i.e.: retweets, favorites) giving more importance to the number of retweets
- **Search by users top words preference:**
  Textual search on tweets *content* with a preprocessing strategy of personalization (i.e.: expanding queries with top words and top entities)
- **Search by users top hashtags preference:**
  Textual search on tweets *content* with a preprocessing strategy of personalization (i.e.: taking into account tweets *hashtags* with different boost levels according to the rank)

- Additionally for each query is possible to filter out tweets of a specified user.

Demo

Deployed on Github at the following link

Thanks for the attention!