

UNIVERSITÀ DEGLI STUDI DI
MILANO-BICOCCA

INFORMATION RETRIEVAL
PROJECT A

Personalized Search Engine for microblog

Authors:

Christian Bernasconi - 816423 -
c.bernasconi11@campus.unimib.it

Marco Ripamonti - 806785 -
m.ripamonti@campus.unimib.it

16/02/2021



1 Introduction

In this project a search engine has been built with the usage of Elasticsearch to implement basic and personalized documents retrievals. A document repository has been created by collecting tweets about politics and science from 6 different users. To achieve personalization in the retrieval process, users profiles are automatically generated from the content of the tweets published by them.

Starting from basic fuzzy queries, the implemented search modalities allow to query the dataset with various settings. For instance retrieval can be influenced by popularity (*i.e.*, *retweets and favorites count*) or by taking into account users profiles (*i.e.*, *user's most used words and hashtags*).

Python has been used to collect tweets, extract users and create indexes.

Finally, an Angular demo application has been developed to show the results obtained with the different retrieval modes of the implemented search engine. The demo is set to be publicly available on GitHub at the following link <https://marcoripa96.github.io/>¹ with an Elasticsearch cluster instantiated on the cloud.

2 Dataset

A document repository has been built with 7800 tweets from six different users gathered using the Tweepy² library. Exactly 1300 tweets per users, starting from the 25/01/2021, have been downloaded. Users have been selected with the intent of generating two possible topics in the documents repository: *Joe Biden*, *Bernie Sanders* and *Alexandra Ocasio-Cortez* for the topic of politics and *Neil deGrasse Tyson*, *Brian Greene* and *Michio Kaku* for the topic of science (mainly Theoretical Physics and Astrophysics). In figures 1 and 2 two word clouds highlight the most common words used in those two topics. The longest tweet has 75 words while the shortest has 1 word. The average number of words between all documents is 28.

¹The demo has been tested on Google Chrome browser. The use of other browsers may introduce bugs.

²<https://www.tweepy.org/>

A user archive has also been defined to store the user profile with the following information:

- Username: username of the user
- Top Words: top 7 most common words used in his tweets
- Top Entities: top 7 most common entities extracted from his tweets
- Top Hashtags: top 3 most common hashtags used in his tweets

Those documents are stored to achieve personalization during search time, a detailed explanation will be presented in section 3. In figure 3 a diagram represents the structure of the documents as just described.

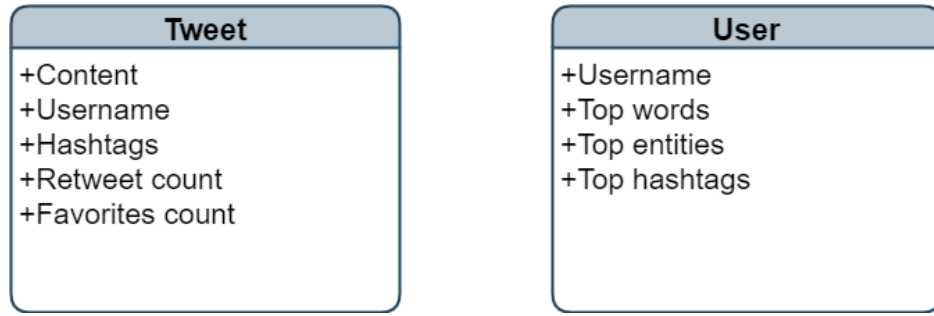


Figure 3: Entities describing documents in repository

3 Search Engine

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

Following the documents structure previously illustrated, tweets are then indexed in the *documents_index* with respect to the mappings listed below:

- **username:** *keyword*
- **content:** *text*
- **hashtags:** *keyword*
- **retweet_count:** *long*

- **favorite_count:** *long*
- **retweet_count_rf:** *rank_feature*; a log transformation³ has been performed in order to use *retweet_count* to rank queries results
- **favorite_count_rf:** *rank_feature*; a log transformation has been performed in order to use *favorite_count* to rank queries results

While *username* and *hashtags* fields are preprocessed only with a lowercase normalization, the *content* of a tweet and queries are involved in the following steps according to the custom made *content_analyzer*:

1. **chars filtering:**

- (a) '#' replacement with 'symbolhash_'
- (b) '@' replacement with 'symbolat_'
- (c) strip HTML entities and replacement with decoded values

2. **tokenization:** standard tokenizer⁴

3. **tokens filtering:**

- (a) ASCII folding converts characters that are not in the Basic Latin Unicode block to their ASCII equivalent.
- (b) stemming of english possessive forms
- (c) remove disturbing special characters following emojis that could negatively impact matches
- (d) synonyms expansion of english emojis using an auxiliary file⁵ to map emojis with their meanings
- (e) remove english stopwords
- (f) expand hashtags and mentions with relative texts

³ $\log(x+1) + 0.001$ to shrink the original value and ensure positiveness of *retweet_count*

⁴<https://unicode.org/reports/tr29/>

⁵<https://github.com/jolicode/emoji-search/blob/master/synonyms/cldr-emoji-annotation-synonyms-en.txt>

An example of how the *content_analyzer* works is shown in figure 4.



Figure 4: Example of a tweet processed with *content_analyzer*

Users profiles, used to personalize retrieval results, are extracted considering all the tweets of a user and are stored in the *users_index* with the fields shown in the list below with a brief explanation of how they have been computed:

- **username:** *keyword*
- **top_words:** *keyword*; extracted by counting words frequencies after a text preprocessing performed with *content_analyzer_aux*⁶
- **top_entities:** *keyword*; extracted with spaCy's⁷ *en_core_web_sm* model after a text preprocessing constituted by the removal of '#', '@', links and retweets abbreviation
- **top_hashtags:** *keyword*; extracted by frequencies

All the fields above are preprocessed only with a lowercase transformation.

The entire data flow of the system is illustrated in figure 5. Please notice that while standard queries involve only the *documents_index*, user based queries use also the *users_index* to get users preferences.

⁶same as *content_analyzer*, except steps 1.a, 1.b and 3.d

⁷<https://spacy.io/>

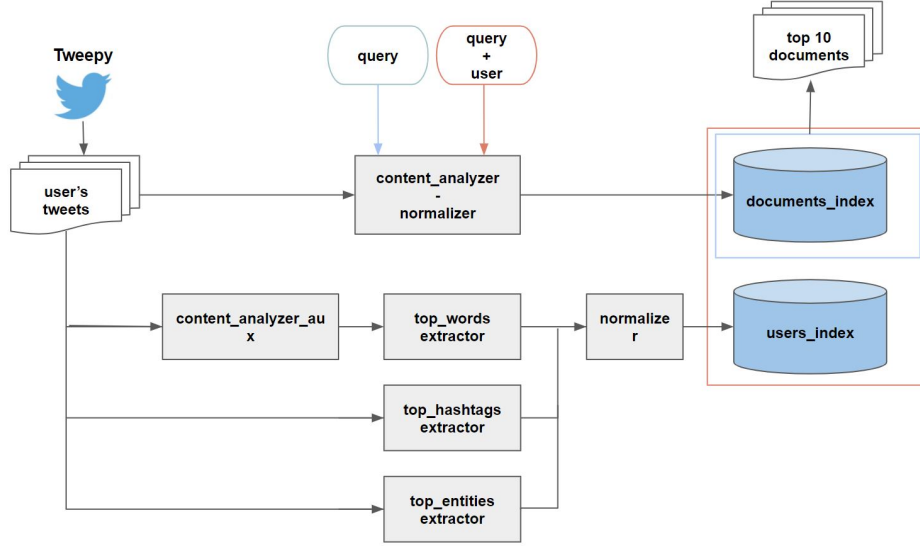


Figure 5: Data flow of the process of index creation and query execution

4 Demonstration plan

For this project four different functionalities have been implemented to cover all the requirements:

1. **search:** basic textual search on *content* field with a fuzzy match
2. **searchByPopularity:** textual search on *content* field with a fuzzy match; takes into account *favorite_count_rf* and *retweet_count_rf* to boost scores and retrieve more popular tweets; while the boost of the query match has been decreased, *retweet_count_rf* has been considered more significant than *favorite_count_rf*, so a higher boost has been set for it
3. **searchWordsPreference:** textual search on *content* field with a fuzzy match; takes into account the union of the user's *top_words* and *top_entities* to boost scores with an additional exact match on *content*
4. **searchHashtagPreference:** textual search on *content* field with a fuzzy match; takes into account user's *top_hashtags* to boost scores with a term match on the *hashtag* field; each user's hashtag has a boost

value which depends on his rank of the top 3 most frequent hashtags (*i.e., the 1st most used hashtag has a higher boost than the 2nd and the 3rd*)

All the above functionalities retrieve the top 10 more relevant documents. Each functionality also allows restricting results to a single user with an optional filter that is enabled when a user is selected. Scores are not influenced by the user selection. Emoji's expansion is done by default for each query. Finally, the matching positions of both query terms and user's profile top words and entities are returned for each query to highlight them in the original tweets.

All the above functionalities has been implemented in a demo developed using Angular and Elasticsearch-js. A Elasticsearch cluster has been instantiated on the cloud and the demo has been deployed on GitHub at the following link <https://marcoripa96.github.io/>.

In the next sections different examples of retrievals will be presented. For each example the top 5 retrieved documents are shown.

Basic search

In figure 6 is shown an example of basic query with fuzzy matching of terms. From this example matches are highlighted and in particular you can notice some fuzzy matches like *"thing"* and *"starting"* with the query term *"string"*. Another aspect to notice is that even though the second most relevant document returned by the query has more matches, the relevance score is similar to the most relevant document because with BM25 a match has a higher impact when occurring in shorter documents.

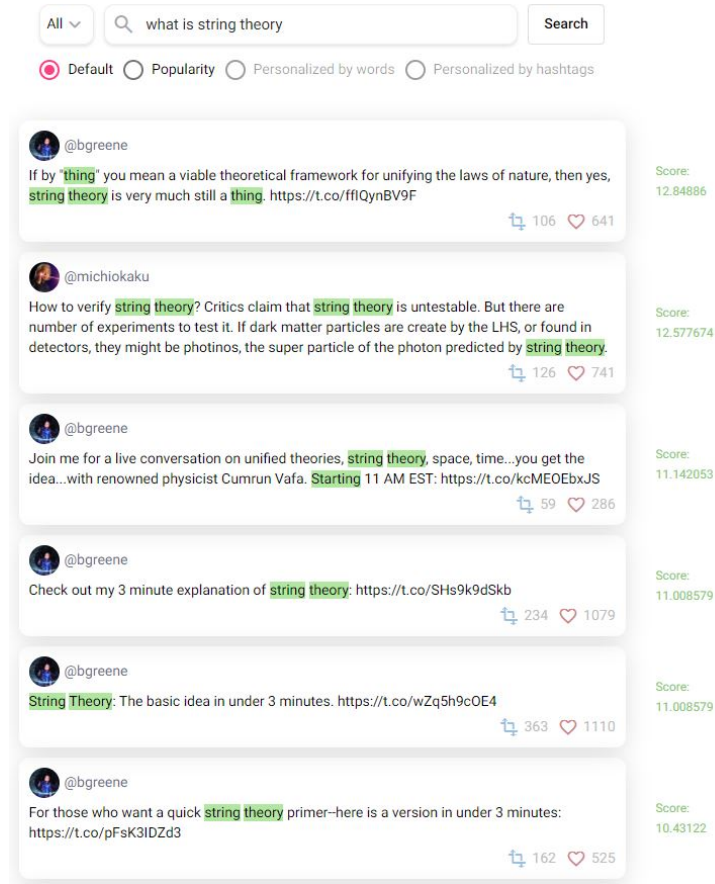


Figure 6: Basic fuzzy search

Query expansion of emojis

In figure 7 is shown an example of a query that includes emojis. As you can see, thanks to the query expansion mechanism, the search engine retrieves the same set of relevant documents for both textual and emoji queries.

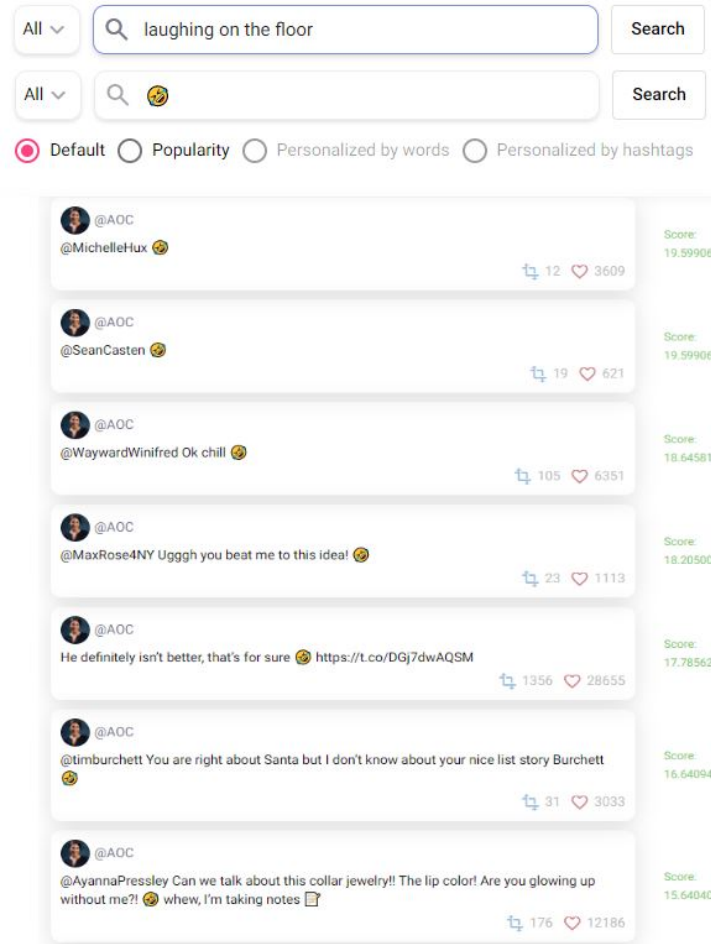


Figure 7: Emoji query expansion search

Search by popularity

The figures 9 shows an example of a query that takes into account tweets popularity. Comparing with the results shown in figure 8 you can see how a basic search with the same query terms produced different rankings. For example, the 5th more relevant document retrieved by the basic search is boosted to the 2nd place by the popularity. The 1st tweet of basic search, instead, decreases by two positions.

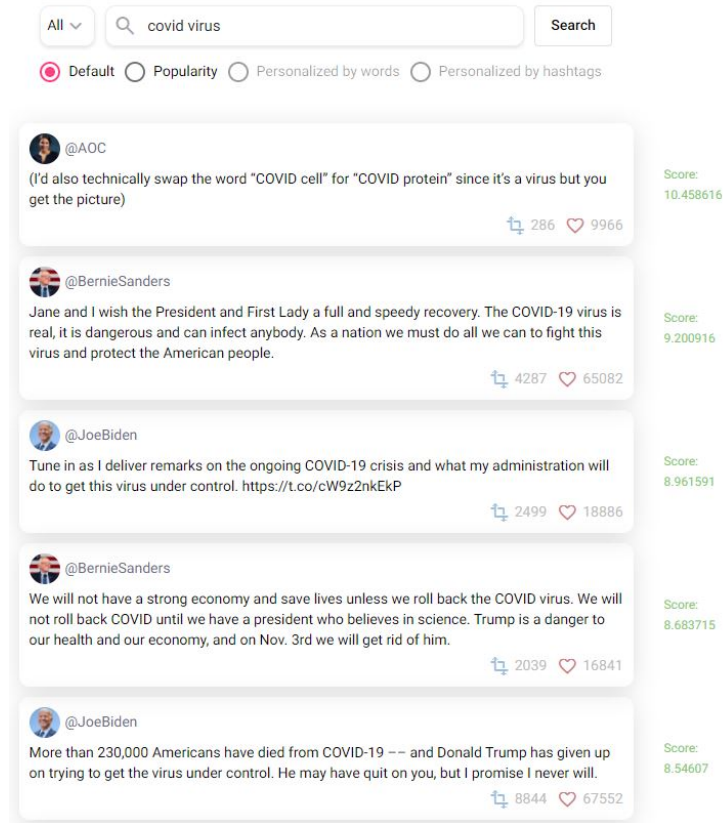


Figure 8: Basic search - example popularity

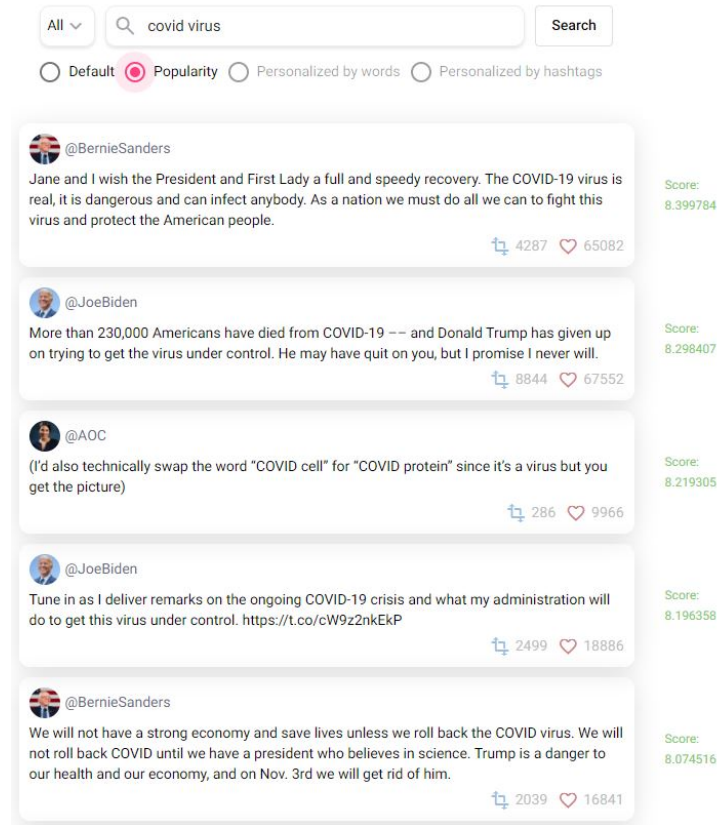


Figure 9: Search influenced by popularity

Personalized search - Top words

To show how the retrieved documents are affected by users profiles in a personalized search that uses top words, the same query has been performed for Brian Greene and Alexandria Ocasio-Cortez. Looking at the highlighted terms in figures 10 and 11 you can notice that documents retrieved are respectively about science and politics according to the users interests.

User profile

Top words: otd time quantum born science one universe

Top entities: otd einstein quantum mechanics america light falls quantum nyc

Top hashtags: #OTD #WSF18 #Eclipse2017

All Search

☐ Default ☐ Popularity ☒ Personalized by words ☐ Personalized by hashtags

@bgreene

Marie Curie, born OTD 1867, first woman to win the Nobel Prize. Kamala Harris, OTD first woman vice president-elect of United States. A great day for women, science, America, and the future of the planet.

262 2110

Score: 11.233868

@bgreene

In quantum mechanics, we often say that a particle can be two places at once. It is more precise to say that there are situations when a particle may not have a location at all. Which is deeply weird. <https://t.co/dbPpS8mAs5>

493 2422

Score: 10.872695

@bgreene

In quantum mechanics, we often say that a particle can be two places at once. More precise to say that there are situations when a particle may not have a location at all. Which is deeply weird. <https://t.co/w9y0aZ8pTo>

440 1943

Score: 10.872695

@michiokaku

Great news! A planet with liquid oceans was discovered. This is something straight out of science fiction. Only 22 light years away.

894 148

Score: 8.751744

@bgreene

"Our planet is a lonely speck in the great enveloping cosmic dark...There is no hint that help will come from elsewhere to save us from ourselves."-Carl Sagan, born OTD 1934 <https://t.co/wwKhyPeExY>

763 1717

Score: 8.028038

Figure 10: Personalized search by top words - Brian Greene

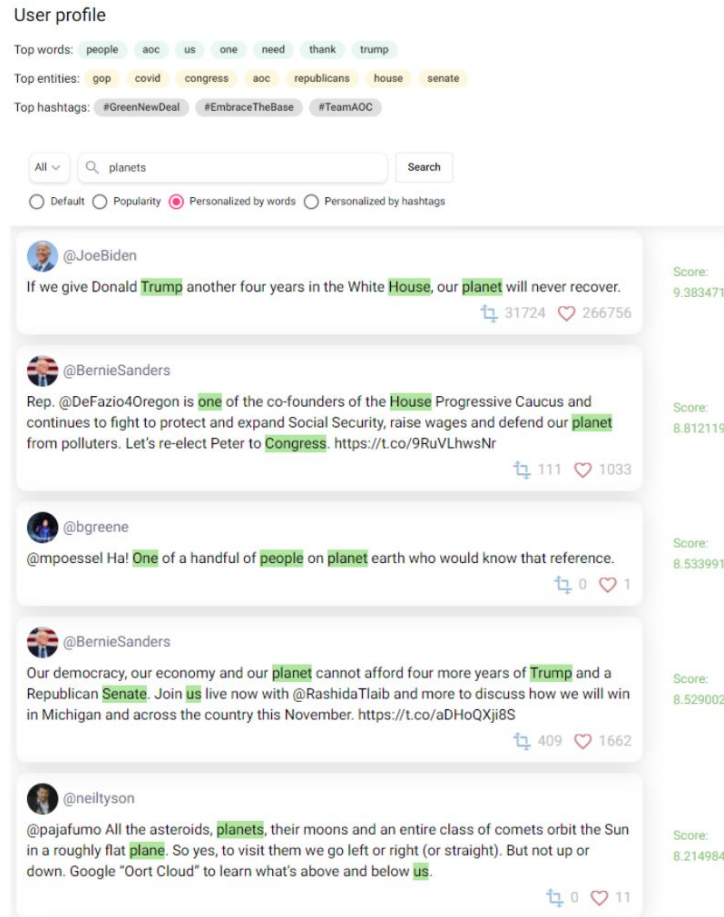


Figure 11: Personalized search by top words - AOC

Personalized search - Top hashtags

In figures 12 and 13 you can see how most used hashtags of a user influence a query. The example shows that '*#GreenNewDeal*' hashtag from Alexandria Ocasio-Cortez's profile boosts the relevance of the document which contains it, producing a score which ranks it to the 1st position. Meanwhile, the same tweet doesn't even show in the documents retrieved by the basic search. In general, not having a large dataset makes finding useful examples of this type harder. The collected data does in fact have low frequency of hashtags, which makes it more difficult to discover hashtags shared between the users.

User profile

Top words: people aoc us one need thank trump

Top entities: gop covid congress aoc republicans house senate

Top hashtags: #GreenNewDeal #EmbraceTheBase #TeamAOC

All ▾

🔍 president election

Search


☒ Default ☐ Popularity ☐ Personalized by words ☐ Personalized by hashtags

 @BernieSanders

No, Mr. **President**. We're not delaying the **election**. The American people are sick and tired of your authoritarianism, your lies, your racism. On November 3, 2020 democracy will prevail and your disastrous **presidency** will end. Bye-bye. <https://t.co/wmBYriSxGs>

Score:
7.4739733


🔗 47986 ❤️ 264972

 @BernieSanders

Yes, this is the most important **election** in history. And you are the most dangerous **president** in the history of our country. That's why you're going to lose. #PromisesBroken

Score:
7.457851

🔗 1495 ❤️ 10242

 @BernieSanders

This is not just an **election** between Trump and Biden. This **election** is about democracy vs. authoritarianism – and democracy must win. The first way to do that is to get out and vote in overwhelming numbers and defeat Trump – the most dangerous **president** in American history.

Score:
7.0130744


🔗 19549 ❤️ 107042

 @BernieSanders

No, Mr. **President**. You're not going to postpone the **election**. You're not going to dismantle the Postal Service. You're not going to undermine the vote-by-mail system. This **election** will have the largest voter turnout in history, and you're going to lose. <https://t.co/8FxiYXu6CG>

Score:
6.78706

🔗 3121 ❤️ 16784

 @BernieSanders

This **election** is about immigration reform and whether or not we continue to have a **president**, who in a disgraceful and racist manner, demonizes the Latino community. <https://t.co/9V0jX22CoG>

Score:
6.7498417

🔗 256 ❤️ 1487

Figure 12: Basic search - example top hashtags

User profile

Top words: people aoc us one need thank trump

Top entities: gop covid congress aoc republicans house senate

Top hashtags: #GreenNewDeal #EmbraceTheBase #TeamAOC

All

president election


Search

☐ Default

☐ Popularity

☐ Personalized by words


☒ Personalized by hashtags

 @BernieSanders

California is on fire. Once in a lifetime hurricanes are sweeping the Gulf Coast. The Arctic topped 100 degrees. The future of our planet is at stake this election. We can no longer have a president in office who rejects science and calls climate change a hoax. #GreenNewDeal

Score: 17.084469


26089 110016

 @BernieSanders

No, Mr. President. We're not delaying the election. The American people are sick and tired of your authoritarianism, your lies, your racism. On November 3, 2020 democracy will prevail and your disastrous presidency will end. Bye-bye. <https://t.co/wmBYriSxGs>

Score: 4.484384


47986 264972

 @BernieSanders

Yes, this is the most important election in history. And you are the most dangerous president in the history of our country. That's why you're going to lose. #PromisesBroken

Score: 4.4747105


1495 10242

 @BernieSanders

This is not just an election between Trump and Biden. This election is about democracy vs. authoritarianism – and democracy must win. The first way to do that is to get out and vote in overwhelming numbers and defeat Trump – the most dangerous president in American history.

Score: 4.2078447

19549 107042

 @BernieSanders

No, Mr. President. You're not going to postpone the election. You're not going to dismantle the Postal Service. You're not going to undermine the vote-by-mail system. This election will have the largest voter turnout in history, and you're going to lose. <https://t.co/8FxiYXu6CG>

Score: 4.072236

3121 16784

Figure 13: Personalized search by top hashtags - AOC