Guillem Morgo 204564
Joan Medina 204561

FINAL PROJECT
REPORT

SECTION 1: Data collection

1.Statistics

The tweet with more likes are 600916
The tweet with more retweet are 104892
Number of tweets: 215664
Number of users: 19878

2.Keywords

the key words are:
"#covid19", "#Covid19", "#COVID19", "#covid", "#coronavirus", "#pcr", "#Covid-19", "#quarantine", "confinement", "Covid19", "coronavirus", "COVID19", "PCR", "Covid-19", "#Covid-19", "Coronavirus", "Antibody", "Social distancing", "epidemic", "pandemic", "Asymptomatic", "vaccine"

3.Time

The approximate time needed to collect the data has been around 2 hours.

Guillem Morgo 204564
Joan Medina 204561

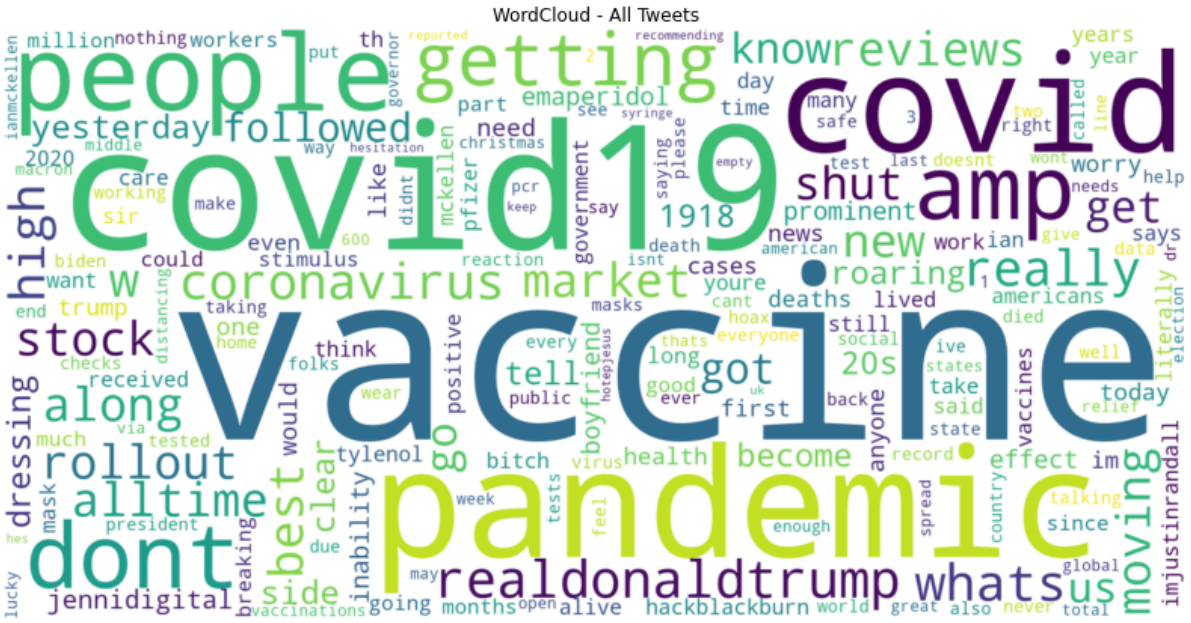
SECTION 2: Search engine

1.Preprocessing

For cleaning the data we apply the following functions:

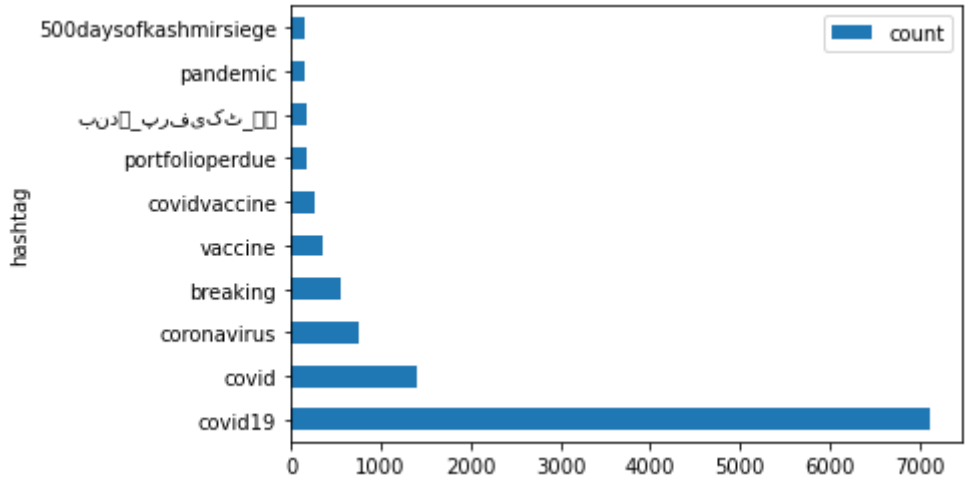
```
text = remove_links(text)
text = remove_hashtags(text)
text = text_to_lower_case(text)
text = remove_text_marks(text)
text = remove_accents(text)
text = remove_emojis(text)
text = split_text_and_numbers(text)
text = remove_alone_numbers(text)
text = remove_multiple_whitespaces(text)
text = remove_punctuation_marks(text)
text = remove_stopwords(text)
text = stemming(text)
```

2.Wordcloud



Guillem Morgo 204564
Joan Medina 204561

3.Barplot



4.Ranking

Our developed ranking consists on making a tf-idf score but instead of returning the top k elements we only return the first one (text column) and we define it as a new query. We have developed a new function in order to make a kind of cluster (similar tweets) to the initial one, returning the top 19 nearest ones. Thus, we have developed another score ranking different from the initial one (tf-idf) and having as an output 20 similar tweets that match with the query.

The screenshots showing the comparison of the two score rankings is shown in the code, printing as an output the returned documents of each query.

Guillem Morgo 204564
Joan Medina 204561

SECTION 3: RQ1

1.Ten selected queries

queries:
'trump has covid',
'impact of coronavirus',
'how to face off covid',
'masks and lockdowns don\'t work',
'how good is the vaccine?',
'vaccine of pfizer',
'vaccine 2021',
'no more masks',
'I think I have covid',
'what are covid symptoms'

2.Can you imagine a better representation than word2vec? Justify.

Word2vec is a method to efficiently create word embeddings. The Word2Vec Algorithm builds distributed semantic representation of words. However, we are trying to get similar tweets which can be considered as documents. So it could be better to use a sentence2vec, where instead of learning feature representations for words, you learn it for sentences or documents.

1.RQ1A: Are you able to detect some subgroups within your tweets representation? Are you able to perform some clustering over the tweets and detect some topics within the conversation? How do you choose the best possible number of clusters?

Implementing the word2vec python function we can realize that there are some words with a high similarity, that depending on the context that they appear in the documents we can conclude that they are subgroups of tweets inside our tweet topic. Due to the executing time of our python function, we have not shown an output defining the tweet clusters, but we have assimilated the theory.