UNIL | Université de Lausanne
HEC Lausanne

ANALYSIS OF TWEETS REGARDING THE 2020 US ELECTION

TEXT MINING – FALL 2020

EDITION DATE, 14TH DECEMBER 2020

Group D

Gaëtan Lovey Cédric Vuignier Michael Vo-Ngoc

Executive summary

The U.S. presidential elections have been in the news consistently in recent months and President Donald Trump has long been known for his extensive use of Twitter. We are therefore interested in how he and other American political figures use this social network on the fringes of elections.

In this study, we apply text mining tools to real political tweets during the American presidential election of 2020. In that framework, we make an analysis of the patterns present in the campaign messages. As we consider that social networks play a key role in elections, we try to demonstrate signs of language specific to candidates and political parties. Three underlying questions linked to the available data are raised. How do politicians use Twitter for their campaign? Is there a linguistic difference before and after the election? Is it possible to predict the author of a message and the person's political party based on their tweets?

In order to analyze this data, we will divide our analysis into four parts. The introduction explains the concept of this work and the methodology used. In the section on data acquisition, we will show how we retrieve thousands of tweets from American politicians before and after the election day and how we will create the corpus to analyze it. Then, we will make an exploratory data analysis which relates various concepts used in text mining such as a sentiment analysis, similarities, topic modeling and unsupervised learning. Finally, we will build different supervised prediction models.

Our analysis underlined a clear difference between politicians and between political parties. Joe Biden and Donald Trump tweeted massively before the election before reducing the use of social networks at the close of the polls. Republicans used a more patriotic vocabulary. They put forward America, its greatness, its people and Trump's upcoming victory. Concerning the Democrats, they focused on more topical issues such as the coronavirus, the climate or the healthcare.

However, there was one glaring result of our analysis. The main use of Twitter is to criticize and destabilize the opposing camp. Indeed, the most used words from presidential candidates are related to the opposite political party.

Finally, our supervised models predicted the author of a tweet and his political party. The different combinations of features used in our models as well as the results obtained showed that in predicting the author of a tweet and his political party, the support vector machine outperformed all the other models studied.

Overall, the study allowed us to find significant patterns among all the tweets. However, for many reasons, this study could not be applied to other geographical scope. Nevertheless, our analysis opened the door to further investigations focused on how the messages can influence the final results of the election.

Table of contents

List of figures

1. Introduction

1.1 Research questions

Social networks are becoming more and more important in our western societies. They even influence the way we do politics. Therefore, it is essential to study this phenomenon in order to understand how it influences future votes. The purpose of this analysis is to study the use of the social network Twitter during the American election campaign. We will try to find patterns among the politicians. Are tweets similar across political parties? Do they vary a lot from one candidate to another? Did the election results influence the candidates' tweets?

We decided to study political tweets because they are unavoidable. All the media read tweets. Their information is often based on these small messages. Furthermore, a candidate should be on Twitter in order to influence the electorate and thus, win the election. We chose to base ourselves on six American politicians involved in the campaign:

- Donald Trump (US president and presidential candidate)
- Mike Pence (VP, US vice-president)
- Joe Biden (presidential candidate)
- Kamala Harris (vice-presidential candidate)
- Barack Obama (former president)
- Alexandria Ocasio-Cortez (AOC, member of the U.S. House of Representatives)

All these people are active on Twitter and have a tremendous influence. Their messages are addressed to millions of followers.

1.2 Methodology

It is important to mention that this study is the result of group work and therefore of a collaboration between Cédric Vuignier who took care of the EDA and the data analysis, Gaëtan Lovey who took care of the un/supervised learning part and Michael Vo Ngoc who built the study report.

For this analysis, we used the R software because we are used to work with it during our Business Analytics courses at HEC Lausanne.

To begin, we will start by extracting the data from Twitter website, then we will transform and load them in order to perform text mining analysis. Finally, we will build prediction models in order to predict the author of a tweet and his political party.

2. Data acquisition

In this part, we will explain how we extracted, transformed and loaded our data. First, we created a Twitter account. Then, from the Twitter developer site, we set up the API accesses to download the tweets. We based ourselves on the "rtweet" package which allows us to link the API and the program R.

```
## load rtweet
library(rtweet)

## store api keys (these are fake example values; replace with your own keys)
api_key <- "cFRZi4lsJGD4DxrtLT24za5UH"
api_secret_key <- "iHBlU1fA5WdQ17N3udi8FWg8TlwugXb8e6c34MNRGmKXNJFIQY"

## authenticate via web browser
token <- create_token(
  app = "TextMiningHEC",
  consumer_key = api_key,
  consumer_secret = api_secret_key)
```

Figure 1: API access

2.1 Tweets scraping

We collected the 3,000 last tweets from Donald Trump, Joe Biden, Kamala Harris, Mike Pence, Barack Obama and Alexandria Ocasio-Cortez. The data frame has 90 variables.

```
tweet_trump <- get_timelines("realDonaldTrump", n = 3000, include_rts = FALSE)

#vice president tweet
tweet_VP <- get_timelines("VP", n = 3000, include_rts = TRUE)
```

Figure 2: Scraping example for Trump

Tweets contain a lot of unhelpful information and it is therefore necessary to clean up the text. Here, we selected only the most important variables for the purpose of readability (Figure 3). The final data set contains 14,297 tweets.

Initial data base

username	created_at	tweet	source	length	retweet	favorite
JoeBiden	2020-11-21 17:00:01	Anyone who wants a COVID-19 test should be able to get one. Period.	TweetDeck	67	53803	1002521
JoeBiden	2020-11-21 00:08:00	This afternoon, @KamalaHarris and I met with @SpeakerPelosi and @SenSchumer to discuss how we'll get this virus under control, deliver much-needed relief, and build back better than before. We're getting right to work for the American people. https://t.co/jbeNpsimdI	Twitter Media Studio	244	12341	176275
JoeBiden	2020-11-20 20:07:00	Here's the deal: Because President Trump refuses to concede and is delaying the transition, we have to fund it ourselves and need your help. If you're able, chip in to help fund the Biden-Harris transition. https://t.co/apJMrdpoSS	TweetDeck	231	17566	115387
JoeBiden	2020-11-20 17:41:56	Tune in as my team provides an update on the Trump Administration's efforts to delay ascertainment and how we are moving forward to ensure a smooth transition of power. https://t.co/In0kywJf3a	Periscope	192	4391	36196
JoeBiden	2020-11-20 15:55:22	To transgender and gender-nonconforming people across America and around the world: from the moment I am sworn in as president, know that my administration will see you, listen to you, and fight for not only your safety but also the dignity and justice you have been denied.	Twitter Web App	274	11837	130060

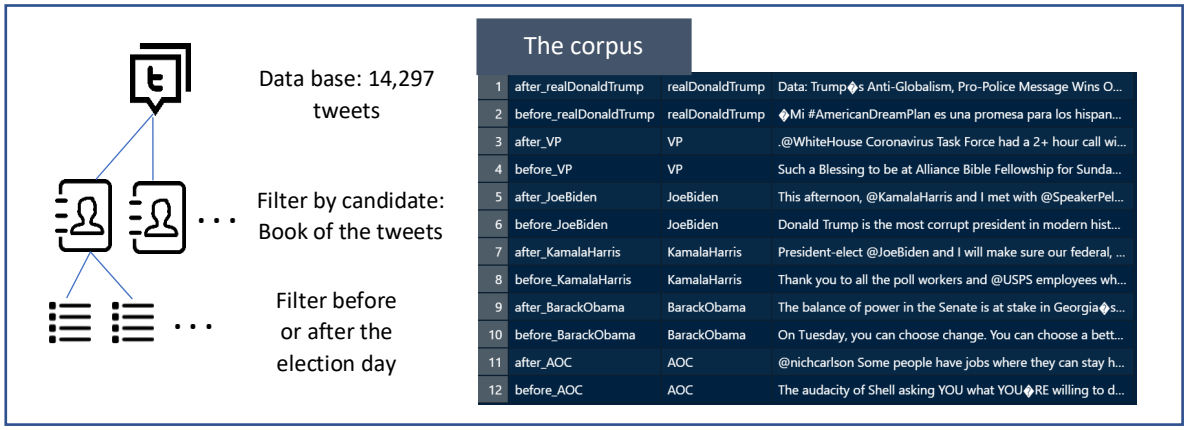
Figure 3: Data base example

2.2 Creation of the corpus

We created a first corpus gathering all the available tweets. Each document represents a different politician without any temporal distinction (6 documents). This database will be mainly used for the un/supervised learning part.

Assuming that the nature of the tweets may change between before and after election day and thus allow us to find different patterns specific to the period analyzed, we created another corpus by separating the pre-election tweets from the post-election tweets. Therefore, we divided the documents in our corpus according to the date of the election. This corpus is composed of the tweets of each candidate before and after the election (12 documents).

In both corpora, we filtered the size of the tweets by keeping only the messages with more than 100 characters.



1	after_realDonaldTrump	realDonaldTrump	Data: Trump€™s Anti-Globalism, Pro-Police Message Wins O...
2	before_realDonaldTrump	realDonaldTrump	€Mi #AmericanDreamPlan es una promesa para los hispan...
3	after_VP	VP	.@WhiteHouse Coronavirus Task Force had a 2+ hour call wi...
4	before_VP	VP	Such a Blessing to be at Alliance Bible Fellowship for Sunda...
5	after_JoeBiden	JoeBiden	This afternoon, @KamalaHarris and I met with @SpeakerPel...
6	before_JoeBiden	JoeBiden	Donald Trump is the most corrupt president in modern hist...
7	after_KamalaHarris	KamalaHarris	President-elect @JoeBiden and I will make sure our federal, ...
8	before_KamalaHarris	KamalaHarris	Thank you to all the poll workers and @USPS employees wh...
9	after_BarackObama	BarackObama	The balance of power in the Senate is at stake in Georgia€™s...
10	before_BarackObama	BarackObama	On Tuesday, you can choose change. You can choose a bett...
11	after_AOC	AOC	@nichcarlson Some people have jobs where they can stay h...
12	before_AOC	AOC	The audacity of Shell asking YOU what YOU€™RE willing to d...

Figure 4: Creation of the corpus

Here we have, as an example, the tweets of Donald Trump. First, we divided the tweets in two parts according to the date of the election. Then, we made a first cleaning of the data by limiting the size of the tweets (100 characters).

```
#get only the tweets after the election
Trump_after_analysis <- TRUMP_all_tweet %>%
  mutate(date = date(created_at)) %>% filter(date >= "2020-11-03",
                                             display_text_width > 100)
#get the tweet before
Trump_before_analysis <- TRUMP_all_tweet %>%
  mutate(date = date(created_at)) %>% filter(date < "2020-11-03",
                                             display_text_width > 100)
```

Figure 5: Data filtering

Next, we compiled all the tweets from Trump into two different documents in order to get one book of his tweets before and one after the election. Thereafter, we repeated this process for other politicians and built the corpus of 12 documents.

```
#book of tweets "after"
Trump_after_analysis <- Trump_after_analysis %>%
  select(text,screen_name, when) %>%
  group_by(screen_name, when) %>%
  summarise(paste(Trump_after_analysis$text, collapse = '. '))

Trump_after_analysis <- rename(Trump_after_analysis, text = `paste(Trump_after_analysis$text, collapse
= ". ")`)
#book of tweets "before"
Trump_before_analysis <- Trump_before_analysis %>%
  select(text,screen_name, when) %>%
  group_by(screen_name, when) %>%
  summarise(paste(Trump_before_analysis$text, collapse = '. '))

Trump_before_analysis <- rename(Trump_before_analysis, text = `paste(Trump_before_analysis$text,
collapse = ". ")`)
```

Figure 6: Book creation

3. Cleaning and exploratory data analysis

3.1 Twitter account statistics

Before starting the text mining analysis, we extracted interesting information from the scraped data. There are several variables that give information on the style of writing and the impact of the message.

General statistics

name	avg_like	avg_retweet	avg_tweet_length
BarackObama	168524	20309	194
JoeBiden	155558	16598	151
realDonaldTrump	151669	29660	124
KamalaHarris	124607	13772	161
AOC	107233	10065	141
VP	0	7223	127

Figure 7: General Statistics

For this part, we analyzed the data in an interval of two weeks before and after the election. On average, Obama's tweets are the most liked and Donald Trump gets the most retweet. Concerning the length of the tweets, Republican tweets are on average shorter. Finally, the vice-president (VP) Mike Pence does not have any like. Indeed, he does not write any message but only retweets.

Concerning the number of tweets per day, there was a strong increase in the last days before the election (3rd November 2020). After this date, the number of daily tweets dropped sharply. We remark that Donald Trump wrote more than 60 tweets on the 1st of November.

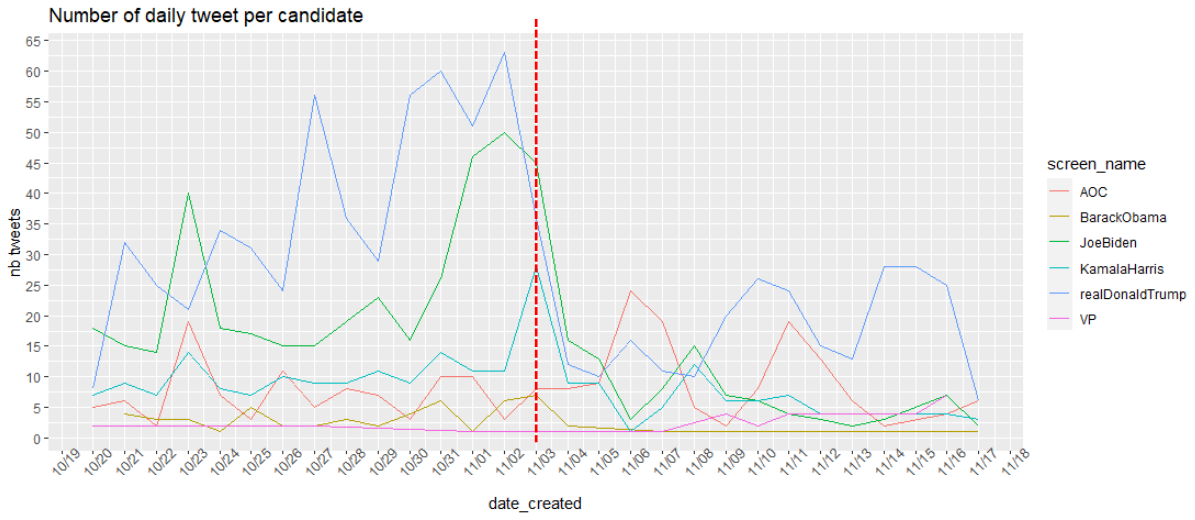


Figure 8: Daily tweets

We observe that both presidential candidates tweeted a lot during the two weeks before the election. We also notice that the vice-president is practically not campaigning on Twitter. The previous graph shows that this social network is an important tool for the election. The candidates widely used it.

3.2 The cleaning process

Data cleaning was very important in our analysis. Many tweets had numbers, special characters or unusable words. Indeed, even if messages are concise and impactful, one does not focus on the writing style when tweeting. Therefore, we removed special characters from our entire corpus using the "gsub" function. Then, we removed some useless words by using the "stop_words" lexicon.

3.3 The most used words

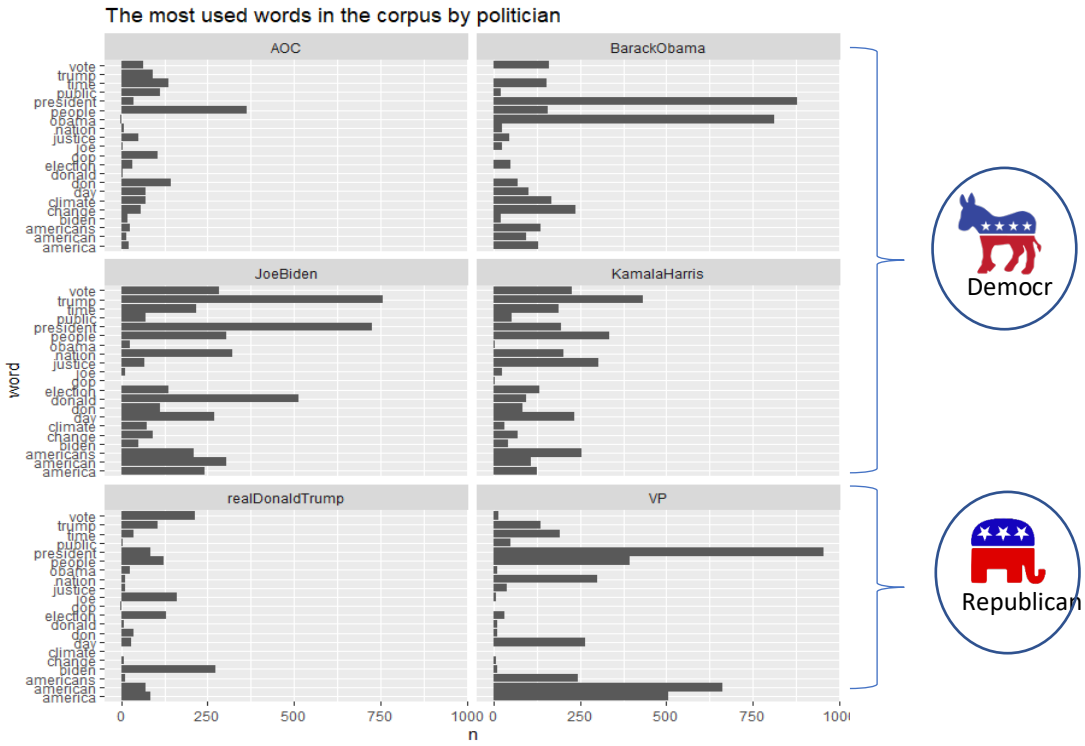


Figure 9: Most used words analysis

This first plot analyzes the most frequently used words among all the candidates. We notice big differences in the vocabulary of the politicians. We can see that the Republicans hardly talk about climate change, but rather populist words are often used. There is a lot of talk about America, the president, the people and the country. The purpose of tweets is to bring voters together with simple messages.

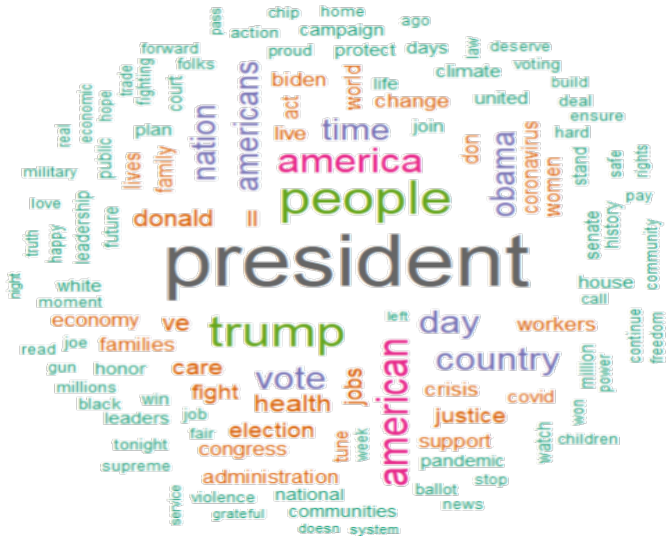


Figure 10: Global wordcloud

3.4 Comparaison between Trump and Biden

We compared the most 15 used words by the two presidential candidates. Results show a lot about their strategy and the topics they address.

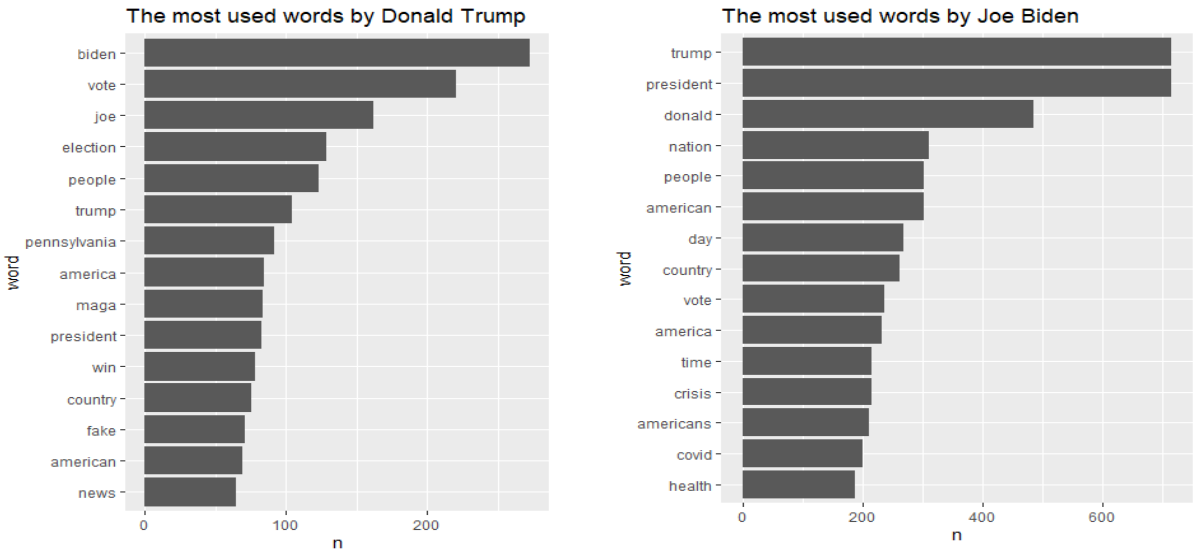


Figure 11: Most used words of Trump and Biden

We can see that a lot of tweets focus on the opponent. Discrediting is a very present characteristic in this campaign. It is also interesting to analyze the candidates individually.

Trump: He does not often address specific topics in his tweets. He promotes topics like America, victory, his slogan "MAGA" and finally he talks about fake news.

Biden: The scope of the topics covered is broader. He talks about health, coronavirus but also about the nation and the people.

3.5 Corpus analysis

Yule's Index

We have seen that the vocabulary used varies a lot from one Twitter account to another. Are the documents containing all the tweets very different? Thanks to the Yule's index, we determined if a candidate's vocabulary is richer than another.

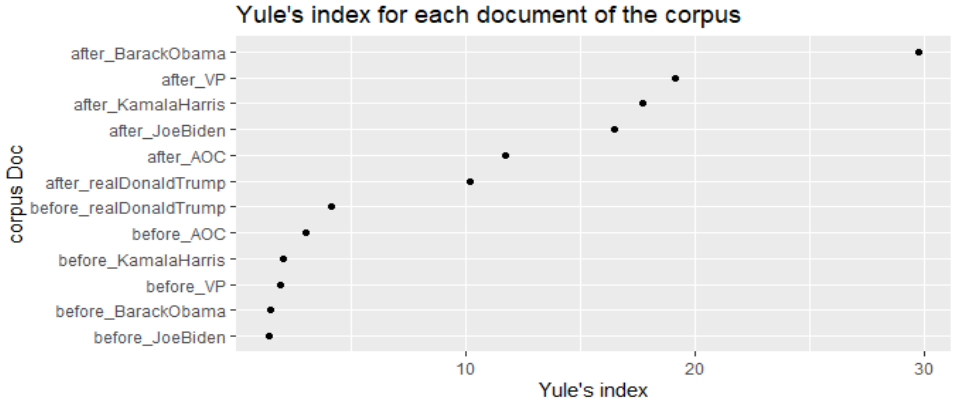


Figure 12: Yule's Index

We see a big difference between before and after the election day. Indeed, the richness of the vocabulary increases for all politicians after the election. Barack Obama uses the most different tokens. We also notice that before the election, the two presidential candidates have the poorest vocabulary. We can imagine that this choice is deliberate. Simple, short and repetitive messages help to captivate and motivate the voters. This is also in line with the previous results. The most used words of Biden and Trump are about opponents. We can therefore state that the use of a broad vocabulary is not useful for American political campaigns.

X-ray plot

The following graphs allows us to inspect where a specific token is used in each document. We analyzed six different words that demonstrated a potential political strategy.

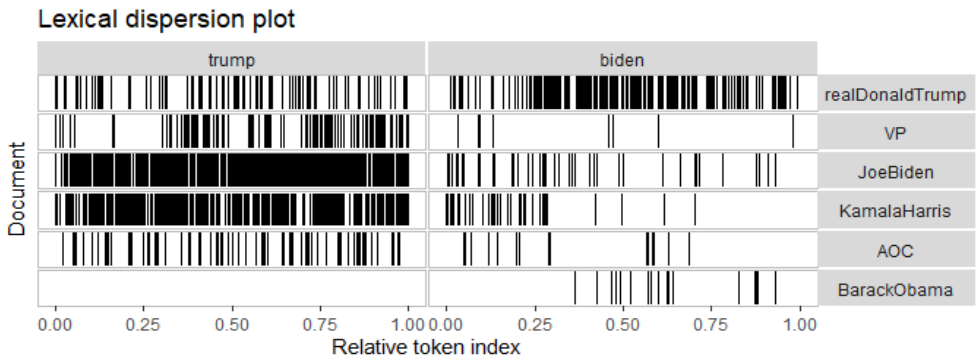


Figure 13: X-ray Trump vs Biden

This graph confirms our main hypothesis. Candidates focus on their opponent. The words "trump" and "biden" are widely used by each party. We even notice that Biden uses the word "trump" almost daily.

We also analyzed the words "covid" and "health" referring to important topics in the campaign. The crisis of the virus is strongly affecting the country. In addition, the question of an accessible health system is a big debate.

We remark that Donald Trump does not mention these subjects. He even fights them. As for Biden, we see that the coronavirus was a campaign subject. He uses it to criticize the Trump administration.

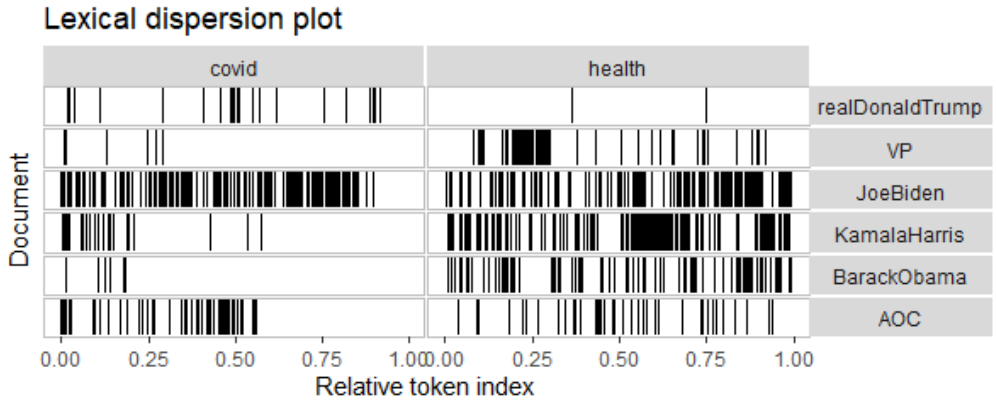


Figure 14: X-ray covid vs health

Keyness analysis

- Terms specific to the Democratic campaign: We see that the current topics of society are specific to democrats (climate, change, justice, black, job, pandemic, health, care).

- Terms specific to the Republican campaign: More patriotic terms are used (America, great, military, border, honor or amendment). Tweets talk about the opposing candidate.

- Terms specific before the election: Few tokens are specific to this period.

- Terms specific after the election: The results are more interesting. We observe terms specific to the counting process (election, count, observer, poll, fraud, signature, watcher, recount, win). This shows that the candidates' Twitter accounts are used to criticize the elections.

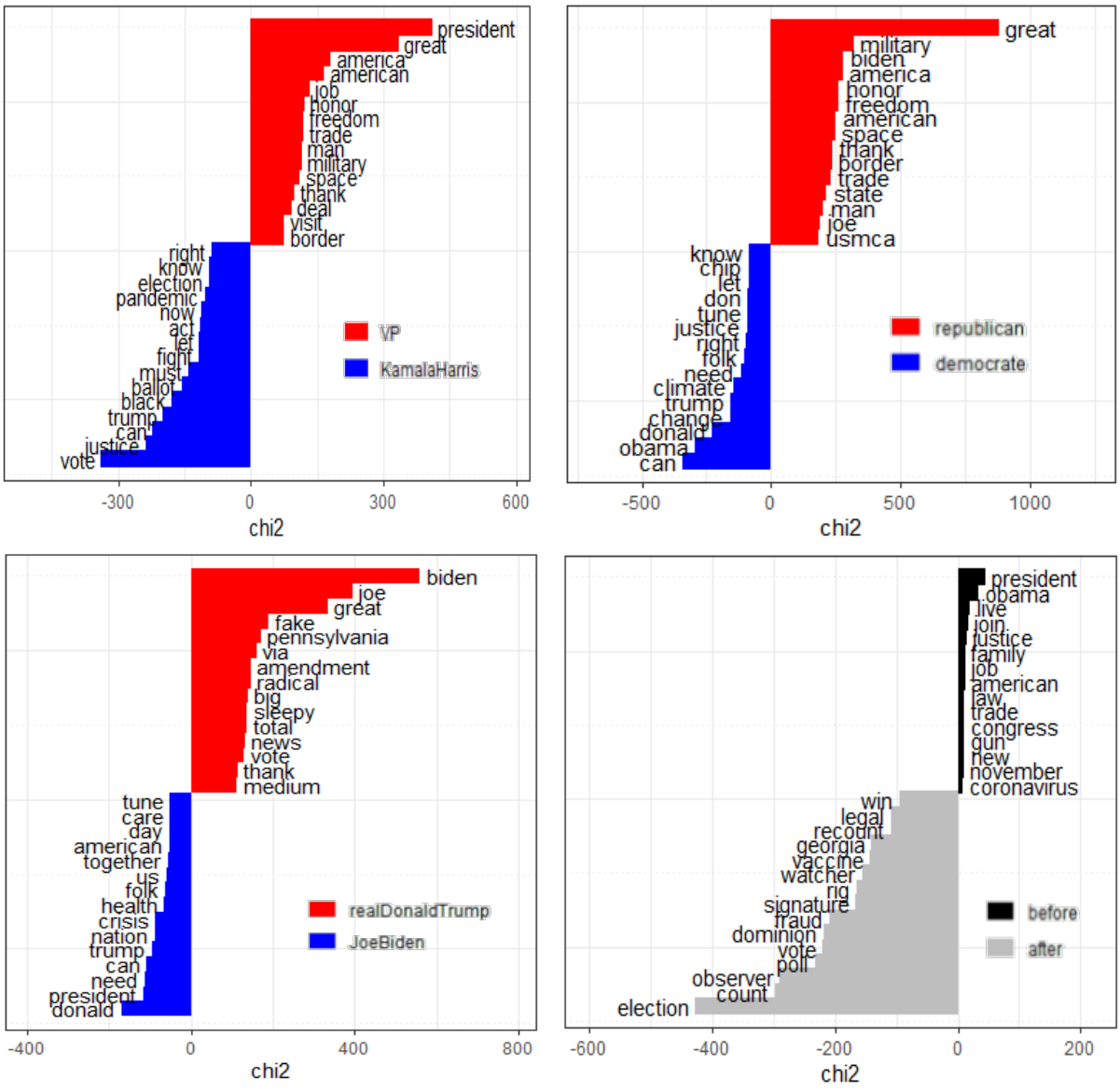


Figure 15: Keyness analysis

3.6 Sentimental analysis

Feelings do not vary according to the period. We therefore analyzed the corpus only by separating the politicians. As the length of the documents was very different, we analyzed the feelings based on their frequency. We observe that sentiment analysis is similar from one account to another. It is above all positivity and trust that are emphasized. In addition, all the documents are positive. Indeed, it makes sense to use this feeling to convince of a future victory.

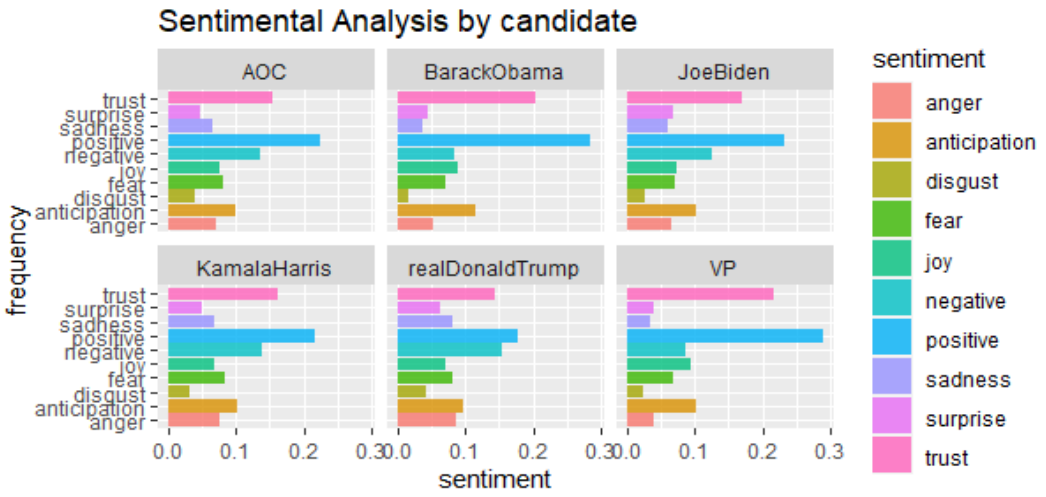


Figure 16: Sentiment-based analysis

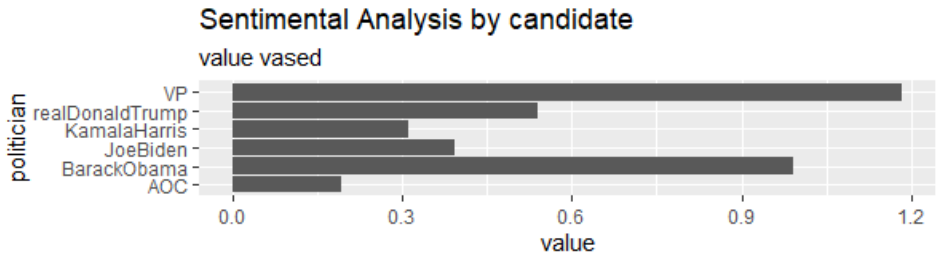
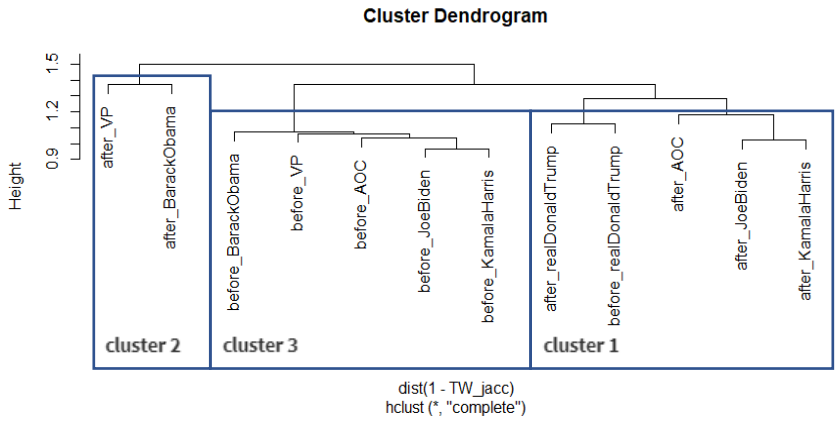


Figure 17: Sentiment value-based analysis

3.7 Similarities

By creating a TF-IDF matrix, we measured the similarity of the vocabulary in the documents. Then, we computed the Jaccard index matrix to define a distance between documents.



clustering

cluster 1	cluster 2	cluster 3
sleepy	astronauts	crisis
pennsylvania	warp	trump
fake	admiral	obama
biden	playlist	nation
radical	moderna	climate
via	vaccine	donald
amendment	songs	live
observers	mission	leaders

Figure 18: Similarities analysis

It is difficult to establish a list of words specific to each cluster. However, based on a before-after analysis, we can see that pre-election documents are closer together as are post-election day documents.

3.8 Topic modeling

In this part, we analyzed the topics of the tweets of six politicians (Joe Biden, Donald Trump, Kamala Harris, Barack Obama, Alexandria Ocasio-Cortez and Mike Pence) during the period of the election of the new President of the United States. We used the corpus containing only six documents and for which there is no difference between pre-election and post-election tweets.

Then, we proceeded to the cleaning of the corpus by removing words whose length was smaller than three characters and we computed the document-term matrix (DTM) which was composed by the words frequency in documents. Punctuations, symbols, URL, numbers and stop words from the dictionary "english" were removed from this DTM.

In order to model the topics and illustrate them, we used two different methods: latent semantic analysis (LSA) and latent Dirichlet allocation (LDA).

3.8.1 LSA

We chose to set the dimensions of the LSA to four. We also removed insignificant and useless words ("u", "t", "co", "https", "http", "s", "amp", "re", "m", "ll", "co", "we", "obama1").

LSA on frequency of words (TF)

We analyzed the different dimensions of the LSA. As the first dimension is often associated with the document length, we display their relation in the next graphic. Even if we have only six documents, it seems that the dimension one is negatively correlated to the length of the documents.

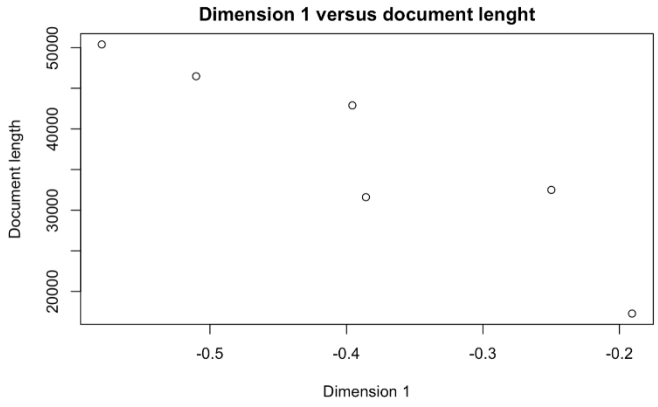


Figure 19: Dimension versus document length

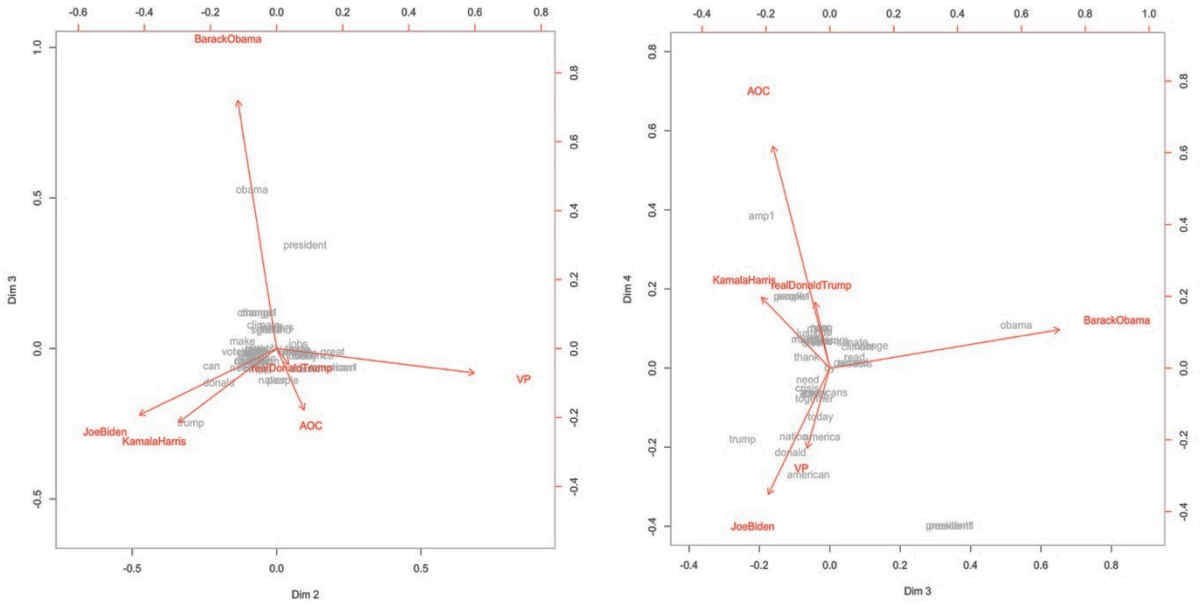


Figure 20: Link between the topics and the documents

The relationships are quite difficult to interpret. Barack Obama is well related to topic three, topic two is linked with Mike Pence (VP for vice president) and topic four is linked to Alexandria Ocasio-Cortez (AOC). Joe Biden is not represented by a specific topic unlike Donald Trump who is presents in nearly each of them.

Then, we extracted the words related the most to each topic. It is difficult to find paterns within topics. However, we observe some overlapping topics with common words (president, people, can, etc.).

Main words of topic 1		Main words of topic 2		Main words of topic 3		Main words of topic 4	
president	0.2426097	american	0.1966990	obama	0.5297437	people	0.1809539
people	0.1229029	great	0.1941972	president	0.3420220	obama	0.1100319
trump	0.1208943	america	0.1324309	change	0.1176331	don	0.1028106
today	0.1170286	thank	0.1002552	climate	0.0804606	now	0.1023835
can	0.1134796	president	0.0987490	leaders	0.0723450	many	0.0963700
american	0.1129068	today	0.0970025	read	0.0706253	justice	0.0871714
america	0.0953403	military	0.0760543	senate	0.0640832	gop	0.0867775
day	0.0790945	jobs	0.0748599	garland	0.0602790	can	0.0754525
nation	0.0765409	freedom	0.0737003	progress	0.0594181	must	0.0738859
one	0.0761340	trade	0.0725157	add	0.0588602	healthcare	0.0712795

Figure 21: Words in each topic (TF)

LSA on weighted frequency (TF-IDF)

We computed the weighted frequency and repeated the same process as previously to analyze the four LSA dimensions. We remark that the dimension one is not correlated to the length of the documents.

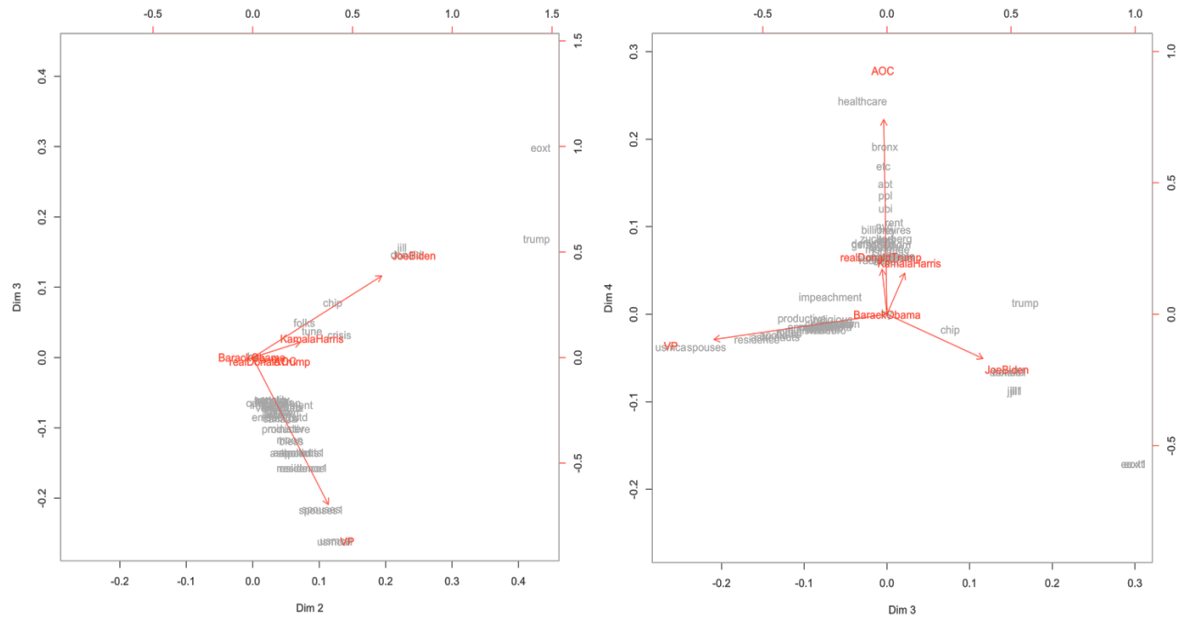


Figure 22: Link between the topics and the documents

Joe Biden is more related do dimension three than other documents, Alexandria Ocasio-Cortez (AOC) to topic four which seems to be linked with healthcare.

Again, it is difficult to find patterns in those topics as words seems to be quite disparate within dimensions. Moreover, topic four seems to gather several abbreviations.

Main words of topic 1		Main words of topic 2		Main words of topic 3		Main words of topic 4	
garland	0.1196494	trump	0.4272242	trump	0.1673693	healthcare	0.2436844
enter	0.0378948	donald	0.2310667	donald	0.1467370	bronx	0.1915900
add	0.0279887	crisis	0.1304762	chip	0.0764761	etc	0.1689918
weekly	0.0279782	chip	0.1203609	folks	0.0495330	abt	0.1490145
merrick	0.0236669	spouses	0.1030687	tune	0.0375859	ppl	0.1348226
climate	0.0225307	tune	0.0890906	crisis	0.0321753	ubi	0.1206308
leaders	0.0193235	folks	0.0777641	fiction	0.0309764	rent	0.1049098
boards	0.0169313	residence	0.0743053	reserve	0.0309764	nyc	0.0984589
consecutive	0.0157779	astronauts	0.0671718	que	0.0303948	billionaires	0.0966353
vacancy	0.0152615	apollo	0.0647175	clearer	0.0281603	hey	0.0947294

Figure 23: Words in each topic (TF-IDF)

3.8.2 LDA

As the LDA method is limited to a document-feature matrix (DFM), we only applied the method on the frequency of words (TF). We set the number of dimensions at eight and computed the LDA.

The words most associated with the topics

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
president	https	country	amp	biden	president	president	https
amp	can	fight	people	vote	obama	today	donald
american	people	justice	can	great	http	now	can
today	make	help	one	joe	change	let	american
america	today	families	just	amp	can	take	need
great	now	must	get	thank	make	together	one
people	act	women	don	election	today	get	vote
nation	americans	time	now	people	climate	make	nation
thank	vote	first	need	big	vote	can	people

Figure 24: LDA: words in each topic (TF)

Below, we computed the beta's representing the per-topic-per-word probabilities revealing the main terms associated to each topic. Like with the LSA method, some topics are overlapping. Topic five seems related to the election, topic three to the work and topic six to the climate change. However, patterns are not clearly defined.

Figure 25: Beta's analysis

Then, we computed the gamma's. They are the per-document-per-topic probabilities and represent the proportion of each topic within each document.

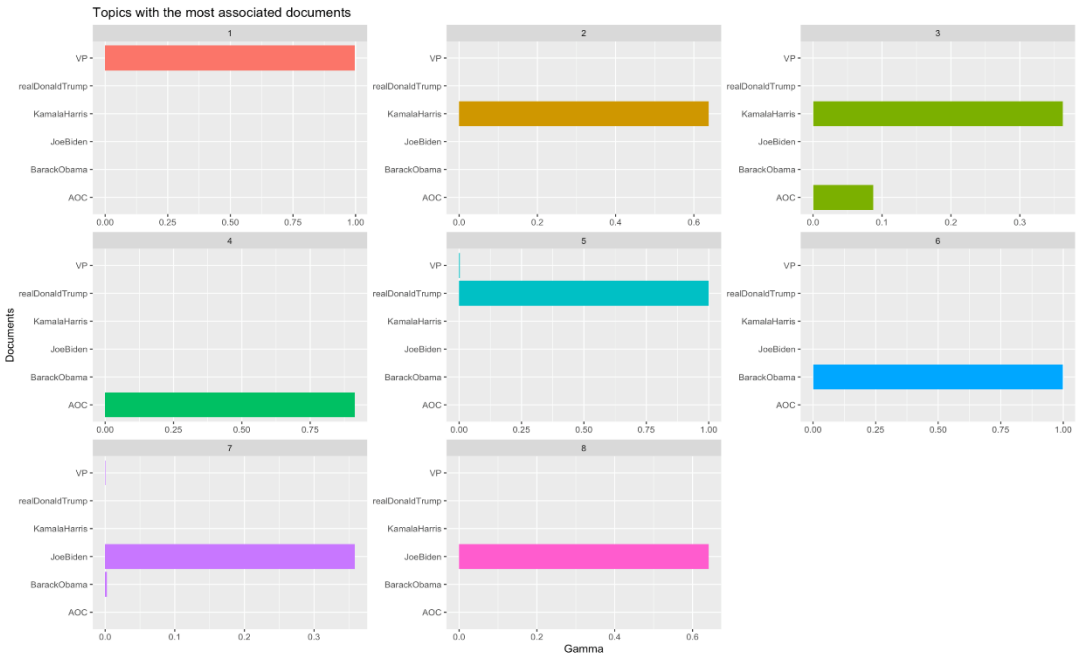


Figure 26: Gamma's analysis

As we can see in the next graph, topic three is used in more than one document. In addition, only Kamala Harris (topic 2 and 3), Joe Biden (topic 7 and 8) and Alexandria Ocasio-Cortez (AOC) (topic 4 and 3) are present in more than two topics.

With the following table, we notice that each document is mainly related to a different topic. Therefore, topics could be useful to classify the author of the documents.

Topics most associated with each document	
realDonaldTrump	5
VP	1
JoeBiden	8
KamalaHarris	2
BarackObama	6
AOC	4

Figure 27: A Topic for each document

4. Unsupervised and supervised learning

In this part again, we used the including the six documents. First of all, the aim was to predict the person who wrote a tweet. Then, we built models to predict the political party. For both predictions, we tried various prediction models throughout our analysis. However, only the two best performing models which are the random forest and the support vector machine (SVM) are presented in this study.

We first added the variable corresponding to the political party of the tweet's writer to the corpus.

screen_name	created_at	text	source	display_text_width	favorite_count	retweet_count	party
JoeBiden	2020-08-18 01:36:24	https://t.co/4vQXPuDSgr	Twitter Web App	0	5784	1040	democrate
JoeBiden	2020-08-18 01:16:28	https://t.co/MgJ45HeB2a	Twitter Web App	0	8073	1667	democrate
KamalaHarris	2019-12-04 21:53:56	https://t.co/qFoI4xgowF	Twitter Media Studio	0	22141	2070	democrate
realDonaldTru...	2020-11-19 21:21:28	https://t.co/IJa6syVoqN	Twitter for iPhone	0	121642	28459	republican
realDonaldTru...	2020-11-18 17:43:52	https://t.co/WqwBtqRH1I	Twitter for iPhone	0	144449	39511	republican

Figure 28: Database

From this corpus, we removed words with less than three characters. Next, we proceeded to the tokenization to split the corpus into tokens. In order to keep only significant tokens, we made different operations. First, we removed the punctuation, the symbols and the URL. Then, we used the lemmatization method to reduce the vocabulary. Uppercase letters were replaced by lowercase letters and stop words from dictionary "english" were removed as well as insignificant tokens. Finally, we also deleted tokens containing numbers and sentences with less than three tokens.

4.1 Word embedding

In this part, we performed a word embedding. It was useful for purposes of unsupervised learning such as clustering but also to compute centroids in order to have a document embedding for supervised learning models. A words vector representation was created by computing the co-occurrence matrix and by using the GloVe method. Then, we chose a two word embedding dimensions in order to be able to display results in a graph. The following figure shows the 100 most frequent words used in the tweets.

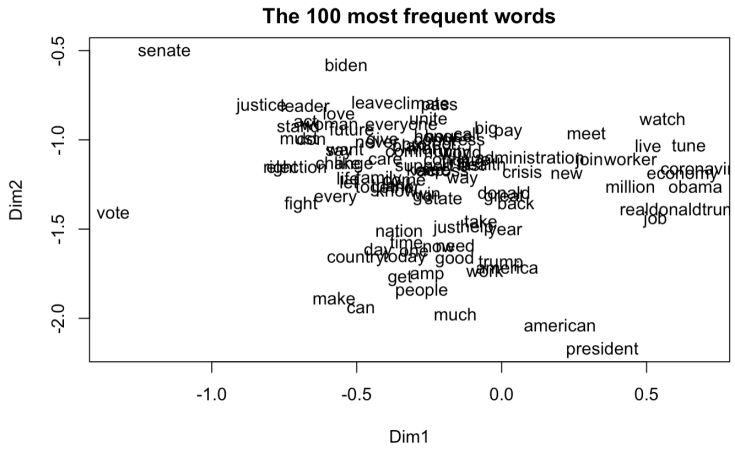


Figure 29: Most frequent words

Afterwards, we computed the distance from the document-term matrix (DTM), composed by the words frequency in documents, by using the RelaxedWordMoversDistance (RWMD). This distance was applied to make clusters. As there were so many different words, dendrogram was completely illegible. To solve this problem, we extracted the 10 most frequent words appearing in five clusters.

Clust.1 <fctr>	Clust.2 <fctr>	Clust.3 <fctr>	Clust.4 <fctr>	Clust.5 <fctr>
president	president	president	president	amp
american	american	can	obama	que
much	trump	american	american	los
trump	today	vote	amp	para
can	can	much	much	get
amp	one	trump	people	one
people	get	amp	work	work
today	need	make	make	people
make	much	people	can	trump
vote	people	get	get	can

Figure 30: Clustering list (WE of 2 dimensions)

Even if it is difficult to find patterns in those clusters, we remark that the fifth cluster seems to be somewhat related to words of Spanish origin. Others are composed by more common words.

To compare our clustering, we repeated the previous stage but with a 25 word embedding dimensions.

Clust.1 <fctr>	Clust.2 <fctr>	Clust.3 <fctr>	Clust.4 <fctr>	Clust.5 <fctr>
president	president	senate	happy	job
american	american	doyourjob	birthday	actonclimate
much	amp	judge	president	president
can	much	leader	obama	american
amp	trump	court	day	amp
people	today	garland	wish	much
get	can	supreme	celebrate	month
make	people	hear	today	economy
vote	day	fair	year	year
trump	make	add	team	clean

Figure 31: Clustering list (WE of 25 dimensions)

Clusters seem more significant. The two first clusters are quite similar, the third one is related to the supreme court and the fourth one links words related to joy and happy events (birthday, happy, wish, celebrate). The last cluster gather words linked to economy and climate.

4.2 Prediction of the author of the tweet

We built different models to classify the tweets according to their author (y = screen_name) and to compare them. We first built a random forest by trying out different combinations of features to find out which ones were most useful.

TF and LSA

To build the features, we computed a DTM matrix composed by the word frequency. We also applied the latent semantic analysis (LSA) on the matrix, which is a reduction dimension technique, in order to obtain less features. We tried different dimensions for the LSA (2, 5, 25, 50, 100, 500, 1000) and kept the one with the best accuracy after having trained the models and made the predictions. We also had logarithm of the number of retweet (logretweet) and the logarithm of the token length as features (length). Therefore, we got a data frame composed by the class to predict (screen_name), the dimensions of the LSA, the logretweet and the length.

We split this data frame into a training (80%) and a test set (20%) and trained the random forest on the training set. Then, we made the predictions and computed the accuracy according to the different LSA dimensions.

```
set.seed(2020)
index.tr <- sample(size=round(0.8*length(y)), x=c(1:length(y)), replace=FALSE)
df.tr <- df[index.tr,]
df.te <- df[-index.tr,]
set.seed(2020)
tweets.fit <- ranger(Class ~ ., #use a random forest
                     data = df.tr)
pred.te <- predict(tweets.fit, df.te)
acc.vec[j] <- confusionMatrix(data=pred.te$predictions, reference = df.te$Class)$overall[1]
```

Figure 32: Random Forest with TF and LSA

We reached an accuracy of 0.686 related to a LSA of 500 dimensions as we can see in the next image.

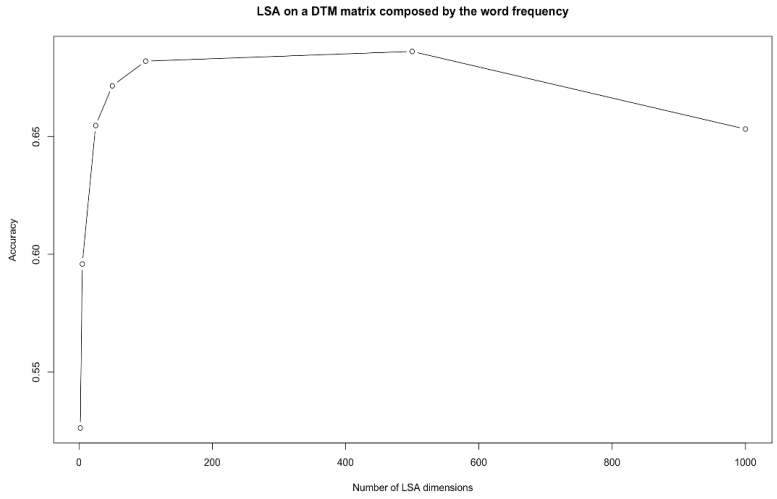


Figure 33: Accuracy of the random forest with LSA on the word frequency (TF)

TF-IDF and LSA

Here, we built the features using the weighted frequencies (TF-IDF), the LSA with different dimensions as previously, the logretweet and the length. We trained a random forest model and scored the predictions to find an accuracy of 0.738 when using a LSA with 50 dimensions. Therefore, the model is doing a better job with TF-IDF instead of TF.

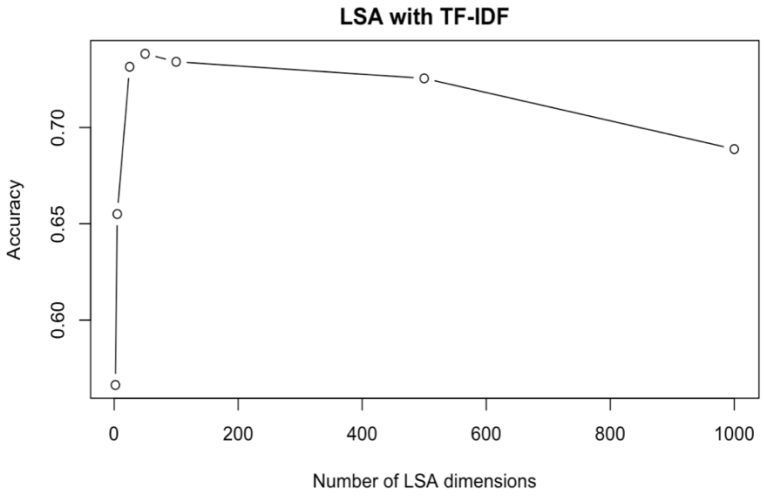


Figure 34: Accuracy of the random forest with LSA on the weighted frequency (TF-IDF)

Word embedding and centroids

In order to transform the word embedding that we presented before into a document embedding, we computed the centroids. Those centroids are the features for our random forest model. This time, we found an accuracy of 0.2764 when using a word embedding of two dimensions and an accuracy of 0.621 for a word embedding of 25 dimensions. This number of dimensions improved largely our score which is still disappointing. However, we could use those centroids in combination with other variables to improve our model.

```
tweets_coo <- fcm(tweets.tokens, context = "window", window = 5, tri = FALSE)
p <- 25 #word embedding dimension
tweets.glove <- GlobalVectors$new(rank = p, x_max = 10) #GloVe method
tweets.we <- tweets.glove$fit_transform(tweets_coo) #central vectors
word_vectors_context <- tweets.glove$components #context vectors
twitter.glove <- tweets.we + t(word_vectors_context) #match central vectors and context vectors

ndoc <- length(tweets.tokens) #number of documents in the corpus
centers <- matrix(nrow = ndoc, ncol = p) #create a matrix for the centroids
for(i in 1:ndoc){ #compute the centroids
  words_in_i <- twitter.glove[tweets.tokens[[i]],, drop = FALSE]
  centers[i,] <- apply(words_in_i, 2, mean)
}

row.names(centers) <- names(tweets.tokens) #name the columns of the centroid matrix

df <- data.frame(Class=y, X=centers)
df.tr <- df[index.tr,]
df.te <- df[-index.tr,]

set.seed(2020)
tweets.fit <- ranger(Class ~ .,
                     data = df.tr)

pred.te <- predict(tweets.fit, df.te)
confusionMatrix(data=pred.te$predictions, reference = df.te$Class)
```

Figure 35: Random Forest with centroids from word embedding part

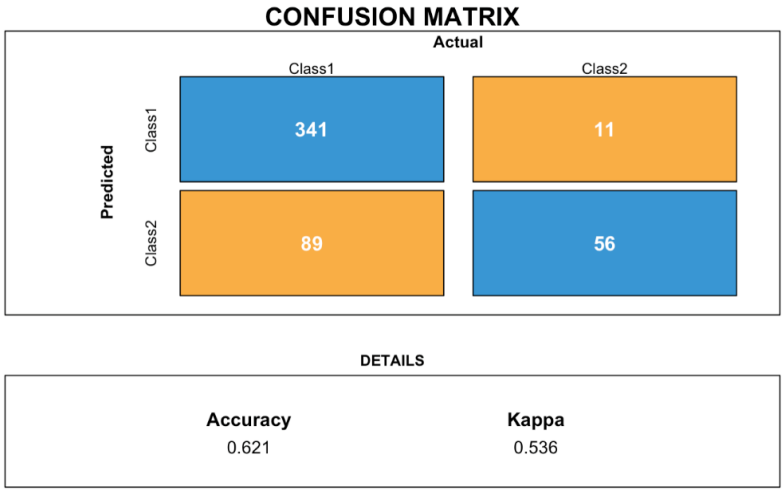


Figure 36: Confusion matrix of the Random Forest with centroids

Combining centroids with TF-IDF and LSA

As TF-IDF did better than TF, we combined the weighted frequencies with a 50 dimensions LSA, the logretweet, the length and the centroids.

```r
tweets.tfidf <- dfm_tfidf(tweets.dfm) #TF-IDF
tweets.lsa <- textmodel_lsa(tweets.tfidf, nd=50)
df <- data.frame(Class=y, X=tweets.lsa$docs)
df <- cbind(df,
            logretweet=log10(docvars(tweets.tokens, c("retweet_count"))),
            length = log(sapply(tweets.tokens, length)))
df <- cbind(df, Cent=centers) # add the centroid from word embedding part
df.tr <- df[index.tr,]
df.te <- df[-index.tr,]
set.seed(2020)
tweets.fit <- ranger(Class ~ .,
                     data = df.tr,
                     importance = "impurity")
pred.te <- predict(tweets.fit, df.te)
confusionMatrix(data=pred.te$predictions, reference = df.te$Class)
```

Figure 37: Random Forest with TF-IDF, LSA and centroids

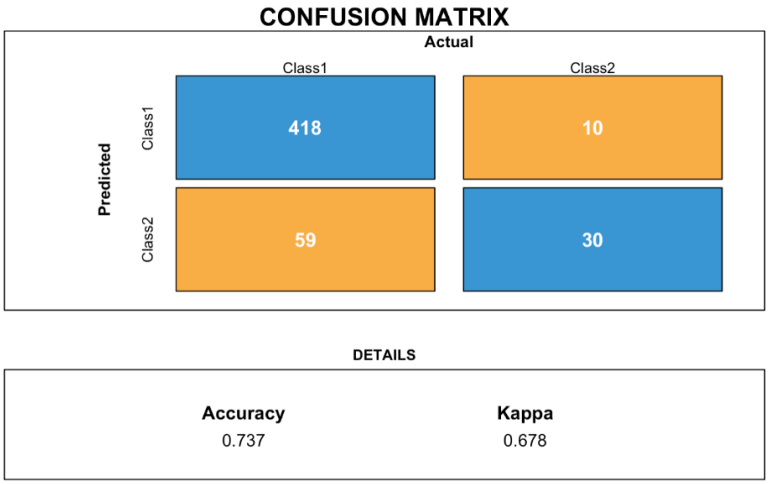


Figure 38: Confusion matrix of the Random Forest with TF-IDF, LSA and centroids

	Class: JoeBiden	Class: KamalaHarris	Class: realDonaldTrump	Class: VP	Class: BarackObama
Sensitivity	0.7518	0.5744	0.75115	0.8434	0.7620
Specificity	0.8992	0.9254	0.98614	0.9435	0.9614
Pos Pred Value	0.6624	0.6304	0.82741	0.8059	0.8271
Neg Pred Value	0.9323	0.9076	0.97816	0.9559	0.9434
Prevalence	0.2082	0.1813	0.08127	0.2176	0.1951
Detection Rate	0.1566	0.1041	0.06105	0.1835	0.1487
Detection Prevalence	0.2363	0.1652	0.07378	0.2277	0.1798
Balanced Accuracy	0.8255	0.7499	0.86865	0.8934	0.8617

	Class: AOC
Sensitivity	0.6817
Specificity	0.9572
Pos Pred Value	0.6773
Neg Pred Value	0.9580
Prevalence	0.1165
Detection Rate	0.0794
Detection Prevalence	0.1172
Balanced Accuracy	0.8194

Figure 39: Statistics by class for the Random Forest

Still using a random forest model, we obtained an accuracy of 0.737. The balanced accuracy is quite high but, there is a huge gap between sensitivity and specificity in some classes, especially for Kamala Harris. Overall, the model is better at predicting true negatives (higher specificity) and has difficulty recognizing AOC and Kamala Harris.

The model did not surpass the predictions made with only TF-IDF and LSA (0.738). Nevertheless, thereafter we kept this combination of features to build another model and compare the scores.

Support vector machine (SVM)

The model, considered as a separation method, consists in looking for the linear optimal separation of the hyperplane in order to classify the observations. We used a radial kernel and a 5-fold cross-validation applied on the train set to tune the hyperparameters.

```
train_control <- trainControl(method = "cv", number = 5)
metric <- "Accuracy"

set.seed(2020)
fit_svm <- train(
  form = Class ~ .,
  data = df.tr,
  trControl = train_control,
  tuneLength = 5,
  method = "svmRadial",
  preProcess = c("center","scale"),
  metric = metric,
  na.action=na.exclude
)
```

Figure 40: SVM with TF-IDF, LSA and centroids

With the CARET package, we tuned the cost hyperparameter controlling the tolerance to bad classification and corresponding to the smoothing of the border that classifies the observations. The larger is the cost (border is not smooth), the fewer misclassifications are allowed. If the cost is too large, it can lead to overfitting.

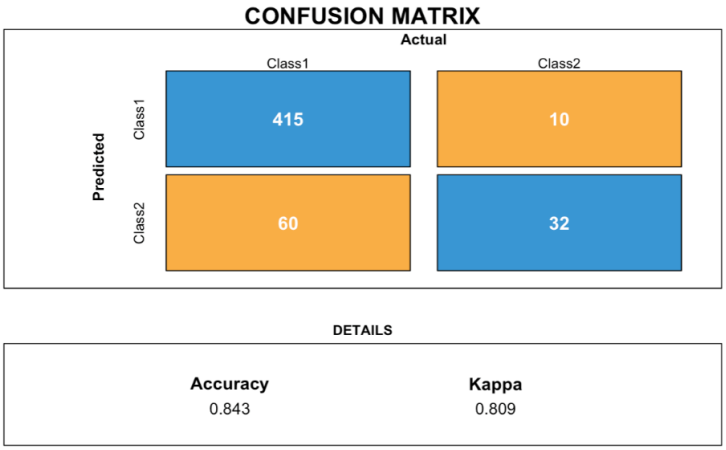


Figure 41: Confusion matrix of the SVM model using TF-IDF, LSA and centroids

	Class: JoeBiden	Class: KamalaHarris	Class: realDonaldTrump	Class: VP	Class: BarackObama
Sensitivity	0.7518	0.5723	0.91589	0.9741	0.9465
Specificity	0.9402	0.9619	0.99183	0.9587	0.9734
Pos Pred Value	0.7684	0.7694	0.90741	0.8679	0.8967
Neg Pred Value	0.9349	0.9101	0.99264	0.9925	0.9867
Prevalence	0.2088	0.1817	0.08036	0.2178	0.1964
Detection Rate	0.1570	0.1040	0.07360	0.2122	0.1859
Detection Prevalence	0.2043	0.1352	0.08111	0.2445	0.2073
Balanced Accuracy	0.8460	0.7671	0.95386	0.9664	0.9599

	Class: AOC
Sensitivity	0.9281
Specificity	0.9762
Pos Pred Value	0.8353
Neg Pred Value	0.9905
Prevalence	0.1149
Detection Rate	0.1066
Detection Prevalence	0.1277
Balanced Accuracy	0.9522

Figure 42: Statistics by class for the SVM

With SVM, there is a huge improvement in the balanced accuracy compared to the random forest, especially for AOC. In fact, sensitivity and specificity are much well balanced, except for Kamala Harris and Joe Biden where the model is better at predicting true negatives.

The accuracy of the prediction is 0.843. The SVM model outperforms the random forest model and thus, is better at predicting the author of a tweet.

4.3 Prediction of the political party

To build features and model in order to predict the political party to which the author of a tweet belongs, we followed exactly the same process as the one used to predict the author of a tweet. Again, the combination of TF-IDF, LSA and centroids received the highest accuracy score when training a random forest. Finally, we compared this score with several models using the same combination of features but only the SVM model outperformed the random forest model.

Support vector machine (SVM)

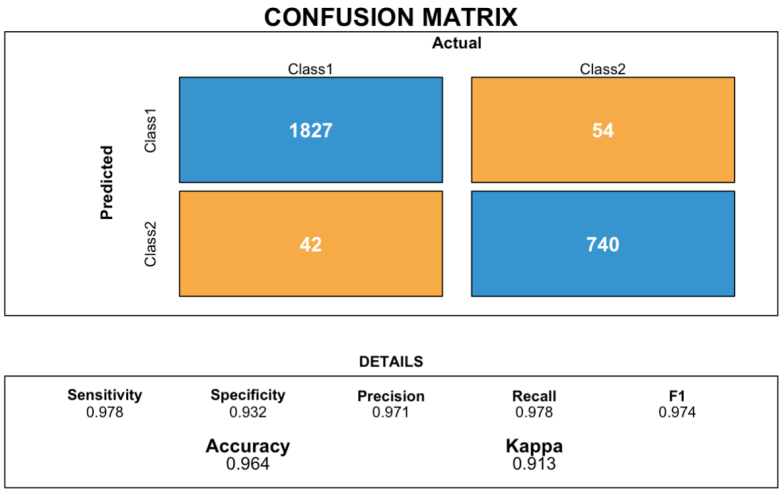


Figure 43: Confusion matrix of the SVM model (Political party)

Obtaining an accuracy of 0.964 but also high and balanced specificity and sensitivity, the SVM model can be considered as a very interesting tool to predict a political party.

5. Conclusion

Throughout this study, we were able to determine several patterns among all tweets collected. First of all, we can say that Trump is the most popular figure among other politicians in terms of the number of retweets despite not doing better than Obama in terms of liked tweets. Trump is also the most active on this social network.

We also realized that the number of tweets is growing as the election approaches and then decreasing drastically in the following days. The use of this social network varies a lot depending on the politician and his political party. We noticed very different strategies in the two political sides. Certain terms are more used either by the Republican camp or the Democratic camp. In this way, we can somewhat guess the electoral program of each party.

When analyzing the sentiments, it was delicate to spot differences between documents. The predominant feelings were associated to positivity and trust. In addition, all documents were related to an overall positive feeling. It was also complicated to establish well-defined clusters when analyzing the similarity between the documents. Nevertheless, we found that documents including the pre-election tweets were distinguishable from the post-election ones. But still, we saw that by using word embedding, we could create more meaningful clusters allowing us to group certain words according to a specific theme.

As regards to the topic modeling part and more specifically the proportion of each topic in each of the documents, we observed that every document is mainly related to a different topic and could have been useful for the supervised learning.

Finally, we took note that the best combination of prediction models features was related to the weighted frequency, the latent semantic analysis (LSA) and the centroids. When comparing our models, it was the vector machine support that allowed us to better classify the author of a tweet as well as his political party. On the other hand, the prediction was more accurate for the classification of the political party than for the author of a tweet. Imagining not knowing the political party, the scores would have been much more disappointing when it came to predicting the author of a tweet. The same goes for predicting the political party without knowing the author of the tweet. Thus, we can say that our models cannot be used in combination and that in order to predict the author of a tweet, it is certainly essential to know the political party and vice versa.

In closing, this analysis showed us that it is possible to analyze Twitter messages and that this approach could be used in many ways. The use of Twitter in politics is very widespread in the USA, unlike in other countries where this analysis would not make much sense. But as a follow-up to this study, we could imagine, on a very large scale and with a huge amount of data, analyzing the opinion of the American people via Twitter and knowing which candidate has the best chance of winning the elections.