French Presidential Election Candidates Tweets

Nicolas Derumigny Emma Kerinec

May 23, 2017

1 Introduction

We have worked on a dataset of more than 3‘000 tweets of the 11 French presidential election candidates posted during few years. We have worked in Python 3 and used the following packages: Sklearn, Numpy, Matplotlib.

2 Preprocessing

We have chosen to keep only relevant words, in order to not give any power to a lot of very used words that do not bring any information (e.g. "le"). For that, we used a list of uninterested words (“stopwords”), improved with some specific words like “https”. We have also distinguish hashtags and other words as they do not have the same meaning in a tweet. Because meaning of words is really context dependant and difficult to process we have choose to not consider semantic.

2.1 Simple Data

Our first idea has been to compute frequencies of each word and each hashtag for each candidate. We have seen that for a lot of candidates the few most frequent words confirm the view that people usually have about their opinions. We can see it with Philippe Poutou (cf images 1).

3 Distance

Definition 3.1. Distance bewteen two sets of words:
Measure the proportion of words that are different between two set of words S_1 and S_2:

$$d(S_1, S_2) = \frac{1}{2} \cdot \Big(\sum_{\substack{w \in S_1 \\ w \notin S_2}} f(w) \quad + \quad \sum_{\substack{w \in S_2 \\ w \notin S_1}} f(w) \Big)$$

Word	Frequency
contre	1.25%
gouvernement	0.49%
travail	0.45%
paris	0.40%
droite	0.38%
solidarité	0.38%

Hashtag	Frequency
#npa	11.03%
#loitravail	2.98%
#grèce	1.73%
#migrants	1.66%
#poutou2017	1.45%
#hollande	1.18%

Figure 1: Most used words and hashtags for Phillipe Poutou

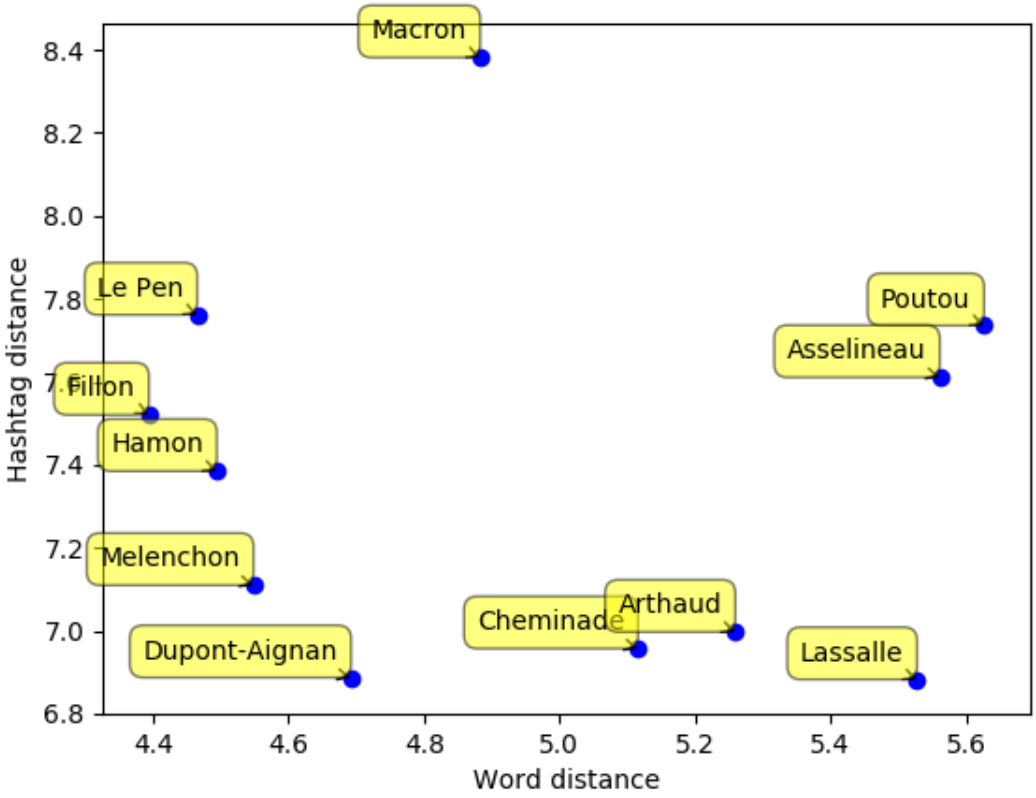

Figure 2: Sum of the distance to the other candidates, all words on x and hashtags on y

where $f(x)$ is the frequency of apparition of the word x.

In that way we can compute the distance with respect to words and the distance with respect to hastags between two candidates.

It is interresting to see that Emmanuel Macron is the most different in hashtags and that Marine LePen et Jean-Luc Melechon often seen as extreme candidates are really common in the sense of the words they use (cf images 2).

Poutou	Melenchon
Cheminade	Fillon
Arthaud	Hamon
Lassalle	Le Pen
Asselineau	Macron
	Dupont-Aignan

Figure 3: Words Similarities

Marcon	Melenchon
	Fillon
	Hamon
	Le Pen
	Dupont-Aignan
	Poutou
	Cheminade
	Arthaud
	Lassalle
	Asselineau

Figure 4: Hashtags Similarities

Melenchon	Fillon
Poutou	Le Pen
Cheminade	Macron
Hamon	Asselineau
Arthaud	Dupont-Aignan
Lassalle	

Figure 5: Hierarchical clustering on hashtags before the campaign

Melenchon	Poutou
Fillon	Cheminade
Hamon	Arthaud
Le Pen	Lassalle
Macron	Asselineau
	Dupont-Aignan

Figure 6: Hierarchical clustering on hashtags during the campaign

4 Data Mining

4.1 Kmeans and Hierarchical clustering

In order to partition candidates we have implemented k-means and Hierarchical clustering. For two clusters on all tweets we obtain the same result with both methods (cf images 3 and 4).

We can find that for the words there is a repartition minor/leading candidates and that for the hashtags Macron is alone (cf images 3 and 4).

4.2 Variation over time

We have observed similar partitions for different periods, since our data cover a few years.

We can see that before the campaign the clustering is right/left candidates but during the campaign it is major/minor candidates (cf images 5 and 6). We can observe this evolution during the campaign by divided it in some periods.

4.3 A priori algorithm

We have use the a priori algorithm in order to see the most frequent words associations for each candidate.

We have observed that there is a huge number of auto-citations and a difference in the number of items for each candidates (cf images 7).

```
 Melenchon
item:  ('c',) , 0.120
Poutou
item:  ('#poutou2017',) , 0.140
item:  ('c',) , 0.112
item:  ('soir',) , 0.105
Fillon
item:  ('fillon2017_fr',) , 0.129
item:  ('france',) , 0.114
item:  ('francoisfillon',) , 0.184
item:  ('veux',) , 0.105
Cheminade
item:  ('#cheminade2017',) , 0.422
item:  ('#cheminade2017', 'jcheminade') , 0.182
item:  ('#presidentielle2017',) , 0.137
item:  ('jcheminade',) , 0.412
Hamon
item:  ('veux',) , 0.101
Arthaud
item:  ('c',) , 0.132
item:  ('lutteouvriere',) , 0.112
item:  ('lutteouvriere', 'n_arthaud') , 0.102
item:  ('n_arthaud',) , 0.592
Le Pen
item:  ('#debattf1',) , 0.109
item:  ('#debattf1', '#legranddebat') , 0.106
item:  ('#legranddebat',) , 0.108
item:  ('france',) , 0.112
item:  ('francais',) , 0.113
item:  ('nos',) , 0.100
Macron
item:  ('c',) , 0.123
Lassalle
item:  ('#presidentielle2017',) , 0.330
item:  ('france',) , 0.106
item:  ('jean',) , 0.109
item:  ('jeanlassalle',) , 0.249
item:  ('lassalle',) , 0.115
Asselineau
item:  ('#asselineau2017',) , 0.240
item:  ('#frexit',) , 0.229
item:  ('#frexit', '#asselineau2017') , 0.161
item:  ('#presidentielle2017',) , 0.120
item:  ('asselineau',) , 0.207
item:  ('francois',) , 0.207
item:  ('francois', 'asselineau') , 0.180
item:  ('upr_asselineau',) , 0.229
Dupont-Aignan
item:  ('#nda2017',) , 0.129
item:  ('#nda2017', 'dlf_officiel') , 0.117
item:  ('dlf_officiel',) , 0.677
item:  ('dupontaignan',) , 0.801
item:  ('dupontaignan', '#nda2017') , 0.121
item:  ('dupontaignan', '#nda2017', 'dlf_officiel') , 0.116
item:  ('dupontaignan', 'dlf_officiel') , 0.660
item:  ('dupontaignan', 'nicolas') , 0.145
item:  ('dupontaignan', 'nicolas', 'dlf_officiel') , 0.127
item:  ('francais',) , 0.108
item:  ('nicolas',) , 0.160
item:  ('nicolas', 'dlf_officiel') , 0.127
```


Figure 7: Use of the a priori algorithm

5 Observations

We have been surprised that Emmanuel Macron who win the election extremely distinguish himself in the use of hashtags but not a lot in the use of words. It has also been interesting to see that the distinction rigth/left is present during common periods but erase in favour of a partition minor/major candidates during the campaign. It is important too to notice that the number of auto-citation and citation of other candidates is extremely important. However, we have to deal the drawback of not using semantic indeed there is no link between different words with the same meaning, and even between singular and plural words. We have also use the sklearn.feature_extraction module in order to represent data as a sparse matrix of tokens. Sadly, we could only measure similarities between candidates without further explanation, as the original meaning of te tweet is lost.