Classification of US senator tweets
Machine Learning for Natural Language Processing, 2021

Justin Aguenier
justin.aguenier@ensae.fr

Violaine Courrier
violaine.courrier@ensae.fr

Abstract

For this project, we have built a classification algorithm that aims to recognize US senator political party based on their tweets. We show got no [1] results. On top of that, we used Word2Vec to understand semantic and structural differences between the tweets of democrats and the republicans.

1 Problem Framing

We wanted to use NLP in politics. By looking at the French presidential elections, we found some ideas. We decided to search for an algorithm that could predict political preferences based on textual data. As it was time-consuming to gather french data [2], we decided to use US tweets. In such a context, we wanted to answer the following questions : **Are we able to detect political preference based on a tweet ?**. Solving this problem may be interesting to lead a political analysis. For instance, it could be useful to identify political trends, to convey surveys, etc. Classifying is interesting, nevertheless, finding language differences between different political parties could be even more powerful. Therefore, we quickly tried to identify structural disparities in the language structure between political preferences.

2 Experiments Protocol

Our main problem matches a **classification task**. The different step of the protocol are fairly simple. We wanted to :

- Discover the unexpected specificities of our dataset [3] thanks to exploration and data visualization
- Clean our data
- Tokenize our data
- Choose an embedding for the data
- Set up our network
- Adjust network parameters
- Explore results (and repeat . . .)

We intended to test the performances of our model by looking at different classification metrics : precision, accuracy, F1-score.

To quickly understand disparities in the language structure between political preferences, we wanted to create Word2Vec. To do such a comparison, we wanted to look for the closest word (in terms of cosine similarity) to a target word. We excepted differences.

3 Data description

After loading the datasets (see part I), we first looked for missing values and duplicates (III). Then, we searched for corrupted data (IV). We found tweets that did not respect the normal tweet structure, and we deleted them. Some questions about our dataset needed to be answered before training any classifier. We found that our dataset was balanced between Democrats and Republicans. We set the Independents as Democrats, as the two Independents senators has left wing ideas (see Bernie Sanders for example). Therefore, our problem is a **binary classification task**. Then, we plotted the tweet distribution over time (see Figure 5), which is not well-balanced. By interest, we looked for the most active senators in our dataset (see Figure 2 and interactive figures on the notebook).

4 Preprocessing

For the preprocessing part (V), we deleted **#, HTML and URL** from the tweets. We also

[1] Our algorithm only predicts the same class for the inference part. We will send you updated results as soon as possible.

[2] No dataset available

[3] Link to data

deleted the **stop words** as they do not seem interesting for a classification task. We removed the different encoding errors and smileys. We search for **multi-word** expression. We use the TweetTokenizer from gensim.

To observe the vocabulary difference between parties, we made a word cloud for each party. We observe clear differences between the two word clouds (see Figure 6 and 8).

We also created a Word2Vec structure for each party. We looked for the closest word for some target word. We identify a clear distinction(see Table 2). For instance, the closest word to the word *environment*" are *planet*, *public lands*, *ecosystems* for the Democrats and *consumers*, *reliable*, *job creation*for the Republicans. This result seems to make sense for us knowing global political ideas. Thanks to a PCA, we plotted both Word2Vec vectors spaces (see Figure **??**).

5 Classification

We followed the lab 3-4 pipeline to train our neural net (VI). We transformed our tweets to numerical values. We also needed to make our dataset PyTorch compatible. We then used the train/test/validation framework to split our dataset. The network was set up for binary classification. We used a long-short term memory structure (LSTM), as we want to capture some *long term* dependencies in the sentences. The loss function is a binary cross entropy (VII). We looked for the training and validation losses to see how our model reacted to the training (see Figure ...)

To improve our network, we explored different techniques. We changed the learning rate, added a dropout. We tried to improve our embeddings. We change the optimizer (we eventually selected the Adam optimizer).

The results are the following ones (IX) :

Accuracy	Precision	F1-score
0	0	0

Table 1: Results

Unfortunately, our algorithm always return the same value. We will try to debug the code, and we will send you the results as soon as possible

6 Discussion/Conclusion

To put it into a nutshell, we used a common pipeline to classify senator tweets. The final model used is an LSTM network. We obtained 0. [insert results when available]. To go further, we thought about merging the datasets below to train our network over more data :

- Obama's tweet
- Partisanship tweet
- Democrat vs Republican

To improve our model, more data and more training are obviously necessary. We could also add some layers to our model (the lack of speed in Colab was problematic to make even more complex model and train them in an acceptable time). We could have used simpler classifying algorithm such as logistic regression, SVM or random forest. We should have done this

Other approach are possible to solve this classification problem. We could rely on sentiment analysis for some hashtag. For instance, a hashtag linked to guns may imply bad sentiment for the democrats.

7 Appendix

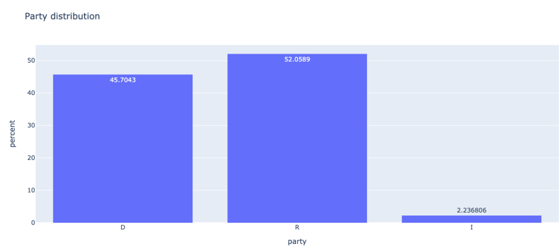

Figure 1: Tweet distribution

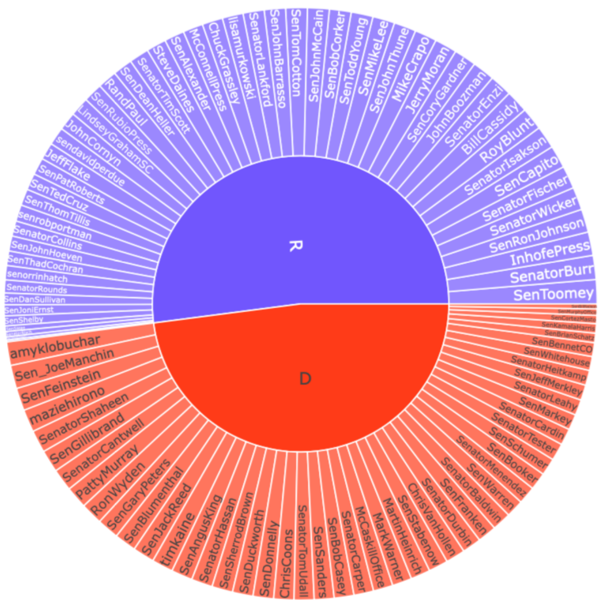

Figure 2: Number of tweets per senator

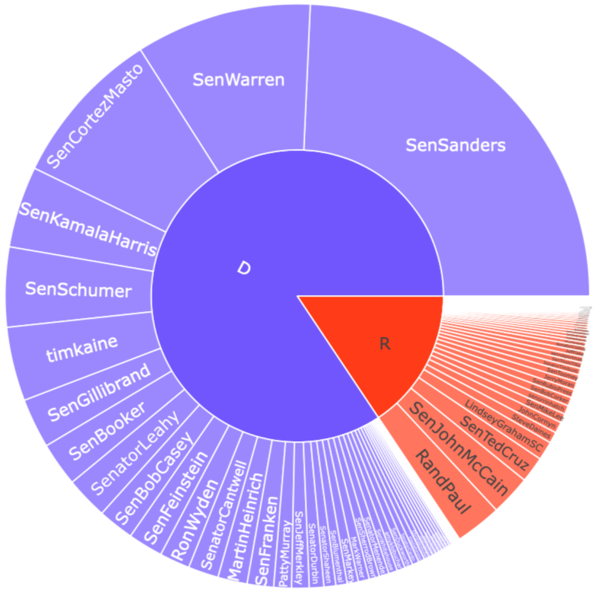

Figure 3: Number of retweets per senator

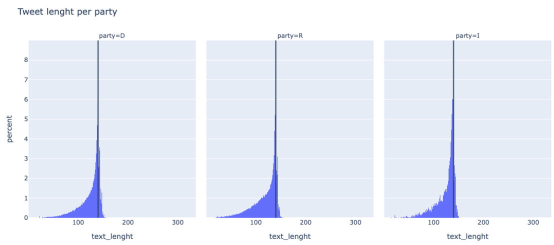

Figure 4: Tweet lenght for the different parties

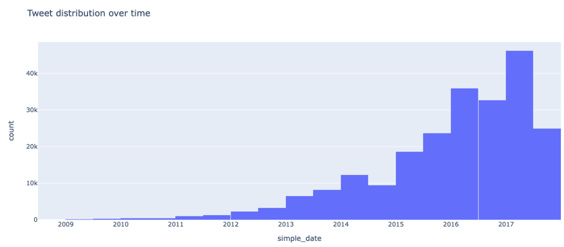

Figure 5: Distribution of tweets over time

Figure 6: Democrat word cloud

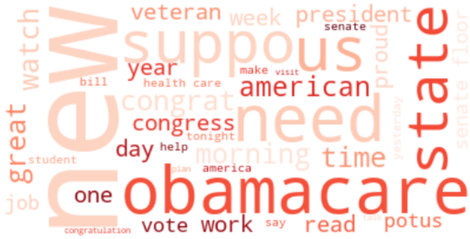

Figure 7: Republican word cloud

Figure 8: ACP

Target word	Closest Democrat	Closest Republican
Business	Biz, bank, small biz	biz, entrepreneurship, economic development
Science	Scientific, denier, scientific research	math, academic, engineering
Welfare	educational, educational opportunities, maternal	lowers, medicaid, dodd frank
Finance	finance committee, comm, epipen price	cmte, banking, comm
Environment	planet, public lands, ecosystems	consumers, reliable, job creation
American	America, Cuban, nation's	British, America's, Cuban
Obamacare	estate tax, individual market, aca	obamacare's, aca, ocare
Family	loved ones, parent, child	condolences family, jane, loved ones
Politics	partisan politics, aside, democracy	political, rather, agenda
War	iraq, syria, civil	aleppo, declaration, syria's
Gun	commonsense gun, background checks, common sense]	gun owners, due process, sanctuarycities
Security	security advisor, prayer breakfast, defense authorization	registry, interest, defense authorization
Economy	economic growth, econ, local economies	economic growth, job creation, growth
Health	healthcare, hlth, mental health	hospice, behavioral health, chronic
Democarcy	elections, democracies, principle	strength, europe, principles

Table 2: Word2Vec closest word for the two parties