RER B Tweet analysis
Machine Learning for Natural Language Processing 2020

Charly Andral
ENSAE
charly.andral@ensae.fr

Colombe Saillard
ENSAE
colombe.saillard@ensae.fr

The notebook can be find on Charly's github [1]

Abstract

We analyze the tweets of the official RERB account @RERB as well as the replies of other users to this account. We first perform a basic data analysis of the data by using descriptive tools such as temporal analysis. Then, we clusterize the 2019 tweets into 18 clusters. We train a classifier on a hand-labelled set to detect the sentiment of the tweet : negative, neutral, positive. Finally we study the influence of the clusters on the average replies they create.

1 Problem Framing

As the famous French journal "Le Point" recently titled : "Le RER est le transport le plus anxiogène devant le métro" (Trevert, 2020). Indeed, the daily experience of every Parisian - and especially of the ones who are lucky enough to have to commute every day from the inner city to the suburbs - is made of delays and "pannes de signalisation" which are sometimes psychologically costly. This sensitive topic is the subject of many daily informal conversations in which we can often hear that the problem is not the traffic incidents themselves, but also the lack of information surrounding them, or its inaccuracy. This study is an attempt to look at this criticized line of communication with the objective glasses of data science : by analyzing the tweets from the RERB account and their replies with Machine Learning tools, we aim to understand the dynamics and the characteristics of the communication set up by the RERB institution, and to have a broad comprehension of the reactions it generates.

Taking a quick look at the tweets and their replies, we see that their frequency, their length and the size of the vocabulary they mobilize are strongly time dependent on different scales and in quite comparable ways :

- they are increasing throughout the years between 2013 and 2019,
- they are almost constant throughout the months, except a decrease in august and a slight increase from september to january,
- they are strongly decreasing in the week-ends relatively to the worked days,
- there are peaking in the morning (especially between 7 and 9 am) and in the evening (between 4 and 6 pm).

But the descriptive tools do not really allow to go beyond these remarks, and especially to make relevant conclusions on the content of the tweets. We tried to have a look at the most frequent words of the vocabulary grouping words according do different variables, but these groups were not relevant and we did not obtain any interesting results. That is why the ML analysis was needed to construct relevant and easily analizable clusters of words and to assess the reactions they generate.

2 Experiments Protocol

2.1 Data

The data were scraped from Twitter using the Python package Twint. We constituted two databases : one with all the tweets written by @RERB and another with all the tweets addressed to @RERB. For the classificaton task, we manually classified around 3600 replies in three classes : negative, neutral or positive replies. The graphs are in the notebook.

2.2 Model used

The model we used for the NLP core of this project is camemBERT (Martin et al., 2019). The model

[1] https://github.com/charlyandral/NLP-ENSAE-2020

is not retrained on our data. For the clustering and classification task, we use the embedding of the tweet (the pooled vector) through camemBERT, then respectively K-means and XGBoost.

2.3 Implementation

All the implementation is in Python. In particular, camemBERT is loaded with Huggingface's *transformers* package. K-means is from the *scikit-learn* library.

3 Results

3.1 Clusterization

We clusterize the data using K-means with 18 clusters. This number, chosen with the help of differents metrics, is also chosen not to be to large (too many clusters to analyze) neither to large (not enough variability to exploit). To visualize the clusters we plot the data in the 2D plane using TNSE and one wordcloud for every cluster. This allows us to interpret the clusters.
We can conclude from our clusterization that some kinds of messages are quite clearly distinct from the others : the ones from the Community Managers,, the one announcing planned works... But the messages announcing daily incidents causing perturbations and the traffic going back to normal are much more indistinct from each other.

3.2 Classification

We classify the 3600 hand-labelled by keeping 80% for training XGBoost. We obtain an accuracy of 75% on the test set. The results are quite good knowing that even by hand the tweets were difficult to classify.

3.3 Sentiments in the clusters

With the classifier we built we can now guess the sentiment of the replies to @RERB. We compute the mean of the sentiment for every cluster and see if the mean sentiment is coherent with the interpretation we made of each cluster.
The average grades of the clusters suggest that more personalized messages, like the one from the Community Managers are more appreciated than general indistinct information about daily incident. Yet, the average impression is that overall, the reactions are very negative.

4 Discussion/Conclusion

Our analysis is based on several assumptions : the number of cluster (which is rather subjective), the way we labelled by hand the dataset (different people classify differently the same tweets). The result also depend on randomness as consecutive runs of the same algorithms (including k-means) lead to different clusters that can be interpreted differently. These few limitations could explain some slight incoherences in our results, like for example the fact that the clusters associated to the traffic going back to normal are relatively badly graded.

References

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Émilie Trevert. 2020. Le rer est le transport le plus anxiogène devant le métro. *Le Point*.