ILLINOIS INSTITUTE OF TECHNOLOGY

Online Social Network Analysis

work realised by

Agnes Gaspard
Alan Collet
Rémi Blaise

Do people believe in fake news?

Taught by
Dr. Aron Culotta

Master of Computer Science, Fall 2019

1 Abstract

While the usual approach in fake news related machine learning is to build fake news detector, we decided to engage in a new challenge: analysing people's belief of an article based on their comments. Basing our work on a Politifact dababase found on Kaggle, we made a collection script retrieving comments from Reddit about politic news. We then manually labeled 3,000 of these comments and programmatically augmented our feature set with sarcasm and sentiment analysis. We trained 3 classifiers: one Logistic Regression, one Naive Bayes and a 2-layer Neural Network classifier. We hence obtained different accuracies we can compare to a baseline method. We used confusion matrices as complementary metrics to evaluate the relevance of our results. If our accuracies are not so above of the baseline as we initially expected, they still represent an improvement. Many biases in our data are also impacting the obtained classifiers.

2 Introduction

Fake news is a plague of the Internet. In response, research on fake news detection has exploded in the few last years. Very few research however has been made to analyse readers' belief of the news. Building a classifier identifying people's belief in fake articles could help analyse fake news propagation, people's credulity and critical thinking about these news.

Applications coul be: identifying critical topics whose reader could have use of helper tools; analysing the impact over time of fake news debuking: are there group movements and changes of mind, what influence can popular people have.

The goal of our project is to make a classifier able to guess if a user believes or not to a news article via one of his comments. It will involve NLP methods similar to sentiment analysis and machine learning classification methods.

3 Related work

The essential subject about Fake News detection in our papers and more generally is to determine if a news is fake or not with only his content wihtout taking care about people reactions over this. There is different way to detect it , some with a bag-of-word principles, some with LSTM neural Networks and other using a k-means clusterig over a vectorial representation of the content with word2vec. Our goal vary from these articles because our goal is to use the poeple reaction in order to determine if a news is fake or not. Another advantage is that with this technique we can observe how the belief is evolving during the time.

4 Approach

We used scientific programming tools along this project: pandas for data handling, numpy, sklearn for machine learning tasks, Keras for neural network tasks and textblob for NLP.

In order to get features, we chose bag of words representation for sentiment analysis by an existing model (Textblob) and by lexicon based method with naive Bayes. First method gave subjectivity and polarity score while second only considered polarity. For a sarcasm indicator provided by an existing model, we used Tf*idf representation. Tf*idf is a technique to give importance to words according to their number of occurrences in a document.

To classify our comments, we considered 3 models: naive Bayes, logistic regression and simple neural network.

- Naive Bayes works as follow: we count word occurrences, total number of words and number of comments per class. We then use Bayes theorem to labelize a comment with the probability of a class given this comment.

$$\log\big(P(Y = y, X)\big) = \log(P(X|Y = y)) + \log(P(Y = Y))$$
$$\log\big(P(Y = y)\big) = \log \frac{d\_count[y]}{d\_count}$$
$$\log\big(P(X|Y = y)\big) = \sum_{w \in V} \log \frac{w\_count[w][y] + \alpha}{w\_count[y] + \alpha|V|}$$

$$P(Y = y|X) = \frac{P(Y = y, X)}{\sum_{y' \in Y} P(Y = y', X)}$$

Figure 1: Equations used in naive Bayes classification

- With Logistic regression, the main goal is to find the best sigmoïd fonction that split the data in two differents classes. With more than 2 original classes (as it is here), the whole job is divided in three different subproblems (One versus all approach). 3 clasifiers will be created to separate a class with the other ones.

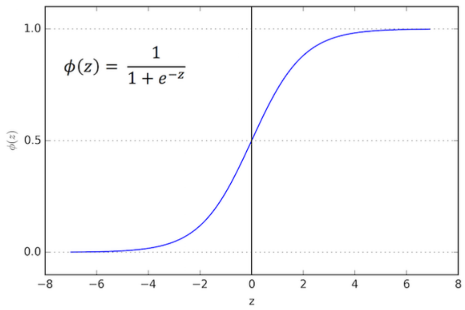

Figure 2: Sigmoid function used in logistic regression

- We also tried a very simple neural network model. Our model is compound of 2 layers of about 20 neurons and try to predict 3 probabilities (one for each predicted class). We used the accuracy curve on the set dataset to find a good architecture.

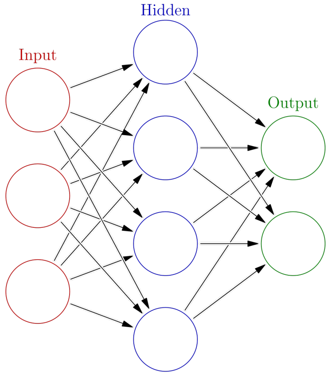

Figure 3: Simple neural network with one hidden layers

Both features (sarcasm and sentiment score) were used in logistic regression and simple neural network models. Naive Bayes relies only on words as features.

5 Experiment

5.1 Data

A database of Politifact fake news (mostly false and pants on fire) including articles' url, healines and id, among other details, was used to collect comments on Reddit. We gathered every first-comments (no responses to comments) associated to each post containing a Politifact article's headline. Thus, a database of more than 13000 comments was built. Then we labelized 3000 random comments among these 13000 with the labels: -1 for non-believer; 1 for believer; 0 for non-belief related or ambigious comments.

When we analysed our data, we realized that it was unbalanced. Indeed, numbers of labelized comments per class were:

- 204 comments for category "non-believer"
- 1992 comments for category "believer"
- 798 comments for category "none of the above"

It probably hindered the training of our classification. We thought of labelizing more data but considering our results (described below), it would have required a large amount of labelized data, in particular in "non-belief" class, in order to improve our models. Hence, we tried to SMOTe technique to create fake data quickly. It is a method representing data graphically and predicting similar data through k-nearest neighbours algorithm. Unfortunately, we did not succeed in using it, so we attempted to reduce other classes' size and add weight to "non-belief" class to improve our resutls.

Furthermore, some articles had lots of comments related, hence we feared subject-oriented vocabulary and we shuffled data before labelizing it. We also found lots of fights between users and non-related comments during labelization. As to not disturb the training of our models, those comments were put in "none of the above" class. Finally, the fact that authors did not comments on their belief but on the content of the article itself probably interfered with our classification.

As no related works/science papers to our project have been published, we have no model to compare our results with. As we do basic classification, we consider the randomness of a classifier. For 3 classes, randomness is 33% for our models and 50% for the binary class (Naïve Bayes Classifier).

5.2 Results

We chose to evaluate our results with mean accuracy and confusion matrix on our 3 models.

- For naive Bayes, our best reult is 56% mean accuracy for two classes (belief and non-belief), which is almost random classification, when both classes have the same size. We also tried for 3 classes but unfortunately, results were even worse.

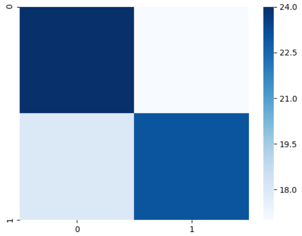

Figure 4: Naive Bayes confusion matrix

- Logistic regression have 49% of accuracy for 3 classes, and is our best classifier. It is still better than a random classifier by more than 15% which is a good improvment.

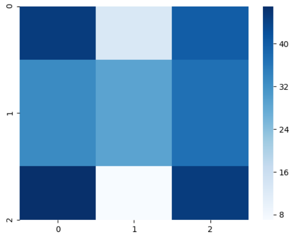

Figure 5: Logistic regression confusion matrix

- The neural network model gives an accuracy of about 47% on 3 classes, with an accuracy on the training which quickly reach 100% (end of training / overfitting).

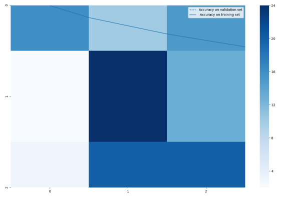

Figure 6: Neural network confusion matrix

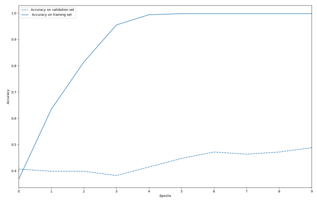

Figure 7: Training and validation curves

We represented word occurences in each class with WordCloud:

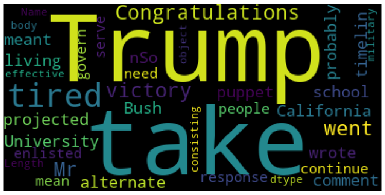

Figure 8: Class "1"

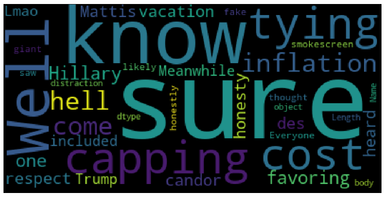

Figure 9: Class "-1"

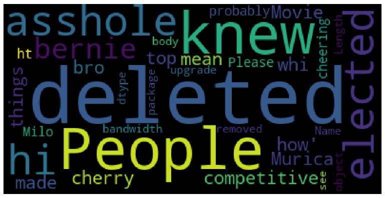

Figure 10: Class "0"

Our comments are realted to Politifact news so it seems logical to see "Trump" as one of the largest number of appearances in "belief" class. Same for deleted or some improper words in class 0 because it contains deleted comments and fight between authors. The biggest word in "non-belief" word is "sure", which could come from the added sarcasm feature we can also find words "Fake" or "smokescreen" which are really revelant of a no trust class .

6 Conclusion

We realized how a balanced and quality database is important to make a good classifier. We can modify our models as much as we want, our database prevented us from improving our result.

Labelizing 3000 comments also showed us how people tend to react quickly to facts, without always checking the source, and can be sometimes gullible and vehement.

To go further, we thought about part of speech analysis with Textblob and unsupervised learning methods as k-mean clusterization.But most importantly, this project needs more labelized and quality data.

7 References

References

[1] *Kaggle Fake News Database*

[2] Kai Shu, Amy Sliva. *Fake News Detection on Social Media: A Data Mining Perspective*
2017

[3] Tencent Inc. *Recurrent Attention Network on Memory for Aspect Sentiment Analysis*
2017

[4] Rowan Zellers, Ari Holtzman. *Defending Against Neural Fake News*
2019

[5] Alec Radford, Rafal Jozefowicz, Ilya Sutskever. *Learning to Generate Reviews and Discovering Sentiment*
2017

[6] Devamanyu Hazarika, Soujanya Poria. *CASCADE: Contextual Sarcasm Detection in Online Discussion Forums*
2018

[7] Bjarke Felbo1, Alan Mislove. *Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm*
2017