Sentiment analysis to propose articles adapted to the reader's mood Machine Learning for Natural Language Processing 2020

Bertille Daran
ENSAE
bertille.daran@ensae.fr

Laure Heidmann
ENSAE
laure.heidmann@ensae.fr

Abstract

As the number of articles published increases, it would be interesting for online newspapers to add a recommendation parameter that takes into account the mood of the reader. For example, a reader feeling sad would be more interested in articles that could cheer him up. So, we undertook, using several NLP methods applied to a set of New York Times comments, to predict the dominant emotion triggered by an article as it could be in the comments. We proceeded to an analysis of feelings and emotions. Our results showed a resurgence of topics with mostly negative comments. We conclude by questioning the use of comments as an accurate means of measuring the effect of an article on readers.

1 Problem Framing

The distribution of newspapers online has led to a quasi-permanent flow of information. However, this constant flow of both positive and negative articles can become overwhelming. The times we live in are a good example of this problem: some people can be drowned out by the incessant news updates on Covid-19. This could affect them and make their thoughts even more negative. Thus, a good feature of online newspapers would be to offer articles to the reader, taking into account the mood previously determined by the reader. From this perspective, the use of NLP could improve the prediction of which articles to recommend to the NYT based on readers' feelings. Our hypothesis is that the feelings expressed in the comments are likely to predict the general feeling that is generated when reading the article.

2 Experiments Protocol

Data - Our data of interest are the comments published on the New York Times website during the month of April 2017[1]. The data set contains the full commentary, as well as the names of the sections to which the article belongs. By evaluating the dominant emotion expressed in each commentary, we will be able to indicate which feeling is most often triggered by a specific topic. We also used a data set of tweets categorized into 13 emotional categories to train our models.

Models - The difficulty of our problem relies in two major tasks: labelling our data and predicting labels. In regard to labelling, we explored three different methods: manual labelling, automatic labelling leveraging sentiment dictionaries and the use of an additional dataset already labelled. Concerning the predictions, we also set up different models. We applied classification models such as SVM and Random Forest. To finetune our predictions, we used BERT for sentiment analysis with negative and positive labels. Finally, we implemented Keras algorithm for predicting sentiment analysis on multi-labelled data.[2]

Implementation - To label our data, we decided to work either with four emotions - *anger*, *sadness*, *trust* and *joy* - or with only *positive* and *negative*. To label in binary data - *negative* or *positive*, we used the lexicon opinion: a list of English positive and negative opinion words or sentiment words (around 6800 words)[3]. For all the comments, we counted a score: -1 for a *negative* word and +1 for a *positive* word. The result was rather harmonious, with a Gaussian concentrated in 0. Finally, we gave the label 0 for *negative* comment to the comments with a score lower than 0 and the label 1 for *positive* comment to the comments with a score higher than 1. To label the data with a set

[1] https://www.kaggle.com/aashita/nyt-comments/data/

[2] `https://colab.research.google.com/drive/1egpnBmcI--AxjPpatoDzK3Qwe0RgX3_p`

[3] https://www.cs.uic.edu/~ liub/FBS/sentiment-analysis.htmllexicon/

of sentiments, we used the NRC Emotion Lexicon: a list of English words and their associations with eight basic emotions and two sentiments[4]. We calculated a score for each emotion and feeling. We then selected the emotion or feeling with the highest score, and merged the different categories for clarity, keeping the emotions *anger*, *joy*, *trust* and *sadness*. For both procedures, we did the same by selecting only the first sentence of each comment. In regard to the predictions, we first implemented algorithms using SVM and Random Forest as classifiers. To preprocess the data, we tokenized the comments, and then we vectorized the text with the function *CountVectorize*. We keep the number of features equal to the size of the whole vocabulary for this initial analysis. We train the SVM and Random Forest classifiers first with the dataset of labelled tweets, and second with our own dataset labelled with sentiment dictionaries. In a second step, we implemented a fine-tuning on the original version of BERT. We made a preliminary treatment of the data by creating the *SSTDataset* class which allows to get, tokenize and pad the comments. Then, we define the sentiment analysis for negative and positive labelled comments model using pytorch. Here, we used the pretrained *Masked Language Model* as one module of our sentiment analysis model. Then, we trained and evaluated our data set. Finally, we implemented another model with LSTM, using the Keras library to build a multi-label sentiment analysis model. Before that, we used the GloVe algorithm to get word embeddings.

Model training - In the part using BERT, we trained our databases using different parameters. We used Binary Cross Entropy with logits loss as loss function and the 'bert-base-uncased' as model. The ones we found the most optimal were a batch size equals to 128, a learning rate equals to 1e-5 and 5 epochs. In the part using Keras, we set the parameters as following: we set at 128 the dimensionality of the output space, at 6 the batch size and at 5 the number of epochs. However, we did observe some overfitting in both models that we prevented reducing the network's capacity.

3 Results

The first challenge we faced was to construct a dataset with labelled comments that we could use to train our models. Our work on the dataset of labelled tweets resulted in disappointing results. With simple models, it was very difficult for our classifiers to distinguish the different emotions, even when we reduced them to only *positive* and *negative*. We nevertheless tested the models thus trained on the comments of our data of interest that we manually annotated. Although the negative emotions was rather well predicted, the classification was completely confused in front of positive ones. So, we concluded that the dataset of tweets was not suitable for our project. Then, we trained the SVM and Random Forest models with the comments that we labelled with dictionaries. The results were very encouraging even with a simple prepocess of the text that kept the whole vocabulary in the vectorization - we limited the analysis here to the comments' first sentence to adapt to the simplicity of the methods. The results were particularly good for SVM (F1-score : 0.9 for positive and negative). We tried to optimize the Random Forest model, but the usual search for better parameters was not productive. We interpreted this as a consequence of the very large size of features, that could be reduced with better embedding methods. Results were rather good using BERT implementation, for both training on all the comment (F1-score: 0.8) and on the first sentence of the comments (F1-score: 0.9). However, results were disappointing for Keras implementation because the algorithm was overfitting too much. So we didn't use it for the final results.

Finally, with our different predicting algorithms, we could assess which emotion was predominantly present in the comments for every topic. We observed a reassuring coherence through the different methods - Random Forest seemed slightly out of tune as we could presume from its accuracy.

4 Discussion/Conclusion

We found that almost all topics trigger negative emotions in the comments. This could stem from Internet users' tendency to comment when they have something negative to say. Thus, focusing only on comments could introduce a bias in predicting the emotion generated by the article.

For further work, we would improve our method of labelling data with dictionaries and we would like to implement sentiment analysis jointly on the comments and on the corresponding article.

[4] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm