Airline sentiment analysis
Machine Learning for Natural Language Processing 2022

Hashley RAMANANKATSOINA
MS Data Science - ENSAE
hashley.ramanankatsoina@ensae.fr

Hilary SEUGUE NONDA
MS Data Science - ENSAE
hilarymichellecephora.seuguenonda@ensae.fr

Abstract

Based on a Twitter database about airline companies, we detect sentiment automatically without human input. Rather than simply using the database at our disposal, we also proposed a method to enrich it by scrapping ourselves new tweets and labeling them thanks to semi-supervised learning. After that, we trained our deep learning model DistilBert on this database and achieved F1-score of 0.75 (positive class), 0.60 (neutral class) and 0.90 (negative class).

1 Introduction

Sentiment analysis is extremely useful in social media monitoring for any companies as it provides an overview of public opinion. Thus, they can react accordingly and improve their services. In this project, we help six major U.S airlines companies to get meaningful insights about their custormers' feelings. To do so we use the Kaggle database "Twitter US Airline Sentiment" [1]. Our work can be found in this Github.

2 Dataset description

Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as "late flight" or "rude service"). Sentiments are classified as following: negative, neutral, positive. We have an unbalanced dataset with more negative tweets (63%). Appendix A describes the data analysis details.

3 Experiments Protocol

To go further than just modeling, we enriched the database with tweets that we scrapped ourselves thanks to the Twitter API. Semi-supervised learning has been used to label the newly scrapped data. Moreover, the use of this technique is supported by the fact that after performing some test, we could see that scrapping new data permitted us to improve the results performance of our deep learning model DistilBert compared to only training on the initial database.

3.1 Twitter scrapping

We used the Twitter API V2 to scrape tweets (appendix B details some limitations of this free API) [2]. To target relevant tweets, we used the keywords: @VirginAmerica, @united, @SouthwestAir, @Delta,@USAirways, @AmericanAir (which represent the airlines Twitter account). In the end, we scrapped 25,880 tweets dating from March 10th 2022 to March 15th 2022.

3.2 Embedding technique

Word2Vec is a common method of generating word embeddings. It has several applications such as text similarity, sentiment analysis... [3]. In this project, we use Word2Vec CBOW after cleaning up the tweets, removing stopwords and tokenizing the text. In our experiments, we also compared Word2Vec with TF-IDF. The results obtained were slightly better with TF-IDF so this is what we present in part 4 with the SGD Logistic Regression.

3.3 Semi-supervised learning

Semi-supervised learning is a case in Machine Learning in which, in our training data, not all samples are labeled. We use the implementation Self-Training Classifier of the Scikit-Learn library for our semi-supervised learning [4]. Appendix C describes the framework of the algorithm. The basic classifier chosen is the SGD (Stochastic Gradient Descent) version of the Logistic Regression. In

the end, thanks to semi-supervised learning, our labeled dataset grew from 10,980 samples to 12,801 samples.

3.4 DistilBERT Model for sentiment classification

DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base [5]. It makes it possible to reduce its size by 40% compared to a BERT model while keeping 97% of its language understanding capabilities and being 60% faster [6].

4 Results

		Precision	*Recall*	*F1-Score*
	Negative	0.82	0.90	0.86
(1)	Neutral	0.62	0.52	0.56
	Positive	0.75	0.64	0.69
	Negative	0.88	0.92	0.90
(2)	Neutral	0.65	0.55	0.60
	Positive	0.76	0.73	0.75

Table 1: Results of SGD Classifier (1) and DistilBert (2) on test data.

As shown in table 1, our deep learning model does better in analyzing the underlying sentiment than our baseline model. Moreover, we can see that the neutral class is difficult to detect for both models, and negative tweets are more easily recognized.

What we can also analyze are the errors made by DistilBert. If we take an example of a tweet predicted positive whereas it was negative:

> @USAirways Forget reservations. Thank you to the great leadership at your company, I've Cancelled Flighted my flight. Once again, thank you.

Actually, the tweet is sarcastic. It is really challenging for machine to understand sarcasm and it is a real subject of study in Machine Learning. The model still hesitated with label negative or positive (31% vs 40%). By looking at the attention layers, we better understand this result. As we can see on Figure 1, attention is focused on the words "thank you" which are positive.

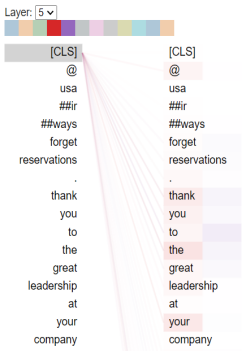

Figure 1: 4th and 5th head of the attention layer 5 on the sentence "@USAirways Forget reservations [...]".

We also decided to test our model on a new dataset. The dataset is about various products such as Tablet, Mobile so it's a whole other subject [7]. The F1-Score were respectively 0.26, 0.37 and 0.14 for the negative, neutral and positive class. Unfortunately, the results have not been conclusive.

5 Discussion/Conclusion

Being able to quickly understand consumer sentiment and react accordingly is extremely helpful for businesses. In this project, we proposed a Machine Learning model to predict users feelings from tweets regarding six major American airlines. Rather than simply using the database at our disposal, we enriched it with reviews scrapped ourselves from Twitter and performed semi-supervised learning to label our new data and learn the best out of it.

The experimental results obtained showed that our deep learning model DistilBert do a better job than our baseline model SGD Logistic Regression. Both models had much less difficulties to predict negative opinion. However, we should keep in mind that our database is highly unbalanced. Additionally, we could see with the analysis of the errors made by the model that some tweets might show ambiguity, sarcasm so questions may arise about the nature of the data.

We may question here the possibility of putting the deep learning model into production knowing that it is more time and space consuming than Logistic Regression. Indeed, performances obtained were almost similar for both models. However, DistilBert training time was thirteen times longer.

References

[1] Twitter us airline sentiment. https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment.

[2] Twitter. Twitter api documentation.

[3] Word2vec. https://towardsdatascience.com/word2vec-explained-49c52b4ccb71.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Distilbert. https://huggingface.co/docs/transformers/model_doc/distilbert.

[6] Julien CHAUMOND Thomas WOLF Victor SANH, Lysandre DEBUT. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. 2020.

[7] Product sentiment classification — kaggle. https://www.kaggle.com/datasets/akash14/product-sentiment-classification.

[8] Duck Young Kimb Sujeong Baeka, Hyun Sik Yoonb. Abnormal vibration detection in the bearing-shaft system via semisupervised classification of accelerometer signal patterns. 2021.

A Data analysis

The goal of data analysis is to extract information that allows to identify more precisely the profile of the data. This section describes the work we have done before forecasting to better understand the database we had at hand.

A.1 Word clouds

Word clouds are a powerful way to visualise what your audience really thinks about a topic. Figure 2 presents the word clouds per sentiment.

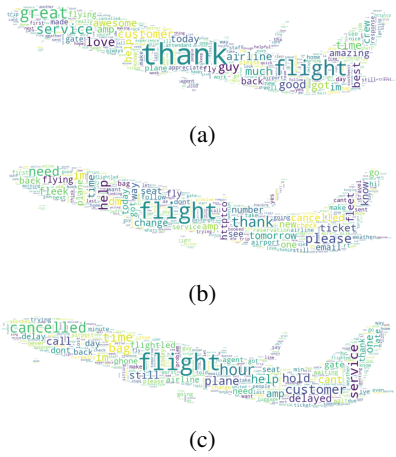

Figure 2: Word clouds per sentiment. **(a)**: Positive tweets. **(b)**: Neutral tweets. **(c)**: Negative tweets.

As shown on Figure 2, frequent positive words like ”thank”, ”great”, ”love” are related to tweets labeled positive. On the other hand, frequent positive words like ”cancelled”, ”delayed”, ”help” are related to tweets labeled negative. Neutral tweets are more tricky because it is a mix of the two.

A.2 Airlines

In the dataset, six major U.S airlines companies are mentioned: Virgin America, Delta, Southwest, American, US Airways and United. However all airlines are not mentioned equally (cf Figure 3). Meaning not all companies are comparable in our analyses. For example Figure 4 presents the sentiment proportion per airline.

Regarding Figure 4, it is true to say that most of tweets are negative for everyone but if Virgin America is doing better than the other companies, isn't it because we lack of feedbacks about them compared to United? It is a possibility. We have to keep that in mind.

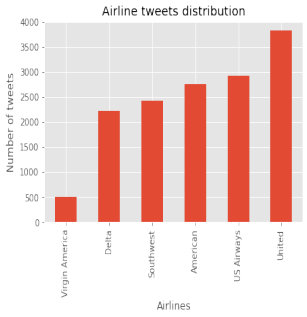

Figure 3: Number of tweets per airline.

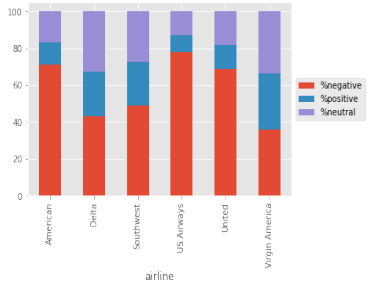

Figure 4: Proportion of POS/NEUT/NEG tweets per airline.

A.3 Sentence level analysis

The most frequent words and bi-grams of the tweets are presented in the table 2.

#	Words	Bi-grams
1	Flight	Customer service
2	Get	Cancelled flightled
3	Thanks	Late flight
4	Cancelled	Cancelled flighted
5	Service	Flight cancelled

Table 2: Top 5 of the most frequent words and bi-grams.

Besides the spelling mistakes, we can see that "customer service" and "cancelled flight" are the reason why people are complaining. This observation is confirmed by looking at the column "negative reason" provided (cf Figure 5).

For the sake of the demonstration, we went into the syntactic analysis of a random sentence. Figure 6 shows the POS-Tagging of positive tweet.

We can see on the Figure 6 that the model understands well which word is referring to which and the grammatical category of each word.

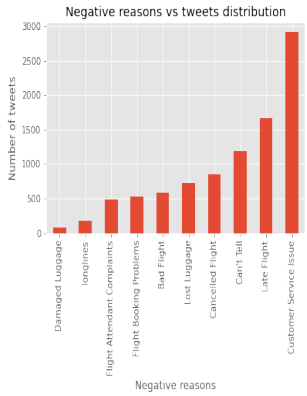

Figure 5: Distribution of the unsatisfaction of customers.

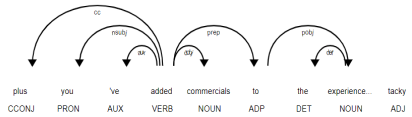

Figure 6: Syntatic analysis of a random positive tweet.

B Twitter API V2

The Twitter API lets us read and write Twitter data. Thus, we can use it to compose tweets, read profiles, and access a high volume of tweets on particular subjects. To scrape tweets we used the free access Twitter API v2. However, this API has some limitations. First, only recent tweets over a 7-day sliding window can be retrieved. Second, the number of tweets per page is 100, so we had to develop a code that automatically takes into account the transition from one page to another. Finally, the maximum number of requests is 450 every 15 minutes. It was therefore necessary to develop a code which takes into account this pause time in an automatic way in order not to obtain any error.

C Self-Training Classifier

Figure 7 illustrates the framework of the Self-Training Classifier.

At each iteration, the base classifier predicts labels for the unlabeled samples and adds a subset of these labels to the labeled data set. This subset is selected such that the prediction probabilities are above a threshold. In our case, we set this threshold to 99% in order to be sure that the newly labeled samples are those on which the model is confident. These steps are performed until a maximum number of loops, predefined by the user, is

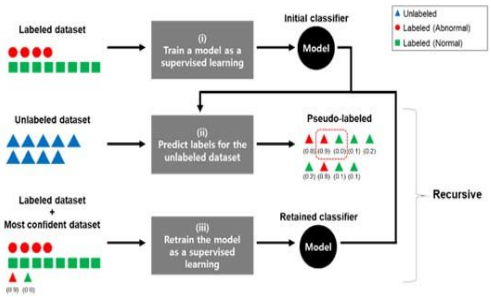

Figure 7: Self-Training Classifier framework [8].

reached or when no sample respects the selection criterion. Obviously, the stricter the selection criterion is (high probability threshold), the less enriched our final dataset will be. There is a trade-off.