Using context-aware NLP on D. Trump's tweets to predict the movement of the S&P 500 Index
Final Project Report

Ruixu Chen
MSc Data Sciences and Business Analytics
ruixu.chen@student-cs.fr

Camille Morand-Duval
MSc Data Sciences and Business Analytics
camille.morand-duval@student-cs.fr

Antoine Ohleyer
MSc Data Sciences and Business Analytics
antoine.ohleyer@student-cs.fr

Ismail Zizi
MSc Data Sciences and Business Analytics
ismail.zizi@student-cs.fr

ABSTRACT

Social media is a common communication tool to convey emotions and opinions. It has grown in popularity among political figures and some now use Twitter to convey their thoughts and beliefs. The effect of the written statements of influential figures can be measured socially and economically. President D. Trump posts regularly and more than 8,000 tweets have been recorded since he was sworn in. His social outbursts have a clear effect on the movement of the stock market.

This report details machine learning models designed to predict the rise and fall of the S&P 500 from D. Trump's tweets since 2017. It describes Natural Language Processing methods to analyse text and extract relevant features. In addition to sentiment-scoring the tweets, most common named entities are extracted, and a term frequency inverse document frequency is used to create features that are more specific. These two methods are combined with Random Forest and KNN algorithms respectively and output accuracies of 0.584 and 0.575 with a propensity to overfit. Doc2Vec methods are studied in a second set of models, are included Distributed Bag of Words, Distributed Memory and the concatenation of both. Classification is done by logistic regression and yields accuracies of 0.505, 0.538 and 0.544 respectively.

The accuracies obtained are better than random guessing but still low. The performance of the TFIDF with KNN classifier model is explained by the specificity of the features,where most of the noise is ignored. Even though it has the lowest score, the Doc2Vec logistic regression model seems to be the one with the most potential as it takes into account the context of the tweets context.

ACM Reference Format:
Ruixu Chen, Camille Morand-Duval, Antoine Ohleyer, and Ismail Zizi. . Using context-aware NLP on D. Trump's tweets to predict the movement of the S&P 500 Index Final Project Report. In . ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

In an era of post-truth politics, political figures are increasingly using social media to exert their influence whereby the repeated assertion of talking points to which factual rebuttals are ignored. Post-truth differs from traditional contesting and falsifying of facts by relegating facts and expert opinions to be of secondary importance relative to appeal to emotion. The latter makes an increasingly relevant case using sentiment analysis to predict social dynamics. A noticeable example would be Donald Trump, the current president of the United States who posted over 2,000 tweets every year over the past three years of his presidency. In his case, tweets serve as presidential statements and reveal his reaction to and opinion of current news. Even though he cultivates the reputation of being unpredictable, his tweets serve as a proxy for his stance on the world - a great contributing factor of the fluctuating of the stock market which, arguably, falls in the span of social dynamics.

The effect of Trump's tweets on the stock market is short-termed but noticeable as shown by statistical analyses [Born et al. 2017; Tom et al. 2018]. The semantic analysis of the content of his tweets could therefore be used, to some extent, to predict the movement of the stock market. The validity of the model will be determined by the accuracy of its predictions i.e. if it is better than random guessing, which corresponds to an accuracy greater than 50%.

This report summarises recent literature predicting the stock market using Natural Process Language (NLP). This review focuses on sentiments analysis to estimate the rise and fall of the stock market. Furthermore, this report details two methods to classify the movements of the S&P 500 (up or down) from D. Trumps tweets. It includes a description of the data collection and processing, its processing as well as the feature engineering led and the different models tested. Finally, it discusses the limitations of the models based on their experimental accuracy and concludes on potential improvements.

2 PROBLEM DEFINITION

The aim of this project is to, given the twitter feed of Donald Trump, use NLP and binary classification to predict the movement of S&P 500. It intends to aggregate his tweets on a particular unit-time basis and train a model to predict the rise or fall of stock market of the next time unit. The accuracy of a model relies on its ability to output correctly the positive (stock goes up) or negative (stock goes down) impact of a tweet. The machine learning framework can be found in figure 1.

Machine Learning, December 7, 2019, CentraleSupelec
©

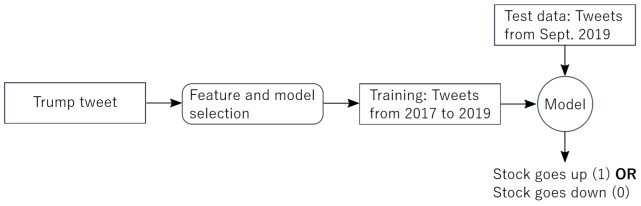

Figure 1: Diagram of the machine learning basic pipeline

It has been shown statistically that the stock market does react positively and negatively to D. Trump's tweets ; this project revolves around the possibility of predicting its movement in a reliable way i.e. if the algorithm predicts the stock will go down, traders can use this information to short their stocks. Potential applications include high frequency trading, and similarly structured models could be built to predict the movements of other financial markets using different social medias or news inputs.

A tweet can be defined as a short message posted on Twitter, which is composed of less than 280 characters (limit changed in 2017 from 140) even though the average number of characters used is 33. Organisation, person and event's names following the symbol "@" and "#" are used to refer to specific topics, some incorporate emojis or links. D. Trump uses them to convey his emotions and opinion on various subjects and matters. The use of NLP i.e. algorithms and techniques used to quantify text and words is therefore appropriate. Additionally, the S&P 500 is a stock market index that tracks the stocks of 500 largest U.S. companies. It is a good indicator of the economical trends in the U.S. and tends to react to various political, economical and other news. The closing and opening balances can be used to define the variations from one time-unit to the other. Hence, the output of the problem is binary (up or down).

3 LITERATURE REVIEW

Yang and Yang have attempted to solve the problem addressed in this report, using D. Trump's tweets to predict market movement [Yang and Yuxin 2017]. The result obtained however is not promising as the accuracy of their models is at most 48%. Given their data set was limited (only one year of tweets) and they used exclusively Word2Vec models, their result can be improved.

Our methodology follows the model built by Hagenau et al. [Hagenau et al. 2012]. This paper aims at determining whether textual information in financial news can be used to improve stock price predictions. The model includes NLP context-based features to transform the text gathered into exploitable information. The extracted information is used as input for Support Vector Machines (SVM). The model is trained to select relevant features.

There are multiple methods to deal with text (qualitative content). Schumaker [Schumaker and chin Chen 2009] outlines three main approaches: Bag of Words, Noun Phrases and Named Entities. Most papers build their classification algorithms on Bag of Words approaches. However, content analysis, as described by Velay [Velay and Daniel 2018], is a promising technique, which is also used by Hagenau et al. [Hagenau et al. 2012]. The development of Natural Language Processing and the emergence of such context aware algorithms means powerful tools have been created. Yu et al. describes such an algorithm as an accurate way to analyse sentiments in stock market news [Yu et al. 2013]. These methods are further explored by Kraus and Feuerriegel [Kraus and Feuerriegel 2017] in their deep neural network model which uses financial disclosures as inputs.

In the same way, classification can be done using various methods: SVM, k-Nearest Neighbour classifier, neural network, linear least squares fit, logistic regression, random forest, Naïve Bayes classification for example [Yang and Liu 2003]. Some of these methods are evaluated by Velay [Velay and Daniel 2018]; the NLP approach is not specified for all results however (the most accurate model described has an accuracy of 57%; it is a combination of Bag of words with Logistic Regression). Over the last few years, new methods have emerged such as Multiple Kernel Learning (MKL) [Shynkevich et al. 2015] and Extreme Learning Machine (EMT) [Velay and Daniel 2018].

The performance of a model is difficult to evaluate from literature as most of the stock prediction data sets are biased (proportion of positive / negative news). Additionally, the papers usually combine different NLP methods with different machine learning algorithms, making it even more difficult to evaluate the performance of each respectively.

The models derived in this report follow the literature mentioned in this section. It combines frequency and TFIDF language processing and sentiment analysis with Machine Learning algorithms such as Random Forest and KNN classifiers. It looks into Doc2Vec promising results when it used with logistic regression.

4 METHODOLOGY

4.1 Data collection

4.1.1 Tweets Data. The tweets are collected from the Twitter Archives database. The extracted data includes the time of the tweet, number of retweets, whether it is a retweet or not and its text.

4.1.2 S&P 500 Data. To predict the rise and fall of the S&P 500 from the D. Trump's tweets, two data sets are necessary, the S&P 500 opening and closing balance at a given time and D. Trump's tweets during that period.

The S&P 500 data is collected daily from Yahoo finance. The extracted data includes the date, opening and closing balance. Given the volatility of the stock market, additional data is collected hourly. This second data set is used as for comparison, to determine whether the accuracy of our final model is greater when tweets and the movements of the S&P 500 are aggregated and calculated daily or hourly. Literature is not consistent as to the effect of D. Trump's tweets on the S&P 500 time wise. It agrees that it is noticeable at an interval of one hour and one day.

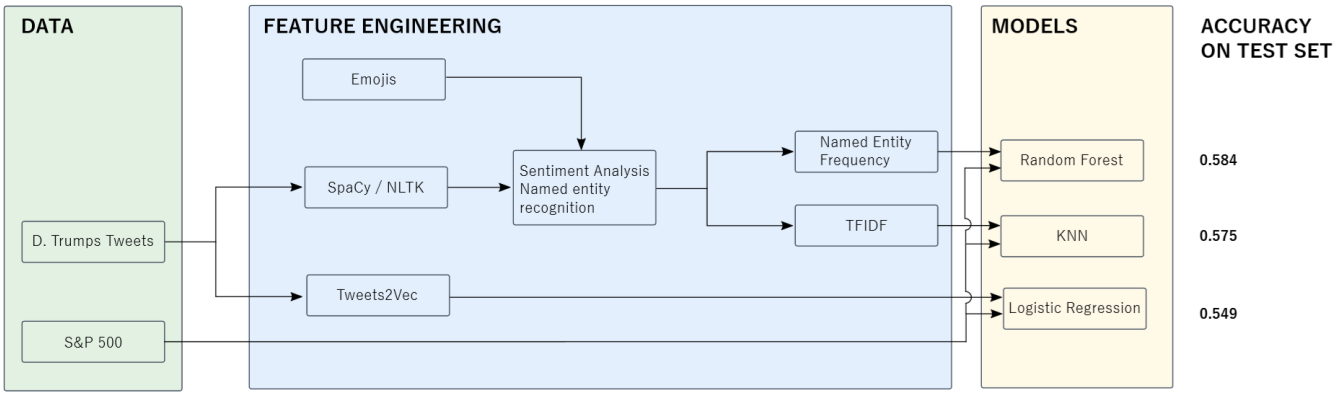

Figure 2: Diagram of the pipelines and models used

4.2 Data cleaning

4.2.1 Tweets Data. The number of words per tweet is calculated for each tweet and added as a feature. Links and special characters are removed from tweets. A set of frequent words deprived of contextual meaning is defined and they are removed from the tweets, so are stop words found in the built-in library. Punctuation and digits are removed from the tweets. The text is converted to lower case. A function is created to combine tweets into a corpus or split them into individual tweets i.e. to aggregate tweets based on a chosen unit-time, in this case hourly and daily. Note that tweets posted outside the trading hours are removed, since there is no stock price information available during those hours.

Following the standard text initial processing, a few steps are taken to clean the data i.e. to filter out trivial tweets corresponding to the ones which are less likely to have an impact on the stock market. The first step consists of filtering out all the tweets that were posted before Trump became the president, the 20th January of 2017. The second step is to remove retweets. They tend to contain less relevant information than the original tweets. Finally, tweets whose word counts are smaller than 5, most of which are noisy text such as such as 'Thanks', are removed from the data set as well.The tweets are then converted into EST time to align with the time zone of the stock market data obtained from Yahoo Finance.

4.2.2 S&P 500 Data. The stock market data is aggregated into a daily and an hourly basis. For each day, the closing balance is compared with the closing balance of the previous day. If the closing price of the current day is higher a target label of 1 is attributed, otherwise 0 is given. Considering the hourly data, the one-hour earlier and one-hour-later stock prices are extracted for each time-unit, and then the labels are assigned accordingly, following the method described for the daily data. Note that the S&P 500 opens at 9:00 AM and closes at 4:00 PM.

4.2.3 Dataframe creation. Tweets and stock data sets are 'innerly' joined after initial processing (cleaning). The merged and initial data frame has three columns: time period, aggregated text, and next day's stock market performance in the form of a binary measure indicating whether the stock goes up (1) or down (0).

4.3 Data processing

Two pipelines are used for text processing: SpaCy and NLTK. For text analysis and models using text as a predictor a few preparation steps are necessary: text segmentation (in this case, it has already been done since tweets have a given number of words), word tokenisation (dividing text fragments into words and symbols if any), prediction of parts of speech (noun, verb, etc.), lemmatization (finding root words - this is done using a built-in dictionary), stop words recognition, dependency parsing and named entity recognition.

4.3.1 NLTK. Following the initial cleaning of the tweets, the python NLTK library can be used to further process the data. It is tokenized and the extracted list of words is associated with the up or down label. The NLTK library allows for tagging, lemmatisation, identification of named entities and display of a parse tree showing common sentence patterns. A sentiment analyser can be imported from the NLTK library to quantify the sentiment of each tweet.

4.3.2 SpaCy. A second method includes the use of SpaCy, a python NLP library built to extract information from unstructured data. It is a rule-based matcher which mines the subject, object, modifiers and part-of-speech of a sentence. The nature and grammatical function of each word can be extracted, grammatical patterns such as noun + verb identified. The pattern recognition is then run though the data and all the sentences matching this pattern are extracted (example: I love + noun). Even though this tool is very powerful, it is difficult to generalize patterns.
SpaCy functions provide tokenization, lemmatization and labelisation of the the tweets. For each word a label is added providing information on the word nature, function and entity (whether it refers to an organization, person, political or an event).

4.3.3 Emojis. Some tweets contain emojis which influence the value of the sentiment calculated. An emoji database is incorporated into the final models to measure more accurately the sentiment of the tweet. A sentiment score ranges from -1 to 1 figures in this additional data set. Measured this way, the emojis are classified into three categories: positive, negative and neutral - emojis whise sentiment score is between -0.05 and 0.05 fall into the last category.

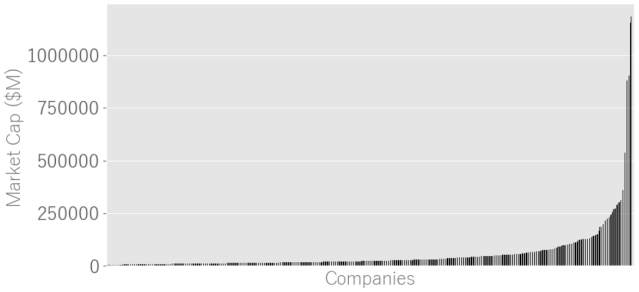

Figure 3: S&P 500 Companies Market Cap Distribution

The sentiment is raised by 0.1 for positive emojis and decreased by 0.1 for negative emojis.

4.3.4 Companies. Additional data is collected for the S&P 500 companies such as their names, price, dividend yield, market cap. Exploratory data analysis figure 3 shows that the market cap in millions of dollars of the 500 companies is exponential. Only a few companies have very high market cap. It is assumed the mention of these companies in a tweet will have a great effect on the S&P 500 index (valid based on [Born et al. 2017]). Companies with market cap above 180 million are recorded as large companies.

The following results are derived using the hourly data.

4.4 Named Entity Frequency Model

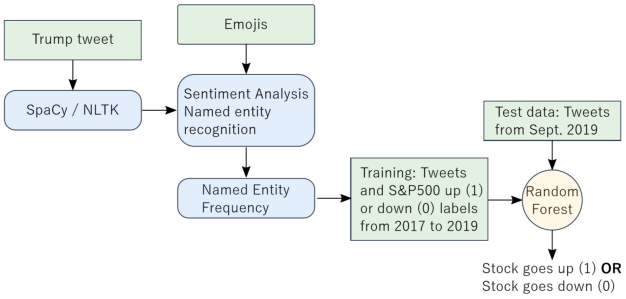

Figure 4: Pipeline of the Named Entity Frequency Model

4.4.1 Mathematical background. The Entity Frequency is defined as follow,

$$f_{i,j} = \frac{d_{i,j}}{N}$$

Where f_i is the frequency of i (word corresponding to a specific entity) in a tweet, d_i the number of occurrences of i and N the number of tweets aggregated per unit-time.

4.4.2 Feature engineering. A first model is designed to understand the importance of the various entities on the stock market and whether mentioning these entities has an impact on the model output. It includes features such as sentiment scores, average numbers

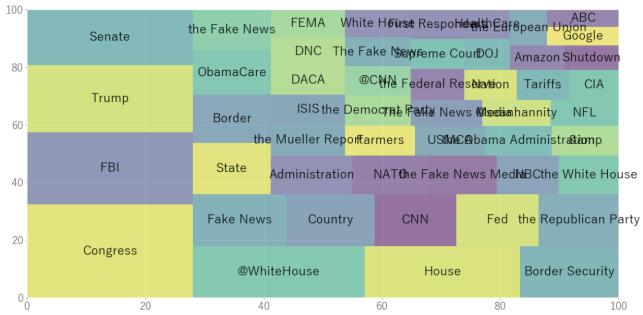

Figure 5: Frequency of organisations names in tweets

of occurrences of organizations, persons, law entities, events and NORP (nationalities or religious or political groups). A representation of the term frequency within the entity "Organisations" can be found in figure 5.

The distribution of the sentiment scores is shown in figure 6. The representation takes into account the variations caused by the presence of emojis in a tweet. It illustrates the strong emotions conveyed in D. Trump's tweets ; very few tweets are neutral.

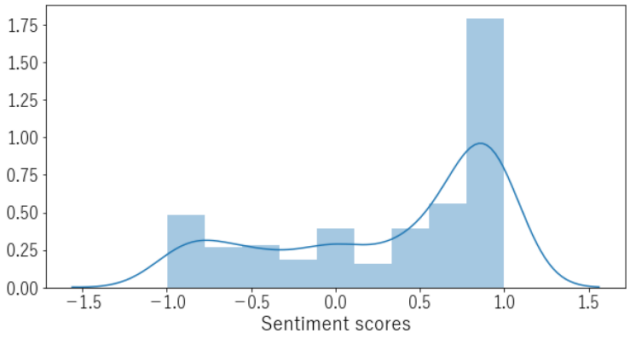

Figure 6: Kernel distribution of the recorded sentiment scores

4.4.3 Modelling. These features are used in a Random Forest model. Random Forest is a relatively robust algorithm and usually performs well without much feature engineering. In this case, the model used includes 300 estimators.

4.4.4 Evaluation. The 5-fold cross validation accuracy of this model averages at 0.526. Even though it is slightly higher than 50%, the model is clearly overfitting. The calculation of the accuracy on the training set averages at 0.976 ; hence it will not generalize well to unseen data.

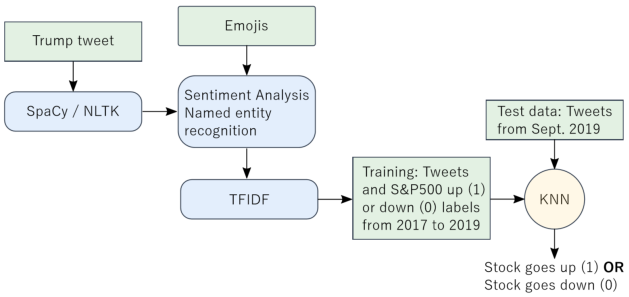

Figure 7: Pipeline of the TFIDF Model

4.5 TFIDF Model

TFIDF refers to Term Frequency Inverse Document Frequency. It reflects the importance of a word in a document or corpus. If a word is present multiple times it will have a low TFIDF weighted score.

4.5.1 Mathematical background. The model relies on the same sentiment scores as in the previous model section 4.4. However, an inverse term frequency weighting is used here. The weights are defined by:

$$w_{i,j} = tf_{i,j} \times log(\frac{N}{df_i})$$

Where $w_{i,j}$ is the weight of i in j, $tf_{i,j}$ the number of occurrences of i in j, df_i the number of documents containing i and N the total number of documents.

4.5.2 Feature engineering. This model implies the creation of a different set of features than the ones used in the model described section 4.4. The 100 most frequent words inside each semantic group are extracted and combined into a custom vocabulary list (duplicates and overlaps between entities are removed). For example, within the organisation semantic group, words such as 'NATO', 'Google' and 'FBI' are added to the custom dictionary. With the help of the sk-learn TF-IDF function, the tokenized words of the training set are vectorized based on a custom vocabulary list. Each words is attributed an IDF weight. The trained algorithm is then applied to the tweets in the training and test set, resulting in a TFIDF matrix.

Since there are fewer than 1000 rows in the training data set, in order to prevent overfitting due to the increased complexity of our feature space, Principal Component Analysis (PCA) is applied and only the 80 principal components are kept. They account for about 90% of the variance. We also include the sentiment score. Because the features are roughly in the same scale, we do not normalize our features.

4.5.3 Modelling. A KNN algorithm is employed using this set of features. Since it is a scale-sensitive model and from literature, it is assumed sentiment scores have a greater influence on the rise and fall of the S&P 500 than the other features. This feature is boosted by multiplication with a factor of 5.

4.5.4 Evaluation. KNN algorithm is highly sensitive to the choice of k and other parameters such as distance metric and neighbor weight methods. To find the most suited parameters, a grid search is applied for k values from 1 to 10, uniform and distance weights and euclidean and manhattan distances.The grid searcher returns a model with the best performing parameters on a 5-fold cross-validation set.

The best model has an accuracy of 0.584. It uses k = 3 neighbors, weights by distance and euclidean metrics. Even though the results are better, the accuracy on the training set is 0.973 when using a 5-folds cross validation hence the model is still overfitting.

4.6 Tweet2Vec

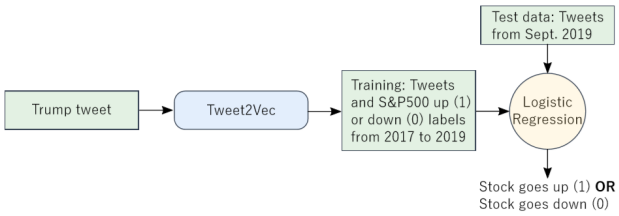

Figure 8: Pipeline of the Tweet2Vec Model

4.6.1 Mathematical background. Doc2Vec relies on the same equations as Word2Vec with an added classifier identifying the document. There are two approaches to Doc2Vec, without and with distributed memory. The difference lies in whether the context is used to predict the centre word (without distributed memory) or the centre word is used to predict the context (with distributed memory). For each center word w in a given position t, the likelihood of a second word being in another position is calculated.

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \leqslant j \leqslant m}^{j \neq 0} P(w_{t+j}|w_t)$$

For θ is the number of parameters, where t is first center word, T the total number of words, j the position from t within a $2m$ window.

The objective function to minimize is therefore,

$$\begin{aligned} J(\theta) &= \frac{-1}{T} log(L(\theta)) \\ &= -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leqslant j \leqslant m}^{j \neq 0} logP(w_{t+j}|w_t) \end{aligned}$$

The probability of one word being at a given position given the centre word is calculated by defining two representations per parameter, no matter whether the word is a center word v_w or it is a context word u_w.

$$P(o|c) = \frac{exp(u_0^T v_c)}{\sum_{w \in W} exp(u_w^T v_c)}$$

This equation correspond to the softmax function.

The optimisation is done by stochastic gradient descent.

$$\theta_{new} = \theta_{old} - \alpha \nabla J_{old}$$

The Doc2Vec algorithm uses two types of vectors: a paragraph vector (unique among paragraphs) and a word vector (shared among paragraphs). The vectors are concatenated during the analysis and the paragraph token is considered as another word acting as memory of the topic of the paragraph. When using Doc2Vec on unseen paragraphs, it requires a second inference step occur. While the word vectors and softmax weights calculated are fixed, a gradient descent is performed to compute the paragraph vectors for new paragraphs [Le and Mikolov 2014].

4.6.2 Feature engineering. In recent years, the micro-blogging site Twitter has become a major social media platform with hundreds of millions of users. The short (140 character limit), noisy and idiosyncratic nature of tweets make standard information retrieval and data mining methods ill-suited to Twitter. Consequently, there has been an ever growing body of IR and data mining literature focusing on Twitter. However, most of these works employ extensive feature engineering to create task-specific, hand-crafted features. This is time consuming and inefficient as new features need to be engineered for every task.

The Tweet2Vec implementation generates general-purpose vector representation of tweets that can be used for any classification task. Tweet2Vec removes the need for expansive feature engineering and can be used to train any standard classifier (logistic regression, svm, etc). Our method is especially useful for natural language processing tasks on Twitter where it is particularly difficult to engineer features. The data pre-processing and cleaning has been kept minimal on this pipeline since we want to pick up on weak signals in the embedding including but not limited to typos, abreviations, etc.

Regarding the financial data, an absolute difference in opening and closing balance feature is created. Some tweets have more impact on the S&P 500 than others; hence a plot (figure 9) of the amplitude of the absolute difference against each tweet is studied. It is part of exploratory data analysis and a way to determine which tweets should be kept for training and which are noise.

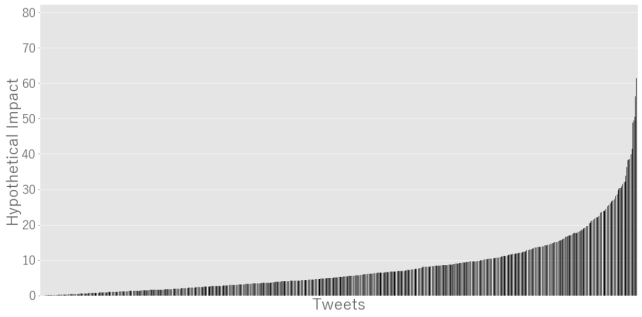

Figure 9: Hourly amplitude distribution of S&P 500

Following the cleaning process described section 4.2, the Beautiful Soup Python library is used to parse the data. Tokenization is done using the NLTK library and tagging is done using the binary output

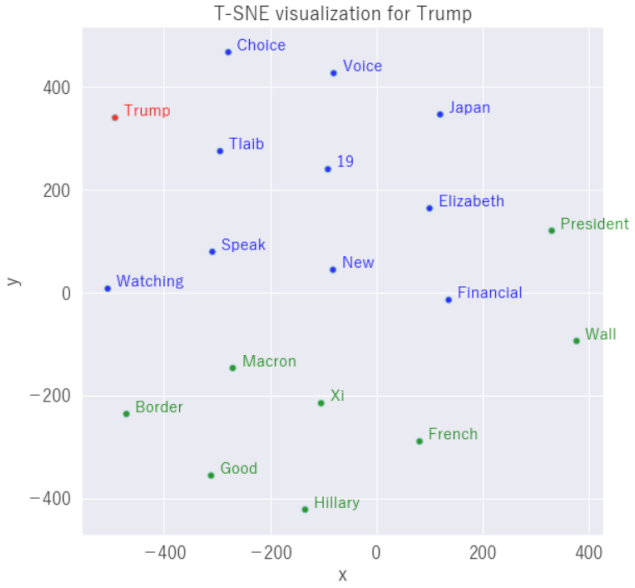

Figure 10: 2-D representation of 'Trump', in green the word from the entry list studied and in blue the closest match found using Tweet2Vec

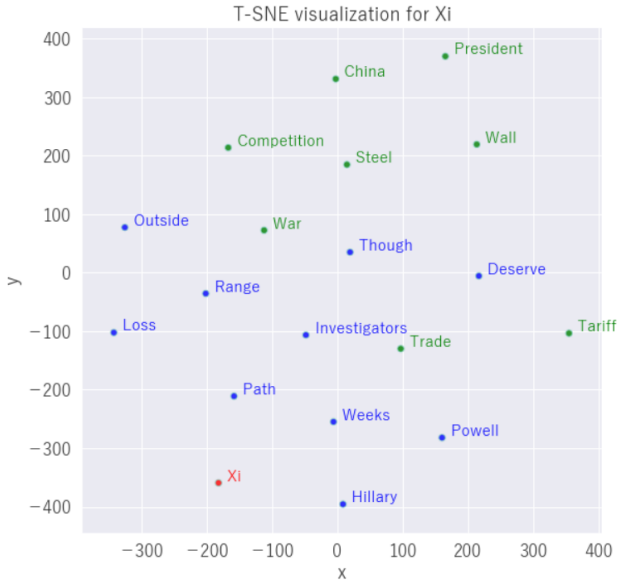

Figure 11: 2-D representation of 'Xi', in green the word from the entry list studied and in blue the closest match found using Tweet2Vec

given in the data frame i.e. each tweet is divided into a set of words. The set is tagged 1 or 0.

The Word2vec algorithm is built on both continuous Bag of Words and Skip Gram models. It is a numeric representation of words capturing various relations between words such synonym, antonyms

etc. Continuous bag of words represents each word as a feature vector. The relationship between vectors is obtained from a sliding window around the word in its context (surrounding words). Skip gram predicts one word based on its surroundings. Doc2vec method combines the words included in the document with additional information unique to the tweet. The words are used as classifiers. The idea behind such method is to group the words of each tweet so the words present in a tweet are used as context to predict the potential words in another tweet. It is a neural network algorithm.

4.6.3 Modelling. There are two steps to implementing Doc2Vec in python : the first is to build the vocabulary from the documents and the second to train the model to vectorize the documents.

The first algorithm tested is set up without a distributed memory parameter, meaning the model relies on a distributed bag of word approach (the context words in the input are ignored). The vocabulary definition uses a feature vector of size 100 (a feature vector of 300 is tested as well but yields an even lower accuracy), 5 noise-words drawn from each sample (i.e. negative sampling) and a minimum of two words. A model is then trained on the vocabulary defined. The tweets of the training data are vectorized using the built-in neural network algorithm of Doc2Vec. The parameters defined in this model are the learning rate, which drops linearly by 0.002 at each iteration, the number of times it goes over the corpus, here once. The data is reshuffled 30 times. A vectorized representation of a group of words is obtained. Each tweet is divided into words and tagged with a number.

Given the number of words and features used, an accurate representation of the vectorized data is difficult. Neverthless, it is possible to visualise specific clusters of the data using a t-distributed Stochastic Neighbor Embedding graph (T-SNE). This methods preserves only small pairwise distances or local similarities. Figures 10 and 11 display the 2-D word repartition of a center word (in red), a list of 10 random words (green) and the 10 closest word to the center word (blue).

The classifying model used is a logistic regression. It takes the vectorized tweets as inputs to predict whether the stock will go up (1) or down (0). The inverse regularization parameter is set up to 1e5.

4.6.4 Evaluation. The non distributed memory model yields an accuracy of 0.505 and an F1 score of 0.504. The same process is followed using a distributed memory parameter (this approach takes into account the order of words). The accuracy of the model is not much better, only 0.539 and its F1 score 0.538. Finally, the process is run for a combination of both vectorization of the tweets. It leads to a final accuracy of 0.543 and F1 score of 0.543, which is still very low.

5 DISCUSSION

5.1 Evaluation

The evaluation of each method was done using a pre-defined test-set containing one fifth of the data of the training set. Their accuracy is given in table 1. It is calculated as follows,

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is true positive, TN true negative, FP false positive and FN false negative.

Table 1: Accuracy of each model on the test set

	Accuracy
Frequency of Named Entity & Random Forest	0.584
TFIDF & KNN	0.575
Tweet2Vec (DM) & Logistic Regression	0.549

The first model is designed to give insight as to the relationship between the occurrences of important entities and the performance of the S&P 500 index. The accuracy on the test set is 0.584, which is an acceptable in a financial setting (D. Trump's tweets are not the only determinant of the stock market). However, as previously mentioned the model will overfit the data (pointed out by the results obtained from the 5-fold cross-validation). It suggests that the relationships looked into here, are weak.

The second model is more in line with literature. The combination of TFIDF and KNN optimised with a grid-search is a standard methods which in this case yields decent results (Table 1). The selection of the 100 most frequent terms in each entity to train the algorithm allows for the removal of unwanted noise. Nevertheless, there may still be some overfitting as the training accuracy is very high compared to the cross-validation accuracy (0.973 average). Theoretically a better accuracy could be reached but due to the lack of training data (the ratio between the number of rows and columns is less than 10), the model does not generalize better than the first model.

Table 2: Accuracy and F1 score of the Tweet2Vec models

Tweet2Vec & Logistic regression	**Accuracy**	**F1 Score**
Distributed Bag of Words (300 dimensions)	0.495	0.495
Distributed Bag of Words (100 dimensions)	0.505	0.504
Distributed Memory	0.538	0.537
Concatenation of both	0.544	0.543

The last set of models uses Doc2Vec or, in this case, Tweet2Vec features to predict the movement of the stock market. This method is supposed to be more accurate and mirrors the current NLP trends in that field. The accuracy and F1 scores of the various versions of the model can be found table 2.The F1 score is defined as follows,

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

As mentioned in literature, Distributed Memory Doc2Vec approach yields better results both in terms of accuracy and F1-score. Since false positive and false negative have similar cost, accuracy is an accurate evaluation measurement here. However, F1-score provides alternative measurements on the performance of the model.

This model could be improved by training the embeddings on the whole calibration set not just the training set or by using another model than a logistic model. Even though the accuracy is higher for the first two models, the last one is more powerful and should lead to better results over time (with a larger data set). It should be noted that all results are for hourly data. The volatility of the S&P 500 does not allow for accurate predictions to be made on a daily basis.

5.2 Limits

There are limits to the data and model used for this project. The most important one is the size of the data set, there are too few rows to make a robust model. Furthermore, there is the potential biases of the data. D. Trump tends to communicate a great amount through his Twitter account, through which to convey his allegedly true opinion and emotions. However, he is the President of the United States, the sentiment carefully calculated and the words chosen specifically. He has a public image to maintain. To counteract this effect, both the sentiment scores and entity frequency / context are used as features in the models. Nevertheless, there are still biases in the initial data set.

Secondly, our models rely exclusively on NLP. Language processing is a trending subject in data science; hence it is being developed rapidly. There are some limitations to modeling exclusively using NLP. For example, irony or the use of the negative form is not always recognised by the algorithms and may lead to false analysis and conclusions.

Lastly, the S&P 500 index do not solely depend on D. Trump's tweets, for example the upward adjustment of the interest rate by the Fed is independent of his tweets. Even though it could be assumed D. Trump reacts to the News and systematically tweets his opinion during the stock market opening hours, it would not be accurate. He could sign a law in the morning and tweet about it in the afternoon ; the effect would be created by the News. Another option would be to choose a less volatile index such as the market sector index which may better reflect political news.

6 FUTURE WORK

Following the limits described section 5.2, a few steps could be taken to improve the models. For example, the data could be trained on a more accurate (and larger) data base such as Bloomberg financial news. Moreover, additional features such as language processing of the financial news could lead to a more performing model.

The implementation of time series accounts for the stock variations in time would benefit the model. Some models including both forecasting and NLP parameters to provide features to predictive models have been designed [Kelly and Ahmad 2018]. The use of D.Trump's tweets should be studied with such a model however, since D. Trump reacts to the news, the forecasting model may become endogenous hence difficult to implement and less reliable.

7 CONCLUSION

The problem addressed in this report is to determine the reliability of a model using only D. Trump's tweets to predict the movement of the stock market, in this case, hourly. Following a literature review of the existing tweets and financial data analysis to predict the movement of the stock market, three machine learning models are designed. They rely on Natural Language Processing and explore various options to define relevant features: named entity frequency, inverse term frequency of the 100 most important words per entity and Doc2Vec methods all combined, if not included in the processing, with sentiment scores. They are respectively coupled with a random forest, a KNN and a logistic regression algorithm and yield accuracies of 0.584, 0.575 and 0.549.

The results obtained are better than random guessing, however, the random forest model is overfitting. Predicting reliably the movement of the S&P 500 based solely on D. Trumps tweet is only feasible to an extent which is clearly displayed in the results.

Additional training data and training the tweet2vec embedding on more specific word data bases such as financial news would allow the model to perform better. A strong complement to the initial models would be a time series forecasting model. Such model would account for the time dependency of the stock market and lead to potentially better predictions.

REFERENCES

Jeffery Born, David Myers, and William Clark. 2017. Trump tweets and the efficient Market Hypothesis. *Algorithmic Finance* 6 (11 2017), 1–7. https://doi.org/10.3233/AF-170211

Michael Hagenau, Michael Liebmann, Markus Hedwig, and Dirk Neumann. 2012. Automated News Reading: Stock Price Prediction Based on Financial News Using Context-Specific Features. *Decision Support Systems* 55 (01 2012), 1040–1049. https://doi.org/10.1109/HICSS.2012.129

Stephen Kelly and Khurshid Ahmad. 2018. Estimating the Impact of Domain-Specific News Sentiment on Financial Assets. *Knowledge-Based Systems* (03 2018). https://doi.org/10.1016/j.knosys.2018.03.004

Mathias Kraus and Stefan Feuerriegel. 2017. Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems* (10 2017). https://doi.org/10.1016/j.dss.2017.10.001

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.

Rob Schumaker and Hsiu chin Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Trans. Inf. Syst.* 27 (02 2009). https://doi.org/10.1145/1462198.1462204

Yauheniya Shynkevich, Sonya Coleman, T.M. Mcginnity, and Ammar Belatreche. 2015. Stock Price Prediction based on Stock-Specific and Sub-Industry-Specific News Articles. https://doi.org/10.1109/IJCNN.2015.7280517

Allen Antony Tom, B Com F&A, Govind Vijayakrishnan, and Ravi Thangjam. 2018. EFFECT OF TWITTER TWEETS ON THE SHORT TERM STOCK PRICES AFTER DONALD TRUMP'S PRESIDENCY. (2018).

Marc Velay and Fabrice Daniel. 2018. Using NLP on news headlines to predict index trends. *CoRR* abs/1806.09533 (2018).

Tong Yang and Yand Yuxin. 2017. Predict Effect of Trump's Tweets on Stock Price Milestone. (12 2017).

Yiming Yang and Xin Liu. 2003. A Re-Examination of Text Categorization Methods. *Proceedings of the 22nd SIGIR, New York, NY, USA* (01 2003). https://doi.org/10.1145/312624.312647

Liang-Chih Yu, Jheng-Long Wu, Pei-Chann Chang, and Hsuan-Shou Chu. 2013. Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news. *Knowledge-Based Systems* 41 (03 2013), 89–97. https://doi.org/10.1016/j.knosys.2013.01.001