IDP: The Role of Media News and Analysts' Revision on Stock Returns

- Report and Documentation

The goal is to mechanically and automatically download reports and analyse them using different textual analysis algorithm including Naive Bayes', Textblob (from NLTK) and powerful and sophisticated Deep Learning architecture like BERT to get sentiments, and to evaluate and create a trading strategy given the results.

References:

- https://github.com/google-research/bert
- https://nlp.stanford.edu/seminar/details/jdevlin.pdf
- https://sraf.nd.edu/data/
- https://meridian.allenpress.com/accounting-review/article/89/6/2151/54156/Evidence-on-the-Information-Content-of-Text-in
- https://sraf.nd.edu/textual-analysis/resources/
- https://github.com/sloria/TextBlob/blob/dev/textblob/

Students :

- Rajdeep Surolia (Mat No. - 03711368, MSc Informatics)

Project Supervisor - Vitor Azevedo

Content

1. Introduction

This study investigates the reciprocal relationships between the fluctuation of the stock returns of the companies listed on the S&P 500 and Dow Jones exchange index. The study advances past research in showing that the prediction of stock price fluctuation based on media coverage can be improved by including sentiment and emotionality. The findings inform strategic communication, and particularly investor relations, in suggesting that media attention, sentiment, and certain corporate topics are crucial when managing media relations and with regard to securing a fair evaluation of listed companies. Furthermore, the innovative research methods are useful for researchers and practitioners alike in showcasing how media coverage related to firms and their stock fluctuations can be identified and analyzed in a reproducible, hands-on and efficient manner.

We first downloaded reports from the Eikon Datastream from Thomson Reuters. In that, we mechanically doenload the PDF reports by controlling and manoeuvre the mouse cursor using Robotic Process Automation (RPA). We then analysed the reports using different algorithms and finally infered a trading strategy to buy/sell stocks based on our analysis.

We used Python as the primary and only language to code, using different tech-stack along the way. These include PyAutoGui and PyGetWindow libraries for RPA; NLTK, numpy, pandas, textblob, Pytorch and sklearn for handling data and applying textual analysis algorithms.

2. Downloading Data and Pre-processing

2.1 Robotic Process Automation

After much research, we decided that the best way to collect data from such a compex platform like Eikon would be to use the state-of-the-art RPA (Robotic Process Automation) instead of using the Eikon API which would not give us the desired number of reports without actually getting paid subcription. RPA is the process by which a software bot uses a combination of automation, computer vision, and machine learning to automate repetitive, high-volume tasks that are rule-based and trigger-driven. It is an intelligent way of automating tedious,repititive, manual tasks thus saving cost, time and energy.

2.2 Method

In this approach of ours, the mouse cursor is moved around on the screen by supplying the coordinates where the desired action needed to take place like left-click, scrolling, double-click, right-click, etc.So, through a series of mouse click and move to events we have been able to develop an automation framework which downloads reports without any human intervention.The reports downloaded were catalouged under the 'Research' tab of the S&P-500 (.spx) and Dow-Jones (.dowj) windows. The code was run for an extended period of time so that the mouse cursor could continuously to the task of downloading and saving the reports. This was a relatively time-taking process even though the whole process was done automatically. However, in comparison, RPA saved us a lot of time when we were able to get access and run the code for long.

In total, reports of about 100 companies were downloaded from each index. The size of the final data collected was close to 250 GB. The reports were then saved onto the local TUM machine. We have since then, gottehn the data on a flash drive for usability.

2.3 Pre-processing

The extracted data have been tokenized using an nltk tokenizer and then extracted text has been cleaned by stemming, lemmatizing, removing punctuations which is now ready for analysis.

Tokenization- It is the breaking of a sentnces into words or tokens.

Stemming- It is the process of reducing a word to its stem or root eg running is reduced to run.

Lemmatization- It is the morphological analysis of words which have a similar meaning like went, go are similar type of words. The past tense of go is went.

3. Textual Analysis and Experiment

We trained the collected Data using 3 classifiers -

1. Naive Bayes's,
2. BERT and
3. Textblob (NLTK)*

This is based on multi-class classification.

3.1 Naive Bayes' Approach

This model applies Bayes theorem with a Naive assumption of no relationship between different features. According to Bayes theorem: Posterior = likelihood *proposition/evidence or P(A|B) = P(B|A)* P(A)/P(B)

For ex: In a deck of playing cards, a card is chosen. What is the probability of a card being queen given the card is a face card?

This can be solved using Bayes theorem.

P(Queen given Face card) = P(Queen|Face) P(Face given Queen) = P(Face|Queen) = 1 P(Queen) = 4/52 = 1/13 P(Face) = 3/13 From Bayes theore: P(Queen|Face) = P(Face|Queen) P(Queen)/P(Face) = 1/3

For an input with several variables: P(y|x1, x2, … xn) = P(x1, x2, … xn|y) *P(y)/P(x1,x2, …xn) with Naive Bayes we assume x1, x2 … xn are independent of each other, i.e: P(x1, x2, … xn|y) = P(x1|y)* P(x2|y) … * P(xn|y)

The assumption in distribution of P(xi|y) give rise to different NBM. For example assuming Gaussian distribution will give rise to Gaussian Naive Bayes (GNB) or multinomial distribusion will give Multinomial Naive Bayes (MNB). Naive Bayes Model works particularly well with text classification and spam filtering.

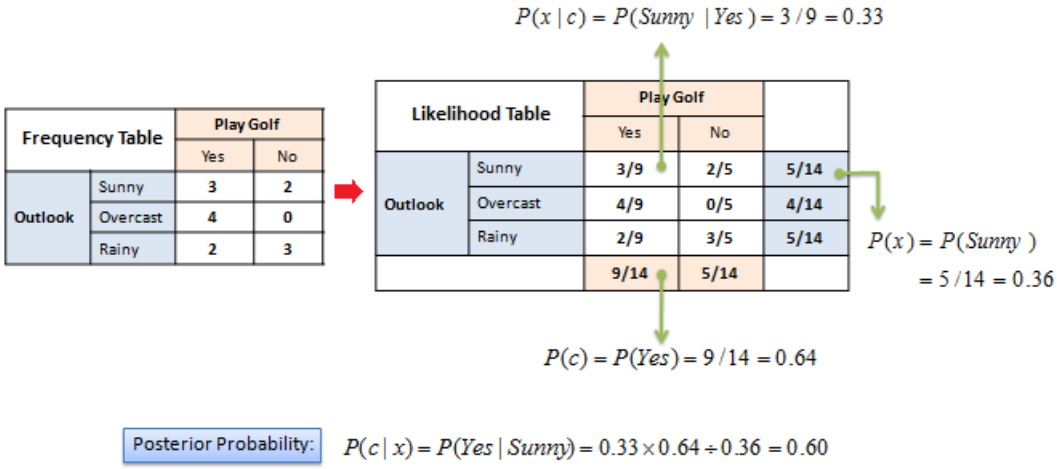
Advantages of working with NB algorithm are:

- Requires a small amount of training data to learn the parameters
- Can be trained relatively fast compared to sophisticated models

The main disadvantage of NB Algorithm is:

- It's a decent classifier but a bad estimator
- It works well with discrete values but won't work with continuous values (can't be used in a regression)

We used a pre-trained NB model for this task. It was the 'bag of words' (BOW) model and two different ways of creating BOW using CountVectorizer and TfidfVetorizer. Naive Bayes is a simple but useful technique for text classification tasks. We can create solid baselines with little effort and depending on business needs explore more complex solutions.



3.2.1 BERT

In Natural language processing one of the greatest challanges is the availability of sufficient amount of training data. Bert is a pre trained model which has been trained on a huge wikipedia corpus which will help us in the sentiment classification task we are going to perform. In Naive Bayes' the entire sentence which is to be classified as positive,negative or neutral is is treated as a bag of words model where the order of the words doesnt matter. But Bert is a deep learning based architechture which leverages the working principle of transformers, attention mechanism and masking.

The sentence to be classified is padded witrh a [CLS] token at the first which will help in classification and between sentences there are [SEP] tokens. Each of the sentences are tokenized using a bert based tokenizer and along with the token embeddings, segment embeddings and position embeddings are fed into the layered Bert architechture.

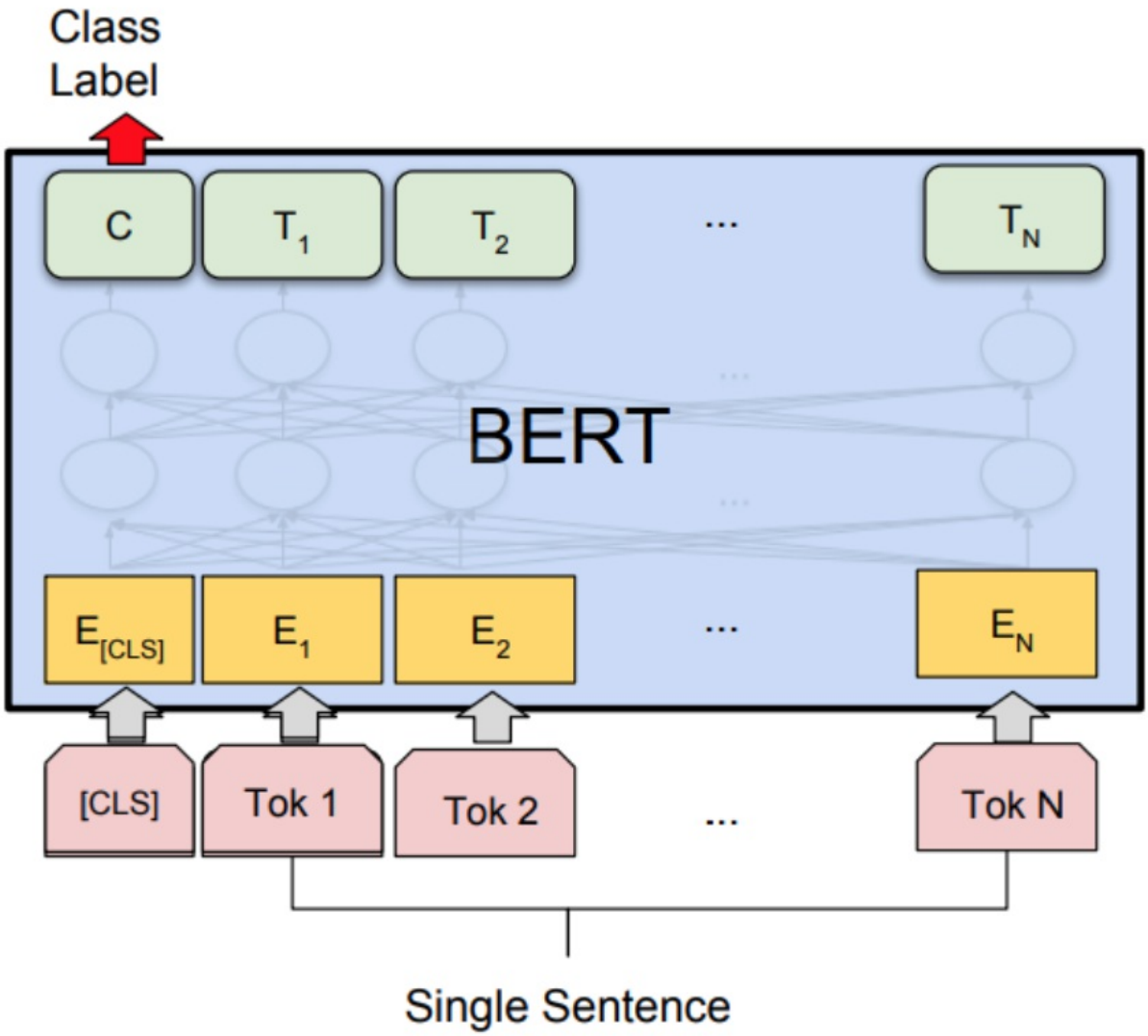
Token embeddings: A [CLS] token is added to the head of first sentence and a [SEP] token is put after the end of a sentence.

Segment embeddings: It indicates which sentence(A or B) is added with which token. It helps in distinguishing sentences.

Positional embeddings: It indicates the position in a sentence

The entire context of the sentence flows through the layers and finally the [CLS] token helps in the sentiment classification of the sentence in the classification layer. Thus we get a positive, negative or neutral class label for each text we pass as an input to BERT. This is the sentiment we want to get about the sentence.

We have used a fine-tuned BERT which has been fine tuned on Financial PhraseBank dataset which provides sufficiently good results for financial text analysis.



3.3 Textblob (NLTK)*

TextBlob is a python library for Natural Language Processing (NLP).TextBlob actively used Natural Language ToolKit (NLTK) to achieve its tasks. NLTK is a library which gives an easy access to a lot of lexical resources and allows users to work with categorization, classification and many other tasks. TextBlob is a simple library which supports complex analysis and operations on textual data.

For lexicon-based approaches, a sentiment is defined by its semantic orientation and the intensity of each word in the sentence. This requires a pre-defined dictionary classifying negative and positive words. Generally, a text message will be represented by bag of words. After assigning individual scores to all the words, final sentiment is calculated by some pooling operation like taking an average of all the sentiments.

TextBlob returns polarity and subjectivity of a sentence. Polarity lies between [-1,1], -1 defines a negative sentiment and 1 defines a positive sentiment. Negation words reverse the polarity. TextBlob has semantic labels that help with fine-grained analysis. For example — emoticons, exclamation mark, emojis, etc. Subjectivity lies between [0,1]. Subjectivity quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information. TextBlob has one more parameter — intensity. TextBlob calculates subjectivity by looking at the 'intensity'. Intensity determines if a word modifies the next word. For English, adverbs are used as modifiers ('very good').

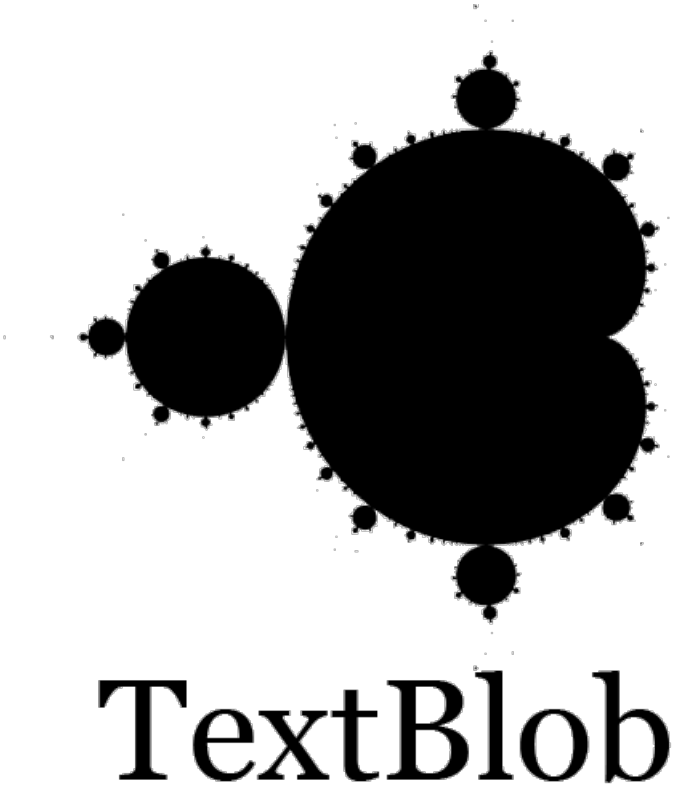
For example: We calculated polarity and subjectivity for "I do not like this example at all, it is too boring". For this particular example, polarity = -1 and subjectivity is 1, which is fair.

However, for the sentence "This was a helpful example but I would prefer another one". It returns 0.0 for both subjectivity and polarity which is not the finest answer we'd expect.

It is expected that if the library returns exactly 0.0 either if your sentence didn't contain any words that had a polarity in the NLTK training set or because TextBlob uses a weighted average sentiment score over all the words in each sample. This easily diffuses out the effect of sentences with widely varying polarities between words in our case : 'helpful' and 'but'.

More often than not, for simple cases, TextBlob works just fine.

*We used this approach not on the collected reports but on real-time data collected from the Eikon API.



4. Trading Strategy

We used sentiment analysis as a measure to predict the rise and fall of stocks. We can actually see that the nature of sentiments of the comapany's news and reports have an effect on the value of the stocks with time. Using this knowledge, one can create a trading strategy to buy/sell stocks.

Let's take a look at it -

```
In [22]: pd.set_option('display.max_colwidth', None)
         df_apple[['text','basic_sentiment','sentiment_from_bert','2M','10M','15M','30M','45M','60M']].head(12)
```

Out[22]:

	text	basic_sentiment	sentiment_from_bert	2M	10M	15M	30M	45M	60M
0	Dow Jones Selected Stocks - September 12	neutral	neutral	0.000000	-0.125011	-0.062506	-0.026788	NaN	NaN
1	Apple finally lets users change default browser from Safari to Chrome -here's how to do it The change comes with iOS 14, which is also introducing widgets and new privacy features	neutral	neutral	-0.035708	-0.026781	-0.151759	-0.080343	-0.053562	NaN
2	Apple (AAPL) Stock Sinks As Market Gains: What You Should Know	neutral	neutral	-0.008931	-0.017862	0.008931	0.008931	0.044655	-0.107172
3	Apple designs face mask for its employees	positive	negative	-0.008932	0.080386	0.044659	0.080386	0.044659	-0.008932
4	KShark's Carbon Scanner app comes to Apple App Store	positive	neutral	-0.008932	0.080386	0.044659	0.080386	0.044659	-0.008932
5	UKRAINE: APPLE-RUSSIA-APPS-1: Apple doesnt take down Russian TV channels apps from Ukrainian AppStore	neutral	neutral	-0.187149	-0.187149	-0.160414	-0.053471	-0.213885	-0.222797
6	UKRAINE: APPLE-RUSSIA-APPS: Apple doesnt take down Russian TV channels apps from Ukrainian AppStore	neutral	neutral	-0.080350	-0.044817	-0.009106	-0.026783	0.080350	0.133917
7	Reuters Insider - Apple is the bellwether of the FAANG stocks: Shannon Saccocia	neutral	neutral	-0.388651	-0.210694	-0.219168	-0.210248	-0.076446	-0.005084
8	Apple to remove NTV app from Appstore in Ukraine at Security Service's behest	neutral	negative	-0.387131	-0.369332	-0.396031	-0.413830	-0.262537	-0.235839
9	UPDATE 2-Apple revises App Store guidelines, loosening some in-app payment rules	neutral	neutral	0.062911	0.140671	0.477005	0.467105	-0.018990	0.463505
10	Reuters Insider - Apple App store policy update will affect Microsoft and Google game streaming apps	negative	neutral	0.089339	0.057760	-0.071975	0.890688	0.296896	-0.166442
11	Reuters Insider - The Pre-Market Rundown 2: September	positive	neutral	0.130654	-0.578663	-0.101099	-0.006758	0.594702	0.584790

Above, we have sentiments from 2 different methods. On the right, we can see price of stocks at time 2', 10', 15', 30', 45' and 60' minute marks. As we can clearly see, the price varies somewhat closely according to the way the nature of the report is, i.e., sentiments. Given this, a trading strategy which automatically buys stocks when the stock is predicted to rise and sell when the stock is predicted to fall.

5. Conclusion

The aim of this project was to explore the interrelationships between stock market fluctuations of specific companies and media coverage characteristics, particularly the association of these companies with sentiment and emotions. Furthermore, we wanted to find out whether sentiment and emotionality in media coverage can improve the prediction of stock market fluctuation above and beyond predictive models that are merely based on media attention or sentiment. It is important to know how media coverage of companies triggers share price reactions, and if so, what kind of coverage impacts the evaluation of the company on the stock market most. The results of this study do not only tells us what kind of impact the media news' and analyst reports have, but also help in creating a trading strategy to buy/sell stocks.

- Our project consisted of 2 parts:
  - Creating a tool for automated collection of data
  - Performing textual analysis on the data using various algorithms
- We used Robotic Process Automation for collecting data from Eikon datastream
- The data consisted of reports of 100 different international companies listed in the S&P 500 and Dow Jones listing respectively
- After this, 3 different techniques to analyse the collected data - Naive Bayes', Textblob (NLTK) and BERT
- Sentiment analysis gave us a measure what the nature of the report/news is
- We used this measure to create a trading strategy to buy/sell stocks

Final Remarks

- Other analysis could be done on the reports other than Sentiment Analysis
- More data points could be extracted with the use of the Eikon API and integrated into the collected data
- This would create better Deep Learning models, and hence, more accurate predictions
- More accurate predictions would enhance the trading strategy
- This could create an end-top-end pipeline for trading stocks

Documentation

Steps to run the code :

- Set the desired filters in the Eikon datastream desktop application.
- Create a python virtual environment and install all the software packages.
- Run the RPA_data_download.ipynb notebook to start downloading reports from the Eikon datastream.
- Keep the code running until you have the desired number of reports.
- Start the FinBert_Model_Example.ipynb or Eikon.ipynb notebook to automatically import, serialize, pre-process and analyse the reports and get sentiments.