UNSUPERVISED SENTIMENT ANALYSIS ON STOCK MARKET TWEETS

Archit Sangal
IMT2019012
archit.sangal@iiitb.ac.in

Pratyush Ranjan
IMT2019065
pratyush.ranjan@iiitb.ac.in

Abstract

Most sentiment analysis methods applied to stock market predictions are supervised. The dataset usually contains a large corpus of tweets or snippets from news articles, and a label denoting whether the tweet would have a positive or a negative impact on the corresponding share. However building a model for stock market sentiment analysis is a labour intensive and expensive process. It requires manual analysis of every tweet or news article and assignment of a correct label. On top of that, inorder to keep up with time, the labelling activities need to be performed very frequently as hundreds of news articles are published everyday and thousands of tweets are made every hour. This project attempts to solve this issue by implementing an unsupervised method of sentiment analysis on stock market data. Word embeddings are trained from the training data and clustered into two classes. Sentiment scores of individual words are calculated. It is followed by calculating the sentiment of an entire tweet or a text by combining the sentiment scores of its individual words. CLS token obtained from Bert[5] embeddings would also be explored to cluster a tweet or a text as a whole, into a positive or a negative class. Data mining related to stock markets would be done using Twitter's APIs inorder to train the model on the latest data.

1 INTRODUCTION

Sentiment Analysis through business articles and social media platforms like twitter is widely used to predict the movements of the stock market. By using sentiment analysis, investors can attempt to determine if the market is driven by emotion so that they can pick up changes in sentiment and take appropriate steps before there is any news to explain the behaviour of stock prices. Sentiment analysis for stock market is done through Natural Language Processing methods to decide whether some piece of text is positive or negative (or neutral). It is mostly done through supervised learning where the dataset contains a corpus of tweets/news-articles with a label describing the sentiment of the text. However, labelled data for tweets and business articles is limited. Every day there are numerous tweets made, hundreds of business articles published. Manually labelling each and every tweet and article is a cumbersome process, and requires a lot of human effort.

In this project, we propose an unsupervised method of sentiment analysis on stock market tweets and business news. The high level idea is that most words for a particular sentiment (positive or negative) tend to appear together. Word embeddings (eg: skip gram Word2Vec [1]) can be trained on a large corpus of training data. It would be followed by clustering the words in the corpus into two classes - positive and negative. The cosine similarity of each word to its respective class would be calculated. Then the sentiment of a piece of text will be determined by appropriately weighing each word in the text by (eg: Tf-Idf [2]) and multiplying the weight with the cosine similarity for both cluster centres. A set of two scores will be obtained which will be further used to determine the sentiment of the text.

2 DATASET

We will use the stock market tweets data 2021 from kaggle [3]. Although the dataset contains tweet data along with sentiment labels, we will not be using the labels for our task. Instead we would only use the tweets and perform an unsupervised learning as described in the introduction.

3 PROCESS FLOW

- Data cleaning
  - Removing punctuations
  - Removing stop-words
  - Expanding contractions
  - Lemmatising words
  - Removing extra spaces
  - Correcting spellings.

- Training word embeddings (Word2Vec [1]) using text from the training corpus.

- Applying K-means clustering [4] on the word embeddings to obtain two clusters corresponding to the positive and negative sentiments.

- Assigning cosine similarity for each word with respect to its cluster centre.

- Calculating Tf-Idf for each word in the text.

- Multiplying Tf-Idf for a word in a text by its cosine similarity with respect to its cluster. Adding these computations for all the words in the text, to determine the sentiment score of the text for a particular cluster. The procedure is dealt in detail the next section.

4 DETERMINING THE SENTIMENT OF A TEXT

Let the given text be $T < x_1, x_2, \ldots, x_n >$, where x_i is the ith word of the text T. Let $\lambda(x_i)$ denote the Tf-Idf of x_i. Let $S_+(x_i)$ and $S\_(x_i)$ denote the cosine similarity of embedding of x_i with the two cluster centres.

The sentiment score of the text T with respect to the positive class, $\Phi_+(T)$ will be given as:

$$\Phi_+(T) = \sum_i \lambda(x_i) \,.\, S_+(x_i)$$

The sentiment score of the text T with respect to the negative class, $\Phi\_(T)$ will be given as:

$$\Phi\_(T) = \sum_i \lambda(x_i) \,.\, S\_(x_i)$$

By applying softmax over $[\Phi_+(T), \Phi\_(T)]$, we would get the probability of the text belonging to either clusters.

Probability of the text being a positive sentiment $P_+(T)$ can be given as:

$$P_+(T) = \frac{e^{\Phi_+(T)}}{e^{\Phi_+(T)} + e^{\Phi\_(T)}}$$

Similarly the probability of the text being a negative sentiment $P\_(T)$ can be given as:

$$P\_(T) = \frac{e^{\Phi\_(T)}}{e^{\Phi_+(T)} + e^{\Phi\_(T)}}$$

5 BERT EMBEDDINGS

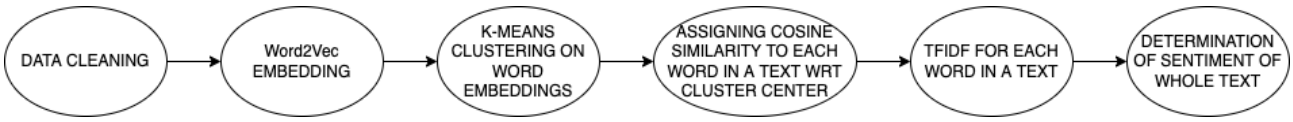
Apart from word2vec embeddings, we will experiment with Bert [5] embeddings too. Bert provides a CLS token which is a pooled embedding of the input text. We will apply K-Means clustering [4] on the pooled embeddings to divide the training corpus into two clusters repressing positive and negative sentiments. Clustering using word2vec embeddings was being done on individual words. However, Bert provides an embedding for an entire text which can be directly used for clustering. Therefore through Bert, a text will be used for clustering as a whole.

6 CHALLENGE PERTAINING TO DATA

The dataset that will be used does not contain recent tweets related to stock market. But inorder for the model to be practical at the present time, more recent data is required. Therefore apart from the dataset that we have, we would perform twitter data mining using Twitter's APIs. We would be mining data related to stock markets.

7 EVALUATION METRIC

The model should be able to deal with both balanced and imbalanced datasets. Discriminating positive sentiment from negative sentiment is important in sentiment analysis concerning stock market predictions so that the investors can take proper actions at the proper time. The evaluation metric would therefore be the area under the receiver operating characteristic curve (roc-auc curve [6]).



Process flow

8 REFERENCES

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781, 2013

[2] sklearn.feature_extraction.text.TidfVectorizer {http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html}

[3] Sohel Rana. Stock Market TWEETS Data-NLP-2021. Kaggle. {https://www.kaggle.com/sohelranaccselab/stock-market-tweets-data-sentiment-analysis}

[4] sklearn.cluster.KMeans {http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html}

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805, 2018

[6] sklearn.metrics.roc_auc_score {https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html}

[7] Rafał Wójcik. Unsupervised Sentiment Analysis. Towards Data Science. {https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483}