AutoPhrase Stock Price Prediction Performance based on Financial News

Yunhan Zhang and **Mingjia Zhu** and **Liuyang Zheng**
[yuz047, mzhu, lizheng]@ucsd.edu

Abstract

Ongoing

1 Introduction

Nowadays, we are seeing a great amount of data generated everywhere. The Data will bring us both challenges and opportunities. As AutoPhrase (Shang et al., 2017) is developed, an automated and domain-independent phrase mining method, we see its potential to be used on extracting quality phrases from numerous daily financial news and editorials, to improve the current data prediction model using bag-of-words and other conventions as its entry. This experiment could be very promising since it can be a quick classifier of long text reports. Analysts then may use the result as a supportive reason for future investments. Thus, adapting this method could improve the efficiency with the same accuracy if used properly.

In this project, we will give a report of comparison of performances among conventional methods and AutoPhrase on predicting the change of stock price (increase or decrease). We will also include the comparison between the performances of AutoPhrase and deep learning methods on predicting the change of the stock price.

2 Dataset

2.1 Financial news dataset

We mainly use reports and news from Yahoo Finance as the input of our models. We found articles that only mention a single company to prevent the noise. For our input, there are two main types of articles: analyst report and press release news. We searched for the articles under "press releases" and "research reports" categories on the web page of a company and used beautiful soup to scrape the articles. We plan to collect news about 30 companies and 30-50 articles within the past one year for each company to test our models.

2.2 Stock price dataset

Our stock price dataset is from a kaggle project that updates all stock prices regularly and has been verified. The raw dataset covers 1657 major nasdaq stock prices from their beginning to present. We are selecting one at a time, and compute its 5-day-average, 10-day-average and so on from the time a particular piece of news is released. We started with Apple Inc. to test our methodology and to try automations. (Mooney, 2020)

2.3 Positive and negative words in financial news dataset

We utilized the "Financial positive and negative terms list" created by Bill McDonald. This dataset contains common positive and negative words in financial news. The dataset contains 354 positive words and 2350 negative words. It helps us to determine the attitudes of financial news articles.

3 Models

Ongoing

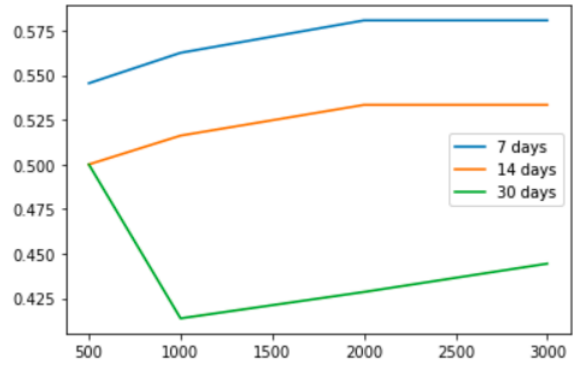
3.1 Baselines

We used the Bag-of-Words method to generate the baseline model. After removing stopwords, removing punctuation, and stemming the text, we performed Bag-of-Words to get the top n words with the highest frequencies. Then we compare the extracted words with positive and negative words in the financial news dataset. If the number of positive words is larger than that of negative words in a document, we give the output as "True". Below is how the model works:

```
#create a dictionary (dict) that
   contains each word (key) in the
   document with the number of times it
    appears in the text (value)
positive_word_count = 0
negative_word_count = 0
for each high frequency word do
  if word in positive_word_list
    positive_word_count += dict[word]
    else if word in negative_word_list
      negative_word_count += dict[word]

if positive_word_count > negative_word_count
  Output: True
else
  Output: False
```

We experimented with different n and tried to predict the change of stock price of Apple Inc. after 7 days, 14 days, and 30 days of the new published date. We used 25 articles to test the model. The graph below shows the change of accuracy. We could see that the baseline model performs the best for n=2000 and the stock price changes after 7 days, with an accuracy of 0.57. In our other model, we will use more sophisticated models to achieve better accuracy.



3.2 AutoPhrase Model

After building the baseline model, we tried to explore more methods to predict the change of stock price. Instead of using native methods like Bag-of-Words to extract the words in articles, we considered to extract high-quality phrases. Therefore, we utilized AutoPhrase to perform phrase mining tasks. AutoPhrase is a framework that extracts quality phrases from text (Shang et al., 2017). After running AutoPhrase, we will get the top high-quality phrases in an article with their scores. We will then use the positive and negative word lists to determine the attitude of the words. Finally, we use the scores and their attitude to predict the stock price change. Below is how we get the score for each extracted phrase:

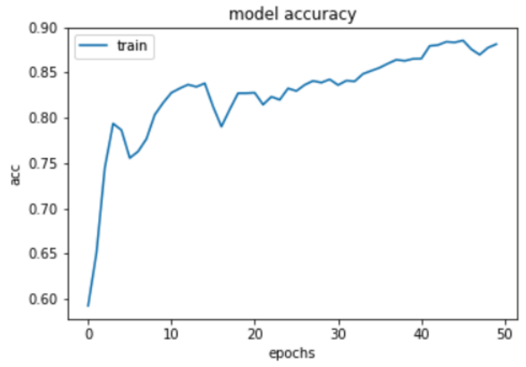
```
for each high quality phrase do
    Split phrase into words
    coefficients = list()
    for each word do
        if word in positive_word_list
            Append 1 to coefficients
        else if word in negative_word_list
            Append -1 to coefficients
    if sum of coefficients = 0
        coefficient = 0
    else if sum of coefficients >= 1
        coefficient = 1
    else
        coefficient = -1
    score of phrase = original score * coefficient
```

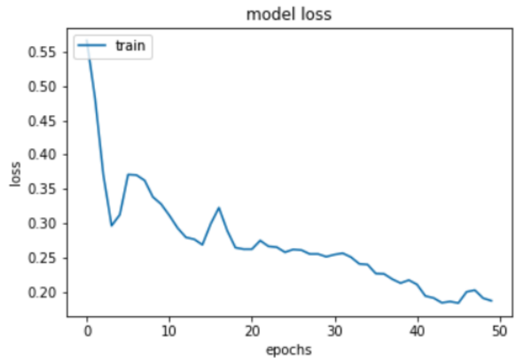
By following the method above, we get an adjusted score for each phrase. We then train a classifier to make the prediction.

This model is a work in progress. For now, we only found positive and negative word bank in financial news instead of phrase bank, so we are only able to split the extracted phrases into words for preprocessing. If we find more resources, we might improve this method.

3.3 Doc2vec and LSTM Model

Given that the board of the company will always try to make a development plan, no matter short term or long term, we know that there might be some connections between each financial report in the analysis of these financial decisions. Thus, we tried to use the Doc2vec model to create numerical representation of documents. Compared to the Bag-of-Words model, it was designed to address the problem of word order and syntax that BoW has, and after the Doc2vec model is trained, we can also fit in a new sentence and predict its paragraph vector, and that will be very useful in our stock price movement prediction because that means every time we have a new financial article, we can find its corresponding paragraph vector and used as the input for our prediction model. We are going to use LSTM for our prediction model. LSTM stands for Long Short Term Memory Network, and it's a special version of Recurrent Neural Network(RNN). The LSTM architecture has a large range of modules for each time step update, and the output of each step update is controlled by a series of gates, which either add or remove the information from being updated to the cell state. With that being said, we will first obtain our vector embedding matrix from our Doc2vec model, and then we will fit in the embedding matrix as a weight to train our LSTM model, and from that we are able to predict the movement of stock price from specific time span given the financial news articles. But the problem right now is we do not have a large enough news article dataset to do proper neural network training, and the performance that we got from the current dataset that consists of a 10-ish number of financial reports is relatively trivial. However, our team has already found a new way to scrape the news articles and financial reports from the website and we should be able to test the model on a large scale of news sources. Even though we don't have the result on our own dataset, we did test the model on a sentiment analysis dataset [4] that was found online consisting of only the financial news headlines, and a sentiment(neutral, positive, negative) assigned by a retail investor. The model gives the following results about the accuracy and loss:





Thus, we confirmed the model's capability of predicting binary labels once we can perform the training on our own dataset.

3.4 BERT Model

While the main advantage of LSTM is that it keeps the long-term dependencies, BERT does more so we chose it to be the state of the art of our project. Upon our findings, we find that BERT is extremely helpful on our sentiment analysis model and good for our stock prediction. That is because models before BERT are one-way read only: either read sequentially from left to right, or from right-to-left. BERT is different from them. By masking part of the words, BERT is able to read from left to right at the same time it reads right from left. This feature enables BERT to learn a sentence but not to learn a sequence of words. Thus BERT is able to catch the connection between words and thus it should know exactly what a word means in that sentence.

It is also an unsupervised learning method, meaning that it does not human labeling work. Thus it can adapt to the most updated text data emerging most recently. For example, the previous work of IMDB case, the unsupervised learning method also solves a crucial problem. There are also some critics who are sarcastic that label a thumb up but actually hate the movie when you read the review. Unsupervised learning does not depend on the label or stars the review gives but it learns the similarity among the reviews, so it will still classify this particular situation as a thumb down. In our situation, unsupervised learning relieved us from labeling in the training process and may directly compare the sentiment analysis report to actual stock performance label. That is, if there is a positive sentiment in accordance with the rising label of the stock.

BERT is also the first NLP technique that relies on self-attention mechanisms. This gives BERT the ability to determine the change of meaning of the words. As Lutslevch suggests in the editorial, the word "is" often changes meaning when the paper goes. BERT, however, could often make a good association with the correct one. " That boy says the animal beside his mom is a cat. He is correct. It is an American Short-hair cat." In these sentences, the word "is" changes its meaning three times and BERT is designed for distinguishing these.

As we are using it as a state-of-art method, we are now choosing to directly cite its result as 85% accuracy by Jack-Wells. We are looking forward to further adapting his code and improving it to a high accuracy with our dataset (Wells, 2020).

4 Methodology

Ongoing

5 Discussions

Ongoing

6 Conclusion

Ongoing

References

Paul Mooney. 2020. Stock market data (nasdaq, nyse, sp500). https://www.kaggle.com/paultimothymooney/stock-market-data.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. 2017. Automated phrase mining from massive text corpora. *CoRR*, abs/1702.04457.

Jack Wells. 2020. Stock-market-prediction-nlp-bert. `https://github.com/Jack-Wells/Stock-market-prediction-NLP-BERT`.