Final Report

Final Year Project

Financial News Analytics with Hybrid Sentiment Analysis

A comparison of different techniques on stock prediction

ANGARA Jacky 15101317D

Acknowledgement

This work would not have been possible without the supervision of Prof. Chan Chun Chung Keith, who has been supportive on this project and worked actively to provide me ideas and insight on this field.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents and grandparents, whose love and guidance are with me in whatever I pursue.

Contents

Abstract

Manual news analytics mainly relies on reading long passages, understanding the overall content, and ultimately, predict the stocks based on our opinion. Machine learning techniques have been used to predict various trends including the stock market. However, current financial news analytics have poor accuracy and highly affects the liquidity of the stocks. Current methods have shown poor results because too much focus is centered on the sentiment analysis. By increasing the accuracy of the sentiment analysis and integrating relevant features, we could achieve better results. This study aims to optimize current stock prediction in accuracy and precision.

1 Introduction

1.1 Motivation

Finance-related news are published everyday to the public and data analytics has helped us to understand the large volume of news which is commonly referred as financial news analytics. Given numerous news articles, commentaries from experts and companies' reports, the problem is to come up with an effective methodology to automatically digest this huge amount of information and summarize them into meaningful figures that will help investors in making investing decisions.

There are two main problems with the current financial news analytics: mediocre news analytics provide misleading information, and highly commercialized news analytics disrupt the liquidity of the stock.

To tackle the given problems, this study proposes to increase the accuracy of sentiment analysis and to look for additional features that might impact the stock market (e.g. the number of readers on a given news). This method suggests a different outlook from current methods by searching for additional relevant information that affects the stock market.

1.2 Objective

The aim of this project is to create an accurate predictive model to provide end users with insights for stock trading decisions. To achieve this goal, several objectives have been set as follows:

1. Conduct analysis on graphs and numerical data to obtain information about stock trends
2. Understand and maximize the most suitable artificial intelligence architecture to aid the sentiment analysis
3. Assess the performance of different classifiers in sentiment analysis
4. Develop a flexible system which can work with different datasets
5. Discover relevant features to increase the accuracy of the financial news analytics
6. Evaluate the correlation between sentiment polarity accuracy and stock prediction accuracy

The objectives build up towards a specific outcome, well-functioning system which will compare the performances of different classifiers. Additional features will be extracted to help the adjustment of the sentiment polarity.

2 Related Works

Multiple studies have attempted to increase the accuracy of sentiment analysis. Hybrid sentiment analysis integrates the benefits from both lexicon-based and learning-based sentiment analysis. According to [1], hybrid sentiment analysis is able to outperform the latter with relatively high margin. This study would like to benefit from the performance and consistency of hybrid sentiment analysis in predicting the stock market.

The hybrid technique will be used to achieve better performance and higher consistency. Hybrid sentiment analysis combines high performance lexicon-based sentiment analysis and learning-based sentiment analysis, and expects the combination of the two will increase the accuracy. A study done by [2], reported that 86.6% accuracy can be achieved using lexicon-based technique (on customer reviews) [3] and 95.5% using Support Vector Machine (SVM) which is a learning-based technique (on movie reviews) [4]. The study also suggests that a hybrid sentiment analysis could achieve 85.4% accuracy on Twitter sentiment analysis.

Hybrid sentiment analysis has shown consistent performance throughout different datasets with combination of different techniques compared to both lexicon-based and learning-based sentiment analysis. This study will further explore the capabilities of hybrid sentiment analysis to a different field (finance).

3 Methodology

Lexicon-based analysis will be used to determine sentiment of ”subjective” texts on the news. Then, a Chi-square test will be performed to extract opinionated texts. These texts will then be passed to learning-based analyzer to be computed. Additional features will be examined to adjust the extent of each polarity. After the adjustment, the sentiment polarity will be converted into price change in the stock market.

3.1 Lexicon-based Analysis

As the baseline, this study will make use of a readily available sentiment lexicon dictionary for comparison. This study proposes 2 lexicon-based analysis for performance comparison: SentiWordNet sentiment analyzer and MPQA sentiment analyzer. Both tools serves as a dictionary in which the sentiment polarity is tagged to each word. Both sentiment analyzer will follow the following algorithm:

3.1.1 SentiWordNet and MPQA

SentiWordNet is a popular sentiment-tagged dictionary that is widely used in many sentiment analyzer. A study have shown that MPQA has a better accuracy than SentiWordNet [5]. On this stage, both lexicon dictionaries will be used, and then compared. The better performing dictionary will be used on the hybrid sentiment analysis.

Algorithm 1 Sentiment Analyzer

Require: Movie Reviews M(T,S), where T is the text passage and S is the sentiment polarity (Pos/Neg) **return** sentimentTotal

```
procedure CalculateSentiment($M$)
    $sentimentPositive, sentimentNegative \leftarrow 0,\ 0$
    $sentimentTotal \leftarrow [\ ]$
    for $i \in \{1, \ldots, len(M)\}$ do
        $aList \leftarrow tokenize(T(i))$
        for $j \in aList$ do
            if $j = Positive$ then
                $sentimentPositive \leftarrow sentimentPositive + 1.$
            else
                if $j = Negative$ then
                    $sentimentNegative \leftarrow sentimentNegative - 1.$
                $sentimentTotal(i) \leftarrow sentimentPositive + sentimentNegative.$
```

3.2 Sentence-Level Sentiment Analysis

3.2.1 Valence Aware Dictionary and Sentiment Reasoner

Valence Aware Dictionary and Sentiment Reasoner (VADER) is a readily-available analyzer with high accuracy [6]. VADER is able to grasp the sentiment of a document with high accuracy by using lexicon contextual sentence structure. This study make use of VADER as a comparison to SentiWordNet and MPQA analyzer. This VADER Analyzer is available inside the nltk package and is imported as shown below:

```
from nltk.sentiments.vader import SentimentIntensityAnalyzer
from nltk.sentiment.vader import SentiText
```

Benefits and drawbacks of the VADER analyzer will be discussed in the latter part of this paper. VADER will be used on exactly same dataset for SentiWordNet and MPQA to determine their accuracy and correlation with stock price.

The VADER analyzer includes emoticons identifier which will understand the emoticons and determine the sentiment of it. On a random sample of tweet dataset, it is found that 19.6% of the tweets contain emoticons [7]. Comparatively, other body of text will not have emoticons present in it, especially financial news. This emoticon identification feature will not be beneficial in analyzing news. Therefore, lower classifying accuracy is expected.

According to [2], sentence-level sentiment analysis [3] performs better than other lexicon-based sentiment analysis. The aim of this stage is to create a better performance sentiment analyzer which will make the overall analyzer reliable. Since there are multiple analyzer used in this project, it is important to make sure that all analyzers perform at their best.

Feature selection is important in classifying opinionated text. The followings are the features that will be taken into account:

1. **Bag-of-Words (BOW) or Bag-of-Sentences (BOS)**:

   Bag-of-Words is a simple technique where texts are split into a single word and the frequency of each word will be calculated throughout different texts. Bag-of-Sentences, which is similar to the former, is a term where texts are split into sentences.

2. **Part-of-Speech (POS) Tagging**:

   POS Tagging is needed to label each term to their respective grammatical context. This can be helpful when chunking words or chinking sentences.

3. **Negation**:

   Negation reverse the sentiment polarity of the trailing word. Mishandling negation words will cause the absolute opposite meaning which will result in faulty sentiment polarity.

4. **Context Shifter**:

   Context shifter are mainly conjunctions (but, however, despite, although). Polarity will be recalculated if a sentence contain such words.

5. **Modifiers**:

   There are two types of modifiers: word modifier and noun modifier.

   Word modifiers: slightly, pretty, and extremely

Noun modifiers: a few, a ton of, several

These features will be used to refine selection of opinionated sentence and sentiment accuracy of subjective sentences.

3.3 Knowledge Base for Text Structure and Contextual Details

As listed previously in Section 3.1.2 (feature selection), these features will be used to construct and update the opinions of the sentences.

BOS is used instead of BOW because the position of each word in a sentence matters. A feature vector will be built based on these BOS and the subjectivity will be calculated. Then, sentiments of the subjective sentences will be computed and the polarity of the sentiment will be frequently updated using the sentence structure and the feature vector.

POS Tagging will be used on this to classify whether an adjective can be used as an opinion word when located near a noun, which is a feature word. These opinion words will determine whether a sentence is opinionated or not.

Negation changes the polarity of a word greatly. There are multiple approach in detecting and resolving negation. One of those is by using dependency parsing [8] or maximum-margin approach [9]. Another approach requires attaching the negation to the trailing word.

```
not happy
not_happy
```

Then, reverse the sentiment polarity of the word. Negation detection is a broad topic itself.

Context shifter is important in sentence-level sentiment analysis. It has minor to major effect on a sentence sentiment polarity. It affects the sentence by influencing the polarity of the opinion words.

Unlike negation, modifiers are harder to deal with. Negation change the sentiment to the opposite, but modifiers slightly change the sentiment to a certain degree.

```
very nice
pretty nice
```

Both phrase will have a slightly different sentiment polarity due to the word modifiers.

3.4 Learning-based Analysis

Learning-based analysis will be used to analyze and classify new instances of opinionated sentences. With the help of chi-square test or knowledge base (feature selection), the new instances can be classified and passed to this analyzer.

There are numerous well-performing machine learning algorithms that can extract accurate sentiment, this study will focus on Long-Short Term Memory (LSTM which is a type of Recurrent Neural Network (RNN). Comparisons have been done by other studies [2], [10] and the results have shown that RNN is highly reliable in context of Natural Language Processing (NLP) because text data are sequential in nature.

RNN is an artificial neural network which is capable of detecting and understanding sequences. It is able to grasp context of the given input of sequences. LSTM network and Gated Recurrent Unit (GRU) are the 2 main RNN networks that will be focused when talking about NLP. Both performs excellently when classifying sentiments of a body of text. Multiple studies have argued which one performs better, because theoretically LSTM has a stronger architecture due to the 1 additional gate compared to GRU. LSTM make use of the forget gate, update gate, and output gate during computation, whereas GRU uses only 2 gates, which are update gate and reset gate. GRU is computationally efficient compared to LSTM and requires less computational power.

However, a study proposed performance comparison between LSTM and GRU on NLP and showed that GRU has a slightly higher performance compared to LSTM (Appendix III) [10]. On the same study, we can observe good performance from Convolutional Neural Network (CNN) on some occasion, but consistency and robustness is important when choosing the right network model. Thus, this study will conduct an experiment to compare which of the two is better. Four experiments will be done in the following ways:

1. 1 LSTM layer with 100 units and 1 dropout layer
2. 2 LSTM layers with 100 units each and 2 dropout layers
3. 1 GRU layer with 100 units and 1 dropout layer
4. 2 GRU layers with 100 units each and 2 dropout layers

Among this four techniques, only one will be used for the hybrid sentiment analyzer. Both time and performance are measured and compared to find the most suitable approach. The dropout layers are introduced to the model to prevent overfitting. Overfitting is a detrimental problem because the model will try to perform its best during model training which may result in poor performance when predicting the test data. There are other methods to prevent overfitting including cross-validation and training with more data. However, using dropout works really well and consume less time. Thus, the method of preventing overfitting in this project is using dropout layers.

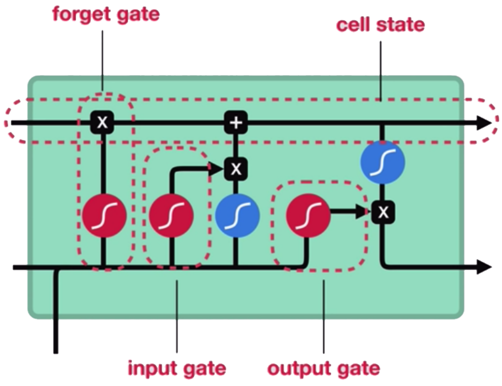


Figure 1: LSTM network architecture

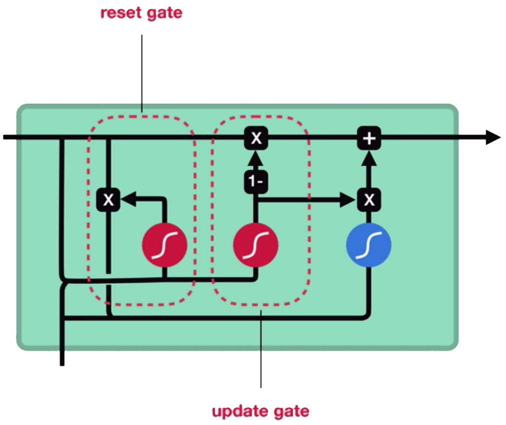


Figure 2: GRU network architecture

Figure 1 and 2 shows the architecture of both LSTM and GRU. LSTM use the cell state to tackle short-term memory problem in typical RNN. Information will be passed through all the gates to narrow down the amount of information going to the cell state. The forget gate is used to choose which information should be kept. Every information are vectorized before processed. Information from the previous hidden state and information from current state is combined and pass through all the gates. The cell state connect throughout the LSTM network. The output gate and the cell state will perform multiplication and then pass additional information to the next LSTM unit, which we previously called ”information from the previous hidden state”.

GRU uses the hidden state instead of the cell state and only has 2 gates, reset gate and update gate. Update gate acts similarly as the forget gate and input gate. The reset gate determines how much passed information to be forgotten. This removal of cell state and gate reduces the the amount of computation. Operations in each unit of both network includes *tanh* function (denoted as the red operation) and sigmoid function (denoted as the blue operation). Additions and multiplication of matrices are also present in each unit.

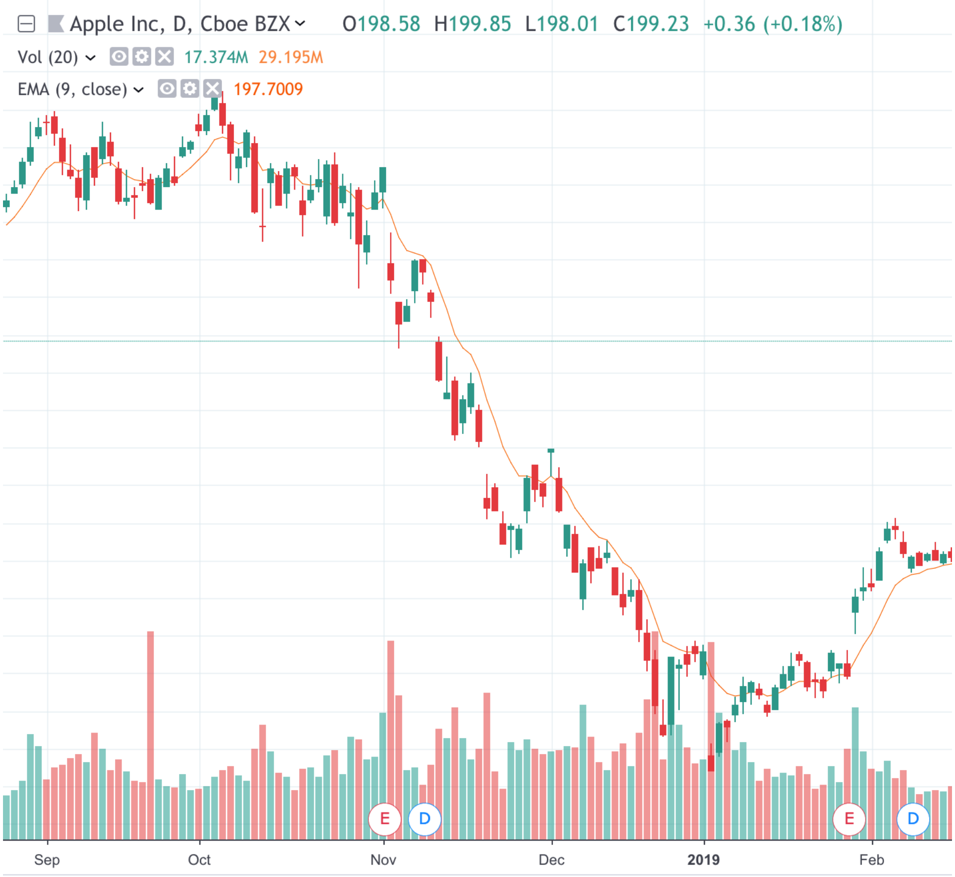
There are numerous variables to adjust when working with neural networks. Number of layers, number of neurons, training batch size, and number of epochs can be adjusted to increase the accuracy. This project will try to adjust these variables to find the best performing neural network.

3.5 Technical Analysis

There are a lot of features hidden behind the stock market chart such as price action and movement. The price action and movement tells whether price goes up or down. This is important since people trade with regards to the price action. Moving average is calculated based on the price action.

3.5.1 Moving Average

Moving average is an indicator which shows the trend in which the market is going. There are two types of moving average, simple moving average (SMA) and exponential moving average (EMA) (Appendix V). The following is an example of moving average in a stock chart. The moving average line is shown in red line chart.



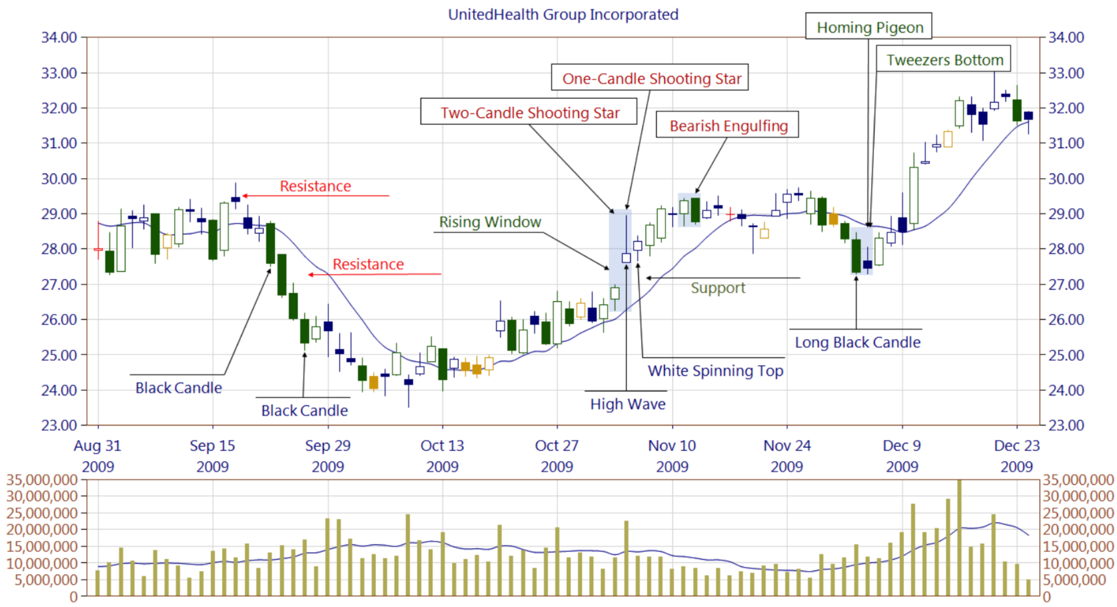
Both approached the problem differently. However, the latter performed better when there is a quick change in price. Thus, EMA will be used in this study to predict the stock.

Moving average is known as a lagging indicator because it calculates the average of previous days. Therefore, it is arguable that EMA can predict the change in the stock price. EMA will mainly be used to aid the sentiment analyzer to increase the accuracy of the stock price prediction.

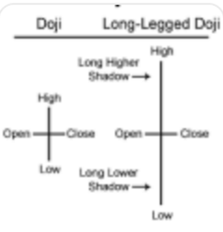
Since moving average is a minor feature in this project, simple experiments were performed to find out which EMA works the best for next 1-day prediction. The technique used is to perform training on stock data and its respective EMA value. After changing the parameters, it is found that EMA of 2-days works the best for predicting next day stock price. Thus this project will use 2-day EMA for prediction.

3.5.2 Candlestick Pattern Recognition

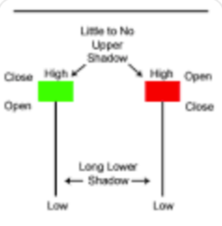
A candlestick shows the opening, closing, high and low price of the unit of time, in this case, daily. Daily candlestick's open starts at the beginning of the day when the stock market opens and closes at the end of the day. High marks the highest price the stock market reached on that day and low marks the lowest price. Traders are using candlestick pattern to trade since they believe that people's action are repetitive. A set of candlestick may form a pattern which indicates a rise or fall in the stock price. Since the occurrence of this patterns are generally followed by a specific trend, traders correlate these patterns to it's respective trend. For example, "shooting star" is often followed by a reversed trend. The following image shows how the patterns correlated to a reversal:



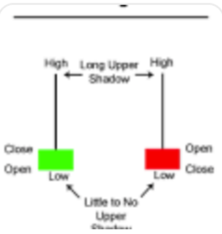
Moreover, numerous traders are using this method in trading which amplifies the price movement. Aside from the "shooting star" there is a huge list of patterns that indicates reversal and continuation. More patterns are as following:



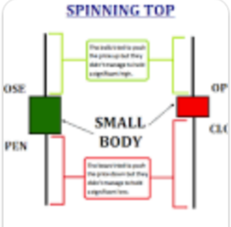
Doji



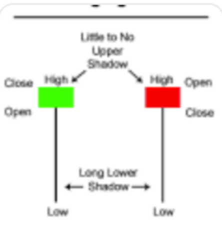
Hammer



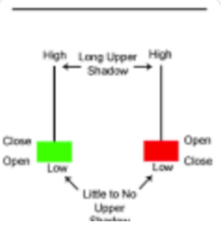
Shooting star



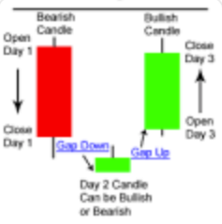
Spinning top



Hanging man

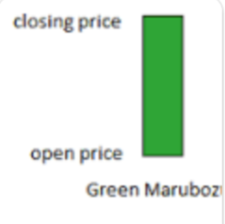


Inverted hammer

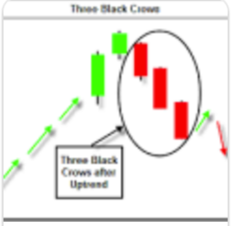


Morning star

Three white soldiers



Marubozu



Three black crows

3.6 Combination

The combination of these features are expected to increase the performance of price predictions. There will be 16 combinations used in this project to find out which one performs the best. The following are the list of combinations that will be used in this project:

1. plain dataset (open, high, low, close data),
2. candlestick pattern (CP),
3. exponential moving average (EM)
4. MPQA sentiment analyzer (MP)
5. VADER sentiment analyzer (VA)
6. hybrid sentiment analyzer (HY)
7. CP + EM
8. CP + MP
9. CP + VA
10. CP + HY
11. EM + MP
12. EM + VA
13. EM + HY
14. CP + EM + MP
15. CP + EM + VA
16. CP + EM + HY

4 Experiments and Results

4.1 Data

For sentiment prediction accuracy, the test data set is the movie reviews provided by nltk package in python. It is a set of sentiment-tagged reviews which is widely used to check sentiment accuracy. Different papers [2], [10] have used this corpus to measure accuracy of their sentiment analyzer.

Some preprocessing has to be done to the data since the data are not normalized. Firstly, all punctuations are removed while still making each word readable. Then, stop words are removed from the each review. Stop words are words that are very common in a piece of text. These words do not contain any sentiment, so removing the stop words will help to normalize the reviews. Next, all words are lowercased, to avoid multiple words of similar letters but different cases to appear at the same time. Lastly, all the words are lemmatized to standardize all form of words. For example, "initializes", "initializing" and "initialized" will be standardized to initialize. This word may or may not contain any sentiment, but a standardize form helps to gather similar words together.

For the learning-based sentiment analyzer, it needs more training data and vocabulary size. Additional data is added from imdb movie reviews. The dataset contains 50,000 data tagged with its respective sentiment polarity. 25,000 of imdb data is appended to 2,000 of the movie reviews for training. The model will be tested against the remaining 25,000 news to calculate the accuracy. The words in this dataset has been embedded to numerical values based on the popularity. For example, the word "the" is converted to 1, and "a" is converted to 2. The reason for this conversion is that to ease the process of transferring it to the machine learning model. Additionally, a function to convert from numbers its respective word is created for converting the 2,000 movie reviews into numerical values.

Another news dataset was parsed using BeautifulSoup4 (a python package) from NASDAQ website (NASDAQ.com). It contains news about Apple of for six months (October to April). It is paired with Apple stock for the same six months for testing purposes.

Additional data is collected from the year 2006 to 2016 containing news on everyday. There is 8% rows out of 2514 data having no news (null) in the dataset. All the news are from the New York times. Since the volume is larger, spreading over the period of ten years, this dataset will be used for training and validation.

4.2 Sentiment Prediction Accuracy

This study would like to test the two readily-available lexicon-based analyzer (SentiWordNet and MPQA), VADER, and hybrid sentiment analyzer on a corpora. This movie review corpora [11] will be used to determine the accuracy of both analyzer. This corpora contains 1,000 negative reviews and 1,000 positive reviews.

The precision is computed as follows:

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

The recall is computed as follows:

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

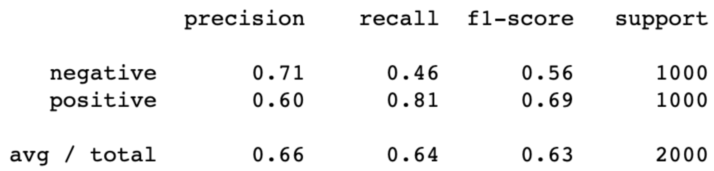
And F1-score uses precision and recall to calculate as follows:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

These indicators will show which analyzer performs the best when predicting sentiment of a text.

4.2.1 MPQA Result

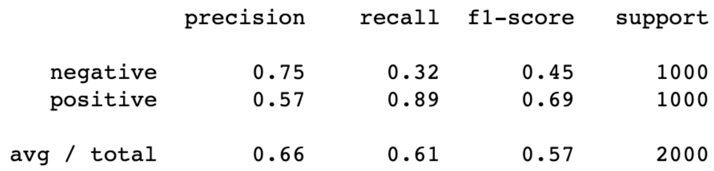
MPQA showed 66% precision and 64% recall on a movie review corpora. Using MPQA, the program was able to identify 60% of positive reviews and 71% of negative reviews correctly. The average f1-score for MPQA analyzer is 63%



```
             precision    recall  f1-score   support

   negative       0.71      0.46      0.56      1000
   positive       0.60      0.81      0.69      1000

avg / total       0.66      0.64      0.63      2000
```

4.2.2 SentiWordNet Result

SentiWordNet has similar precision to MPQA of 66% but falls short in the recall score of 61%. The average f1-score for SentiWordNet analyzer is 57% which is 6% lower than MPQA.

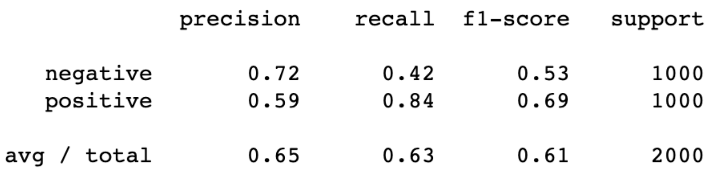


```
             precision    recall  f1-score   support

   negative       0.75      0.32      0.45      1000
   positive       0.57      0.89      0.69      1000

avg / total       0.66      0.61      0.57      2000
```

MPQA performs better than SentiWordNet in predicting sentiment for movie reviews. Therefore, MPQA will be used as the baseline for comparison with other techniques.

4.2.3 VADER Lexicon-based Sentiment Analysis

VADER uses the term booster referring to context shifters previously mentioned. This feature may be useful when calculating sentiment since it better understand the author's sentiment. VADER's performance is on par with MPQA with 65% precision, 63% recall and f1-score of 61%.

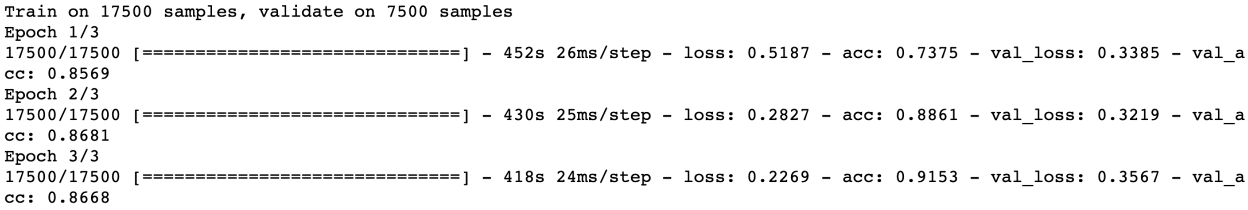


```
             precision    recall  f1-score   support

   negative       0.72      0.42      0.53      1000
   positive       0.59      0.84      0.69      1000

avg / total       0.65      0.63      0.61      2000
```

4.2.4 Learning-based Analyzer

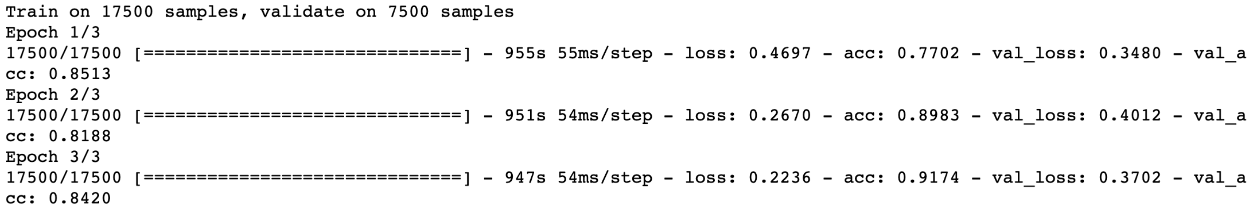
The learning-based analyzer will be first made before creating the hybrid sentiment analyzer. The hybrid sentiment analyzer will make use of MPQA and VADER to improve the accuracy. The following is the training process of all four learning-based analyzers.

Figure 3: 1 LSTM Layer



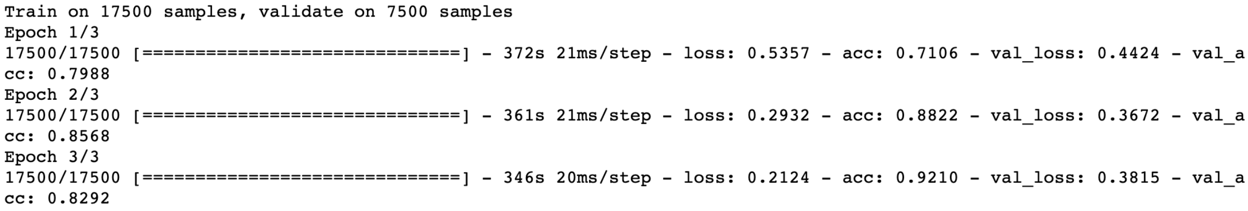
```
Train on 17500 samples, validate on 7500 samples
Epoch 1/3
17500/17500 [==============================] - 452s 26ms/step - loss: 0.5187 - acc: 0.7375 - val_loss: 0.3385 - val_a
cc: 0.8569
Epoch 2/3
17500/17500 [==============================] - 430s 25ms/step - loss: 0.2827 - acc: 0.8861 - val_loss: 0.3219 - val_a
cc: 0.8681
Epoch 3/3
17500/17500 [==============================] - 418s 24ms/step - loss: 0.2269 - acc: 0.9153 - val_loss: 0.3567 - val_a
cc: 0.8668
```

Figure 4: 2 LSTM Layers



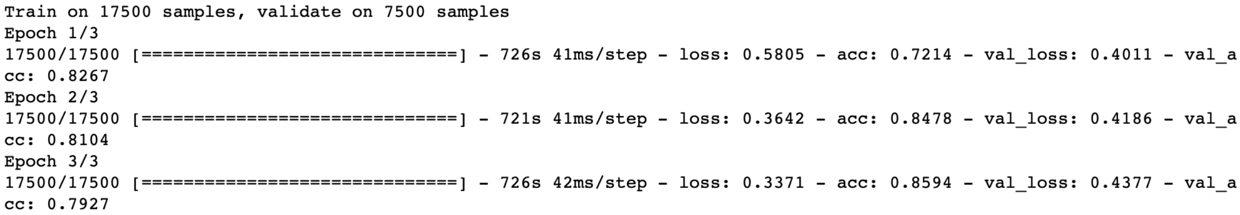
```
Train on 17500 samples, validate on 7500 samples
Epoch 1/3
17500/17500 [==============================] - 955s 55ms/step - loss: 0.4697 - acc: 0.7702 - val_loss: 0.3480 - val_a
cc: 0.8513
Epoch 2/3
17500/17500 [==============================] - 951s 54ms/step - loss: 0.2670 - acc: 0.8983 - val_loss: 0.4012 - val_a
cc: 0.8188
Epoch 3/3
17500/17500 [==============================] - 947s 54ms/step - loss: 0.2236 - acc: 0.9174 - val_loss: 0.3702 - val_a
cc: 0.8420
```

Figure 5: 1 GRU Layer



```
Train on 17500 samples, validate on 7500 samples
Epoch 1/3
17500/17500 [==============================] - 372s 21ms/step - loss: 0.5357 - acc: 0.7106 - val_loss: 0.4424 - val_a
cc: 0.7988
Epoch 2/3
17500/17500 [==============================] - 361s 21ms/step - loss: 0.2932 - acc: 0.8822 - val_loss: 0.3672 - val_a
cc: 0.8568
Epoch 3/3
17500/17500 [==============================] - 346s 20ms/step - loss: 0.2124 - acc: 0.9210 - val_loss: 0.3815 - val_a
cc: 0.8292
```

Figure 6: 2 GRU Layers



```
Train on 17500 samples, validate on 7500 samples
Epoch 1/3
17500/17500 [==============================] - 726s 41ms/step - loss: 0.5805 - acc: 0.7214 - val_loss: 0.4011 - val_a
cc: 0.8267
Epoch 2/3
17500/17500 [==============================] - 721s 41ms/step - loss: 0.3642 - acc: 0.8478 - val_loss: 0.4186 - val_a
cc: 0.8104
Epoch 3/3
17500/17500 [==============================] - 726s 42ms/step - loss: 0.3371 - acc: 0.8594 - val_loss: 0.4377 - val_a
cc: 0.7927
```

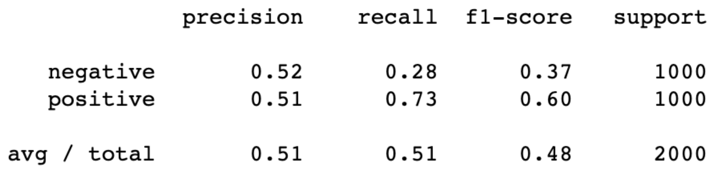
Figure 3 to 6 shows the run time of each technique and the accuracy in every epoch. The LSTM runs slower compared to GRU because more computations are done in each unit. The dataset used for training is from IMDB review dataset consisting of 25,000 news. The training data was then split into 70:30 for validation, 17,500 for training the model and 7,500 for validating the model. In the final epoch, we can see the final accuracy of the model when predicting the 7,500 test data, denoted by "val_acc". "acc" shows the accuracy of the model when calculating accuracy of the sentiment of the train dataset. However, there are cases of overfitting in technique 3 (1 GRU Layer) where the final model can predict the train dataset with 92% accuracy, but it dropped to 82% on the test dataset. After training and validating the loss and accuracy, a final testing will be done on the test dataset. The following table shows the accuracy of each technique on the test dataset.

Technique	Accuracy
1 LSTM Layer	0.84888
2 LSTM Layers	0.83800
1 GRU Layer	0.82632
2 GRU Layers	0.79292

Table 1: Accuracy on all learning-based sentiment analyzer techniques

Table 1 shows the accuracy of different techniques when performing sentiment analysis. GRU performs slightly worse than LSTM in both cases, but performs up to 20% faster compared to LSTM. Since we have discovered that overfitting is the case with GRU layer, it appears that LSTM is better for this project. LSTM will be paired with MPQA and VADER to further improve the accuracy.

Since the learning-based analyzerwas trained and tested on a different dataset, another testing need to be made. The test will be on the same dataset used on MPQA, SentiWordNet and VADER. Performing the test on the same dataset will standardize the performance of all analyzer. The following is the performance of 1 LSTM Layer on the movie review dataset.



```
             precision    recall  f1-score   support

   negative       0.52      0.28      0.37      1000
   positive       0.51      0.73      0.60      1000

avg / total       0.51      0.51      0.48      2000
```

The performance of the learning-based analyzer is very low with 51% precision, 51% recall and 48% f1-score. An investigation was carried out to find out why the performance is not as expected. Different variables were changed and measured to increase the accuracy, without training on the movie review dataset. The first variable adjusted was the dataset volume, doubling the volume from 25,000 to 50,000. However, the result precision, recall and f1-score became lower of around 40%. Then, the vocabulary size was increased to 100,000. The accuracy remained at 51%.

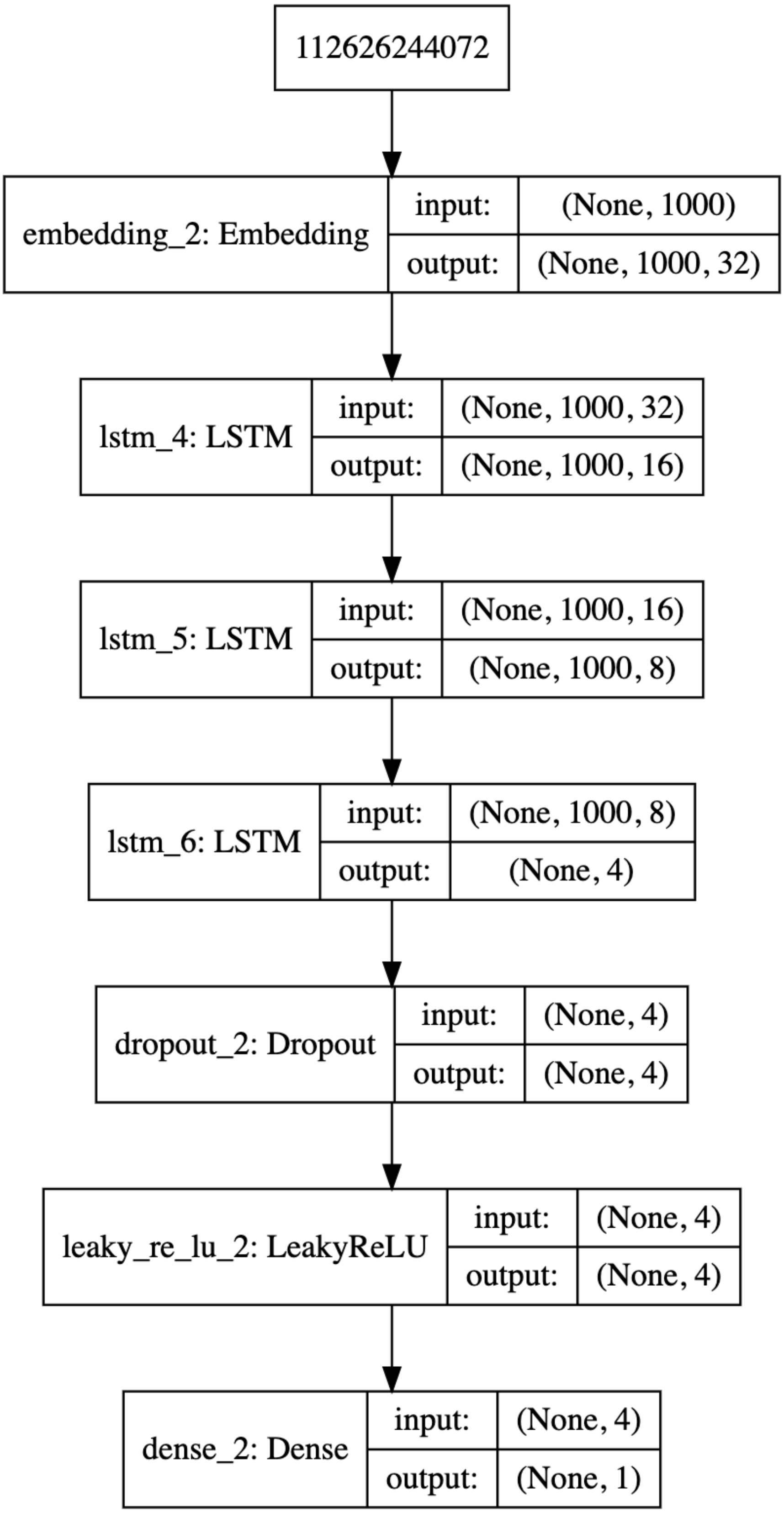
Since changing these variables does not help the accuracy at all, the dataset will be explored to find some clues about this abnormalities. From 2000 data, VADER was able to identify 1256 correctly and learning-based analyzer was able to identify 1018. Calculating their difference was found that from 744 that the VADER could not identify, 520 of them was identified correctly by the learning-based analyzer. Thus, the best possible performance of a hybrid analyzer is at most 89%.

Another reason that could be the cause of the low performance is that from 100,000 vocabulary size, some words are not identified for the test dataset. 37% of the vocabulary in the test dataset are not identified. Thus, lowering the accuracy.

Having more vocabulary size, the train dataset is used for testing and the test dataset is used for training. The model is trained on the 27,000 movie reviews dataset. 2 additional LSTM layer with different number of neurons are added Another layer is added: Leaky ReLU. Leaky ReLU was introduced to overcome the problem of vanishing gradient (Appendix VI).

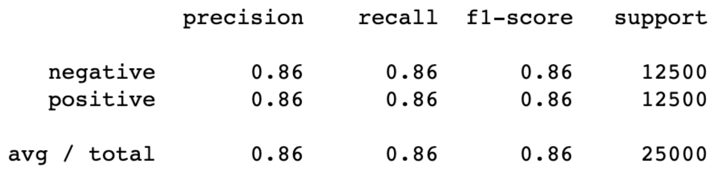
The accuracy increased to 76% on 25000 movie reviews test dataset.

The structure of the final LSTM model used is as follows:



4.2.5 Hybrid Sentiment Analyzer

After obtaining a satisfactory result from the learning-based analyzer, different related studies proposed that hybrid sentiment analyzer works better than plain learning-based analyzer. The test is performed and it achieved 86% precision, recall and f1-score. Similar results were obtained. This classifier will be used to predict the stock.



```
             precision    recall  f1-score   support

   negative       0.86      0.86      0.86     12500
   positive       0.86      0.86      0.86     12500

avg / total       0.86      0.86      0.86     25000
```

4.3 Stock Price Prediction

In time series forecasting, the best indicators of accuracy are mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE). Each of them are better compared to the other depending on the data being analyzed. The data has been scaled into a range of decimals from zero to one using min-max scaler. MAE and RMSE are better indicators for scaled data compared to MAPE because MAPE are scale-independent [12]. Between MAE and RMSE, there is another factor that determines the better fit to this project. MAE is better when there are outliers in the data, and RMSE is better when the possible outliers contain useful information. Since stock data are exact data and outliers are possible depending on many external factors, RMSE is a better fit for this project.

MAE formula is as follows:

$$MAE = \frac{\Sigma_{i=1}^{n} |y_i - x_i|}{n}$$

RMSE formula is as follows:

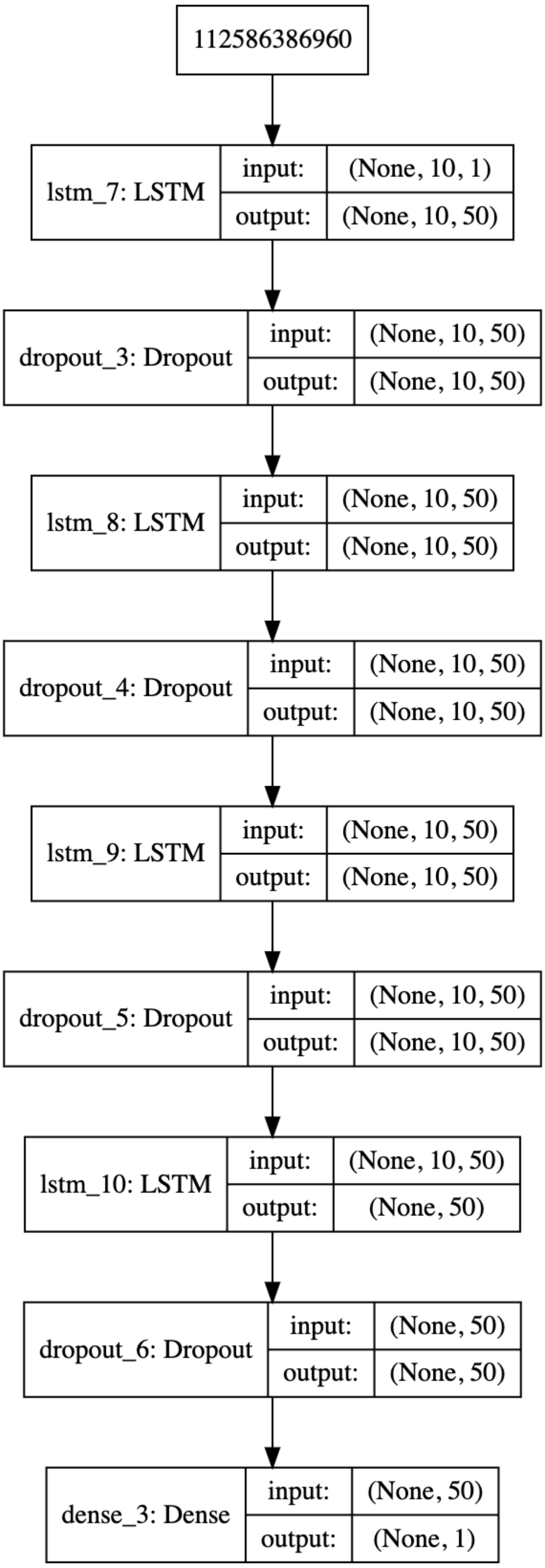
$$RMSE = \sqrt{\frac{1}{n} \Sigma_{i=1}^{n} \Big(\frac{d_i - f_i}{\sigma_i} \Big)^2}$$

A dataset containing stock price data from 2006 to 2016 was used to calculate the accuracy of the prediction. Different features will be padded to the dataset and passed into a deep neural network consisting of 4 LSTM layers with 50 units each layer. Dropout layer is added to prevent overfitting the model on the training data.

The training dataset contains 2514 data with each row representing one day data. The columns consist of the daily opening price, daily highest price, daily lowest price, daily closing price, adjusted closing price and the news for the day. Different sentiment analyzers are paired with this dataset for the prediction. The features that will be tested are:

1. CP: candlestick pattern,
2. EM: exponential moving average,
3. MP: sentiment using MPQA,
4. VA: sentiment using VADER, and
5. HY: sentiment using the hybrid sentiment analyzer

To predict the stock, another model is trained using LSTM layers. The following diagram shows the pipeline of the model.



Using 1750 data to train, the model will be predicting 750 data ahead. The results are as the following:

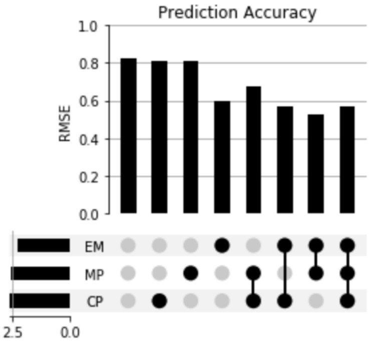


Figure 7: MPQA performance

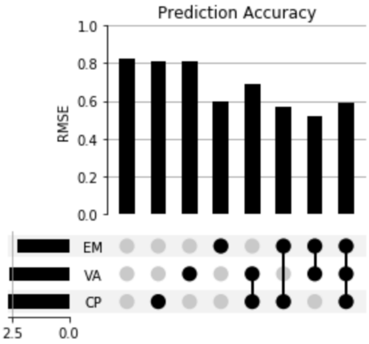


Figure 8: VADER performance

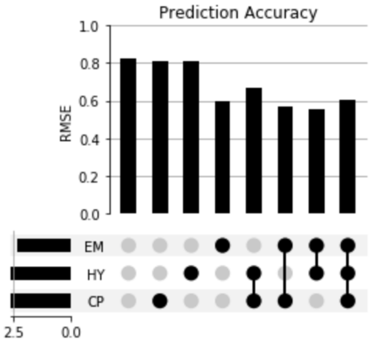
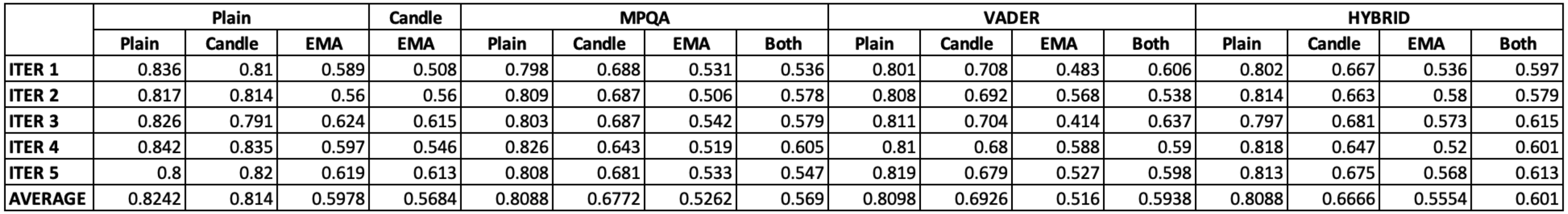
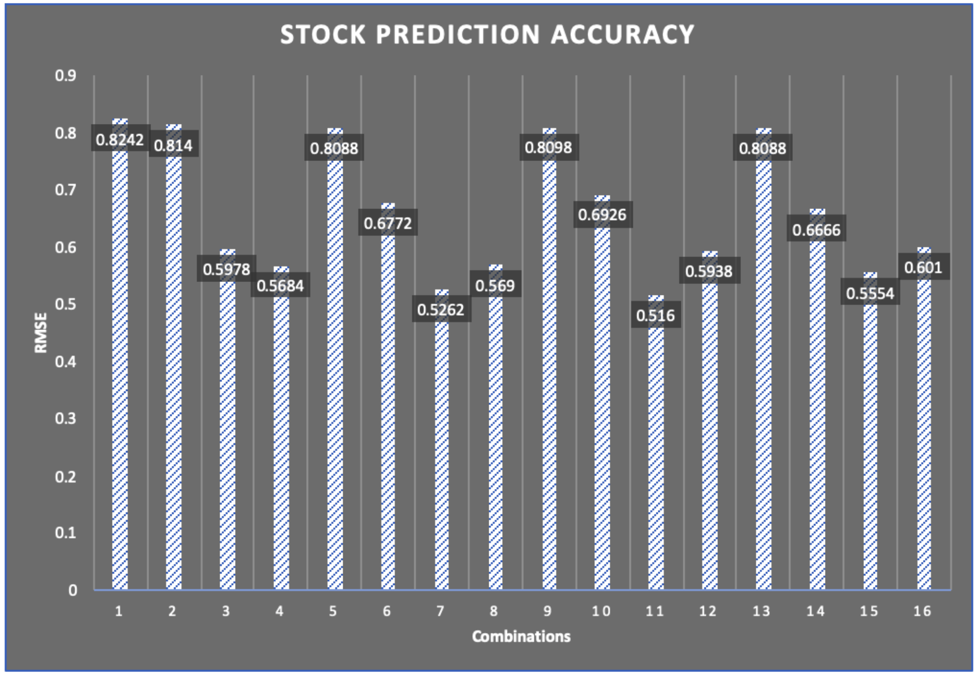


Figure 9: Hybrid performance

Figure 7 - 9 shows performances of the 3 sentiment analyzer being paired with other technical analyzer. From the figure above, the best performance is showed by using Exponential Moving Average and VADER sentiment with RMSE value of 0.516. The worst performance is to use only plain dataset to predict (open, high, low, close data) with RMSE value of 0.824.



	Plain			Candle	MPQA				VADER				HYBRID			
	Plain	Candle	EMA	EMA	Plain	Candle	EMA	Both	Plain	Candle	EMA	Both	Plain	Candle	EMA	Both
ITER 1	0.836	0.81	0.589	0.508	0.798	0.688	0.531	0.536	0.801	0.708	0.483	0.606	0.802	0.667	0.536	0.597
ITER 2	0.817	0.814	0.56	0.56	0.809	0.687	0.506	0.578	0.808	0.692	0.568	0.538	0.814	0.663	0.58	0.579
ITER 3	0.826	0.791	0.624	0.615	0.803	0.687	0.542	0.579	0.811	0.704	0.414	0.637	0.797	0.681	0.573	0.615
ITER 4	0.842	0.835	0.597	0.546	0.826	0.643	0.519	0.605	0.81	0.68	0.588	0.59	0.818	0.647	0.52	0.601
ITER 5	0.8	0.82	0.619	0.613	0.808	0.681	0.533	0.547	0.819	0.679	0.527	0.598	0.813	0.675	0.568	0.613
AVERAGE	0.8242	0.814	0.5978	0.5684	0.8088	0.6772	0.5262	0.569	0.8098	0.6926	0.516	0.5938	0.8088	0.6666	0.5554	0.601



The table and chart above are combinations of all the data acquired. The process was iterated five times to get the average of the prediction.

5 Conclusion

Comparing all the sentiment analyzers, hybrid sentiment analyzer has the best performance of 86%, and SentiWordNet has the worst performance of 57%. Performance has been improved by 20% from lexicon-based and sentence-level sentiment analyzer.

When looking at the performance in predicting the stock, all sentiment analyzer have an equal performance with 0.808 RMSE. EMA has the best performance among all the features used with 0.597 RMSE. The EMA are using 2-day previous data to predict the next day. When adding the news' sentiment to the model, the model is able to have a better accuracy in predicting the stock with average RMSE value of 0.532. Candlestick pattern did not impact the prediction by a lot individually.However, when paired with sentiment analyzer, the RMSE imrpoved by 0.15, which is a large improvement. The performance is bottlenecked when using all 3 features together: candlestick pattern, moving average and sentiment analyzer.

Adding candlestick into the 2-feature model (EMA and sentiment analyzer) reduced the performance of the model. This could imply that the candlestick pattern is not reliable when predicting the stock. Moreover, the candlestick pattern data is very sparse, meaning that it is filled with numerous "False" value and very few "True" value. The performance of the candlestick pattern recognition solely relies on the open-source program. This feature could be improved to further improve the predicting performance of the analyzer.

The EMA feature predicts really well. The main parameter for the EMA function is the numbers of days ahead is used to calculate the moving average. After several experiment, EMA of 2 days works best compared to other time frames. The 2-days EMA is more sensitive to price changes, thus better in predicting the next day. The higher the amount of days, the better it is to predict a bigger time frame. For example, 15-days EMA are normally used to see weekly trend.

Comparing the accuracy of all analyzers, hybrid performs the best. However when predicting the stock, it performs worse than its counterparts. This could imply multiple problems with this approach.

1. The accuracy is only reliable in predicting the movie reviews, but not financial news
2. The higher the accuracy, the more dependent the model is to the sentiment analyzer
3. The lower the accuracy, the more lenient the model is to the sentiment analyzer
4. News volume is too small

Every feature used in this study affect the stock prediction to a certain degree. Thus, no features are found misleading used in this study.

6 Discussion

Further investigation will be needed to check the features needed to predict. There is another technique to predict the stock, which is fundamental analysis. However, in predicting day-to-day stock prices change, fundamental analysis performs poorly since it only predicts the price, but not the time. For example, fundamental analysis is able to predict AAPL stock price to reach US$300, but does not know when it will reach it. Fundamental analysis can be used in a larger time frame, which possibly can be used to predict daily stock predictions.

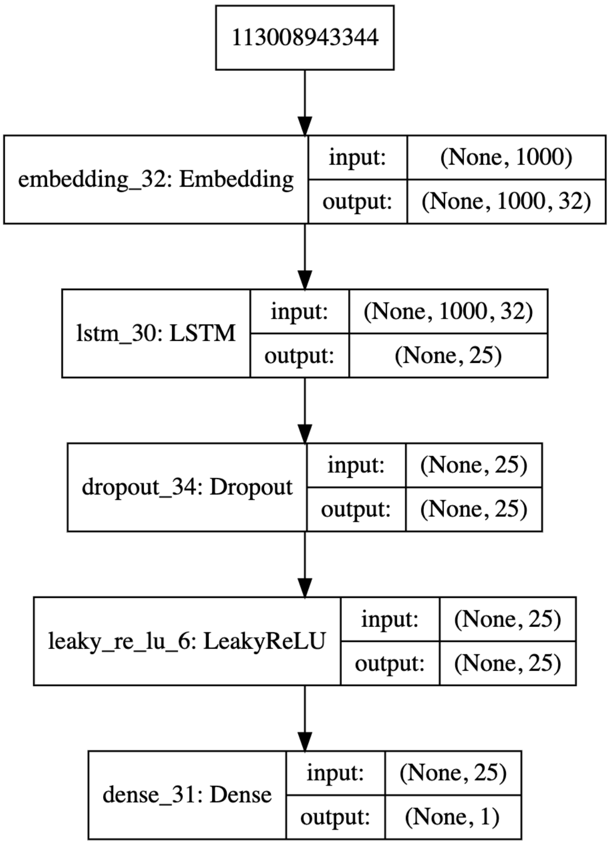
Another improvement can be added to the candlestick pattern. The candlestick pattern recognition was using an open-source package with hard-coded numbers. Ideally, training a model to recognize patterns that is followed by a reversal or continuation will work better than the inflexible hard-coded pattern recognition.

Aside from the adding new features, the data need some improvement. The news volume could be increased to increase the confidence level. Since the news volume is too little, the amount of news pushed to the model is also little. Thus, lower performance is reflected from the prediction model. 8% of 2514 days do not have any news attached to the data. This 8% could increase the performance of the prediction by a little.

Accuracy testing is also important. The sentiment analyzers were tested on movie reviews, not financial news. The testing on sentiment-tagged financial news will increase the credibility of the accuracy. Another natural language processing approach, known as aspect-based sentiment analysis, has a high accuracy when trained on its respective dataset. It might be trained understand the sentiment of financial terms such as surge and plunge. In the MPQA and SentiWordNet dictionaries, the word surge and plunge were not classified properly (in financial terms) which may result in less accurate prediction.

7 Appendices

APPENDIX I

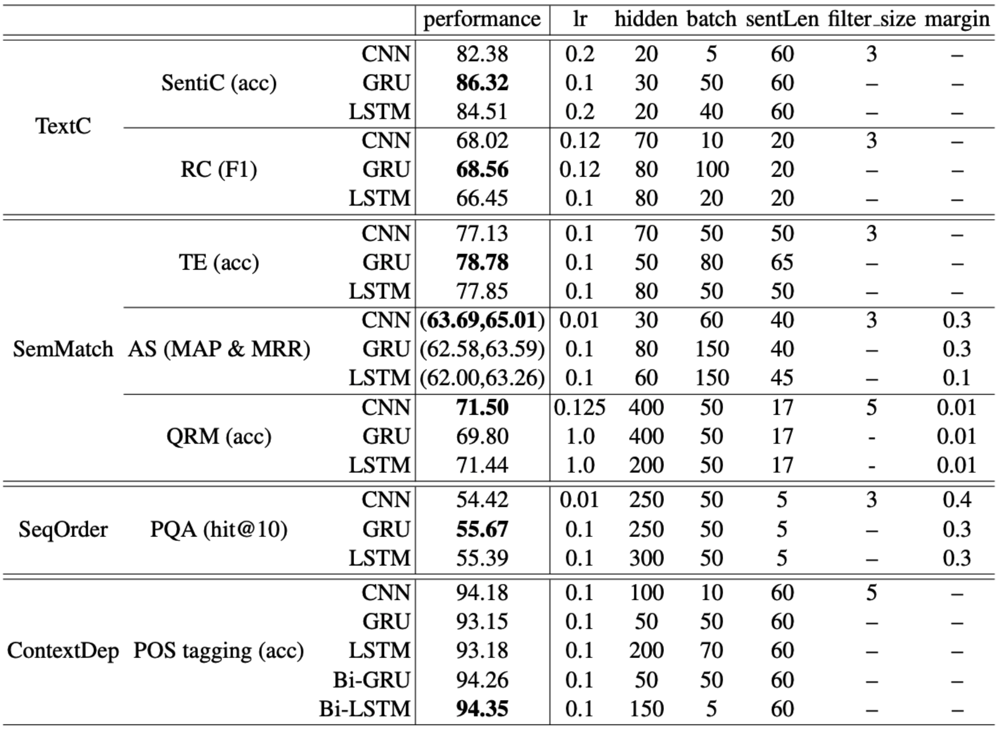


LSTM with LeakyReLU

APPENDIX II

NEGATE = {"n't","aint", "arent", "cannot", "cant", "couldnt", "darent", "didnt", "doesnt", "ain't", "aren't", "can't", "couldn't", "daren't", "didn't", "doesn't", "dont", "hadnt", "hasnt", "havent", "isnt", "mightnt", "mustnt", "neither", "don't", "hadn't", "hasn't", "haven't", "isn't", "mightn't", "mustn't", "neednt", "needn't", "never", "none", "nope", "nor", "not", "nothing", "nowhere", "oughtnt", "shant", "shouldnt", "uhuh", "wasnt", "werent", "oughtn't", "shan't", "shouldn't", "uh-uh", "wasn't", "weren't", "without", "wont", "wouldnt", "won't", "wouldn't", "rarely", "seldom", "despite"}

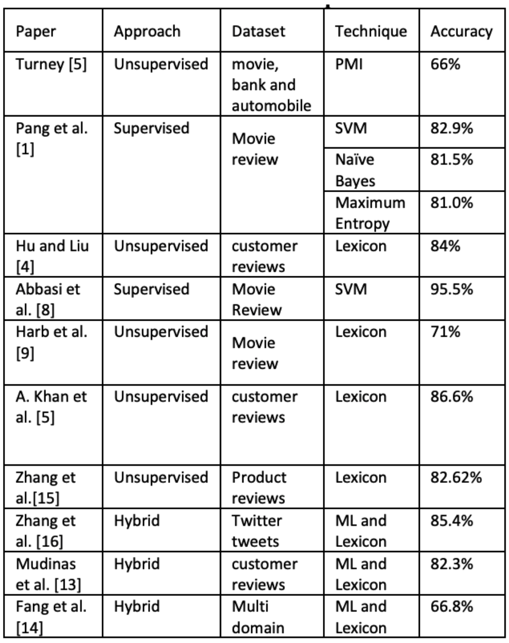
APPENDIX III



			performance	lr	hidden	batch	sentLen	filter_size	margin
TextC	SentiC (acc)	CNN	82.38	0.2	20	5	60	3	–
		GRU	**86.32**	0.1	30	50	60	–	–
		LSTM	84.51	0.2	20	40	60	–	–
	RC (F1)	CNN	68.02	0.12	70	10	20	3	–
		GRU	**68.56**	0.12	80	100	20	–	–
		LSTM	66.45	0.1	80	20	20	–	–
SemMatch	TE (acc)	CNN	77.13	0.1	70	50	50	3	–
		GRU	**78.78**	0.1	50	80	65	–	–
		LSTM	77.85	0.1	80	50	50	–	–
	AS (MAP & MRR)	CNN	**(63.69,65.01)**	0.01	30	60	40	3	0.3
		GRU	(62.58,63.59)	0.1	80	150	40	–	0.3
		LSTM	(62.00,63.26)	0.1	60	150	45	–	0.1
	QRM (acc)	CNN	**71.50**	0.125	400	50	17	5	0.01
		GRU	69.80	1.0	400	50	17	-	0.01
		LSTM	71.44	1.0	200	50	17	-	0.01
SeqOrder	PQA (hit@10)	CNN	54.42	0.01	250	50	5	3	0.4
		GRU	**55.67**	0.1	250	50	5	–	0.3
		LSTM	55.39	0.1	300	50	5	–	0.3
ContextDep	POS tagging (acc)	CNN	94.18	0.1	100	10	60	5	–
		GRU	93.15	0.1	50	50	60	–	–
		LSTM	93.18	0.1	200	70	60	–	–
		Bi-GRU	94.26	0.1	50	50	60	–	–
		Bi-LSTM	**94.35**	0.1	150	5	60	–	–

Comparison between performances of CNN vs. GRU vs. LSTM

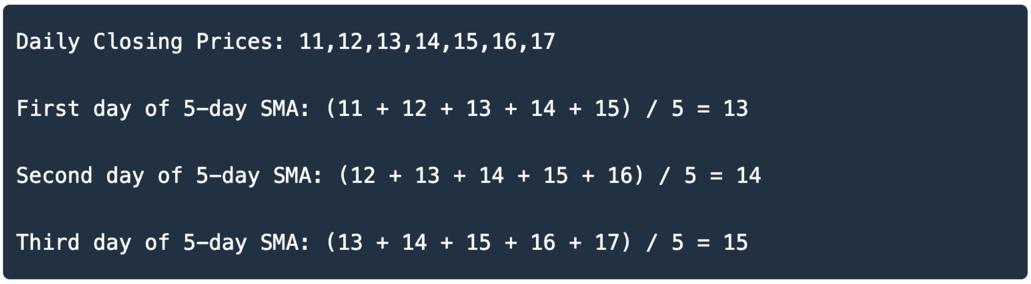
APPENDIX IV



Paper	Approach	Dataset	Technique	Accuracy
Turney [5]	Unsupervised	movie, bank and automobile	PMI	66%
Pang et al. [1]	Supervised	Movie review	SVM	82.9%
			Naïve Bayes	81.5%
			Maximum Entropy	81.0%
Hu and Liu [4]	Unsupervised	customer reviews	Lexicon	84%
Abbasi et al. [8]	Supervised	Movie Review	SVM	95.5%
Harb et al. [9]	Unsupervised	Movie review	Lexicon	71%
A. Khan et al. [5]	Unsupervised	customer reviews	Lexicon	86.6%
Zhang et al.[15]	Unsupervised	Product reviews	Lexicon	82.62%
Zhang et al. [16]	Hybrid	Twitter tweets	ML and Lexicon	85.4%
Mudinas et al. [13]	Hybrid	customer reviews	ML and Lexicon	82.3%
Fang et al. [14]	Hybrid	Multi domain	ML and Lexicon	66.8%

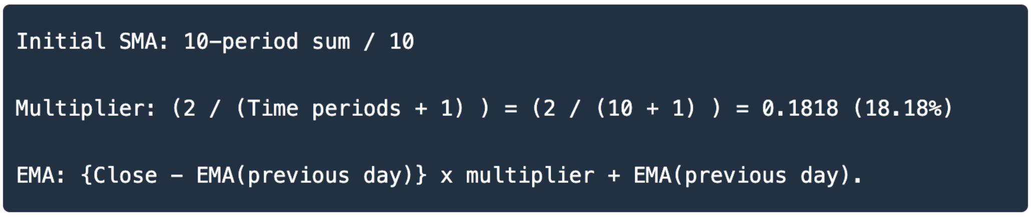
Comparison between performances Multiple Sentiment Analyzer

APPENDIX V



```
Daily Closing Prices: 11,12,13,14,15,16,17

First day of 5-day SMA: (11 + 12 + 13 + 14 + 15) / 5 = 13

Second day of 5-day SMA: (12 + 13 + 14 + 15 + 16) / 5 = 14

Third day of 5-day SMA: (13 + 14 + 15 + 16 + 17) / 5 = 15
```

Example of SMA [13]



```
Initial SMA: 10-period sum / 10

Multiplier: (2 / (Time periods + 1) ) = (2 / (10 + 1) ) = 0.1818 (18.18%)

EMA: {Close - EMA(previous day)} x multiplier + EMA(previous day).
```

Example of EMA[13]

References

[1] H. Thakkar and D. Patel, "Approaches for sentiment analysis on twitter: A state-of-art study," *arXiv preprint arXiv:1512.01043*, 2015.

[2] P. Haseena Rahmath and T. Ahmad, "Sentiment analysis techniques-a comparative study," 2014.

[3] A. Khan, B. Baharudin, and K. Khan, "Sentiment classification from online customer reviews using lexical contextual sentence structure," in *International Conference on Software Engineering and Computer Systems*, Springer, 2011, pp. 317–331.

[4] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, p. 12, 2008.

[5] H. Saif, M. Fernandez, Y. He, and H. Alani, "Senticircles for contextual and conceptual semantic sentiment analysis of twitter," *Lecture Notes in Computer Science The Semantic Web: Trends and Challenges*, pp. 83–98, 2014. DOI: 10.1007/978-3-319-07443-6_7.

[6] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text.," in *ICWSM*, E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, and A. H. Oh, Eds., The AAAI Press, 2014, ISBN: 978-1-57735-659-2. [Online]. Available: http://dblp.uni-trier.de/db/conf/icwsm/icwsm2014.html#HuttoG14.

[7] N. Ljubešić and D. Fišer, "A global analysis of emoji usage," in *Proceedings of the 10th Web as Corpus Workshop*, 2016, pp. 82–89.

[8] S. Sohn, S. Wu, and C. G. Chute, "Dependency parser-based negation detection in clinical narratives," *AMIA Summits on Translational Science Proceedings*, vol. 2012, p. 1, 2012.

[9] M. Enger, E. Velldal, and L. Øvrelid, "An open-source tool for negation detection: A maximum-margin approach," in *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*, 2017, pp. 64–69.

[10] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[11] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the ACL*, 2004.

[12] R. J. Hyndman, "Measuring forecast accuracy,"

[13] stockcharts. (2019). Moving average, [Online]. Available: https://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_averages (visited on 04/10/2019).

The Hong Kong Polytechnic University

Department of Computing

COMP4913 Capstone Project

Name: Jacky Angara (Student No: 15101317d)

Programme Code: 61431-FCS

Project Title: Financial News Analytics with Hybrid Sentiment Analysis

Supervisor: Prof. Chan Chun Chung Keith

Except where reference is made in the text of this Capstone Project Report, I declare that this report contains no material published elsewhere or extracted in whole or in part from any works or assignments presented by me or any other parties for another subject. In addition, it has not been submitted for the award of any other degree or diploma in any other tertiary institution.

No other person's work has been used without due acknowledgement in the main text of the Report.

I fully understand that any discrepancy from the above statements will constitute a case of plagiarism and be subject to severe academic penalties that may lead to deregistration from the programme. The department reserves the right to check the paper and electronic submissions of the Report via various mechanisms, such as Turnitin.

Signature: Jacky Angara, 15101317D

Date: 16 April 2019