Multi-Window Stock Trend Prediction with News and Tweets

Lubingzhi Guo

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

10th December 2021

Abstract

In order to maximize profits and lower the risk of investment, the stock market prediction has been extensively studied. Many studies have shown that the stock market is somewhat predictable, and social media content, including news and tweets, are related to the stock market. And the progress of natural language processing has made it feasible to analyze financial texts such as news and tweets. This project aims to construct a model to predict stock trends built on the Hybrid Attention Network. Furthermore, we develop a new model that incorporates dynamic contextualized embeddings and multi-window structure. This proposed model has shown to be more effective than the compared models in some experiments.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

<Please note that you are under no obligation to sign this declaration, but doing so would help future students.>

Name: Lubingzhi Guo Signature: Lubingzhi Guo.

Acknowledgements

I would like to express my gratitude to Dr. John Williamson for his support and guidance throughout the project. Also, I would like to express my gratitude to my family for their support during this time.

Contents

Chapter 1 Introduction

1.1 Motivation

The stock market prediction has been widely researched to maximize profit, but it is considered as a challenging task due to its unstable nature and volatile behavior (Adam et al., 2016). Various studies have indicated that the stock market is somewhat predictable, along with historical price, the advancement of natural language processing has facilitated studies on financial texts such as news and tweets (Hu et al., 2018; Xu & Cohen, 2018).

Sentiment analysis, the method of extracting sentiments about a topic from a text, has been shown to play a significant role in predicting future market movements (Sohangir et al., 2018). Transformer-based models such as BERT (Devlin et al., 2019) and Roberta (Liu et al., 2019) are able to learn embeddings from unlabeled text inputs and have achieved massive success in assisting different downstream applications. FinBERT was the first application of BERT in financial texts sentiment analysis, and Araci (2019) also suggested employing it on financial news in conjunction with the stock return data to broaden their work. These superior models make the exploration of integrating contextualized embeddings with deep learning models into stock trend prediction tasks possible.

1.2 Purpose

The overall purpose of this project is to develop a model based on contextual embeddings of news and tweets to assist in stock movement prediction. The model would predict the trend of the following day by analyzing the news and tweets released in previous days.

The overall aims of this project are:

1. Reproduce the proposed hybrid attention network structure by Hu et al. (2018).
2. Instead of word embeddings, leverage the contextualized embeddings.
3. Create new models and perform experiments to evaluate effectiveness.
4. Analyze the experiment results and provide critical discussions.

1.3 Report structure

There are five chapters in this report. Chapter 1 discusses the motivation and goals of the project. Chapter 2 discusses the background survey, problem analysis and statement, as well as the requirements of the project. Chapter 3 discusses the general design of the project and the detailed implementation of critical parts. Chapter 4 discusses the testing process in-depth and summaries of evaluation results. Chapter 5 is the conclusion, including achievements, limitations, and future work direction.

Chapter 2 Analysis

2.1 Background Survey

2.1.1 Stock Trend Prediction

In order to comprehend market trends better, many academics in the domains of finance and computer science have worked on accurate prediction of stock movements, which is crucial to stock investment, e.g. lowering the risk and increasing the return. According to the nature of the data and the number of sources employed, the various approaches can be categorized into three types: technical, fundamental, and combined analysis (Nti et al., 2020).

The technical analysis relies on the historical prices and volumes, as well as other technical indicators. Previous studies attempted to integrate machine learning algorithms to predict the profit and loss (Nayak et al., 2015), as well as combine with the trading strategy based on flag pattern to identify the underlying market trend (Cervelló-Royo et al., 2015). In recent years, increasing research has focused on using deep neural networks to model time-series better and capture long-term dependencies. Zhang et al. (2017) proposed the SFM recurrent network to catch the latent trading patterns from historical market data to make long-term and short-term predictions.

However, quantified data cannot adequately depict a firm's financial state; qualitative information included in textual resources such as news and social media supplements the information evaluated by investors and improves model prediction (Li et al., 2015). Hence, news and social media serve as two of the most important data sources of fundamental analysis, and various efforts have been made to mine news or social media data to analyze the semantics and sentiment to improve market trend prediction. Ding et al. (2016) proposed a joint model that includes outside knowledge graphs in the goal function to learn event embeddings from news in order to improve event-driven model prediction. Sul et al. (2017) examined the relationship between sentiment in tweets and stock returns on future stock movements. Similarly, Oliveira et al. (2017) found that microblog sentiment was especially effective for predicting S&P 500 index returns, portfolios of companies in high technology, energy, and telecommunications fields.

Another major approach for market trend prediction is the combined analysis. Xu & Cohen (2018) jointly exploited tweets and historical prices on stock movement prediction, and the model with two market information inputs outperformed the model with single-source input. Additionally, they also found that both tweets and historical prices improve the stock movement prediction, whereas the fundamental analysis model gained much better results than the technical analysis model and competitive results with the assembled model.

Deep learning frameworks such as LSTM and CNN are the most representative models widely used in this field. Rather et al. (2015) developed a hybrid model that generates predictions by combining RNN and two linear models. Besides, Ding et al. (2015) presented a deep convolutional neural network to model the joint effect of long-term and short-term events for event-driven stock market prediction. The attention mechanism has attracted considerable interest as academics have gotten a better grasp of how it improves the performance of processing information (Bahdanau et al., 2016). Therefore, more studies on stock prediction employ this mechanism. For example, Hybrid Attention Network (HAN) was created by Hu et al. (2018) that utilizes recent relevant news as input and three attention layers to learn comprehensive information of stock trends.

2.1.2 Contextualized Embeddings

In NLP, the first process usually encodes words or sentences into fixed-length embeddings before applying the stock trend prediction models.

While Latent Semantic Analysis is a significant count-based technique (Deerwester et al., 1990), the fact that each unique word requires one dimension causes difficulty with excessively high-dimensional vectors. Therefore, much research effort has been spent on lowering the dimension of vectors.

Word2Vec and GloVe (Pennington et al., 2014) are the most frequently used deep learning algorithms for word embedding; for instance, Hu et al. (2018) extracted word embeddings from a pre-trained Word2Vec layer and then averaged the vectors of all words to create a news embedding. However, these algorithms often ignore contextual information and assume that the word embedding stays static over sentences. Although context might alter the meaning of a word, once trained, it generates the same embedding for that word (Chen, 2021).

In recent years, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) have shown the ability to generate contextualized word embeddings. The model takes the whole sentence as input rather than a single word and then produces word embeddings by evaluating it in combination with other words in the sentence. BERT has deeper network architecture ergo includes more parameters than ELMo and consequently better representation capabilities. Furthermore, fine-tuned BERT can be integrated into downstream applications to provide outstanding results when augmented by domain-specific or task-specific data. Araci (2019) showed that the fine-tuned model on the financial domain outperforms general models.

2.2 Problem Analysis

Hence, we propose applying the contextualized embeddings with domain-specific information to complement the existing HAN model structure (Hu et al., 2018) and predict stock trends based on news and tweets data in collaboration with BiGRU and attention mechanisms. Since the fundamental analysis has a competitive result with the model with both qualitative and quantified inputs (Xu & Cohen, 2018), we will focus on the fundamental analysis of financial text data.

Yet there are several issues to be concerned about:

1) The correlation between the stock market and expressed financial information. Lin et al. (2016) and Strauß et al. (2018) observed that the volume of news and tweets is positively associated with stock trading strength, hence, revealed the dynamic interdependency between stock

trend and financial news and tweets. Stock price movements may affect the amount and content of news and tweets; therefore, they are not always the reliable indicator of the overall stock market trend.

2) Experimental bias in various companies. The stock market seems to have a more significant response for assets with more social media attention than those that receive less (Lin et al., 2016). Thus, the trend of large-capitalization stock may be better explained by social media interaction.
3) Experimental bias in the dataset. We intend to train a single model for all stocks in our dataset. However, further analysis of the dataset discloses that it is difficult to obtain sufficient data for a diverse set of stocks; in other words, the textual data of various stocks is imbalanced, e.g. Tesla is expected to have the most news and tweets. Thus, we suspect that the model trained on the dataset will have some form of experimental bias towards stock with most articles.

2.3 Problem Statement

In this part, we first define the problem as the multi-window stock trend prediction. The inputs will be news or tweets, because they represent the views of two distinct types of participants (individuals and institutions), each of whom expresses their sentiments differently (Gupta & Chen, 2020). Therefore, examining the impact of various textual sources may be beneficial for fully comprehending the stock trends.

We regard the stock price movement as the sentiment of the stock market, and hence define the stock trend prediction issue as sentiment analysis. The sentiment is considered as the $Growth$:

$$Growth(t) = MinMaxScaler(\frac{Price(t+1)-Price(t)}{Price(t)}) \quad (1)$$

t denotes a given date and s denotes a given stock. We can classify the $Growth$ into three categories: DOWN (significant decrease), PRESERVE (steady fluctuation) and UP (significant increase). In addition, we observe that it is necessary to scale the $Growth$ feature to a specific range rather than simply employing the raw feature, since the price volume differs among different companies, and thus, the information contained in the price change varies as well. We can apply min-max normalization to reduce the influence of the diverse company scales, and the scaled $Growth$ can better describe the impact of the financial texts on the overall market.

Suppose that we have a list of windows $W = [w_1, w_2, \dots, w_d]$, where d represents the number of windows. The objective is to predict the class of $Growth$ using text sequences from all windows, e.g. time period $t - w$ to $t - 1$, $(w \in W)$, the financial texts can be either news or tweets, denoted as $S_i = [s_{t-w}, s_{t-w+1}, \dots, s_{t-1}]$, $(i \in d)$. Each text sequence s_k includes all the relevant texts released in the same date, denoted as $s_k = [n_{k,1}, n_{k,2}, \dots, n_{k,L}]$, $(k \in [t - w, t - 1])$, where L denotes the number of texts and varies across dates. Note that the single-window model is inspired by the study of Hu et al. (2018).

The text embeddings are created by first transforming each sentence into a fixed-length embedding and then averaging these embeddings over all sentences. We denote the sentence embedding as $emb_r, (r \in m)$, so the text embedding can be written as follows:

$$Emb_{k,l} = Mean([emb_1, emb_2, \dots, emb_m]),\ (l \in L) \qquad (2)$$

where m denotes the number of sentences.

2.4 Research Objective

The objectives of this project can be summarized as follows:

- Objective 1: Contextualized Embedding Generation
    - ◊ Fine-tune the base RoBERTa (Liu et al., 2019) model with the financial domain data.
    - ◊ Extract embeddings from the base and fine-tuned RoBERTa
    - ◊ Evaluate the performance of the fine-tuned contextualized encoder
- Objective 2: Reproduction of the Hybrid Attention Network
    - ◊ Recreation of the HAN (Hu et al., 2018) model
    - ◊ Evaluate the results from comparisons of the model to its variations
- Objective 3: Creation of the Multi-window model
    - ◊ Generate a multi-window model which can handle the un-fixed number and un-fixed length windows, as well as different textual source windows, e.g. news and tweets
    - ◊ Create baseline models
    - ◊ Evaluate the new model performance metrics against baseline models

Chapter 3 Design & Implementation

3.1 Framework

The entire project is implemented in python, mainly based on the PyTorch, Pandas and NumPy packages.

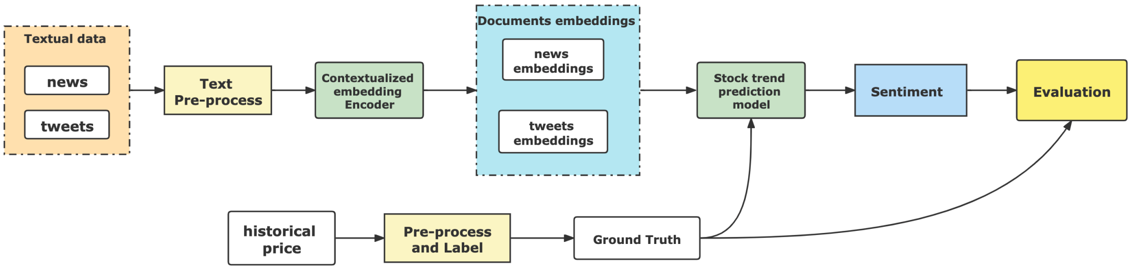


Figure 1: Overall logic of the project.

Figure 1 depicts the overall data flow design of the project. Firstly, the pre-processed textual data will be fed into the contextualized embedding encoder to generate document embeddings. And the historical price data will be processed and labelled to create the ground truth. The more in-depth pre-process is demonstrated in 2.3. To train the stock prediction models, employing supervised learning on the contextual embeddings of texts. As mentioned in 2.4, the model will be four types: 1) baseline of dummy model 2) baseline of GRU model 3) BiGRU and attention model with single window 4) BiGRU and attention model with multiple windows. Finally, the evaluation is conducted using the model output and the ground truth.

3.2 Data Preprocess

3.2.1 Data Collection

The text dataset that we use is a historical news archive of the US equities collected from the investing.com website[1], and financial tweets about the top companies[2]. Hence, in order to be consistent with tweets, we only utilize news and tweets of 5 leading stocks (AAPL, GOOGL, AMZN, TSLA, MSFT) between 2015 and 2020. Each entry in the news dataset has a *release_date* indicating when the news was posted, a *ticker* showing the stock associated with the news, and the *content* and *headline*. Similarly, the tweets dataset includes columns for the *post date*, *ticker*, and *body*, as well as the number of *comments*, *retweets*, and *likes*.

[1] https://www.kaggle.com/gennadiyr/us-equities-news-data

[2] https://www.kaggle.com/omermetinn/tweets-about-the-top-companies-from-2015-to-2020

To perform supervised learning, we should provide the model with ground truth from which to optimize; hence, we use the historical price data collected from Yahoo Finance[3] to label our data and represent the stock sentiment, which is the time series of price and trading volume from 2015 to 2020 on a daily basis.

3.2.2 Preprocess

Pre-processing of news is done using the NNSplit package[4] to accomplish sentence segmentation since punctuation is not included in the raw text. Using contextualized embedding models like BERT, RoBERTa, we don't need to do any text cleaning on news data, such as stop-word removal, stemming, etc., since all of that information will be utilized.

We remove weblinks, the "#" hashtag, the"@" identifier, the "$" ticker symbol and other special characters from the raw tweets dataset before retaining the tweets with the top 35% of tokens, i.e. to ensure that each tweet has a reasonable length and to exclude tweets with fewer words. Then, for each tweet, add up the total number of comments, reposts, and likes as its social attention score, and then filter out the tweets with a score of less than 10 to delete spam and less concerned information.

After this procedure, the total amount of news items is 44,125, the total amount of tweets is 113,032, and AAPL has the most news with 20,231, while TSLA has the most tweets with 77,303.

3.2.3 Data Splitting

In our experiment, we select a period (from 01/01/2015 to 31/12/2018) as our training data, a short period (from 01/01/2019 to 30/06/2019) as our test set and another short period (from 01/07/2019 to 31/12/2019) as the validation set.

3.2.4 Data Labelling

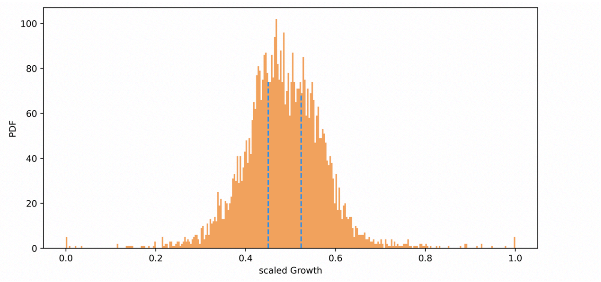


Figure 2: Distribution of the scaled Growth among the training dataset.

As shown in Figure 2, the logic of our labelling strategy is to select two thresholds to evenly divide the scaled Growth of the training set into three categories. The scaled Growth is calculated using Equation (1). We utilize the MinMaxScaler of the sklearn packages to fit and transform the training data. Then, we apply the trained scaler to transform the test and validation data to

[3] https://uk.finance.yahoo.com

[4] https://bminixhofer.github.io/nnsplit/

prevent leaking future information to the model. In the same vein, we use the lower and upper bound obtained in training data to bin the whole Growth to specify the label of stock sentiment. i.e. DOWN (Growth<0.45) labelled as 0, PRESERVE (0.45<=Growth<0.523) labelled as 1, and UP (Growth>=0.523) labelled as 2. Regardless of whether or not there is relevant news or tweets released, we are supposed to label the stock sentiment of this date.

3.3 Contextualized Embedding Encoder

3.3.1 RoBERTa

RoBERTa is a replication study of BERT pretraining that investigates the impact of various critical components of the training process (Liu et al., 2019). Unlike Devlin et al. (2019), they found that removing the next sentence prediction objective resulted in a competitive result or even slightly enhanced the performance of downstream applications. Additionally, more data and longer sequences can be used to train it.

3.3.2 Fine-tune RoBERTa

The RoBERTa model is trained on data of the general domain, and it is usually preferred to fine-tune it on the target domain data for a specific downstream task. For the text classification task, Sun et al. (2019) presented three methods to fine-tune BERT. Due to the problem definition, we are unable to assign a label to each piece of news, which prevents us from performing the supervised multi-task fine-tuning. Therefore, we would further pre-train the pre-trained RoBERTa model on the training data by unsupervised masked language model. This contextualized embedding encoder will only be fine-tuned on the news dataset to simplify the evaluation and reduce the complexity.

Given the conclusion that individual sentence input hampers the performance on downstream tasks (Liu et al., 2019), and the fact that it exceeds the GPU memory limit, we tried two types of inputs. Either of the following two ways may be used:

1) concatenate the segmented sentences from each news item until the predefined maximum length is reached.

2) divide news into fixed-length chunks with 20-word overlaps to avoid omitting the contextual information.

We use the pre-trained base RoBERTa model from Hugging Face[5] as the base model in our implementation, configured with a hidden size of 768, 12 self-attention heads, and 12 hidden layers. After tokenizing the raw text data, we randomly mask 15% of the tokens with 3 [MASK], excluding special tokens, i.e. 0 [PAD], 1 [CLS], or 2 [SEP]. The random masking program runs once at the start, and the processed data is kept for training. Then, we pre-train the model to update the parameters with our domain-specific data, with a maximum text length of 514, a learning rate of 1e-4, and a batch size of 8 (due to the limited memory of GPU).

[5] https://huggingface.co/roberta-base

3.3.3 Contextualized Embeddings

We may extract sentence embeddings from either the base RoBERTa model or the fine-tuned model; throughout this project, the term "sentence" refers to the preprocessed text chunk used as the fine-tuning input rather than an actual semantic sentence. We tokenize the sentences using the RoBERTa Tokenizer and then feed them into the RoBERTa model, from which the first vector of the returned hidden state is used as the sentence embedding, i.e. the embedding of [CLS] token. We produce document embeddings by averaging the embeddings of all the sentences in the same text, as demonstrated in Equation (2).

3.4 Stock Trend Prediction Model

3.4.1 Custom Dataset

Before training the prediction model, we override the pytorch Dataset to generate the dataset dynamically. And with the development of the project, we integrate the implementation of the single-window and multi-window.

The custom dataset is intended to address two issues:

1) Dataset generation

Create the dataset based on the input parameter $windows$; the parameter might be a single integer w or a list of values W. For each date t, each window w, collect texts sequences S_i from $t - w$ to $t - 1$ and store samples and their corresponding labels.

2) Variable-length sequences

Since the quantity of news or tweets posted on a given day varies, the vector containing the text embeddings for each window will be variable in length. We tackle this problem by padding each date vector to a specified maximum length; it may be the maximum or 90% of the length of texts in the training dataset for a particular date. And store the length vector representing the number of texts of each date, along with their corresponding samples and labels.

To improve the execution efficiency and limit the use of memory, the dataset generation happens in the $\_\_init\_\_$ function, and the process of handling variable-length sequences is in the $\_\_getitem\_\_$ function.

3.4.2 Baselines

Dummy Model

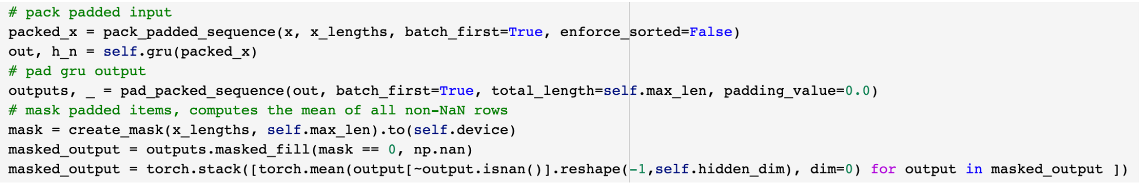
We apply the DummyClassifier from sklearn[6] package as one of our baseline models. Considering it works independently of the input samples, there is no need to mask the padded input data.

GRU Model

[6] https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

Another baseline is a simple GRU model connected with an MLP model to output the final classification result. This baseline enables us to examine some components in the more complex model (described in 3.4.3) more thoroughly.

The corresponding code snippet is as follows:



```
# pack padded input
packed_x = pack_padded_sequence(x, x_lengths, batch_first=True, enforce_sorted=False)
out, h_n = self.gru(packed_x)
# pad gru output
outputs, _ = pad_packed_sequence(out, batch_first=True, total_length=self.max_len, padding_value=0.0)
# mask padded items, computes the mean of all non-NaN rows
mask = create_mask(x_lengths, self.max_len).to(self.device)
masked_output = outputs.masked_fill(mask == 0, np.nan)
masked_output = torch.stack([torch.mean(output[~output.isnan()].reshape(-1,self.hidden_dim), dim=0) for output in masked_output ])
```

Figure 3: A code snippet of GRU baseline model.

As shown in Figure 4, the baseline data generation procedure is slightly different in terms of data structures; in this scenario, we stack all text embeddings from the same window. After obtaining the GRU model output, we should mask the padded sequences before feeding them into MLP. To construct the masked output, we average all non-padding rows.

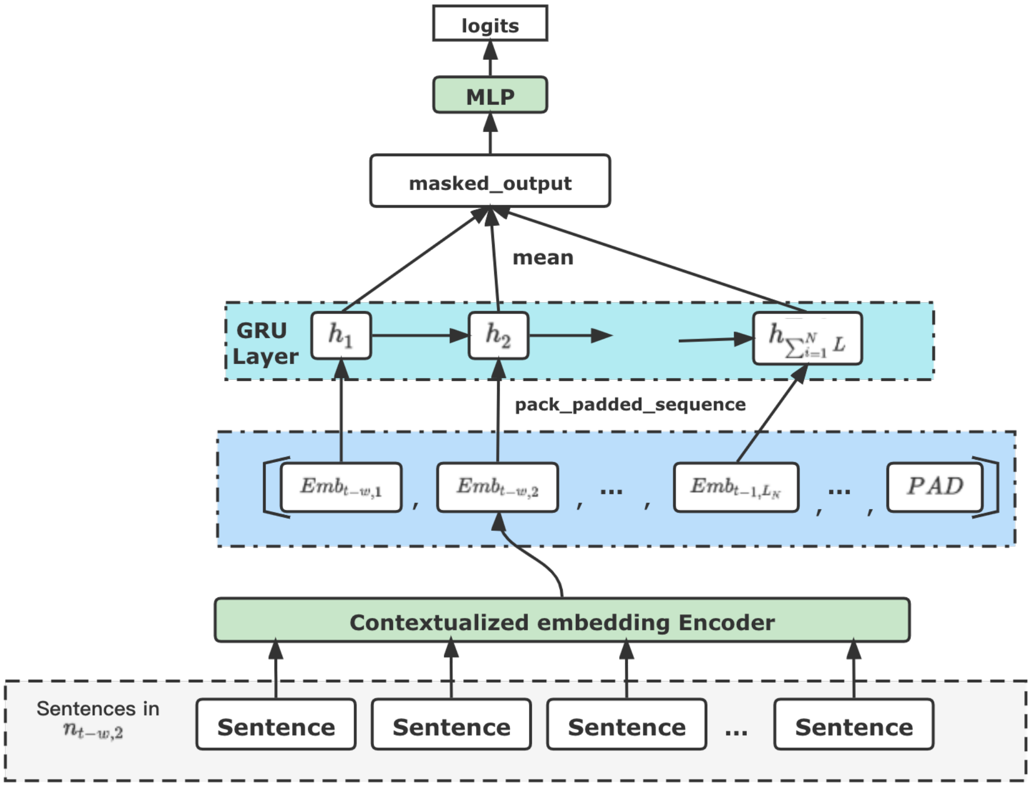


Figure 4: The overall framework of GRU baseline model.

3.4.3 BiGRU + Attention Model

Figure 5 illustrates the overall structure of the Multi-window BiGRU and attention model. With a padded text embedding sequence as input, a masked attention layer assigns an attention score to each text vector in the same date, followed by a masked SoftMax layer that dynamically adjusts the weight for padding and non-padding items, and then a weighted mean vector is leveraged to represent the date. Following that, a bi-directional GRU is employed to encode these dates vectors. Next, a temporal attention score was assigned to each date;

similarly, the weighted mean is calculated as the output of the context information of a specific window. All of the outputs are concatenated and fed into MLP to predict the stock trend.

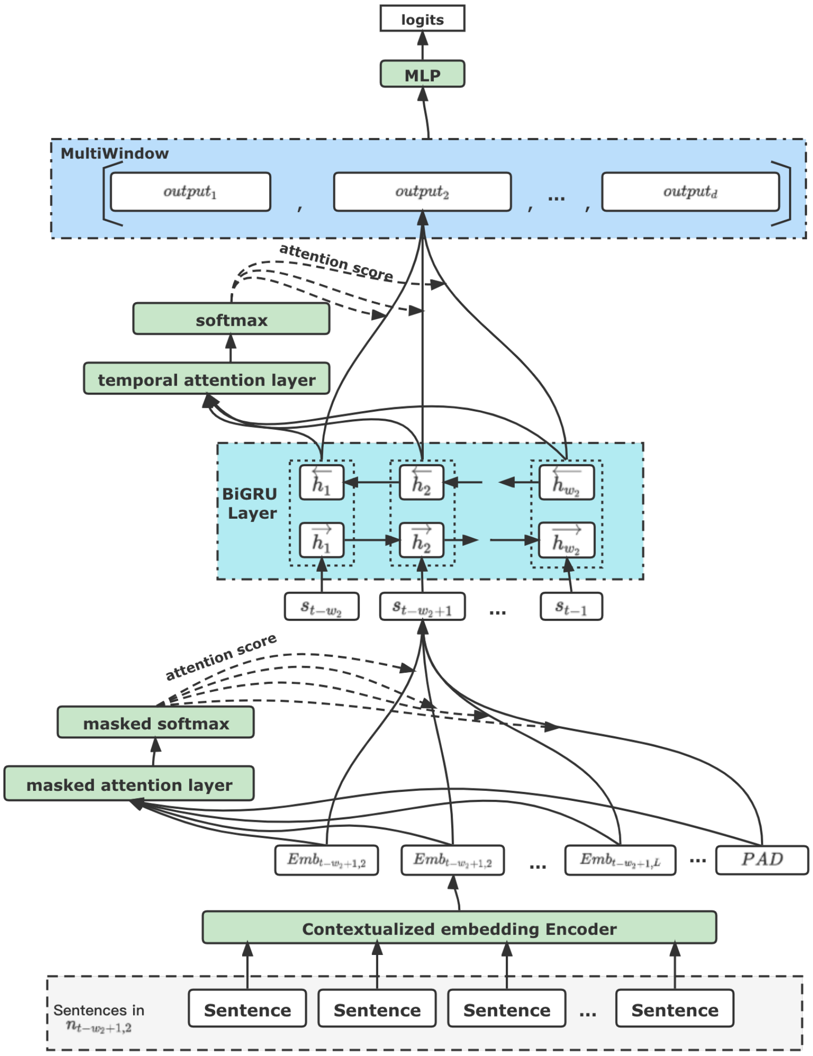
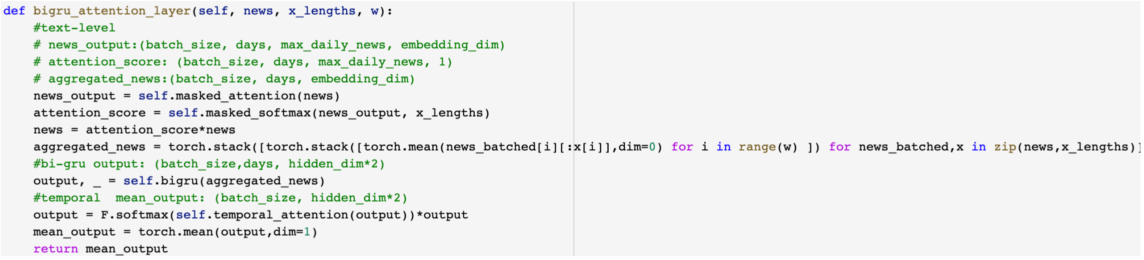


Figure 5: The overall framework of multi-window trend prediction model

The detailed implementation of the BiGRU attention layer is presented in the figure 6.



```
def bigru_attention_layer(self, news, x_lengths, w):
    #text-level
    # news_output:(batch_size, days, max_daily_news, embedding_dim)
    # attention_score: (batch_size, days, max_daily_news, 1)
    # aggregated_news:(batch_size, days, embedding_dim)
    news_output = self.masked_attention(news)
    attention_score = self.masked_softmax(news_output, x_lengths)
    news = attention_score*news
    aggregated_news = torch.stack([torch.stack([torch.mean(news_batched[i][:x[i]],dim=0) for i in range(w) ]) for news_batched,x in zip(news,x_lengths)]
    #bi-gru output: (batch_size,days, hidden_dim*2)
    output, _ = self.bigru(aggregated_news)
    #temporal  mean_output: (batch_size, hidden_dim*2)
    output = F.softmax(self.temporal_attention(output))*output
    mean_output = torch.mean(output,dim=1)
    return mean_output
```

Figure 6: BiGRU and attention layer.

Attention Mechanism: The intuition behind the attention layers is that not all articles contribute equally to prediction due to the various content. And since the stock market is time-sensitive, news or tweets posted on different dates have varying effects (Hu et al., 2018). As shown in Figure 5, The two attention mechanisms are both implemented by a linear and a SoftMax layer, while the masked SoftMax function is used to reweight the padded text sequence.

BiGRU: We employ the BiGRU model and then concatenate the output of both directions to encode the dates vectors in order to incorporate the context information, i.e. in this scenario, the past and future.

Multiwindow: The intention of implementing multiple windows is to confront the issue of inadequate data; since the volume of social media postings and the stock market are interdependent, there may be an empty window for a given date when it comes to low trading strength. Hence, analyze the information included in additional windows as supplementary. Same as the implementation in Custom Dataset, the parameter *windows* can be either a digit or a list. The Multi-window method encodes the embeddings from single source for each window using a shared BiGRU and attention network. Additionally, we consider the combination of news and tweets also as a multi-window issue; in this scenario, two windows will contain text from distinct textual sources, and the concatenated vectors will be fed into the MLP. We implement both a shared and a separate BiGRU and attention network.

MLP: The final discriminative network is a four-layer MLP, with four layers of output sizes 256, 128, 64 and 3; two neighbor layers are connected through a dropout to alleviate the overfitting issue. By applying a SoftMax to the logits, it can produce the possibility of each class.

3.5 Evaluation Metrics

Confusion Matrix

As this is a multilabel classification problem, the confusion matrix is a 3×3 matrix, which contains four terms for each class, i.e. True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). And metrics based on the confusion matrix are used to measure performance of model, such as Accuracy and Mattheus correlation coefficient.

Trading Strategy

To reflect the reality and verify the worthiness of the stock trend models, we leverage the Backtrader[7] package to conduct backtesting to simulate trading on five stocks. The strategy is pretty straightforward; we produce trading signals for each date based on the model predictions, i.e. UP indicates 'buy', PRESERVE indicates 'do nothing', and DOWN indicates 'sell' (if possessed). From 23/01/2019 to 30/12/2019, we invest evenly in five stocks and calculate the cumulative annualized returns of the portfolio.

[7] https://www.backtrader.com/

Chapter 4 Evaluation

4.1 Overview

This chapter discusses the detailed evaluation and analysis of models, and the evaluations are carried out to answer the following research questions:

RQ1: Is using the fine-tuned RoBERTa model as the contextualized embedding encoder an effective approach to improve the performance?

RQ2: Is using the attention mechanism an effective approach to integrate information from the embedding sequences?

RQ3: How well does having a multi-window approach improve the performance?

RQ4: Does combining the two textual sources (news the tweets) improve the performance of stock trend prediction?

RQ5: How well does the stock trend prediction model at discriminating different trends?

4.2 Experimental Setup

We tune the hyperparameters of the model only based on train and test data; the final result will be the prediction of validation data by the model trained on training data, without changing the parameters obtained. Early Stopping allows the model to cease training if the test loss increases for a certain number of epochs and store the model with the best test loss; the preset parameter patience determines the number of epochs.

Set random seeds to facilitate reproducibility of experiment results, then average the experiment performance of five different seeds on each model to obtain the final results. The cross-entropy loss function is utilized as the loss of each iteration and Adam optimizer is employed with a learning rate of 1e-3 for training.

4.3 Compared Models

To examine the effectiveness of our proposed model, we compare the following models.

Dummy Classifier (DC): We choose the *stratified* strategy for the dummy model; the classifier produces predictions based on the class distribution in training data.

GRU Baseline (GRU): The GRU network introduced in section 3.4.2.

Fine-tuned GRU (FT-GRU): The model structure is identical to GRU Baseline, except that we use embeddings extracted from the fine-tuned RoBERTa rather than the base Roberta.

BiGRU: We use the same network as in GRU Baseline but with a bi-directional GRU to examine the performance of bi-directional configuration.

Temporal-Attention BiGRU (TEMP-BiGRU): To analyze the temporal attention mechanism, we employ this model by replacing the averaging method of the BiGRU with the temporal-level attention layer.

Masked-Attention BiGRU (MASK-BiGRU): To analyze the masked attention method, we add the masked attention layer before the BiGRU to process the padded text sequences.

Single-Window BiGRU Attention (SW-BiGRU): The structure of this model is introduced in section 3.4.3, in which the parameter *windows* is a single number.

Multi-Window BiGRU Attention (MW-BiGRU): The structure of this model is introduced in section 3.4.3, in which the parameter *windows* is a list of numbers.

Fine-tuned Multi-Window BiGRU Attention (FT-MW-BiGRU): The model structure is identical to MW-BiGRU, except that we use embeddings extracted from the fine-tuned RoBERTa rather than the base Roberta.

4.4 Result and Analysis

Table 1: Performance of compared models

	News		*Tweets*	
	Acc	MCC	Acc	Loss
DC (w=5)	32.15	-0.0074	32.82	-0.0015
GRU (w=5)	59.19	0.4357	58.78	**0.4464**
FT-GRU (w=5)	59.75	0.4403		
BiGRU (w=5)	59.84	0.4439	58.81	0.4337
TEMP-BiGRU (w=5)	57.13	0.4023	57.53	0.4318
MASK-BiGRU (w=5)	59.16	0.4324	59.34	0.4454
SW-BiGRU (w=5)	58.63	0.4065	59.31	0.4269
SW-BiGRU (w=1)	56.14	0.3426	46.00	0.2235
SW-BiGRU (w=7)	61.64	0.4259	59.10	0.3869
SW-BiGRU (w=14)	62.76	0.4465	60.32	0.4145
MW-BiGRU (W= [1,7])	62.65	0.4399	59.51	0.4048
MW-BiGRU (W= [1,7,14])	63.56	0.4550	**62.08**	0.4443
FT-MW-BiGRU (W= [1,7,14])	**63.98**	**0.4655**		

The performance metrics of compared models is provided in Table 1, where w denotes the window's length and W denotes a list of windows' lengths. For clarifying the naming method, "A_w" denotes "a single-window model A with a window length of w", "A_w1_w2" denotes "the multi-window model A with two windows lengths of w1, w2, respectively".

Effects of fine-tuning base RoBERTa (RQ1)

Because the base RoBERTa model was pre-trained on news corpora, we would only compare the performance on news datasets.

As shown in Table 1, FT-GRU and FT-MW-BiGRU perform better than GRU and MW-BiGRU in terms of accuracy and MCC score, with a 0.56% and 0.42% increase in accuracy, respectively. Furthermore, in Figure 7, we notice that when directly employing the embedding from base RoBERTa in simulated trading, there is an obvious disadvantage in overall cumulative returns. Thus, applying contextualized embeddings from the fine-tuned model, which includes financial domain information, effectively improves the performance of the stock trend prediction, especially the GRU baseline.

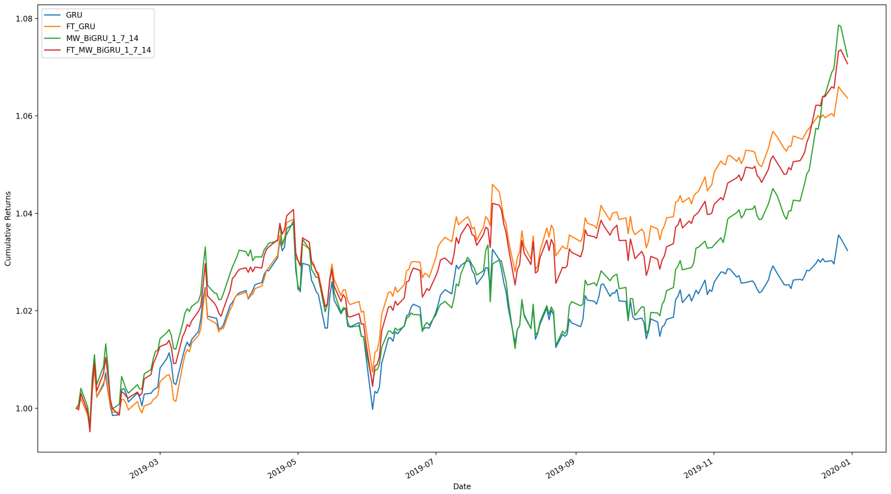


Figure 7: The backtesting result of GRU, FT-GRU, MW-BiGRU and FT-MW-BiGRU models on the news dataset.

Effects of Attention Mechanism (RQ2)

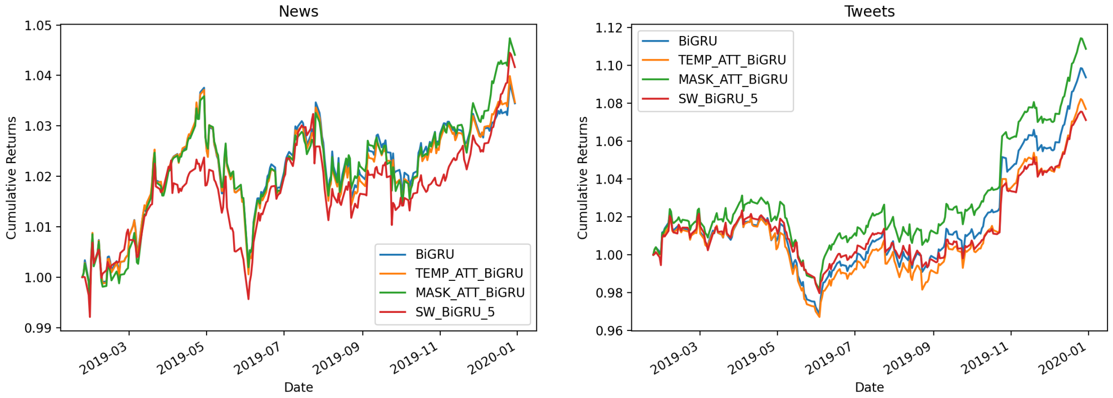


Figure 8: The backtesting result of different attention models on news and tweets.

As shown in Table 1 and Figure 8, the TEMP-BiGRU with a temporal attention layer and the SW-BiGRU with a combination of two attention layers significantly lower the performance metrics and cumulative returns of BiGRU baseline on both the news and tweets datasets. The MASK-BiGRU model with a mask attention layer degrades model performance on the news dataset but slightly improves it on tweets, boosting accuracy and MCC score by 0.53% and 0.0117, respectively, and providing higher returns throughout the year. This might be due to the dataset's nature; tweets include more noise than news since they are often posted by individual users; thus, assigning an attention score to each tweet in the text sequence could improve overall performance.

Together, using the attention mechanism is not effective in improving the performance of BiGRU. However, the results of mask attention on tweets data indicate that further investigation of the attention mechanism is possible.

Effects of Multi-Window (RQ3)

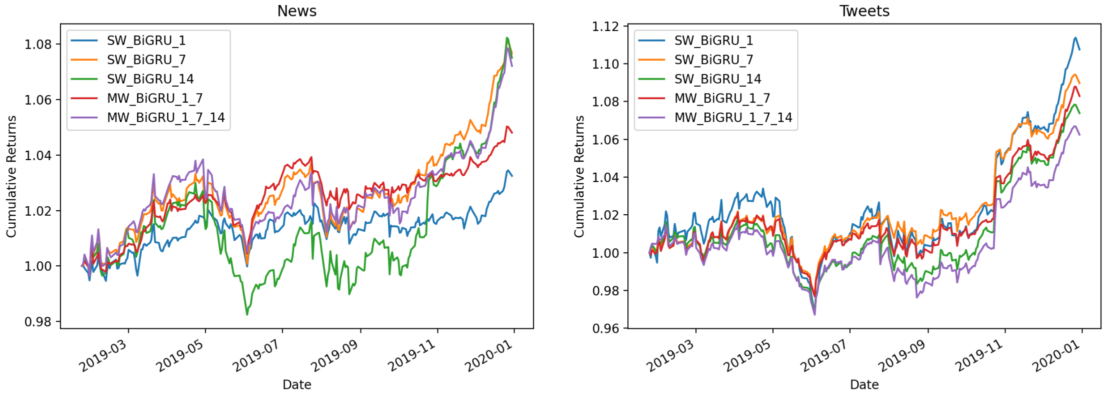


Figure 9: The backtesting result of models contain different windows.

From Table 1, it can be seen that the multi-window model outperforms all compared models on both datasets in accuracy score, and it gains superior accuracy than the model contained one of its single windows, for example, MW-BiGRU_1_7 gains better performance metrics than the models only contains a length of 1 and a length of 7 window. In addition, the model seems to have an improvement in accuracy and MCC scores with a lengthier window.

As shown in Figure 9, when employing news data, the SW_BiGRU_14 generates negative returns between 2019-05 and 2019-10; however, the SW_BiGRU_7 and MW_BiGRU_1_7_14 generate competitive cumulative returns, and both earned significant cumulative returns across the models with varied windows. When employing tweets data, the MW_BiGRU_1_7_14 has the lowest returns, followed by MW BiGRU_1_7.

Additionally, the result implies that the stock market reacts differently to various textual sources; news data has a long-term impact on the stock market, while tweet data has a short-term effect. Extending the period and adding additional windows approaches to the tweet model does not enhance its trading performance, though it does improve accuracy and MCC score.

Such that the answer to RQ3 is, having a multi-window approach can improve the performance metrics on both datasets, but it does not boost the actual returns and even significantly decrease the returns on tweets dataset.

Effects of combining News and Tweets (RQ4)

Table 2: Performance of models based on news, tweets or the combination

	Acc	MCC
News	58.63	0.4065
Tweets	**59.31**	**0.4269**
Shared Strategy	58.87	0.4135
Separate Strategy	58.72	0.4185

A single-window model with a window length of 5 trained on the news and tweets datasets is referred to as News and Tweets. Row 3, 4 are the combination of News and Tweets: Separate Strategy uses a separate BiGRU and attention layer for each textual source, while Shared Strategy uses a shared BiGRU and attention layer.

Table 2 and Figure 29 show that the two ways for merging news and tweets data provide comparable results in terms of performance metrics and cumulative returns. Although the multi-source approach does not improve accuracy and MCC scores, it beats the single-source model in trading simulation from November 2019 to January 2020.

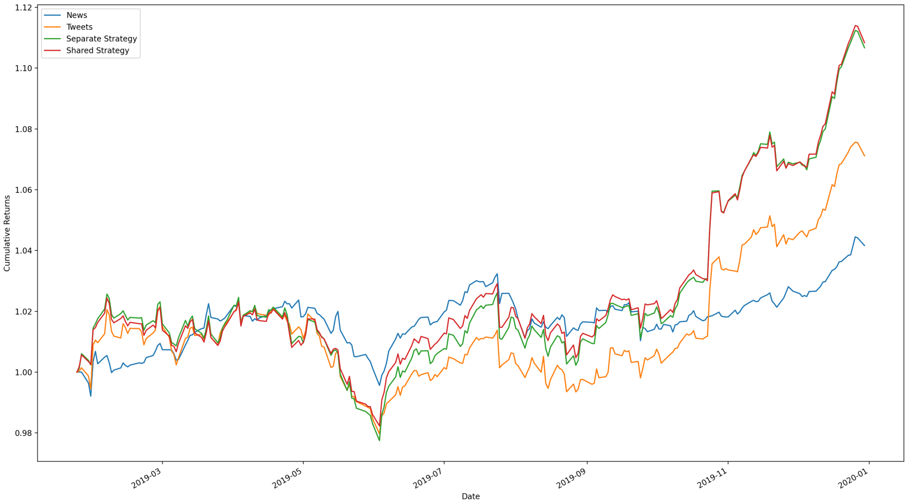


Figure 10: The backtesting result of models trained on different dataset

Discriminative Ability of Stock Trend Prediction (RQ5)

As seen in Figure 11, the model does a better job predicting the UP and DOWN classes than the PRESERVE class, particularly the GRU baseline on tweets data. This is because we set a threshold to equally split the Growth into three categories in order to balance the classes in our dataset. While this strategy addresses the class imbalance issue, it also brings a vague boundary between the UP and PRESERVE classes and between the DOWN and PRESERVE classes, leading our model to have poor discrimination ability in the PRESERVE class. Thus, to answer RQ5, the stock prediction model is generally effective at categorizing the UP and DOWN classes but performs poorly in the PRESERVE class.

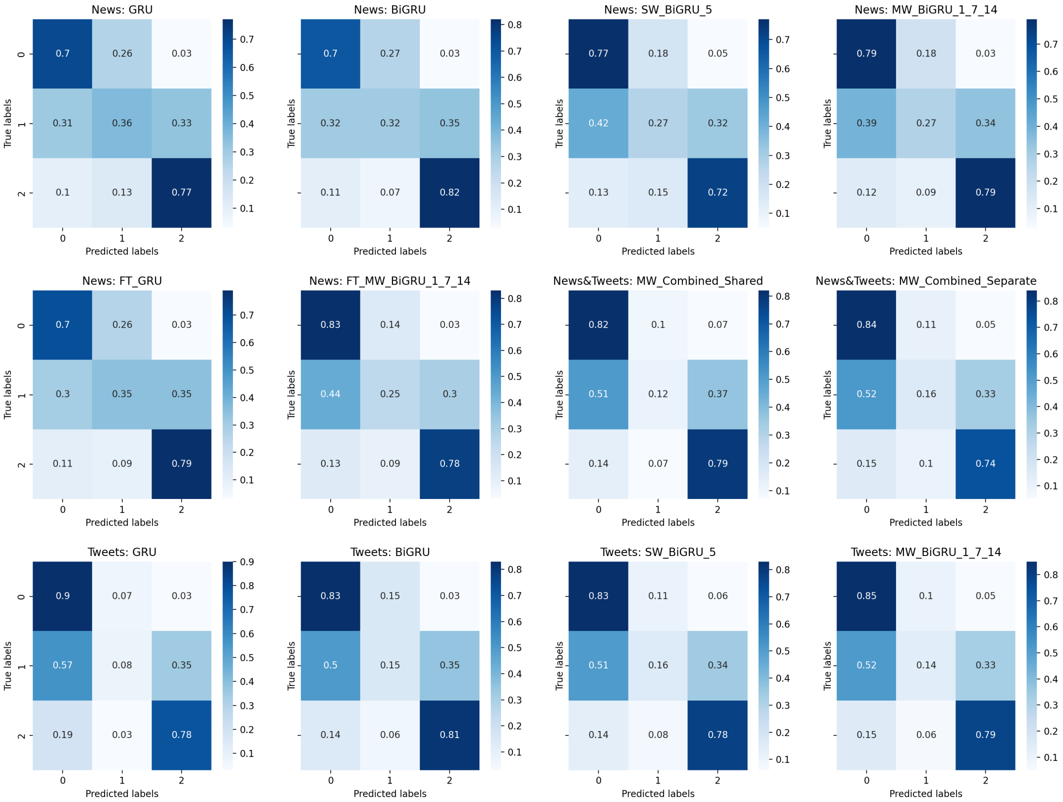


Figure 11: Normalized confusion matrix of compared models

Chapter 5 Conclusion

5.1 Achievements

The primary objective of this project is to develop a model for stock trend prediction. Accordingly, we accomplished three research objectives. Firstly, we investigated the performance of a base RoBERTa model fine-tuned on financial news; we found that fine-tuned contextualized embeddings indeed perform better than general embeddings. Secondly, we reproduced and evaluated the Hybrid Attention Network, where we found the only the mask attention mechanism works on tweets data in the trading simulation of our model. Lastly, to augment the prediction model with additional data, we propose a multi-window framework for combining financial texts released over multiple periods as well as combining textual source data from various sources. The proposed architecture improves the overall performance of the baseline and single-window model. More precisely, the model with multiple windows with the same textual source increases sentiment classification accuracy; the model paired with news and tweets achieves the higher cumulative return in simulated trading compared to models with a single textual input. As previously stated, we believe that further investigation into fine-tuning the contextualized embedding encoder and multi-window structure can lead to better accurate trend predictions and higher returns from the stock market.

5.2 Limitations and Future work

One thing we could not do is to conduct experiments on a diverse range of equities. We employ a subset of the news data to be consistent with the tweets dataset in order to investigate the integration of various textual sources. As a consequence of the insufficient data, the project is based on the top five stocks associated with leading technology companies, which have a high market capitalization and their prices are continuously growing, albeit some volatility may occur. This lowers the generalization of our project and introduces experimental bias into our model prediction. Additionally, due to time constraints, we could not investigate the attention mechanism and parameter settings further, and there is still an optimizing possibility in the original parameter space.

Looking at the discriminative performance of the compared models on sentiment classification, it is obvious that the model requires further refinement. If there were more time and resources, we would investigate the various thresholds for data labelling in order to determine the optimal boundary for accurately dividing the three classes and evaluate a diverse selection of equities to confirm the effectiveness of our proposed structure.

Besides, I would like to do more research in stock trend prediction and the fine-tuned document embeddings based on longer texts (such as financial reports). Then integrate more different textual sources using the multi-window structure based on different deep learning models.

Furthermore, investigate the multi-source abstractive and extractive text summarization algorithms to capture the stock market sentiment from financial texts in a different representation.

Appendix A

The code for this project:
https://github.com/hahuh/Sentiment-analysis-for-financial-news

Bibliography

Adam, K., Marcet, A., & Nicolini, J. P. (2016). Stock market volatility and learning. *The Journal of Finance*, *71*(1), 33–82.

Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *ArXiv:1908.10063 [Cs]*. http://arxiv.org/abs/1908.10063

Bahdanau, D., Cho, K., & Bengio, Y. (2016). Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv:1409.0473 [Cs, Stat]*. http://arxiv.org/abs/1409.0473

Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems with Applications*, *42*(14), 5963–5975. https://doi.org/10.1016/j.eswa.2015.03.017

Chen, Q. (2021). Stock Movement Prediction with Financial News using Contextualized Embedding from BERT. *ArXiv:2107.08721 [Cs, q-Fin]*. http://arxiv.org/abs/2107.08721

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). Deep learning for event-driven stock prediction. *Proceedings of the 24th International Conference on Artificial Intelligence*, 2327–2333.

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2016). Knowledge-Driven Event Embedding for Stock Prediction. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2133–2142. https://aclanthology.org/C16-1201

Gupta, R., & Chen, M. (2020). Sentiment Analysis for Stock Price Prediction. *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 213–218. https://doi.org/10.1109/MIPR49039.2020.00051

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2018). Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 261–269. https://doi.org/10.1145/3159652.3159690

Li, Q., Jiang, L., Li, P., & Chen, H. (2015). Tensor-Based Learning for Predicting Stock Movements. *Proceedings of the AAAI Conference on Artificial Intelligence*, *29*(1), Article 1. https://ojs.aaai.org/index.php/AAAI/article/view/9452

Lin, S., Ren, D., Zhang, W., Zhang, Y., & Shen, D. (2016). Network interdependency between social media and stock trading activities: Evidence from China. *Physica A: Statistical Mechanics and Its Applications*, *451*, 305–312. https://doi.org/10.1016/j.physa.2016.01.095

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv:1907.11692 [Cs]*. http://arxiv.org/abs/1907.11692

Nayak, R. K., Mishra, D., & Rath, A. K. (2015). A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices. *Applied Soft Computing*, *35*, 670–680. https://doi.org/10.1016/j.asoc.2015.06.040

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, *53*(4), 3007–3057. https://doi.org/10.1007/s10462-019-09754-z

Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, *73*, 125–144.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *ArXiv:1802.05365 [Cs]*. http://arxiv.org/abs/1802.05365

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, *42*(6), 3234–3241. https://doi.org/10.1016/j.eswa.2014.12.003

Sohangir, S., Wang, D., Pomeranets, A., & Khoshgoftaar, T. M. (2018). Big Data: Deep Learning for financial sentiment analysis. *Journal of Big Data*, *5*(1), 3. https://doi.org/10.1186/s40537-017-0111-6

Strauß, N., Vliegenthart, R., & Verhoeven, P. (2018). Intraday News Trading: The Reciprocal Relationships Between the Stock Market and Economic News. *Communication Research*, *45*(7), 1054–1077. https://doi.org/10.1177/0093650217705528

Sul, H. K., Dennis, A. R., & Yuan, L. (Ivy). (2017). Trading on Twitter: Using Social Media Sentiment to Predict Stock Returns. *Decision Sciences*, *48*(3), 454–488. https://doi.org/10.1111/deci.12229

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? In M. Sun, X. Huang, H. Ji, Z. Liu, & Y. Liu (Eds.), *Chinese Computational Linguistics* (pp. 194–206). Springer International Publishing. https://doi.org/10.1007/978-3-030-32381-3_16

Xu, Y., & Cohen, S. B. (2018). Stock Movement Prediction from Tweets and Historical Prices. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1970–1979. https://doi.org/10.18653/v1/P18-1183

Zhang, L., Aggarwal, C., & Qi, G.-J. (2017). Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2141–2149. https://doi.org/10.1145/3097983.3098117