北京郵電大學

Queen Mary University of London

Undergraduate Project Report
2019/20

Improving the Stock Market Prediction with Representation Learning

Name:	Fan Ding
School:	International School
Class:	2016215119
QM Student No.:	161188243
BUPT Student No.:	2016213516
Programme:	Internet of Things Engineering

Date: 04-05-2020

Table of Contents

Abstract

Stock market price trend prediction is one of the most significant tasks for stock investors, and it involves various types of data, including quantitative trading data like prices and qualitative descriptions which are un-quantitative such as news, comments and reports. Traditionally, most of the researchers who handled this task in the previous studies mainly focused on the quantitative trading data, like stock prices or quantifiable indicators like that. However, the task of stock prediction only depending on this information, which confines to the quantifiable data, is not enough. The relevance between stocks is also one of the most important factors which contribute to the stock prediction. Moreover, it is vital to incorporate the qualitative data like news and comments of investors in order to improve the stock prediction performance. In this paper, to learn the useful representation of stocks' relevance, I apply the node2vec to learn low-dimensional representations for stock nodes in a stock relationship graph and then calculate and obtain the correlation matrix of different stocks. Next, to fuse price data, news data and stocks correlation matrix into a unified framework, I build a learning model based on RNN-LSTM. To obtain the stocks relationship graph, I design a stock relationship graph construction model based on the principles of word2vec, using fluctuation information of stock prices to construct a stock relationship graph. To get the sentiment score of stock news, I employ the emotional analysis of NLP. The training and testing sets of the learning model are based on data of 50 stocks from the year 2015 to 2017. I design and conduct groups of experiments to evaluate and verify the entire model. Finally, in this paper, I show the working software which can visualize representations and report the stock prediction results.

Keyword: stock market, representation learning, node2vec, word2vec, RNN

摘要

This is the Chinese translation of the Abstract.

股票价格走势预测是投资者的重要任务之一，涉及到各种形式的数据，包括可定量的交易数据，例如股价；和不可定量的定性描述，例如新闻和报道。传统上，以往的研究者在研究中处理这一问题时，大多仅专注于股票价格等可定量的交易数据上。然而，股票预测仅仅依靠局限于可量化的信息是不够的。股票相关性同样也是影响股票预测的最重要因素之一。此外，为了提高预测效果，将新闻和投资者评论等定性的数据纳入其中也很重要。为了学习股票相关性的有用表征，本文利用 node2vec 学习股票关系图中股票节点的低维表示，从而计算并得到股票的关联矩阵。其次，为了将价格数据、新闻数据和股票关联矩阵融合到一个统一的框架中，构建了基于 RNN-LSTM 的学习模型。为了得到股票关系图，根据 word2vec 的原理设计了一个股票关系图构建模型，利用股票价格波动信息构造股票关系图。为了得到股票新闻的情感得分，采用了 NLP 的情感分析方法。该学习模型的训练和测试集都基于 2015 年至 2017 年 50 只股票的数据。同时进行了多组种实验来评估设计的完整模型。最后，在本文中介绍了设计的应用软件，它可以可视化特征表示和展示股票预测结果。

关键字：股票市场，表征学习，node2vec，word2vec，RNN

Chapter 1: Introduction

1.1 Project Research Motivation and Significance

Techniques of Data Mining play an important role in the financial field. Prediction of the stock price trends is one of the significant tasks in terms of data mining.

Being able to accurately predict the trend of stocks has always been the desire of every investor, most investors and a number of researchers have tried different methods to analyse stocks' trends in order to figure out the problem, nevertheless, it is difficult for people to give prediction precisely. After talking with a graduate student in the financial major with respect to that problem, I learned that based on nowadays technology, qualitative analysis is more reasonable and feasible than Quantitative analysis that is very challenging. Also, there is still a lot of room to improve the performance of stock prediction.

Precisely, it is more appropriate to understand stock forecasting as a time-series problem based on historical data, then predicting future price trends. RNN is the main technique, which researchers used to try to figure out the problem of stock prediction. This technique mainly depends on the quantifiable data like the historic stock price and indicators of quantifiable features such as Price to Book Ratio (PB), Price Earnings Ratio (PE), and stuff like that [1].

However, stock price prediction only depending on the information that confines to the quantifiable data is not enough. On one hand, because its decisions and fluctuations are subject to a variety of economic and political factors, as well as investment psychology, it is more suitable for us to incorporate the unquantifiable data like news, reports, and comments of investors into the task of stock prediction, which can reflect a significant positive or negative sentiments contributing to improving the prediction performance. On the other hand, it is beneficial to consider the relevance among different stocks. For instance, "Microsoft successfully completed the acquisition of company xxx" Apart from the stock of Microsoft will rising, companies that have certain relationships with Microsoft such as corporation supply and the customer will also be affected by this news, whose stock will also rise. Therefore, it is necessary to consider the companies relationships in terms of measuring the impact of the news.

The definition of representation learning is: "Learning representations of the data which can make it easier to extract useful information when building classifiers or other predictors"[2]. Obviously, unlike the quantifiable data like stock price which is easier for us to represent using float number, relationships among different stocks such as the stock relationship graph and the sentiment of the stock news, comments or report need special techniques, related to

representation learning, to handle. Specifically, to incorporate the relevance of the stock into stock prediction, we need representation learning techniques like node2vec, embedding the graph of companies (stocks)' relationship into vectors, which means learning low-dimensional representations for stock nodes in a stock relationship graph. Then, we can calculate the similarity between every two stocks using the embedding result and obtain the correlation matrix of all stocks. Apart from that, handling the task extracting sentiment from each stock news also involves the knowledge of representation learning. Usually, we need to apply the knowledge of emotional analysis in the field of natural language processing (NLP).

The point and novelty of the project are that: adopting techniques related to representation learning, learn useful representations of various types of data, including stocks relationship graph and qualitative descriptions such as news. Then design and implement a learning model and fuse these representations and the quantitative data like stock price into the model for stock price fluctuation predictions.

1.2 Project Process Overview and Achievement Summary

Overall, the process of this project consists of 4 phases:

1.2.1 Data collection, Data preprocessing, Feature extraction:

The whole data I used in this project includes the quantifiable price data and news data of 50 Chinese stocks in the year 2015, 2016 and 2017. One point during this phase is that initially, I hope to find and use the ready-make companies' relationship graph from the online resources (such as https://www.tianyancha.com, etc.), which are clear and reliable. However, after investigation, there is no suitable companies' relationship graph for my project. Those companies' relationship graphs mainly focus on some other relationships such as the Tenure relationship, which mainly reflects the economic interest relationship between individuals and companies. There is limited information about the clear interest relationship between different companies. After discussing with my supervisor, I decided to design a model to construct the companies' relationship graph by myself, which based on the principles of word2vec. The vertexes in this graph represent different stocks (companies); the edges between every 2 vertexes are set according to the similarity of stock fluctuation of prices. When pre-processing the dataset, I choose Pandas which is a NumPy-based tool created to address data analysis tasks.

1.2.2 Representation Learning model building:

I decided to use the node2vec to implement the graph embedding model, which can embed the stock nodes of Companies' relationship Graph into the stock vectors (learning low-dimensional

representations for stock nodes in a stock relationship graph) to calculate the similarity between every 2 stocks and obtain the correlation matrix of all 50 stocks. As for extract the sentiment from the stock news, I decided to use the NLP-Python-SDK provided by Baidu-ai, which can judge the category of emotion polarity (positive, negative, neutral) on the text containing subjective information and give the corresponding confidence degree. Represent positive news with 2, negative news with 0, natural news with 1.

1.2.3 Design and Implementation of Learning model, training and experiments of learning model:

I have read relevant literature and papers that focus on stock prediction issues. In short summary, the previous works mainly handled the stock prediction works as a time-series problem based on historical prices and other quantifiable data like that. The main technique used to handle that problem is mainly designed and implemented on the Recurrent Neural Network (RNN). Therefore, in this project, the implementation of the Learning Model is based on RNN, and I will use the Representation Learning Result in the previous phase, combined with the quantifiable data to improve the stock prediction. I decided to choose the Pytorch as the main learn deep learning framework, which I think is more convenient. When training and evaluating the learning model, I set 3 groups of experiments. Group1 (Quantifiable): give prediction only depending on the quantifiable data like stock prices. Group2 (Quantifiable + News): give prediction depending on the quantifiable data like stock prices and the news which reflect the sentiment. Group3 (Quantifiable + News +Stock Relationships): give prediction depending on the quantifiable data like stock prices, the news which reflects the sentiment, and the stock relationships.

1.2.4 Writing software with an interface which can report the results and visualize the representations.

The project has a working software so that readers can view the representation and the output of the learning model, the prediction result of 50 stocks based on the trained learning model. The working software in this project is designed and implemented in python. The python-package and the design tool I used is Pyqt5 and the QT Designer.

1.3 Paper Structure

The rest of this paper structure is organized as follow:

Chapter 1 introduction, this part gives a brief overview of the overall project which includes Project Research Motivation and Significance, Project Process Overview and Achievement

Summary and Paper Structure.

Chapter 2 Background, this part provides some background concepts about algorithms that are used in this project. And it also illustrates the related work of this project by other researchers and the background technology.

Chapter 3 Design and Implementation, this part tells readers the details of design and implement the entire project, which includes the Stock Graph Construction Model, Graph Embedding Model, the Learning Model based on LSTM. It also explains the main part of the working software.

Chapter 4 Result and Discussion, this part tells the details about the dataset, how the project set the experiments and the results of the experiments to verify and evaluate the entire model.

Chapter5 Conclusion and Further work conclude the whole project and discuss the further work of this project.

Chapter 2: Background

2.1 Concept Explanation

In this chapter, all the concepts of the background technology related to this project will be briefly explained. The definition and explanation of Representation Learning, which is easy for the reader to understand. The word2vec algorithm, a type of Natural Language Process (NLP) method, usually is used to produce word embeddings, which, however, this project creatively used and adopted to construct the stocks relationship graph. The node2vec, an algorithmic framework for representational learning on graphs, which is used to learn low-dimensional representations for stock nodes in a stock relationship graph and then calculate and obtain the correlation matrix of stocks, in this project. LSTM (Long Short-Term Memory) a classical Recurrent Neural Network (RNN) which the project used to build the learning model. Moreover, this chapter will give a brief summary of the related work. This chapter will provide all the necessary information for the rest chapter of the paper.

2.1.1 Representation Learning

The definition of Representation Learning is that: learning representations of the data that make it easier to extract useful information, which then can be used for building classifiers, predictors and other downstream machine learning tasks [2]. When people learn a complex concept, people always hope that there is a shortcut to simplify it. Machine learning models are no exception. If a refined and better representation of the raw data exists, people can often make subsequent tasks more efficient. This is also the basic idea of presentation learning, which is to find a better representation of the original data for subsequent tasks (such as classification and regression problem). The development of artificial intelligence, to machine learning, to deep learning has gone through a bumpy and rising process. More and more models have been proposed, however, if hoping to get a good result, still need to pay much more attention to the data. Good data quality, good data characteristics is one of the most significant factors.

In terms of machine learning, if people need to represent thing A, it usually relies on experts to manually extract features to represent A, which is called Feature engineering. Feature engineering is the process of transforming the original data into features that can describe the data well and allow the model to perform outstanding performance. Feature engineering can be considered as applied machine learning itself [3]. From the mathematical view, feature engineering is how to design the input variable X of a model. In the aspect of deep learning, we directly input the A into the model, and the A will be automatically transformed into an efficient

and meaningful representation, which is called representation learning. Table 1 shows the contrast between Feature engineering and Representation learning.

Table 1 Feature engineering and Representation learning

Machine learning	Feature engineering	Manually extract	The performance of the task depends on the experts to extract the explicit features, and the workload of the project is huge
Deep learning	Representation learning	Automatically extract	The implicit characteristics of data are learned automatically by the model, independent of experts' experience

2.1.2 word2vec

In this project, I used the word2vec algorithm to construct the stock graph for later representation learning. In this chapter, it will give a brief explanation of the word2vec concept and principle. How to design and implement the Graph construction model using the word2vec algorithm will be explained in chapter 3.

In the previous and common natural language processing system, the encodings of different words are arbitrary, so it is impossible to provide the system with enough useful information about the possible relationship between symbols, and it can also cause the problem of data sparsity. However, this problem can be overcome by the use of word embedding like word2vec. Table 2 shows the example of tradition one-hot representation of the words and the embedding result of word2vec.

Table 2 Example of different representation of words

word	One-hot	Embedding Result of word2vec
King	0001	[0.99,0.99,0.55,0.7]
Queen	0010	[0.99,0.05,0.93,0.6]
woman	0100	[0.02,0.01,0.99,0.5]
princess	1000	[0.98,0.02,0.94,0.1]

Word2vec is a way of representing the semantic information of a word in terms of word vectors by learning the text dataset, that is, by making semantically similar words close together in an

embedded space (low dimensional) [4][5][6]. Generally speaking, word2vec is usually used to represent each word with a vector (Semantic information is represented by the word vector implicitly), which has a better performance than one hot representation especially when you need to calculate the similarity between different words. The input of the word2vec are many sentences, and it will generate the training samples based on these sentences, each word is represented in one-hot representation. With specific window size, each sentence will generate training samples firstly, then use those samples to train the neural network model of word2vec. Finally, we can get the vector (embedding result) of each word appear in these input sentences. Figure 1 gives an example of how we generate the training samples for the word2vec neural network from each input sentence.

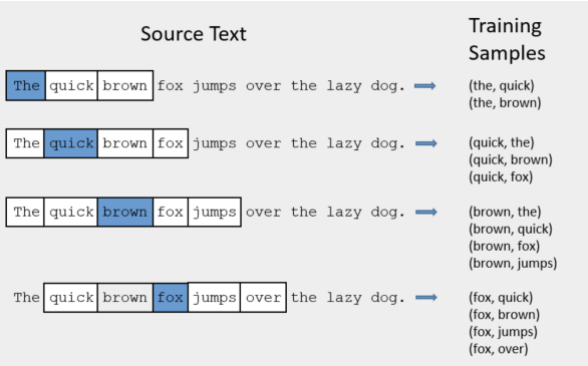


Figure 1 Example of generating training samples from the input sentence with windows size of 2

Figure 2 shows the details of the neural network of word2vec. The weight matrix from the input layer to the hidden layer '*WI*' and the weight from the hidden layer to the output layer '*WO*' are written as matrix respectively.

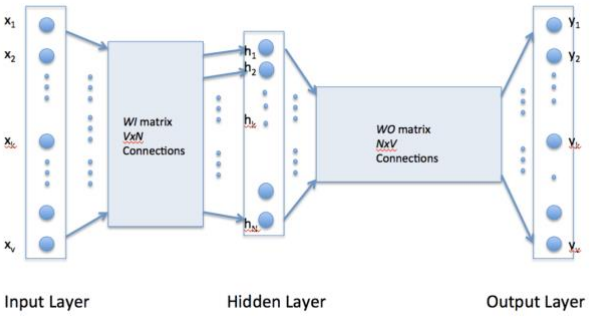


Figure 2 Neural network of word2vec

If there are N neurons in the hidden layer, which means that the input word finally is represented as an n-dimensional vector. The weight matrix in the hidden layer is V*N in size. A v-dimensional one-hot input word is mapped to a N-dimensional vector through the weight matrix, and this vector is the so-called word vector, corresponding to each row on the right in Figure 3.

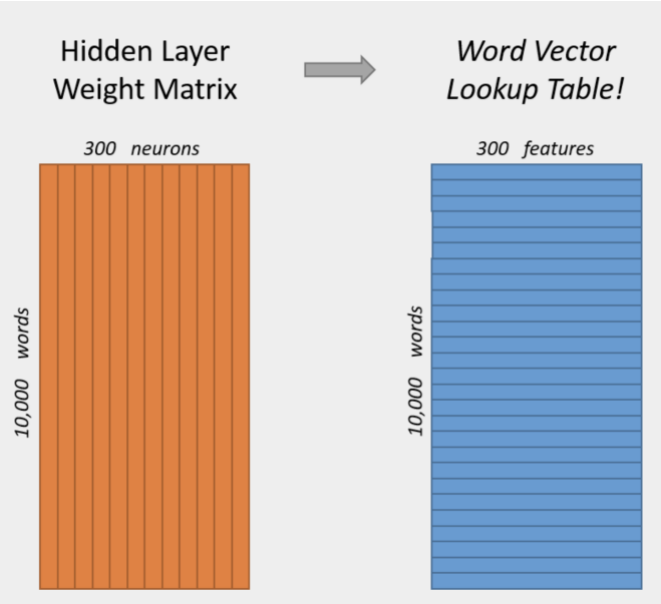


Figure 3 The trained hidden layer matrix is exactly the word vector lookup table (10000 different words, finally each word is represented by a vector with length 300)

The real and final purpose of the wor2vec training model is to obtain the hidden layer weight of the model based on the input training samples. Actually, word2vec doesn't really care about the trained model. Word2vec can be divided into two parts: the model and the word vector obtained by the model. The model is shallow, two-layer neural networks. When the network is trained, we don't use the trained network to handle new tasks. What we really need is the parameters that the model learning from the training samples, the hidden layer weight matrix -- which we see in Figure 3 the hidden layer weight of word2vec are actually the "word vectors" we are trying to learn. The process of training data using the model, we gave it the name "Fake Task", which means that modelling is not our ultimate goal.

As mentioned above, both input words and output words will be encoded by one-hot representation. When we finish training the model and get the hidden layer matrix, equation (1) can explain how we lookup the embedding result of each word. The left-hand side is 2 matrixes in size 1 x 5 (one-hot representation of a word) and 5 x 3 (hidden layer weight matrix of the model), the right-hand side is the vector in size 1 x 3 (the embedding result of the word). It can be easily seen that in this way, the hidden weight matrix in the model becomes a "lookup table". When matrixes are calculated, the corresponding weight values of the dimension with the value of 1 in the input vector are directly checked. The output of the hidden layer is the "embedded word vector" of each input word. The equation (1) shows a word represented by one-hot with length 5 is embedded to a vector with length 5.

$$[0 \quad 0 \quad 1 \quad 0 \quad 0] \times \begin{bmatrix} 17 & 24 & 01 \\ 23 & 05 & 07 \\ 04 & 06 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19] \tag{1}$$

Word2vec mainly contains two models: Skip-Gram and Continuous Bag of Words (CBOW). Figure 4 shows the difference between CBOW and Skip-gram [4].

- Skip-Gram model: given input words to predict context.
- Continuous Bag of Words model (CBOW): given context to predict input words.

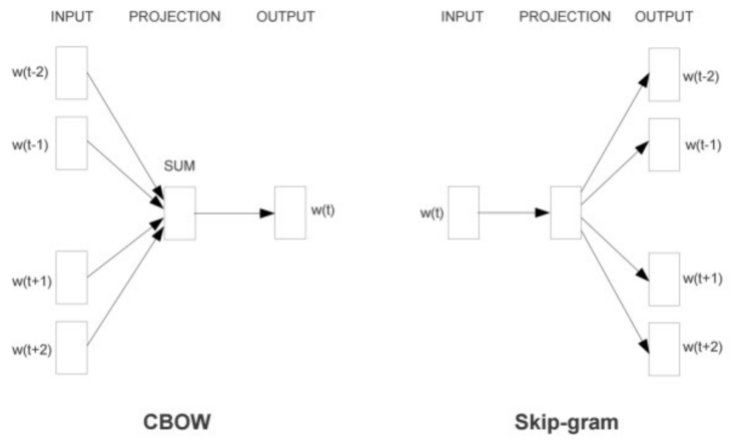


Figure 4 The difference between CBOW and Skip-gram

In this project, I used the Skip-Gram model, the reason Why choosing Skip-Gram model will be explained in chapter 3 when discussing the design and implementation of the graph construing model.

2.1.3 node2vec

Node2vec, which derived from word2vec, is an embedding algorithm used for learning the nodes representations in a graph. Generally speaking, it plays the role of embedding the complex Graph, which is difficult to quantify or represent in the deep learning tasks, into a set of low-dimensional node vectors. Each vector represents a node in the graph, and each vector contains not only the characteristics of the node itself but also the characteristics of the relationship with other nodes [7].

The node2vec method is mainly inspired by and derived from the word2vec method. The node2vec algorithm can be divide into 2 parts. The first part is the biased random walk, the second part is actually the same as the word2vec algorithm. The difference is that node2vec firstly treat biased random walk on input networks or graphs to generate node sentences, then learn node representations with the technique for learning word representation, word2vec. The

input of word2vec is a set of sentences. However, the input of the node2vec is a graph, because of that, firstly, the node2vec algorithm will firstly walk randomly on the input graph to generate many node sequences sentences for training. Each node of a graph is treated as an individual word, and a random walk will generate a sentence made of nodes. As you can imagine, the generated node sentences are like the input sentences in word2vec.

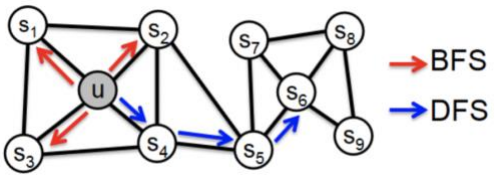


Figure 5 BFS and DFS search strategies from node u (k=3)

node2vec generates the node sentences with a hybrid strategy of Breadth-first Sampling (BFS): homophily; and Depth-first Sampling (DFS): structural equivalence, as we can see in the above Figure 5. And the biased random walk of Node2Vec has two parameters p and q:

• p: controls the probability of revisiting a node in the walk

• q: controls the probability of exploring "outward" nodes

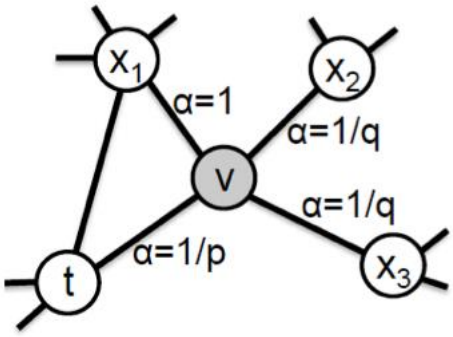


Figure 6 p and q control the biased random walk

For example, in Figure 6, let's start at node *t* with a biased random walk, and if we are at node *v*. Which is the next node? If you use the BFS strategy, you should go to x_1, because *v* and x_1 are direct neighbors of the *t* node; If you choose the DFS strategy, you will go to x_2 or x_3, because they're a step apart from *t*; And, of course, you maybe go back to the *t* node. Therefore, the node2vec designed a second-order transition probability algorithm: equation (2).

$$\pi_{vx} = \alpha_{(t,x)} \cdot w_{vx} \quad \textbf{(2)}$$

Where, w is the weight of the edge of two nodes, which is set according to the scene. Search bias is defined as equation (3).

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & if\ d_{tx} = 0 \\ 1 & if\ d_{tx} = 1 \\ \frac{1}{q} & if\ d_{tx} = 2 \end{cases} \quad (3)$$

The next step of a node depends on the relationship between the previous step and the next step. *v* is the current node, *t* is the node where the previous step of *v* is located, and *x* is the position of the next step. *D (t, x)* represents the shortest distance between *t* and *x*, Table 3 explains how *p* and *q* affect the random work.

Table 3 the weight of different *D (t, x)*

When d=0	returning from v to t node. In this condition, search bias is 1/p, which can be understood as returning the previous step with the probability of 1/p
When d=1	x is the direct neighbor of t, equivalent to BFS, the bias is 1.
when d=2	x is the neighbor of t's neighbor, equivalent to DFS, the bias is 1/q

p is called the return parameter, because it controls the probability of returning to the original node. *q* is called the in-out parameter, because it controls the relationship between BFS and DFS. In this way, if set different *p* and q, it can get different node sequences with a different bias. When training the model, you can select *p* and *q* according to the needs of the scene.

2.1.4 RNN and LSTM

The ordinary neural network can only take and process one input matrix each time, and the previous input and the latter input are completely unrelated. However, some tasks need to be able to handle sequence information better, that is, the previous input is related to the subsequent input.

RNN (Recurrent Neural Network) is a kind of Neural Network used to process sequence information data. After each RNN operation, it will generate a description of the current state. We will use the abbreviation *S(t)*, and then the RNN will start to analyze *x(t+1)*, it will generate *S(t+1)* according to *x(t+1)*, but in this case, *y(t+1)* is actually created by *S(t)* and *S(t+1)* together.

Ordinary RNN has long-term dependency problems, such as long-distance between the useful information and the place where the information needs to be processed. As a result, ordinary RNNs can't learn useful information easily, and the task may get failed.

LSTM was proposed to solve this problem. LSTM has three more controllers gate (input control,

output control, forget control) than normal RNN. The LSTM network can delete or add information to a cell's state through a gate [8].

The first step of LSTM is to determine what information the cell state needs to discard. This operation is handled by a sigmoid unit called a forget gate. It outputs a vector between 0 and 1 by looking at h_{t-1} and x_t information. And the value between 0 and 1 in the output vector indicates what information in the cell state C_{t-1} is retained or how much is discarded. 0 means discard all information, and 1 means retain all information. Figure 7 and equation (4) shows the principle of the forget gate.

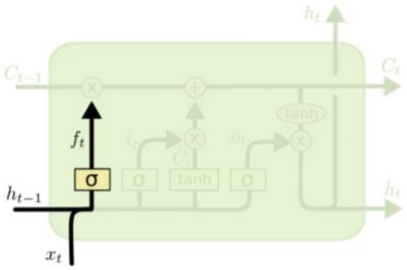


Figure 7 Forget gate of LSTM

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4}$$

The next step is to decide to add how much new information to the cell state. First, using h_{t-1} and x_t through an operation called the input gate to decide what information needs to be updated. New candidate cell information $\widetilde{C_t}$ is then obtained by using h_{t-1} and x_t through a tan layer, which may be updated into the new cell information. These two steps are described in Figure 8 and equation (5)

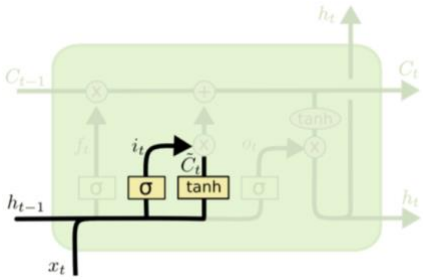


Figure 8 Input gate of LSTM

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \end{cases} \tag{5}$$

The old cell information C_{t-1} will be updated to the new cell information C_t. The updated rule is to select a part of the old cell information C_{t-1} to forget by the forget gate, and select a part of the candidate cell information $\widetilde{C_t}$ to add by the input gate to get the new cell information C_t. The update operation is shown in Figure 9 and equation (6).

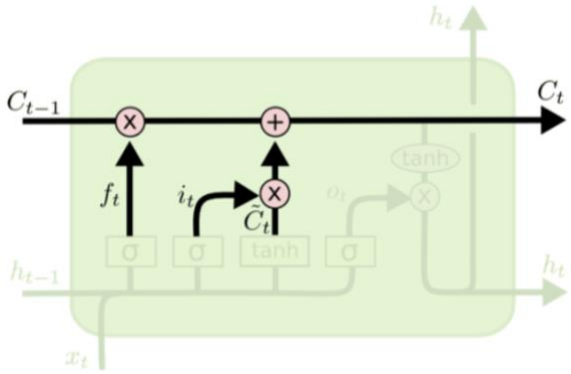


Figure 9 update old cell information C_{t-1} to the new cell information C_t

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (6)$$

After updating the cell state, it is necessary to decide to output what characteristics information of the cell C_t according to the input h_{t-1} and x_t. Here, we need to pass the input x_t through a sigmoid layer called the output gate to get the judgment condition o_t, and then pass the cell state C_t through the tanh layer to get a vector between -1 and 1, which is multiplied by the judgment condition of the output gate to get the final output h_t of the RNN unit. This step is shown in Figure 10 and equation (7).

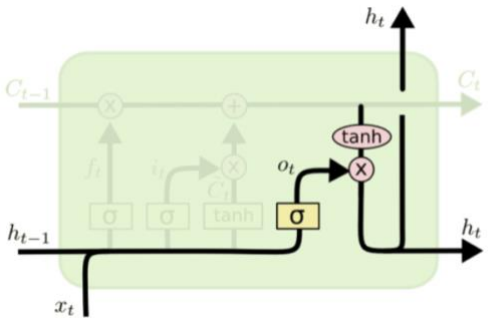


Figure 10 output gate of the LSTM

$$\begin{cases} o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t * \tanh(C_t) \end{cases} \quad (7)$$

2.2 Related Work

Stock market prediction is a popular research topic in data mining, which involves various types of data, including quantitative trading data and qualitative descriptions such as news and reports. I have read several relevant literature and papers that focus on stock prediction issues. In summary, the previous works mainly handled the stock prediction works as a time-series problem. The main technique used to handle that problem is mainly designed and implemented on the Recurrent Neural Network (RNN). Most of the researchers are mainly based on historical prices and other quantifiable data like that. Actually, besides the quantifiable data like prices, a few researchers have already incorporated the qualitative descriptions like news data and investors comment into the stock prediction task. Xi Zhang has introduced an Extended Coupled Hidden Markov Model incorporating the news events with the historical trading data [9]; and proposed a multi-source multiple instance model that can effectively combine events, sentiments, as well as the quantitative data into a comprehensive framework [10]. Also, a few researchers have considered the relationship of stocks. Xi Zhang predicts multiple correlated stocks simultaneously through their commonalities [11]; Jue Liu has used a trans model representing the entities and relations of the knowledge graph in the vector space to measure the correlation between stocks when predicting the stock price [12]. Representation learning has shown its superior performance in many fields such as natural language processing and image processing. For example, the node2vec and deepwalk algorithms shown superior performance when handling the graph embedding task. However, it still lacks research works that attempt to apply this technique to stock market prediction. Therefore, in this project, I will use the Representation Learning method combined with the RNN to improve the stock prediction.

2.3 Background Technology

2.3.1 Program Language

For the sake of programming language's unification, in this project, I choose Python as the programming language. All of the project processes, data pre-processing, design and implement the models, design and implement the working software, are completely done in python. There are several reasons. After a simple investigation, I decided to use Pandas combined with NumPy

to preprocess the data, which are high-performance and easy to use data analysis tools for the Python programming language. Moreover, the famous deep learning framework Pytorch, which I decided to use is based on Python Language. The last reason is that I decided to use Pyqt5 to design and implement my working software, which is a GUI package base on python.

2.3.2 Framework Environment

Programming language: Python

Deep learning Framework: Pytorch

Operation System: MAC OS Catalina

Chapter 3: Design and Implementation

3.1 Model Description

There are several important parts of this project, which are shown in Figure 11, in this chapter, I will explain the design and implementation process of each phase using this graph. The first part is to design and implement the Stock Relationship Graph Construction Model, which is used to generate the companies' relationship graph. The second part is to design and implement the Graph Embedding Model, which is used to learning the low-dimensional representation of the companies' relationship graph and then get the correlation matrix of all stocks. The third part is to design and implement the learning model which is used to train models based on the stock price data, stock news data and stock correlation matrix. One point is that the third part will not discuss the experiments, which are discussed in chapter 4. At the end of this chapter, I will also give an introduction to the design and implementation process of the working software, which is used to show the representation result and the stock prediction result.

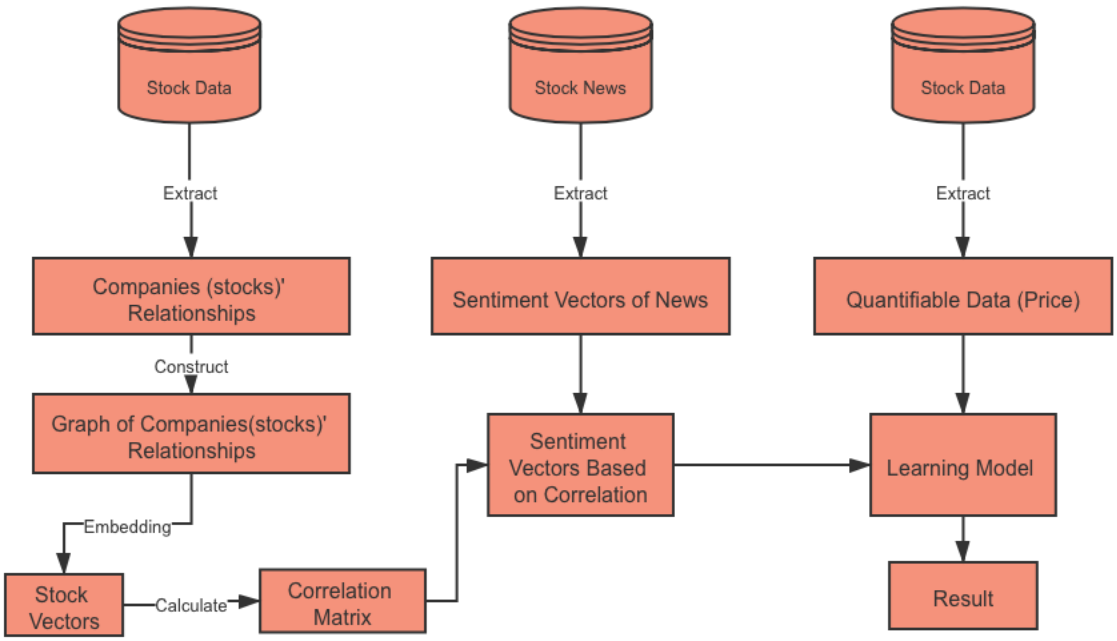


Figure 11 Project framework

3.2 Model Implementation

3.2.1 Stock Relationship Graph Construction Model

As I have said in the 1.2.1, after conducting an initial investigation, I found it is difficult for me to find a suitable companies' relationship graph of the 50 stocks which I choose for my project.

Therefore, after discussing with my supervisor, I decided to design a model to construct the companies' relationship graph by myself, which based on the principles of word2vec. The vertexes in this graph represent different stocks (companies); the edges between every 2 vertexes are set according to the similarity of stock fluctuation of prices. At first, there are only 50 nodes existing in the initial graph, which respectively represent the 50 stocks and there is no edge existing in the graph. Then for every 2 stocks, if the similarity of 2 stocks' price fluctuation exceeds the set threshold value, I will set an edge using this similarity as the weight between the 2 stock nodes. Finally, we can get a stock relationship graph of 50 stocks.

The method I used to get the price fluctuation similarity of each 2 stocks is to use the word2vec algorithm. As I have said in the 2.1.2, word2vec is usually used to represent each word Semantic information with a vector, which shows a better performance than one hot representation especially when you need to calculate the similarity between different words. The principle is that, with a certain window in each sentence, the words show up together frequently in sentences will have a similar embedding result. The more frequently the two words appear together, the closer the cosine similarity of 2 stock nodes' embedding result is to 1.

In the same way, I treat each stock as a unique word. In each day, I will put those stocks in order of price change to one sentence. These sentences are the input of the word2vec, which will be regarded as the input sentences trained by Skip-gram of word2vec. With certain window size in sentences, each stock will show up with highly related stock frequently, because they tend to move their prices together. Therefore, the similarly the stocks' price fluctuate, the more frequently the two stocks appear together in stock sentences, the closer the cosine similarity of 2 words' embedding result is to 1.

Actually, Word2Vec has 2 training models: CBOW(Continuous Bag-of-Words) and Skip-gram, the reason why I choose the Skip-gram model is that Skip-gram is to predict the context based on one input word, which is similar to the project problem condition: predict the close-related stocks based on an input stock.

In this project, the stock price data, which I used to construct the stock relationship graph is in the year 2015 (244 trading days). Using the model, I mentioned above, I generate 244 stock sentences (remove a certain stock with value "Null" in the sentence if it stops trading in that day), as the input of the Skip-Gram model of word2vec to get the embedding result of stocks. Then calculate the cosine similarity to construct the stock relationship graph. The final 50 stock relationship graph is shown in Figure 12.

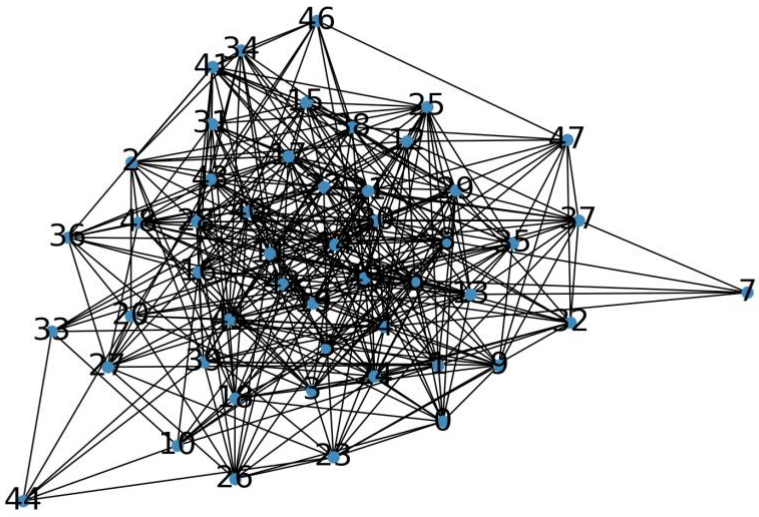


Figure 12 Generated stock relationship graph of 50 stocks

3.2.2 Graph Embedding Model

As I have said in 2.1.3, I decided to use the Node2Vec to implement the graph embedding model, which can embed the stock nodes of Companies' relationship Graph into the stock vectors respectively, and then to calculate the similarity between every 2 stocks and obtain the correlation matrix of all 50 stocks.

The input of this model is a graph edge-list file like the following image Figure 13 (too many edges, so only show the edge of node ‘0’ and node ‘1’): each row is 2 nodes index of the edge and the weight of the edge, like node ‘0’, node ‘1’, edge weight 0.217. The edge-list file is the representation of the graph generated by the Stock Relationship Graph Construction Model in 3.2.1.

```
0 1 0.21719379723072052
0 5 0.2714232802391052
0 6 0.4298326075077057
0 9 0.2532106041908264
0 13 0.2752186954021454
0 15 0.24012543261051178
0 18 0.316873162984848
0 22 0.3498799502849579
0 23 0.4201621115207672
0 24 0.37176772952079773
0 26 0.23880931735038757
0 29 0.2220575213432312
0 45 0.26066869497299194
1 3 0.31941336393356323
1 4 0.2154572755098343
1 6 0.29671818017959595
1 10 0.34046700596809387
1 14 0.4337739050388336
1 22 0.2558542788028717
1 24 0.20863038301467896
1 25 0.24168434739112854
1 27 0.23946022987365723
1 29 0.232477605342865
1 30 0.23528125882148743
1 32 0.37119537591934204
1 37 0.2923225462436676
1 39 0.24328796565532684
1 45 0.21824853122234344
```

Figure 13 Example of an edge list file

The embedding result of the stocks' relationship graph is 50 low-dimensional vectors (vector size: 1 x 30) which represent the 50 stocks. For each stock embedding vector, it contains not only the characteristics of the stock node itself, but also the characteristics of the relationship with other nodes.

In this project, there are 50 stock nodes in the Graph and set the embedding dimension as 30, the embedding layer will be a matrix with size (50, 30). Each row is a vector with size (1, 30). Hence, I can calculate the cosine similarity of each 2 stocks and get the correlation matrix with size (50, 50). In the correlation matrix, each item is the similarity of 2 stocks, such as the item in row 2 column 3 of the correlation matrix represents the similarity of stock 2 and stock 3. Aactually, we have gotten the similarity between every 2 stocks in the 3.2.1 Stock Relationship Graph Construction Model when I am applying the word2vec technique constructing the stock relationship graph. The main reason why I recalculate the similarity between every 2 stocks using the node2vec method is that special information is contained in the graph structure.

The final output of the Graph Embedding Model is the stock correlation matrix, which is the input of the project's final model: Learning Model based on LSTM.

3.2.3 Learning Model Based on LSTM

The input of the Learning Model Based on LSTM is the stock correlation matrix obtained from the 3.2.2 Graph Embedding Model, the quantitative features like prices, and the stocks' qualitative features like news information.

The stocks' news impact for the stock prediction can be positive or negative. In the project's dataset, each trading day there are a lot of news of each stock, all of the news of a specific stock together can give a positive or negative impact for the next day stock prediction. Generally, there are 2 ways to extract the sentiment from the news: one way is to define the positive words bag and the negative words bag, and then calculate the sentiment score based on the occurrences of negative and positive words; another method is to train the emotional analysis neural network model with a large number of labelled corpus, and then use the trained neural network to extract the sentiment from the news. After an investigation, I found the second method shows a better performance than the words bags method. Therefore, I decided to use the trained neural network to extract the sentiment from the stock news. Because the training process of the emotional analysis neural network generally requires extremely large training set with manually labelled labels, and it is difficult for me to get labelled training set like that. I decided to use the NLP-

Python-SDK provided by Baidu-ai, the reason is that it has a well pre-trained corpus model. which can judge the category of emotion polarity (positive, negative, neutral) on the text containing subjective information and give the corresponding confidence degree. Represent positive news with 2, negative news with 0, natural news with 1.

After extracting the sentiment from the stock news, for each stock, we can get stock news sentiment scores for each trading day in the year 2015, 2016, 2017. Therefore, for each stock, the sentiment score of three years can be seen as a quantifiable time-series data like the price data.

Before we set the above news and quantifiable price series data as the input of the LSTM to train each stock's learning model. We need to combine the stock correlation matrix with the stock sentiment data. The reason is as I have mentioned in 1.1: "Microsoft successfully completed the acquisition of company xxx" Apart from the stock of Microsoft will rising, companies having certain relationships with Microsoft such as corporation supply and the customer will also be affected by the news, whose stock will also be likely to rise. Therefore, it is necessary to consider the companies relationships in terms of measuring the impact of the news. When training the learning model of stock "000001", besides the stock news related to the "000001", we can make full use of the news of the other stocks how has a closed relationship with the stock '000001". That's what correlation matrices do.

For the *x*-th stock in 50 stocks, this project denotes these symbols shown in Table 4.

Table 4 symbols of *x*-th stock

For the *x*-th stock in 50 stocks	
news on *i*-th day	$news_i^x$
the news sentiment on *i*-th day	$sentiment_i^x = f(news_i^x)$
the sentiment time-series data in n continuous days	$S = \{sentiment_1^x, sentiment_2^x, sentiment_3^x, \dots, sentiment_{n-1}^x, sentiment_n^x\}$
similarity between *x*-th stock and *y*-th stock (according to correlation matrix)	$S_{x,y}$

Before considering the correlation matrix, the sentiment time-series data of *x*-th stock as is shown as equation (8). When considering the correlation matrix, we need to recalculate the sentiment score. When recalculating the sentiment of the *x*-th stock on *i*-th day, the project

uses the equation (9). Then, the correlation news sentiment time-series of *x*-th stock in n days can be represented as the equation (10) which is used as part of the input to the LSTM.

$$S = \{sentiment_1^x, sentiment_2^x, \dots, sentiment_{n-1}^x, sentiment_n^x\} \quad \textbf{(8)}$$

$$\widetilde{sentiment}_i^x = \sum_{y=0}^{n} sentiment_i^x * similarity_{x,y} \quad \textbf{(9)}$$

$$\tilde{S} = \{\widetilde{sentiment}_1^x, \widetilde{sentiment}_2^x, \dots, \widetilde{sentiment}_{n-1}^x, \widetilde{sentiment}_n^x\} \quad \textbf{(10)}$$

For the *x*-th stock in 50 stocks, the project denotes the quantifiable price data on i-th day as equation (11). The quantifiable price time-series data of the *x*-th stock in n days can be represented as equation (12).

When training the learning model, the final input of the LSTM learning model is the combination of the correlation news sentiment time-series as equation (10) and the quantifiable price time-series data Q as equation (12). The combined time-series can be denoted as equation (13). We will generate the training set based on the combined time-series.

$$Q_i^x = \{q_i^{x_1}, q_i^{x_2}, \dots, q_i^{x_n}\} \quad \textbf{(11)}$$

($q_i^{x_n}$ represents the n-th specific Quantitative indicator like opening price)

$$Q = \{Q_1^x, Q_2^x, Q_3^x, \dots, Q_n^x\} \quad \textbf{(12)}$$

$$\text{LSTM}_{\text{input}} = \{(\widetilde{sentiment}_1^x, Q_1^x), (\widetilde{sentiment}_2^x, Q_2^x), \dots, (\widetilde{sentiment}_n^x, Q_n^x)\} \quad \textbf{(13)}$$

In the LSTM learning model, the length of time sequences is finally set as 4. The experiments and reason why I choose 4 will be discussed and shown in chapter4. It means to predict 5th day from the previous 4 days combined time-series. When training the LSTM model, for every (x,y) pair. The input x is the combined time-series of 4 days, the y is the target (0 indicate the stock decrease on 5th day, 1 indicate the stock increase on the 5th day).

3.3 Working Software

As the final task of this project, it is to design and implement a working software, which can visualize representations and report stock prediction results.

The main window has 2 buttons as shown in Figure 14, which can redirect the user to the window to display Representation of Stock Graph or redirect the user to the window to show

the stock prediction result.

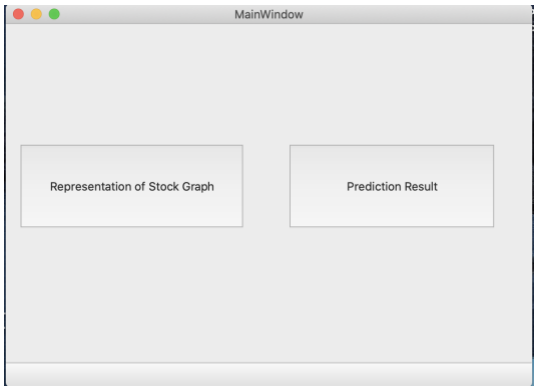


Figure 14 Main window of the working software

3.3.1 Representation of Stock Graph

This is the window to display the Representation of Stock Graph as shown in Figure 15, you can input “all” to display the entire graph which includes all nodes, all edges, however which looks messy.

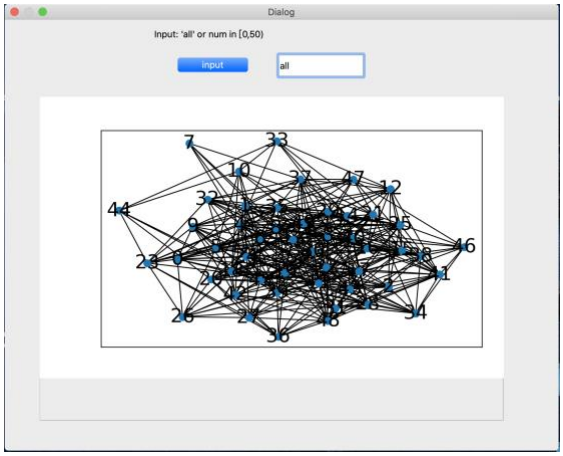


Figure 15 Window to display the Representation of Stock Graph (overall graph)

Instead, you can also choose to display the specific node and other nodes who connect with the specific node, which looks more clearly as shown in Figure 16.

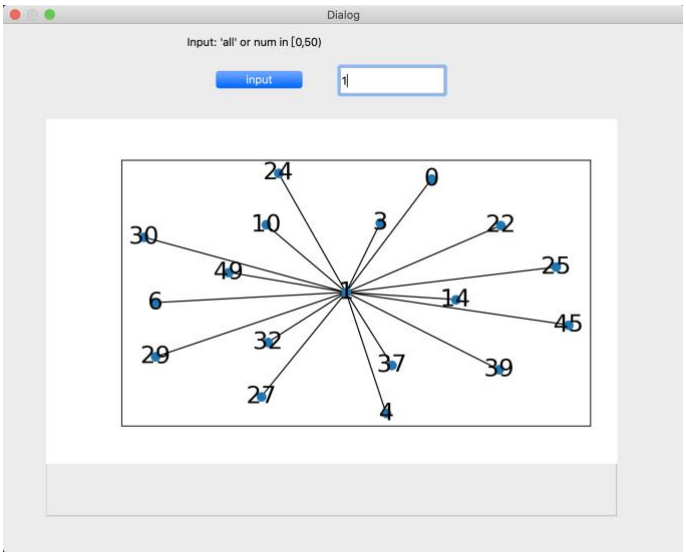


Figure 16 Window to display the Representation of Stock Graph (choose node 1)

3.3.2 Stock Prediction Results.

This is the window to show the prediction result as shown in Figure 17. You can click the pushbutton to input the stock index (0 ~ 49), and then the window will show the prediction result of the specific stock. The prediction result is based on the pre-trained LSTM learning model which I mentioned in 3.2.3 and the testing set. The prediction result includes the target and the prediction (‘0’ represents stock rose on that day, “1” represents the stock go down on that day). For each day, if the target and the prediction are both ‘0’ or both ‘1’, the prediction result on that day is true; if the target and the prediction is: “ target: ‘0’ prediction: ‘1’ ” or “target: ‘1’ prediction: ‘0’ ” , the prediction on that day is false. The accuracy is the ratio of the number of true days contrasts the total days.

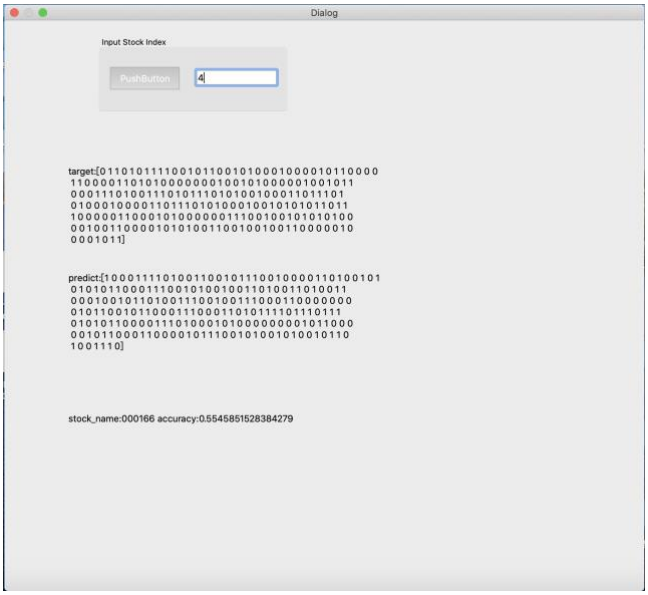


Figure 17 Window to show the prediction result (choose stock 4)

Chapter 4: Results and Discussion

In order to verify the proposed idea: improving the stock prediction with representation learning and to evaluate the prediction performance of the ultimate implemented model. The project has divided the dataset into the training set and the testing set. For each stock, a unique LSTM learning model has been trained with the training set, then test the pre-trained model with the testing set. In this chapter, I will give a clear explanation of the dataset, how the experiment setup and the results of the experiment.

4.1 Dataset

Initially, the entire dataset includes quantifiable data like prices and other indicators, the news of 50 different stocks in the year 2015, 2016, 2017. Each year has 244 trading days, however, some stocks stop trading on specific days, if a stock has 10 ten days stop trading in a year, then the stocks only have 234 real trading days and related trading data. For each stock, we use the data in 2015 and 2016 as the training set, use the data in 2017 as the testing set.

4.2 Experiment setup

In order to verify the proposed improving the stock prediction with representation learning and to evaluate the prediction performance of the ultimate implemented model, I set 3 groups of experiments as shown in Table 5.

Table 5 The introduction of 3 groups of experiments

Group1 (Quantifiable)	give prediction only depending on the quantifiable data like stock prices.
Group2 (Quantifiable + News)	give prediction depending on the quantifiable data like stock prices and the news which reflect the sentiment
Group3 (Quantifiable + News +Stock Relationships)	give prediction depending on the quantifiable data like stock prices, the news which reflects the sentiment and the stock relationships.

According to the principles of controlling variables. We can use group 1 and group 2 to assess and evaluate our model and analyse how incorporating the news into the model help improve the stock prediction performance. In the similar way, we can use group 2 and group 3 to assess

and evaluate our model and analyse how to incorporate the relevance of the stock into the model help improve the stock prediction performance. Each group has 50 different stocks, each stock has its own learning model, so I use the average accuracy (ACC) and average F1-score of 50 stocks as the evaluation metrics. I also set experiments with varied n from 2 to 7 to find the suitable value of the hyperparameter n. Hyperparameter n refers to the historical days as I have explained in 3.2.3. It means we use previous n days to predict n+1-th day.

4.3 Results of Experiment

The following Table 6 shows the experiment's results of three groups. Each group shows the average accuracy (ACC) and average F-1 score of 50 stocks:

Table 6 experiment results (ACC & F-1) of three groups, with varied n

	n=2		n=3		n=4		n=5		n=6		n=7	
	ACC	F-1	ACC	F-1	ACC	F-1	ACC	F-1	ACC	F-1	ACC	F-1
Group 1	0.5209	0.4807	0.5175	0.4891	0.5213	0.4958	0.5203	0.4980	0.5209	0.4883	0.5180	0.4842
Group 2	0.5297	0.5180	0.5313	0.5115	0.5331	0.5123	0.5351	0.5194	0.5402	0.5217	0.5331	0.5170
Group 3	**0.5473**	**0.5438**	**0.5514**	**0.5441**	**0.5547**	**0.5445**	**0.5532**	**0.5432**	**0.5443**	**0.5406**	**0.5386**	**0.5366**

Compared to Group 1 which gives prediction only depending on the quantifiable data like stock prices, Group 2 improves both the accuracy and F-1 score. Such gains mainly come from incorporating each stocks' own qualitative descriptions like news instead of only quantifiable price data. It verifies the idea that incorporating the news into the model can help improve the stock prediction performance.

Compared to Group 2 which gives prediction depending on each stock's own quantifiable data like price data and qualitative descriptions like news, Group 3 improves both the accuracy and F-1 score. Such gains mainly derive from Group 3 incorporating the relevance of the stock between 50 stocks when considering the impacts of stock news. It verifies the ides that incorporating the stocks relevance between different stocks when considering the impact of each stocks' news data, can help improve the stock prediction performance.

Within Group 3, we can see the prediction result includes ACC and F-1 varied with different values of hyperparameter n which refer to the number of historical days as mentioned in 3.2.3. When the values varied from 2 to 7, the ACC and F-1 score firstly increase and then decrease. The time when the model shows the best performance of ACC and F-1 score is set n with 4. One possible reason is that the impact of both the historical price data and the qualitative

description like stocks news gradually decay, as time goes by.

According to the result, we can see the prediction result of Group 3 is better than the other two groups. It can verify the idea of improving the stock prediction with the representation learning is correct. Besides the quantifiable data, incorporating the news and the stock relationship into the model can actually help improve the stock prediction performance.

Chapter 5: Conclusion and Further Work

5.1 Conclusion

In this project, I use the Representation Learning to improve the stock prediction task. Besides the quantifiable data like stock prices, I incorporate the stocks relationships and the news data of the stock into the stock prediction. At the end of the project, I also design and implement a working software that can visualize the representation and report the predicting results. The dataset includes the stock news and quantifiable data of 50 different Chinese stocks in the year 2015, 2016, 2017, which are used to generate the stock relationship graph and divided into a training set and testing set for the LSTM learning model. Because of a problem that it is difficult for me to find a ready-made stock relationship graph, I design and implement a Stock Graph Construction Model based on the word2vec principles, which can generate the stock relationship graphs from the similarity of different stocks' fluctuation. Next, using the node2vec algorithm, I design and implement the Graph Embedding Model which can embed the stock relationship graphs into low-dimensional vectors. Each vector represents a node in the graph, each vector not only includes its own features, but also includes the relationship with other neighborhood nodes. From these embedding vectors, then get the stock correlation matrix by calculating the similarity between every 2 stocks of the 50 stocks. After getting the stock the correlation matrix, I design and implement a Learning Model Based on LSTM. Then, use the stock correlation matrix, stock news data and stock quantifiable data like prices to train the learning model of each stock. I also design and conduct three groups of experiments to verify the implemented model, using the testing set. From the result of the experiments, we can conclude, using Representation Learning, our overall model actually helps improve the stock prediction performance.

5.2 Further work

Critically evaluating my project, there are also some problems and weakness which need to be solved and improved in the future or can be avoided if I could do the project again. One aspect is that although I construct the stock relationship graph based on the similarity of different stocks price fluctuation, it seems not enough to improve the stock prediction performance because it cannot entirely reflect the stocks' relationship or companies' relationship. So, I can try exploring other methods to construct the stocks' relationship or finding a well-ready make graph to improve the stock prediction performance in our model. Moreover, I can try using the bigger dataset to train and test our model to improve and verify the performance of our model.

Apart from that, if I have extra time to extend the project or task, the future work can concentrate on designing and implementing a real-time prediction software which incorporates web crawler. It can give a prediction for the next day based on our pre-trained LSTM learning model, the correlation matrix and the quantifiable data and news data which are used web crawler software to crawl from the Internet every day.

References

[1]. Y. C. Wei, Y. C. Lu, J. N. Chen, Y. J. Hsu. (2017). Informativeness of the market news sentiment in the taiwan stock market. North American Journal of Economics & Finance, S106294081630122X.

[2]. Bengio, Y., Courville, A. and Vincent, P. (2013) Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35, 1798-1828. http://dx.doi.org/10.1109/TPAMI.2013.50.

[3]. Ng, A. (2013) Machine Learning and AI via Brain Simulations. Stanford University.

[4]. T. Mikolov, K. Chen, G. Corrado, Jeffrey Dean. (2013). Efficient Estimation of Word Representations in Vector Space. Google.

[5]. T. Mikolov, Q. Le. (2014). Distributed Representations of Sentences and Documents. Google.

[6]. P. Bojanowski, E. Grave, A. Joulin, T. Mikolov. (2016) Enriching Word Vectors with Subword Information. Facebook AI Research.

[7]. A. Grover, J. Leskovec. (2016). node2vec: Scalable Feature Learning for Networks. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016. Stanford University.

[8]. S. Hochreiter, J. Schmidhuber. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[9]. X. Zhang, Y. X. Li, Wang, S. Z. Wang, B.X. Fang, P. S. Yu. (2018). Enhancing Stock Market Prediction with Extended Coupled Hidden Markov Model over Multi-Sourced Data. Knowledge & Information Systems. Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing University of Posts and Telecommunications.

[10]. X. Zhang, S.Y. Qu, J.Y. Huang, B.X. Fang, P. Yu. (2018). Stock Market Prediction via Multi-Source Multiple Instance Learning. IEEE. Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing University of Posts and Telecommunications.

[11]. X. Zhang, Y. J. Zhang, S. Z. Wang, Y. T. Yao, B.X. Fang, P. S. Yu. (2018). Improving Stock Market Prediction via Heterogeneous Information Fusion. IEEE. Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing University of Posts and Telecommunications.

[12]. J. Liu, Z.C. Lu, W. Du. (2019). Combining Enterprise Knowledge Graph and News Sentiment Analysis for Stock Price Volatility Prediction. Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019. School of Information, Renmin University of China.

Acknowledgement

Finally, I would like to thank and appreciate all the professor and senior who have helped me a lot in this project especially my supervisor. He has given me a lot of critical guidance includes the stock datasets and the reference paper initially; and help me a lot during the process of the design and implementation of the project; and the process of finishing the final report and preparing for the final presentation.

Appendix

北京邮电大学 本科毕业设计（论文）任务书

Project Specification Form

Part 1 – Supervisor

论文题目 Project Title	Improving the Stock Market Prediction with Representation Learning		
题目分类 Scope	Data Science and Artificial Intelligence	Research	Software
主要内容 Project description	Representation learning has shown its superior performance in many fields such as natural language processing and image processing. However, it still lacks of research works that attempt to apply this technique to stock market prediction. Stock market prediction is a popular research topic in data mining, and it involves various types of data, including quantitative trading data and qualitative descriptions such as news and reports. How to fuse these data and learn a useful representation is still unexplored. In this project, it is required design and implement a model to fuse trading data and Web data into a unified framework, and learn useful representations that can be used for stock price fluctuation predictions. Moreover, a software that can visualize the representations is required.		
关键词 Keywords	Stock market, representation learning		
主要任务 Main tasks	**1** Data preprocessing and feature extraction		
	2 Representation learning model building		
	3 Learning method design and implementation		
	4 Software with interface that report the results and visualize the representations		
主要成果 Measurable outcomes	**1** Representations that encode the stock-related information		
	2 Working software that can visualize the representations		
	3 The prediction algorithm and the prediction results		

北京邮电大学 本科毕业设计（论文）任务书

Project Specification Form

Part 2 - Student

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	丁 Ding	**名 First Name**	凡 Fan		
BUPT 学号 BUPT number	2016213516	**QM 学号 QM number**	161188243	**班级 Class**	2016215119
论文题目 Project Title	Improving the Stock Market Prediction with Representation Learning				
论文概述 Project outline **Write about 500-800 words** **Please refer to Project Student Handbook section 3.2**	1. Project description and analysis of user requirements Currently, techniques of Data Mining play an important role in financial field. Prediction of stock price is one of the significant tasks in terms of data mining. Being able to accurately predict the trend of stocks has always been the desire of every investor, most investors and a number of researchers have tried different forms to analyse their trends in order to figure out the problem, nevertheless, it is difficult for people to give prediction precisely. After talking with a graduate student in financial major in respect of that problem, I learned that based on nowadays technology, qualitative analysis is more reasonable and feasible than Quantitative analysis that is very challenging. Also, there is still a lot of room for improving the performance of stock prediction. Precisely, it is more appropriate to understand stock forecasting as a time series problem-based on the historical data, predicting the future price. Since deep learning methods developing by leaps and bounds, RNN is the main technique, which researchers used to try figuring out the problem of stock prediction. This technique mainly depends on the quantifiable data like the historic stock price and indicators of quantifiable features such as Price to Book Ratio (PB), Price Earnings Ratio (PE), and stuff like that. However, stock prediction only depends on the information that confine to the quantifiable data is not enough. On one hand, because its decisions and fluctuations are subject to a variety of economic and political factors, as well as investment psychology, it is more suitable for us to incorporate the unquantifiable data like news and comments of investors into the task of stock prediction, which can reflects a significant positive or negative sentiments contributing to improving the prediction performance. On the other hand, it is beneficial to consider the relevance among different stocks. For instance, "Microsoft successfully completed the acquisition of company xxx" Apart from the stock of Microsoft will rising, companies having certain relationships with Microsoft such as corporation supply and customer will also be affected by the news, whose stock will also rise. Therefore, is necessary to consider the companies relationships in terms of measuring the impact of a news. Obviously, unlike the quantifiable data like stock price which is easier for us using float number to represent, the sentiment of the news and the relation among different stocks need special techniques related to represent learning. Specifically, for one thing we need represent learning techniques embedding the				

news into a score or a vector to represent the sentiment, for another we need represent learning techniques like DeepWalk or Node2Vec embedding the graph of companies' relationships into vector or matrix, then obtain the correlation between each two stocks. Finally, using these two kinds of information build a model and improve the stock prediction performance with the quantifiable data based RNN methods.

2. Data collection

My research needs three different types of data:

- The data used to represent the quantifiable features of stocks.
- The data used to extract the sentiment features of stocks.
- The data used to extract the relationships among stocks.

These three kinds of data are based on the dataset which my project supervisor professor Xi Zhang provides for me. Also, I need to adjust these data sets and combine the datasets from online resources in order to meet my research project.

3. The methodologies algorithms and other techniques to be employed

- Methodologies of this project
  a) Investigate
     Investigate related papers to collect algorithms and models. At the same time search, collect and construct the data sets.
  b) Study
     Study the theory of methods which are popular used in papers.
  c) Design/Improve
     Design/Improve a model with representation learning methods to improve the stock prediction based on RNN.
  d) Experiments
     Improve the performance of my model and prove the project hypothesis.
  e) Conclusion

- Algorithms and other techniques
  a) NLP and representation learning algorithms and techniques to extract the sentiment features of news such as sentiment score or vectors.
  b) NLP and representation learning algorithms and techniques to extract the relationships of stocks like correlation matrix from the raw data.
  c) Deep learning methods like RNN to build the learning model.

4. Experiments.

In order to test the effect of using the representation learning techniques to improve the stock prediction performance, I am going to some experiment to prove the project hypotheses. Specifically, we designed several groups of comparative experiments.

- Group 1 (Quantifiable): give prediction only depends on the quantifiable data like stock prices.
- Group 2 (Quantifiable + News): give prediction depends on the quantifiable data like stock prices and the news which reflect the sentiment.
- Group 3 (Quantifiable + News +Stock Relationships): give prediction depends on the quantifiable data like stock prices; the news which reflect the sentiment and the companies or stock relationships.

	5. Tools （Programming language / database/ software package to be used） Programming language: Python Deep learning platform: TensorFlow/pytorch Operation System: Linux/Windows 6. Background Material/References [1]. Zhang, X., Li, Y., Wang, S. et al. Knowl Inf Syst (2019) 61: 1071. https://doi.org/10.1007/s10115-018-1315-6 [2]. Zhang, Xi et al. “Improving Stock Market Prediction via Heterogeneous Information Fusion.” Knowledge-Based Systems 143 (2018): 236–247. Crossref. Web. [3]. Zhang, Xi & Qu, Siyu & Huang, Jieyun & Fang, Binxing & Yu, Philip. (2018). Stock Market Prediction via Multi-Source Multiple Instance Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2869735. [4]. https://scholarspace.manoa.hawaii.edu/handle/10125/59565
道德规范 **Ethics**	Please confirm that you have discussed ethical issues with your Supervisor using the ethics checklist (Project Handbook Appendix 2). [YES/NO]

3

	Summary of ethical issues: (put N/A if not applicable) 1. Will the participants be exposed to any risks greater than those encountered in their normal working life? N/A 2. Will the participants be using any non-standard hardware? N/A 3. How will participants voluntarily give consent? N/A 4. Are you offering any incentive to the participants? N/A 5. Is there any intentional deception of the participants? N/A 6. Are any of your participants under the age of 16? N/A 7. Do any of your participants have an impairment that will limit their understanding or communication? N/A 8. Are you in a position of authority or influence over any of your participants? N/A 9. Will the participants be informed that they could withdraw at any time? N/A 10. Will the participants be informed of your contact details? N/A 11. Will the participants be debriefed? N/A 12. Will the data collected from participants be stored in an anonymous form? N/A

中期目标 **Mid-term target.** **It must be tangible outcomes, E.g. software, hardware or simulation.** **It will be assessed at the mid-term oral.**	1. Search, collect and construct the data sets, especially the data used to construct the graph of companies' relationships. 2. Design algorithm based on the representation learning which can successfully embedding the graph of companies' relationships into vector to generate the stock vector. 3. Using the stock vector embedding from the graph of companies' relationships obtain the similarity between the stocks and construct the correlation matrix of stocks.

Work Plan (Gantt Chart)

Fill in the sub-tasks and insert a letter X in the cells to show the extent of each task

	Nov 1-15	Nov 16-30	Dec 1-15	Dec 16-31	Jan 1-15	Jan 16-31	Feb 1-15	Feb 16-29	Mar 1-15	Mar 16-31	Apr 1-15	Apr 16-30
Task 1 [Data pre-processing and feature extraction]]												
collect and construct the data sets used to represent the quantifiable features of stocks		X	X	X	X							
collect and construct the data sets used to extract the sentiment features of stocks		X	X	X	X							
collect and construct the data sets used to extract the relationships of stocks		X	X	X	X							
Task 2 [Representation learning model building]												
construct the graph used to represent the relationship among companies			X	X	X	X	X					
identify and design the specific represent learning method used to embedding the graph into vector to represent each stock			X	X	X	X	X					
calculate similarity between the stocks based on the stock vector and obtain the correlation matrix						X	X	X				
identify and design the specific technique used to extract the sentiment feature from the news			X	X	X	X	X	X	X			
Task 3 [Learning method design and implementation]												
identify the specific RNN model used to construct the learning model					X	X	X	X				
design the learning method and implement					X	X	X	X	X	X		
Task 4 [Replace this line with the task 4 from the Spec part 1]												
design a software which user can choose specific stock to give a prediction based on data								X	X	X	X	

6

北京邮电大学 本科毕业设计（论文）初期进度报告

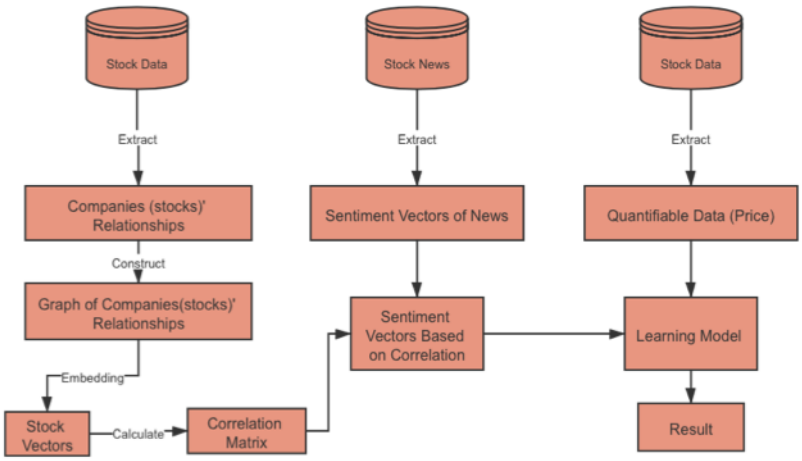
Project Early-term Progress Report

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	丁 Ding	**名 First Name**	凡 Fan		
BUPT 学号 BUPT number	2016213516	**QM 学号 QM number**	161188243	**班级 Class**	2016215119
论文题目 Project Title	Improving the Stock Market Prediction with Representation Learning				

已完成工作 Finished work:

Announcement:

There are several parts of the project, which are shown in the following Graph (A Preliminary Sketch), I announce this for easy understanding of my early-term report.



- **Summary of the material was read and researched**
  - **Papers**

1. I have read relevant literature and papers that focus on stock prediction issues. In short summary, the previous works mainly handled the stock prediction works as a time series problem-based on historical prices and other quantifiable data like that. The main technique used to handle that problem is mainly designed and implemented on the Recurrent Neural Network (RNN). Therefore, in this project, I will use the Representation Learning method combined with the RNN to improve the stock prediction.
2. I have read relevant literature and papers about the Word2Vec. Because, on one hand, one of the possible choices for me to embed the Companies' relationship Graph into the stock vectors is the Node2Vec method, which is mainly inspired by and derived from the Word2Vec method. On the other hand, I have decided the way to extract the stock relations to construct the companies' relationship graph from the stock data, in which I attempt to use the Skip-gram (from Word2Vec) as the method to obtain the relation extent between stocks to construct the companies' relationship graph. The main idea is that each day, I will put those stocks in order of price change to one sentence, which will be regarded as the input sentences trained by Skip-gram. With certain window size in sentences, each stock will show up with highly related stock frequently, because they tend to move their prices together. After training complete, the hidden layer will be used to

calculate the relation extent between stocks to construct the companies' graph. Actually, Word2Vec has 2 training model: CBOW(Continuous Bag-of-Words) and Skip-gram, the reason why I choose the Skip-gram model is that Skip-gram is to predict the context based on one input word, which is similar to the project problem condition: predict the close-related stocks based on a input stock.

3. I have read relevant literature and papers about the Node2Vec and Deepwalk. The reason is that both of the methods are possible choices for me to embed the graph relationships into stock vectors. Apart from that I also read the literature and papers about GCN, the reason is that it is also a possible choice for me to embed the graph relationships into vectors. What's more, some blogs give the conclusion that GCN may have a better performance than Node2Vec and DeepWalk methods.

- **Deep Learning and Machine Learning resource**

Doing this project, I need to have a clear understanding of the fundamental knowledge related to Deep Learning and Machine Learning. Hence, I have watched the online course deeplearning.ai which is taught by Andrew Ng (https://mooc.study.163.com/smartSpec/detail/1001319001.htm). Currently, I have become familiar with the key concepts of Deep Learning and Machine Learning, which is helpful for me to read and learn the papers related to my project.

- **Finished work**

1. Data Pre-processing

Currently, before constructing the overall dataset which contains the daily News and Quantifiable data for each stock in the dataset. I have first constructed a small dataset from online resources which only contains the data used to construct the companies' graph. The reason is that it is easy for me to test my code during the process of designing and implementing the Companies' relationships and embed that graph into stock vectors to calculate the correlation matrix. After initial implementation and get the result of the correlation matrix, I will finish the task of constructing the overall datasets includes the daily news of stocks used for training the learning model for prediction.

2. Design for Graph construct model

I have decided the way to construct the Companies(stocks)' relationship Graph from the online data. I will calculate the similarity score which indicates the relationship between 2 Stocks. If the score exceeds the threshold, I will set an edge between the 2 stock nodes on the graph representing the 2 stocks have a close relationship. For calculating the similarity score I will use the Skip-gram from Word2Vec. In each day, I will put those stocks in order of price change to one sentence, which will be regarded as the input sentences trained by Skip-gram. With certain window size in sentences, each stock will show up with highly related stock frequently, because they tend to move their prices together.

3. **Practice of running the sample code**

I have run some sample codes for practice, which includes Word2Vec training model; GCN model to issue the "Zachary Karate Club" problem; the homework and small project of the online course deeplearning.ai.

- **Problems were faced**

1. Companies relationship' graph

Initially, I hope to find and use the ready-make companies' relationship graph from the online resources (such as https://www.tianyancha.com, etc.), which are clear and reliable. However, there is no suitable companies' relationship graph suitable for this project. Those companies' relationship graphs mainly focus on some other relationships such as the Tenure relationship,

which mainly reflects the interest relationship between individuals and companies. There is limited information about the clear interest relationship between different companies.

- **Solutions were found**

  1. Companies relationship' graph

  To continue the project on track, instead of using the ready-make companies' relationship graph, I decide to design a model to generate the companies' relationship graph by myself. The vertexes in this graph represent different stocks (companies); the edges between each 2 vertexes is set according to the similarity of stocks fluctuation of prices.

是否符合进度？On schedule as per GANTT chart?

[YES/NO]

YES

下一步 Next steps:

1. Implements the companies' relationship' graph construction model.
2. Choose a suitable method to embed the companies' relationship' graph into stock vectors.
3. Obtain the correlation matrix from the embedding vectors.

3

北京邮电大学 本科毕业设计（论文）中期进度报告

Project Mid-term Progress Report

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	丁 Ding	**名 First Name**	凡 Fan		
BUPT 学号 BUPT number	2016213516	**QM 学号 QM number**	161188243	**班级 Class**	2016215119
论文题目 Project Title	Improving the Stock Market Prediction with Representation Learning				

是否完成任务书中所定的中期目标？Targets met (as set in the Specification)?

[YES/NO]

YES. I have completed the targets, which are set in the specification:

- Search, collect and construct the data sets, especially the data used to construct the graph of companies' relationships.
- Design algorithm based on the representation learning which can successfully embedding the graph of companies' relationships into vector to generate the stock vector.
- Using the stock vector embedding from the graph of companies' relationships obtain the similarity between the stocks and construct the correlation matrix of stocks.

已完成工作 Finished work:

1. Required knowledge of Python and Deep Learning

As I mentioned in the Early Term Report, in order to finish the tasks of my project, I need to have a clear understanding of the fundamental knowledge related to Deep Learning and Machine Learning. Therefore, I have watched the online course deeplearning.ai which is taught by Andrew Ng before submitting my early term report.

With the progress of my graduation project, I have learned more new things since my early term report, which I think is the required knowledge to complete my project.

Firstly, for the sake of programming language's unification, I decided to use python to pre-process the data too. After a simple investigation, I decided to use Pandas, which is a high-performance and easy to use data analysis tool for the Python programming language. Hence, I learned the required knowledge of Pandas for data pre-processing according to the official documents and tutorials of Pandas.

Secondly, after comparing Tensorflow and Pytorch by reading official documents and tutorials, I decided to choose the Pytorch as the main learn deep learning framework, which I think is more convenient.

2. Researching on papers

As I mentioned in the Early Term Report, I have read the relevant literature and papers include Word2Vec, Node2Vec. Since the Early Term Report, I read more related blogs of the internet about those, which are other people's interpretations of these papers and study notes, helping me for better understanding.

3. Data Pre-processing

Currently, based on the dataset (price data and news data of 100 stocks in 2016 and 2017) which are provided by my supervisor, I also collected some stock price data from the internet by myself. The main reason is that when I am preprocessing the price data of the stock, which are used to construct the stock relationship Graph, I found some stocks have a different number of

trading days. For example, in the dataset which my supervisor provided for me, the stock "000001" has related data of 243 trading dates in 2016; however, the stock "000002" only has related data of 121 trading dates in 2016. At first, I thought there may be something wrong with this dataset and I decided to collect some extra data to compensate the dataset, which is missing in the original dataset, from the internet. But later when I have collected the related data from the internet and downloaded them, I found the truth is that some stocks stopped trading on certain dates during a year. The number of trading dates, of each stock in the dataset which my supervisor provides for me is right. The main difference between the original dataset and the data of each stock I collect from the internet is that, the original dataset deletes the row of the dates when the stock stopped trading.

I have decided to use 50 stocks in my project. When pre-processing the data for constructing the stocks' relationship graph, I choose Pandas which is a NumPy-based tool created to address data analysis tasks. The quantifiable data (like open price close price) of each stock, of each year are in different ".csv" files. Based on these files, I extract the price change of each stock to make a new file in ".csv" type, each row represents a stock, each column represents the price change of stocks in certain dates. If a stock stopped trading in a certain date, the item is replaced by the value "Null" in the file.

4. Design and implement the Graph construction model in Python

As I have said in the Early Term Report, I have read relevant literature and papers about the Word2Vec. The reason is that I have decided the way to extract the stock relations to construct the companies' relationship graph from the stock data, in which I attempt to use the Skip-gram (from Word2Vec) as the method to obtain the relation extent between stocks to construct the companies' relationship graph.

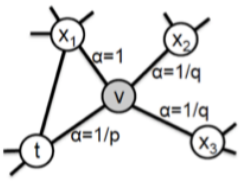
The input of the model is the ".csv" file I mentioned above, each row represents a stock, each column represents the price change of stocks in certain dates.The main idea is that each day, I will put those stocks from high to low in the order of price change to one sentence (and remove a certain stock with value "Null" if it stops trading in that day), which will be regarded as the input sentences trained by Skip-gram. With certain window size in sentences, each stock will show up with highly related stock frequently, because they tend to move their prices together. After training complete, the hidden layer will be used to calculate the relation extent between stocks to construct the companies' graph.

Actually, Word2Vec has 2 training model: CBOW(Continuous Bag-of-Words) and Skip-gram, the reason why I choose the Skip-gram model is that Skip-gram is to predict the context based on one input word, which is similar to the project problem condition: predict the close-related stocks based on an input stock.

5. Design and implement the Graph embedding model in Python

As I have said in the Early Term Report, I have read relevant literature and papers about the Node2Vec and Deepwalk. Finally, I decided to use the Node2Vec to implement the embedding model, which can embed the stock nodes of Companies' relationship Graph into the stock vectors to calculate the similarity between every 2 stocks and obtain the correlation matrix of all 50 stocks.

The Node2Vec method is mainly inspired by and derived from the Word2Vec method. The difference is that Node2vec treat biased random walk on networks as sentences, which means learning node representations with the technique for learning word representations. Node2Vec finds the node context with a hybrid strategy of Breadth-first Sampling (BFS): homophily; Depth-first Sampling (DFS): structural equivalence.



As we can see in the above figure, the biased random walk of Node2Vec has two parameters p and q:

- p: controls the probability of revisiting a node in the walk
- q: controls the probability of exploring "outward" nodes

6. Obtain the correlation matrix from the embedding vectors

Based on the embedding result, which gets from the Graph embedding model, I can represent each stock with a vector. For example, if there are 50 stock nodes in the Graph and set the embedding dimension as 30, the embedding layer will be a matrix with size (50,30). Each row is a vector with length 30. Hence, I can calculate the cosine similarity of each 2 stocks and get a correlation matrix with size (50, 50).

尚需完成的任务 Work to do:

1. Identify and design the specific technique used to extract the sentiment feature from the news
2. Identify the specific RNN model used to construct the learning model
3. Design the learning method and implement
4. Design a software which user can choose specific stock to give a prediction based on data

存在问题 Problems:

Currently, it's hard to decide how to set the appropriate value for the hyperparameters of Graph construction model and Graph embedding model to achieve the optimal result.

拟采取的办法 Solutions:

In order to set the appropriate value for the hyperparameters of Graph construction model and Graph embedding model, I'll have to wait to finish the design and implementation of the learning model. Then, by setting different hyperparameters on many experiments to get the appropriate value of hyperparameters.

论文结构 Structure of the final report:

1. **Abstract.**

   This is a short overview of my whole report.

2. **Introduction.**

   In this part, the following questions should be answered:

   - What is the point of the project?
   - What did you try to do and why?
   - What have you achieved?
   - How is the report structured?

3. **Background knowledge.**

   Including introduction of the dataset, relevant information that explains the background context of my project to the reader.

4. **Design and implementation.**

   The whole process of my project

5. **Result and Analysis.**

   The results of experiments presented in a clear way so that the reader may interpret the information easily.

6. **Conclusion and further work.**

   In this part, I should answer the following questions:

   - What did I try to do?
   - What did I actually achieve?
   - What would I do differently if I could do the project again?
   - What else would I do if I had more time given to me to complete the project?

7. **Acknowledgements**
8. **References**
9. **Appendices (if need)**

Reference:

[1]. Zhang, X., Li, Y., Wang, S. et al. Knowl Inf Syst (2019) 61: 1071. https://doi.org/10.1007/s10115-018-1315-6
[2]. Zhang, Xi et al. "Improving Stock Market Prediction via Heterogeneous Information Fusion." Knowledge-Based Systems 143 (2018): 236–247. Crossref. Web.
[3]. Zhang, Xi & Qu, Siyu & Huang, Jieyun & Fang, Binxing & Yu, Philip. (2018). Stock Market Prediction via Multi-Source Multiple Instance Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2869735.
[4]. https://scholarspace.manoa.hawaii.edu/handle/10125/59565
[5]. node2vec: Scalable Feature Learning for Networks. A. Grover, J. Leskovec. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016.

北京邮电大学 本科毕业设计（论文）教师指导记录表

Project Supervision Log

学院 School	International School	**专业 Programme**	**Internet of Things Engineering**		
姓 Family name	丁 Ding	**名 First Name**	凡 Fan		
BUPT 学号 BUPT number	2016213516	**QM 学号 QM number**	161188243	**班级 Class**	2016215119
论文题目 Project Title	Improving the Stock Market Prediction with Representation Learning				

Please record supervision log using the format below:

Date: dd-mm-yyyy
Supervision type: face-to-face meeting/online meeting/email/other (please specify)
Summary:

Date: 16-10-2019
Supervision type: face-to-face meeting
Summary: discussed the project and the required knowledge which I need to learn

Date: 06-11-2019
Supervision type: face-to-face meeting
Summary: discussed the project specification

Date: 13-11-2019
Supervision type: e-mail
Summary: reported learning progress

Date: 17-11-2019
Supervision type: e-mail
Summary: received reference literature from supervisor

Date: 18-11-2019
Supervision type: face-to-face meeting
Summary: discussed the draft project specification

Date: 09-01-2020
Supervision type: e-mail
Summary: send the draft of early term report

Date: 18-01-2020
Supervision type: e-mail
Summary: reported project progress

Date: 28-02-2020
Supervision type: e-mail
Summary: reported project progress and sent the mid-term progress report

Date: 04-15-2020
Supervision type: e-mail

Summary: discussed the draft of final report and the time of mock viva

Date: 04-19-2020
Supervision type: e-mail
Summary: sent the draft of final report and reported the progress

Date: 04-20-2020
Supervision type: online meeting
Summary: did the mock viva and got feedback

Date: 04-22-2020
Supervision type: e-mail
Summary: sent the updated draft of final report and reported the progress

2

Risk and environmental impact assessment

This project is Improving the Stock Market Prediction with Representation Learning. The risk and impact mainly occur in software aspects. And the assessment is shown in Table 7.

Table 7 Risk and environmental impact assessment

Risk Description	Consequence Description	Likelihood level (L)	Consequence level (C)	Risk Score (R = L * C)	Preventative actions
Machine breakdown	Loss of all the code and software because of the PC crash.	2-unlikely	3-very serious	6-Moderate Risk	Backup data periodically.