Final Viva Presentation

Improving the Stock Market Prediction with Representation Learning

Name: Ding Fan

BUPT number: 2016213516

QM number: 161188243

Class: 2016215119

Contents

1. Introduction: Purpose

Learning representations of the data: make it easier to extract useful information when building classifiers or other predictors. Such as: word2vec (embedding words to vectors), node2vec (embedding graph nodes to vectors)

previous studies: as a time series problem-based on the historical price data (solution: RNN)

NEWS: "Microsoft successfully completed the acquisition of company xxx"

Which stocks' price will be affected?

- Microsoft ⬆️
- The supply companies ⬆️
  The customer companies ⬆️

- **quantitative trading data : trading price**
- **qualitative descriptions(un-quantitative) : news and reports.**
- **relationship information (un-quantitative) : stock relationship Graph**

Learn representations that encode these stock-related information

1. Introduction: Target

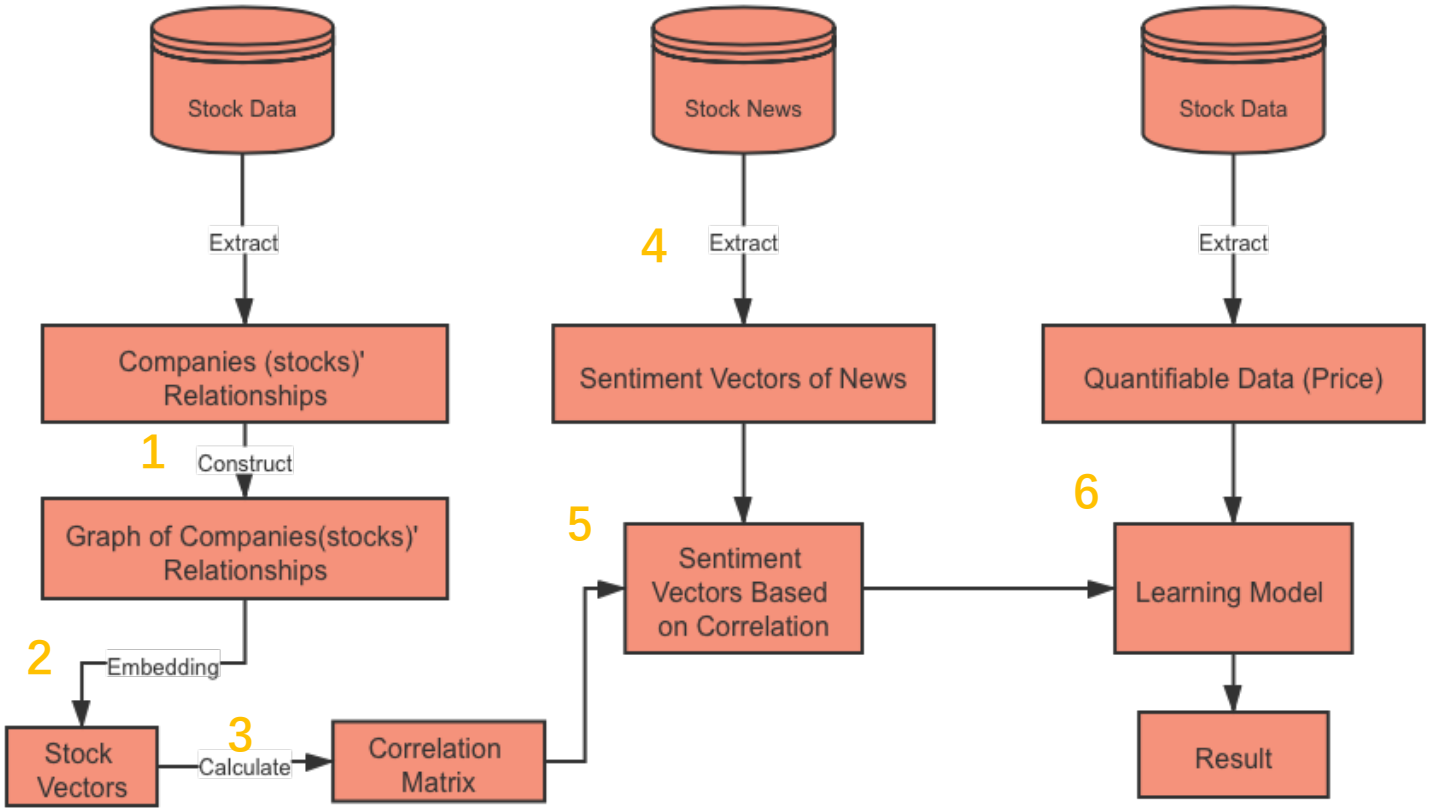
Tasks:

1. Data preprocessing and feature extraction (Finished)
2. Representation learning model building (Finished)
3. Learning method design and implementation (Finished)
4. Software with interface that report the results and visualize the representations (Finished)

Outcomes:

- Representations that encode the stock-related information
- The prediction algorithm and the prediction results
- Working software that can visualize the representations

2. Design and Implementation
Overall Framework



1. Construct the graph of companies' relationships. word2vec principle
2. Embedding the graph of companies' relationships into vectors to generate the stock vectors. node2vec
3. Calculate the similarity between every2 stocks and construct the correlation matrix of 50 stocks.
4. Get the sentiment score of each stock news
5. Use the correlation matrix to update the sentiment score of the target stock
6. Input the quantifiable data and the updated sentiment score of target stock to learning model LSTM

2. Background-1 word2vec

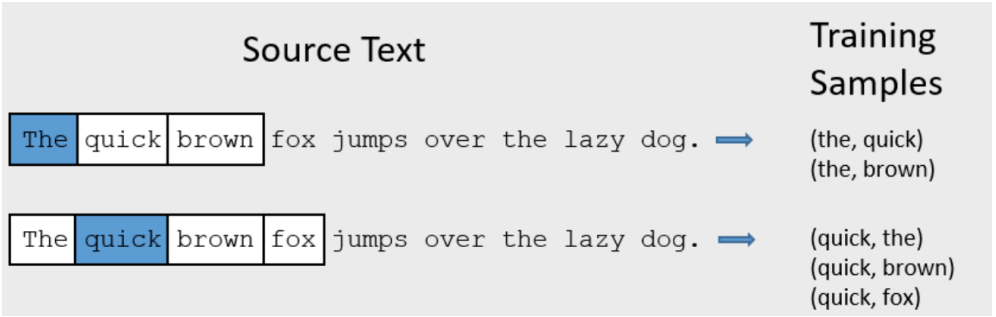
1-Stock Relationship Graph Construction Model

Word2vec: learning low-dimension representation of words

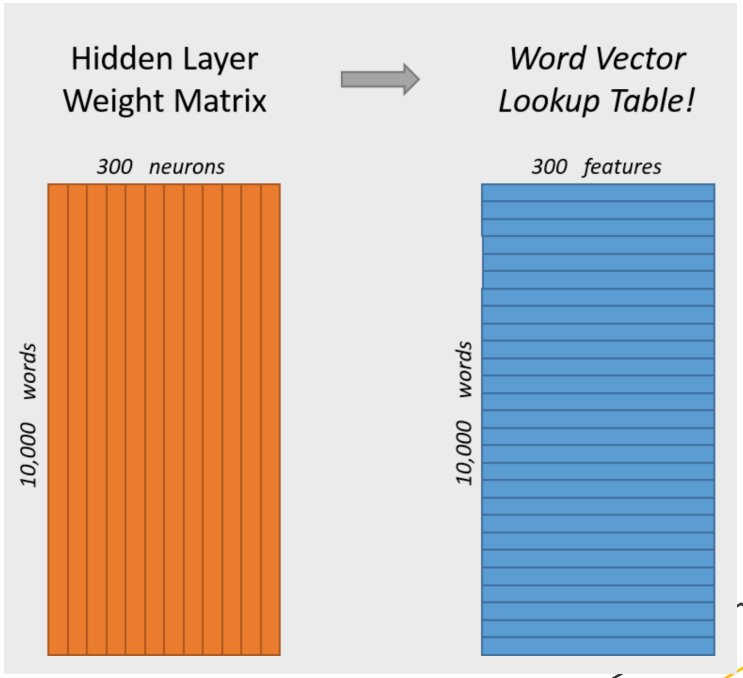
- Represent each word with a vector
- Has better performance when need to calculate the similarity between different words or find relevant words.

word	One-hot	Embedding
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]

- Input: a set of training sentences (**n** different words)
- Step 1: generate training samples from each sentence
- Step 2: train 2-layer network (hidden layer+ output layer)
- Output: **n** vectors represents **n** words (**Hidden layer**)



Trained



2. Design and implementation

1-Stock Relationship Graph Construction Model

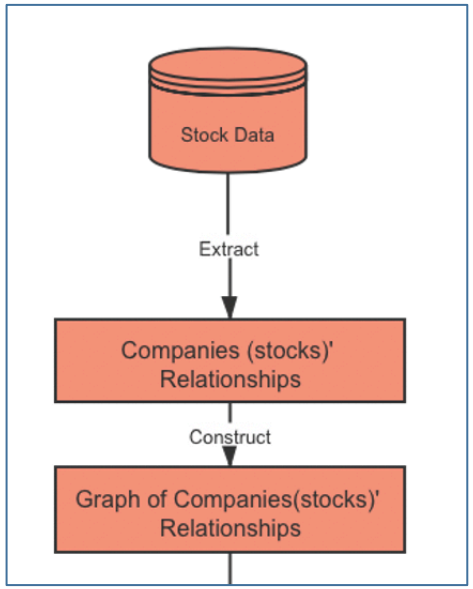
similarity of Stock price fluctuation —word2vec→ **stock relationship graph**

For every 2 stocks: if price fluctuate **more similarly, the 2 stocks** appear together in stock sentence **more frequently**

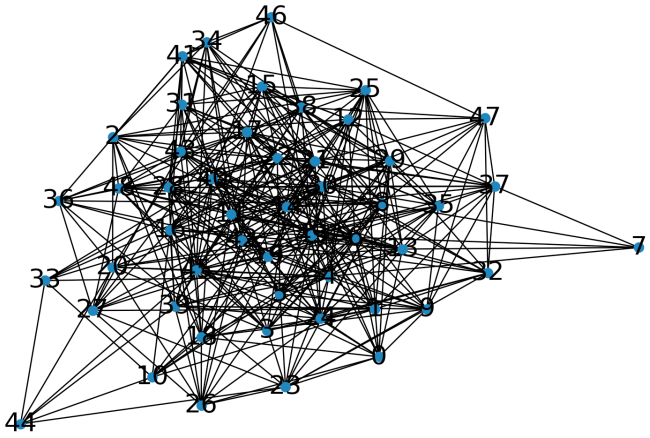
- **Input: 50 stock prices (**244 trading days in 2015)
- **Step1:** Each day, put stocks name **in order of price change** from high to low **to generate 244 sentences**

```
002739 300059 600518 600115 002024 600050 600036 600028 600886 000166 000069 600795 600585 600018 000725 600837 600023 600015 000001 600705 600010 600016 002736 600000 000625 600111 600019 600030 600104
600048 600340 000776 600637 002252 000895 000063 600887 000858 600276 000651 002594 002415 000538 000333 600485 600519 002304
```

- **Step2:** based on 244 sentences, use word2vec to get the **50 vectors (representation of 50 stocks)**
- **Step3:** calculate the cosine similarity **between every 2 stocks** to construct the **stock relationship graph.**



if **similarity exceeds the threshold value**, set an edge between the 2 stock nodes.



2. Graph Embedding Model

2. Background-2 node2vec

Graph Embedding: node2vec (derived from word2vec)

- **Input**: a Graph with **n** nodes
- **Step1: Biased Random walk** on input graph to generate a set of **node sequence sentences for training**
- **Step2: word2vec**
- **Output**: **n** vectors represents **n** nodes in the Graph

- Each vector represents a node in graph
- contains the characteristics of the node itself
- Also,characteristics of the relationship with other nodes

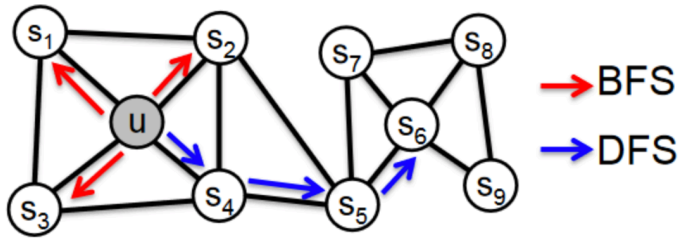
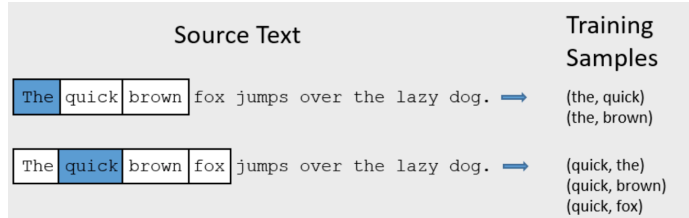


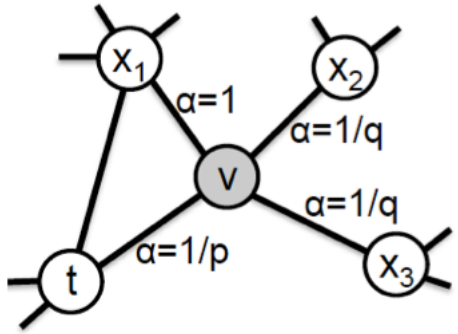
Figure 1: BFS and DFS search strategies from node u ($k = 3$).

Biased Random walk

Sentence_1: u S4 S5 S6
Sentence_2: u S2 S1 S3

like





Biased Random walk

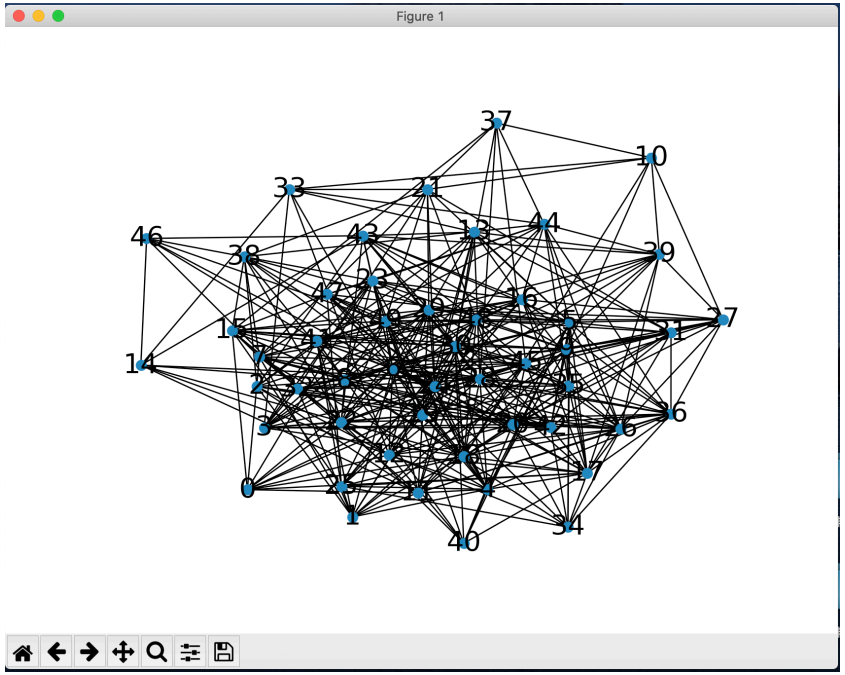
$$\pi_{vx} = \alpha_{pq(t,x)} \cdot w_{vx}$$

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

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

2. Design and implementation

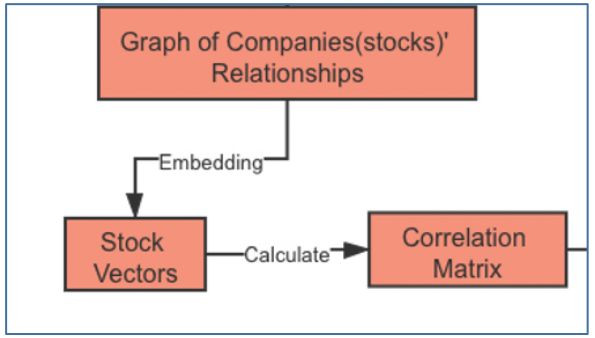
2. Graph Embedding Model

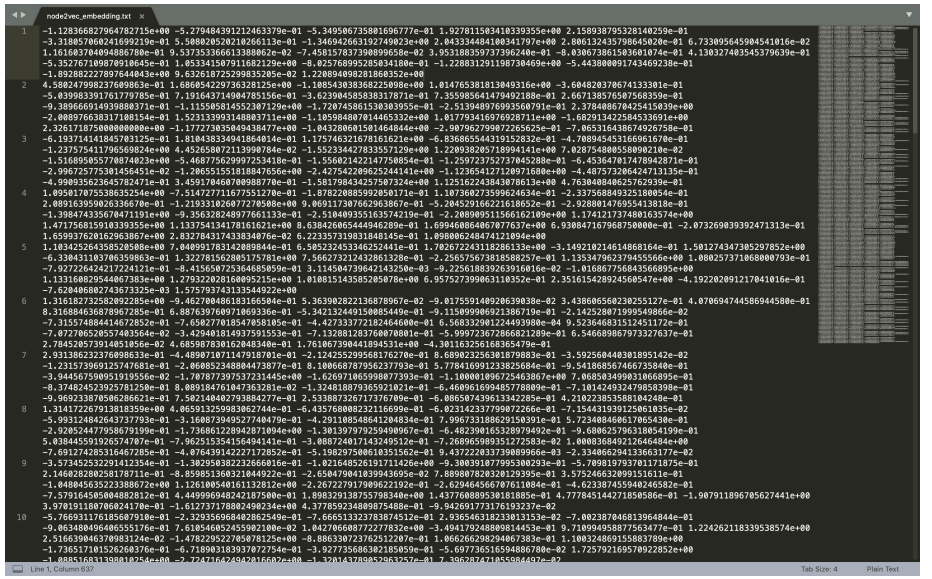


Stock relationship Graph

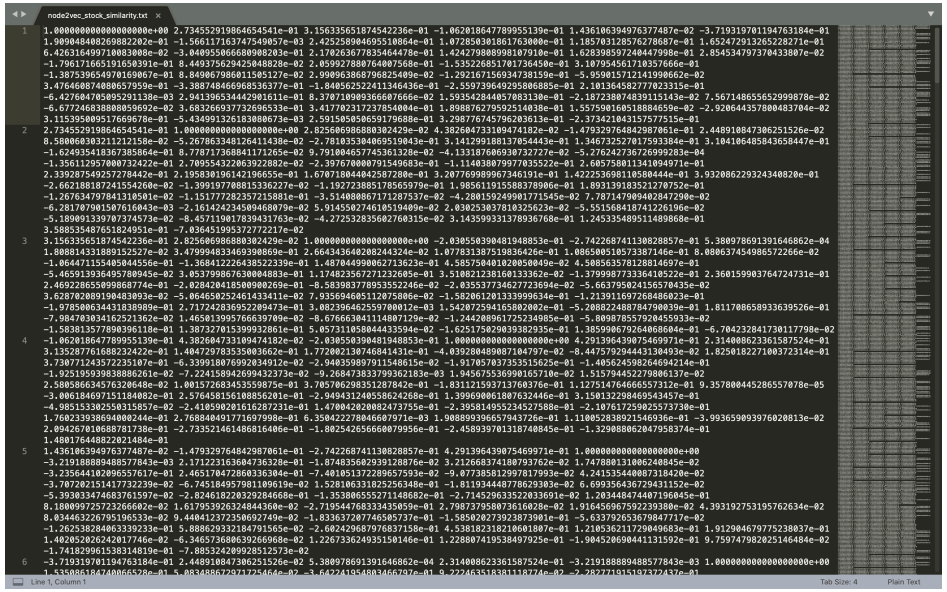
node2vec

Calculate cosine similarity





Node2vec embedding result



Stock correlation matrix

2. Design and implementation

3. Learning Model Based on LSTM

a kind of RNN used to process sequence information data.

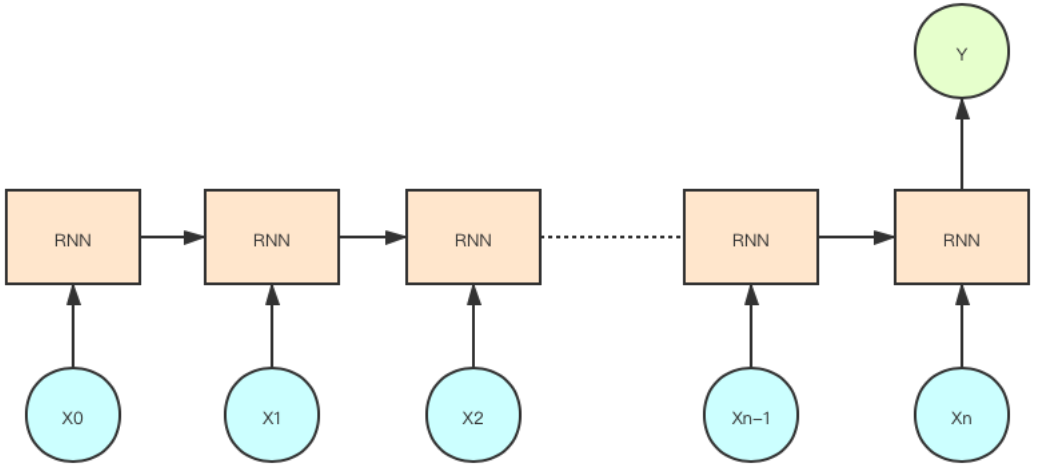
- Quantifiable data
- Correlation matrix
- News data

- Sentiment score of stock x on i-th day is denoted as $\boldsymbol{sentiment_i^x}$
- Quantifiable features is denoted as $\boldsymbol{Q_i^x} = \{\boldsymbol{q_i^{x_1}}, \boldsymbol{q_i^{x_2}}, \dots, \boldsymbol{q_i^{x_n}}\}$
- Similarity between stock x, y is $\boldsymbol{S_{x,y}}$
- Use the correlation matrix to update the sentiment score of stock x by:

$$\widetilde{\boldsymbol{sentiment_i^x}} = \sum_{\boldsymbol{y=0}}^{\boldsymbol{n}} \boldsymbol{sentiment_i^x} * \boldsymbol{S_{x,y}}$$

- $\boldsymbol{LSTM_{input}} = \{(\widetilde{\boldsymbol{sentiment_1^x}}, \boldsymbol{Q_1^x}), (\widetilde{\boldsymbol{sentiment_2^x}}, \boldsymbol{Q_2^x}), \dots, (\widetilde{\boldsymbol{sentiment_n^x}}, \boldsymbol{Q_n^x})\}$

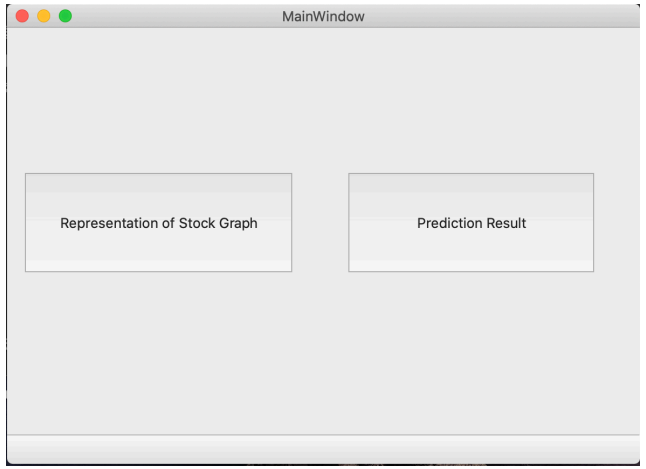
Sentiment score of news: use the NLP Python-SDK provided by Baidu-ai, the reason is that it has a well pre-trained model.

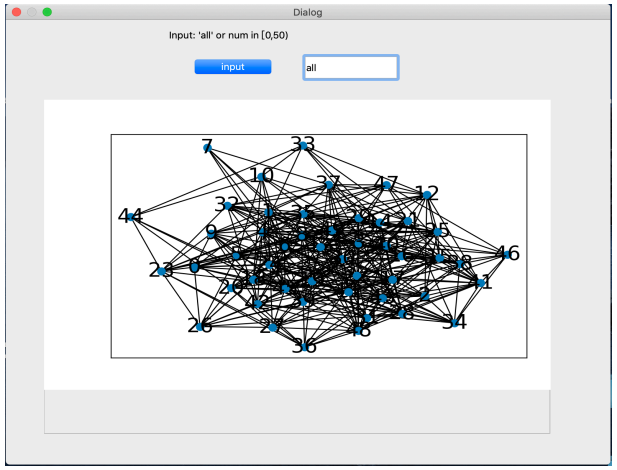


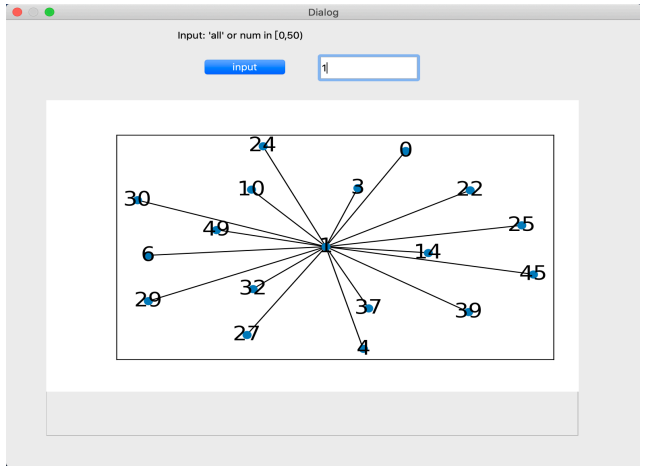
- n refer to the length of the time-series
- Prediction of stock x on n+1-th day, based on n previous days
- $\mathrm{X}_{\boldsymbol{n}} = (\widetilde{\boldsymbol{sentiment_n^x}}, \boldsymbol{Q_n^x})$
- Y is the prediction of the stock x on n+1-th day(1:  ; 0: )

2. Design and implementation

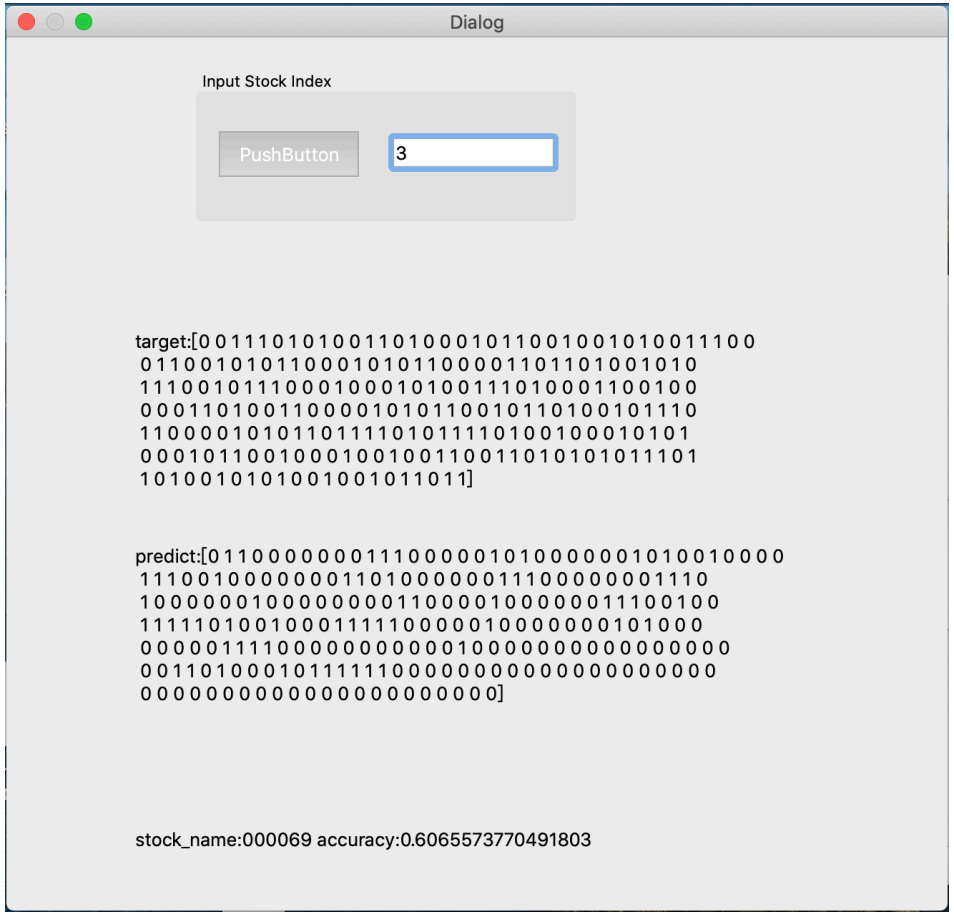
4. Working Software







Window to display the Representation of Stock Graph



Window to show the prediction result

3. Experiments Results

1. Dataset (Price and News on each trading day):

- 50 different stocks in the year 2015, 2016, 2017.
- For each stock, 2015 and 2016 as the training set, 2017 as the testing set.

2. Setup

Group	Description
Group1 (Quantifiable)	give prediction only depends on the quantifiable data like stock prices.
Group2 (Quantifiable + News)	give prediction depends 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.

3. Results of Experiment

- n: Prediction of each stock on n+1-th day, based on n previous days
- Accuracy (ACC) and F-1 score of following table is the arithmetic mean of 50 stock models

	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**

4. Conclusion and Further work

C:

- Using Representation Learning, helps improve the stock prediction performance.
- However, there also are some problems and weaknesses which need to be solved and improved in the future or further work.
- Constructing the stock relationship graph is based on the price fluctuation similarity, which maybe cannot accurately reflect the relationships between different stocks.

F. 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.

Thanks for Watching