Contextual Information Based Stock/Index Market Prediction using Explainable Dynamic Graph Neural Networks

Motivation and Related Works

- Information means fortune
  - Buy long / sell short
    - Both rising price and dropping price means profits
  - Efficient Market Hypothesis (EMH)
    - Assets prices reflect all available information
  - Stock-related News
    - Prevalent and informative
- Previous works are not perfect
  - Traditional ML Techniques
    - Linear regression / naive Bayes / SVM / decision tree …
    - Static features v.s. dynamic trends
  - Deep Learning Techniques
    - traditional CNN / RNN ...
    - Euclidean v.s. Non-Euclidean

Research background

- Daily financial news articles play a vital role for investors while judging the stock prices. However, to fully utilize news materials to infer the stock fluctuation and volatility **accurately and efficiently** using advanced learning techniques is still challenging.
- Leveraging textural information in market prediction is also of high significance due to the embedding experience from human expertise and the investment preference from stakeholders. Such expertise needs **explainable , scalable and analytical** support for both stakeholders.

Related works

- Deep Learning for Event-Driven Stock Prediction (IJCAI, 2015)
- Knowledge-Driven Event Embedding for Stock Prediction (COLING, 2016)

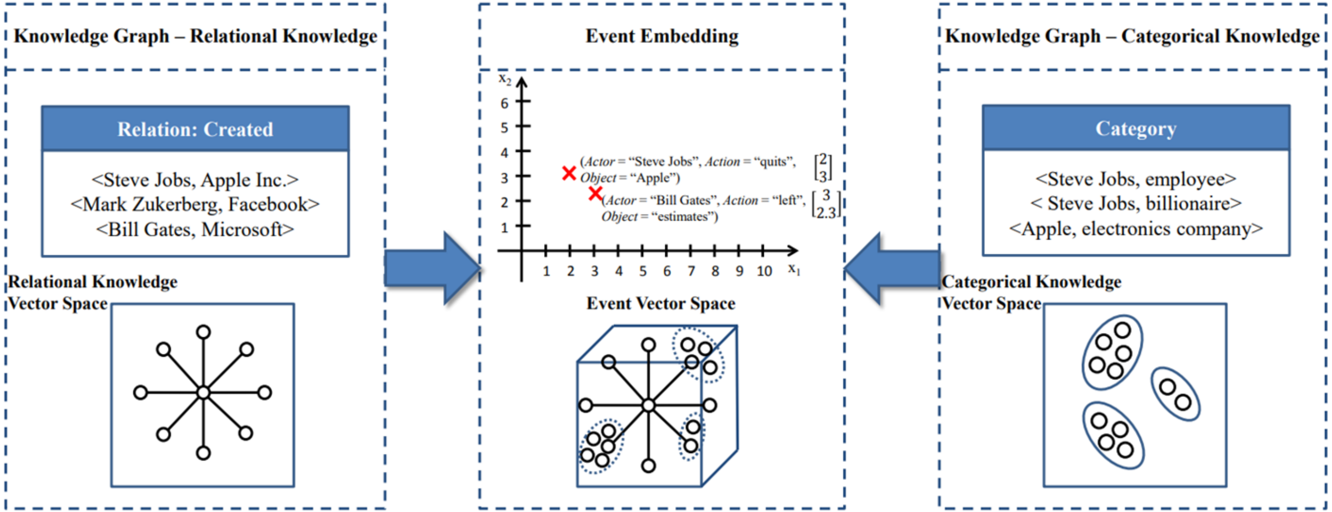
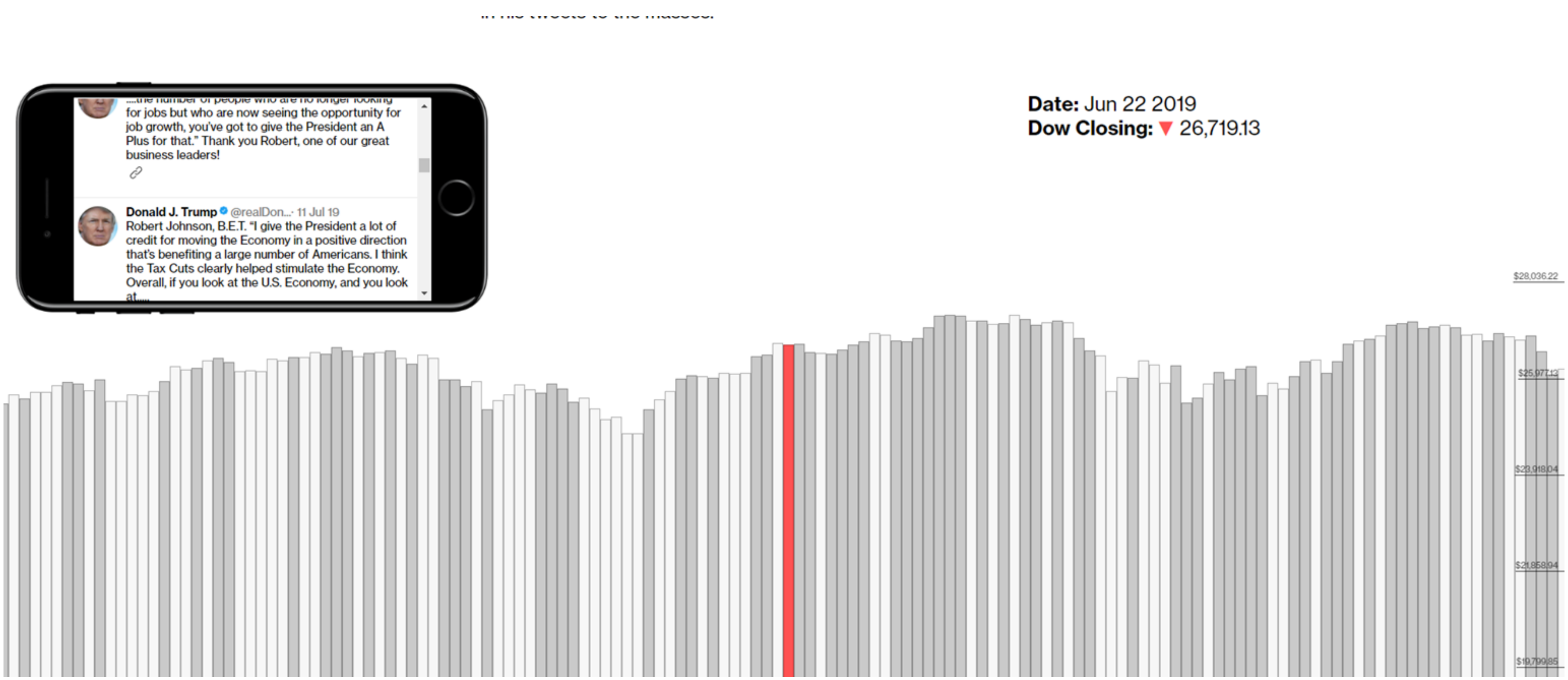


Figure 1: Incorporating knowledge graph into the learning process for event embeddings.

Related works from industry



https://www.bloomberg.com/features/trump-tweets-market/

Related works from peer

Predict Effect of Trump's Tweets on Stock Price Milestone (Stanford)

The Effects of Donald Trump's Tweets on US Financial and Foreign Exchange Markets

- We found all of Trump's Tweets from May 4th, 2009 to Oct 20th, 2017 on www. trumptwitterarchive.com.
- We exported their archive, and got a 5.9 MB csv file with 11, 330 tweets. The second part of our dataset is the outputs (labels). We are going to use ~ 1GB of S&P 500 Index and underlying stocks minute resolution trading prices as an indication of market movements.
- We acquired this data source from Wharton Research Data Services (WRDS), which is a matrix of dimension 146640×486, where 146, 640 refers to the time ticks from the beginning of 2016 up to today.The third part of our data is the pretrained GloVe[3] for word embedding, which will be used in the LSTM model.

Results: Analyzing the problem using different models, here is a brief summary of the prediction accuracy:

Baseline Random Prediction:	**33.3%**	Naïve Bayes:	**42.5%**
SVM classifier:	**46.5%**	LSTM (RNN):	**48.2%**

Moreover, experimental result shows that longer time interval

Their limitations

- These prior models do not provide an approach to **reveal hidden contextual information among entities.** Our proposed model is learned based on graph structures of words. This gives us the benefit of discovering the impact of hidden connections among words for **forecasting future events.**

- Given the limited dataset, we would like to propose an efficient,scalable, explainable method to predict events

“Tensor a mathematical object analogous to but more general than a vector, represented by an array of components that are functions of the coordinates of a space. = Rank in Matrix”

Event Forecasting - we all want to predict future in a way or so

Many event forecasting approaches with spatio-temporal correlations have been applied to **predictions of elections, stock market movements, disease outbreaks, and crimes.**

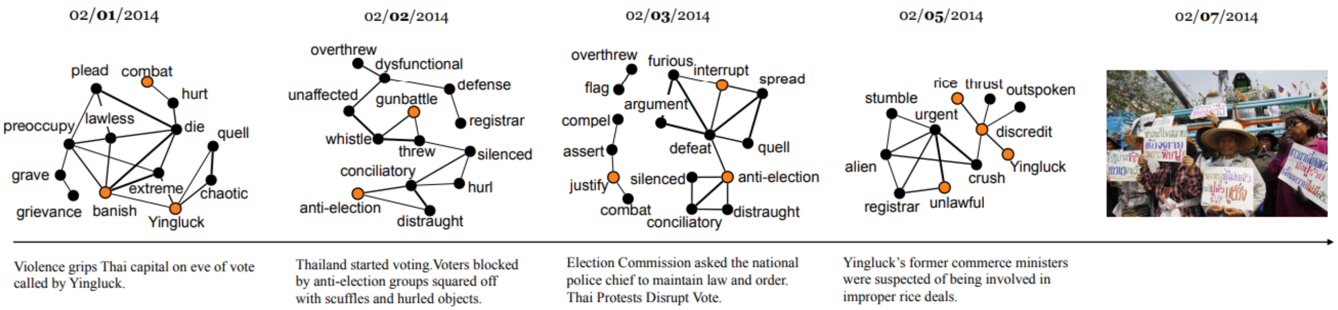


Figure 3: An event in Thailand: Hundreds of rice farmers are heading towards the Ministry of Commerce in Bangkok to demand the government give their rice back and trying to oust Yingluck due to deception and waste in the rice scheme.

Learning dynamic context graphs for predicting social events. KDD, 2019

Learning Dynamic Context Graphs for Predicting Social Events (KDD 2019)

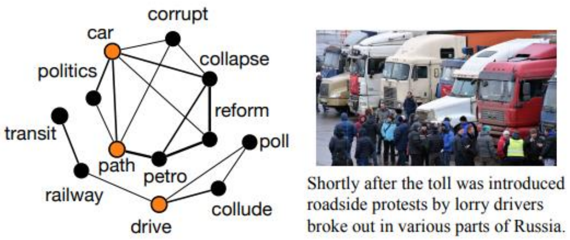


Figure 5: Event summary graph for one event in Russia.

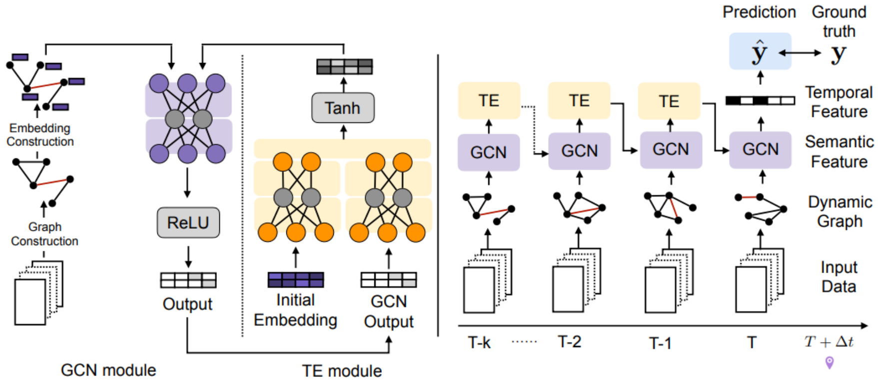


Figure 2: System Framework of the Dynamic Graph Convolutional Network. Input data consists of event related articles ordered by time. We construct dynamic graphs based on input data and feed them into the GCN layers by time. For each GCN layer except the first one, the input features are processed by a Temporal Encoded (TE) module, involving the output of the last GCN layer and the current word embeddings, to capture temporal features. We add a masked nonlinear transformation layer to unify the final output vector from the final GCN layer. The loss is calculated between the model output and ground truth.

pre-trained semantic features usually can not reflect contextual changes over time.

They propose a **temporal encoded feature module** to alleviate this problem. It takes into account both the semantic embeddings and the hidden graph embeddings from previous time periods.

The network consists of three main components:

1) an encoding and extracting graph from news module;
2) an innovative graph neural network;
3) a new mechanism for handling sequential graph representation

Part 1 : Dataset and Evaluation

- News data: The author of the dataset crawled historical news headlines from Reddit WorldNews Channel (/r/worldnews). They are ranked by Reddit users' votes, and only the top 25 headlines are considered for a single date. (Range: 2008-06-08 to 2016-07-01).

- Stock data: Dow Jones Industrial Average (DJIA) is used to ”prove the concept”. (Range: 2008-08-08 to 2016-07-01).

  Task evaluation: the data from 2008-08-08 to 2014-12-31 as Training Set, and Test Set is then the following two years data (from 2015-01-02 to 2016-07-01).

Part 1 : Dataset and RQ

- Macroeconomics: target to predict the daily changes of the Dow Jones Index Average (DJIA) close price given Reddit's Top News.
- To what scale the trend DJIA rises or fall, the ratio
- Sentiment analysis: how information like reddit/top news/representative twitter affect DJI, S&P, certain stock

"Can market sentiment really help to predict stock price movement"?

Market Trend Prediction using Sentiment Analysis: Lessons Learned and Paths Forward   (WISDOM'18)

https://arxiv.org/ftp/arxiv/papers/1607/1607.01958.pdf
https://www.sentic.net/wisdom2018mudinas.pdf

Evaluation methods

Prediction performance including **accuracy, recall, precision, and F1-Score;**
a new indicator?

- Interpretability of graph representation
  - We display the dynamic context graphs indicating the importance of specific factors and discuss the effectiveness of our model.
- Hyperparameter Sensitivity
  - evaluate the performance of the model with several hyper parameters in our model and analyze the influence of varying hyperparameters for experiment results

Learning Dynamic Context Graphs for Predicting Social Events

The general graph-based learning problem

The graph learning problem is formulated as follows:

- we are given a set of nodes, each with some observed numeric attributes x_i.
- For each node we'd like to predict an output or label y_i. We observe these labels for some, but not all, of the nodes.
- We are also given a set of weighted edges, summarised by an adjacency matrix A. The main assumption is that when predicting the output y_i for node i, the attributes and connectivity of nearby nodes provide useful side information or additional context.

Kipf & Welling use graph convolutional neural networks to solve this problem. A good way to imagine what's happening is to consider a neural network that receives as input features x_j from all nodes j in the local neighbourhood around a node i, and outputs an estimate of the associated label y_i. The information from the local neighbourhood gets combined over the layers via a concept of graph convolutions.

The deeper the network, the larger the local neighbourhood - you can think of it as the generalisation of the *receptive field* of a neuron in a normal CNN. This network is applied *convolutionally* across the entire graph, always receiving features from the relevant neighbourhood around each node.

How powerful are Graph Convolutions? (review of Kipf & Welling, 2016)

From Event Embedding to Graph Neural Network

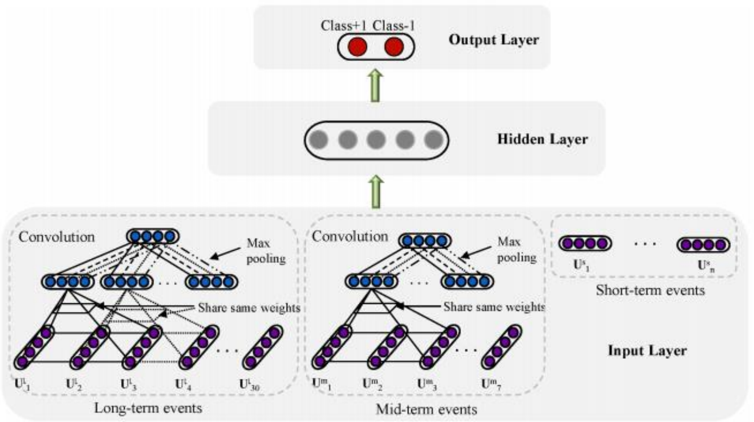


Figure 3: Architecture of the prediction model based on a deep convolutional neural network.

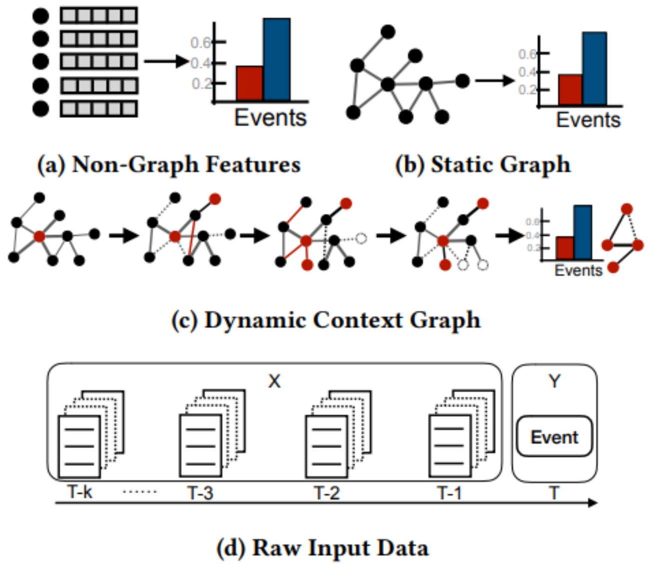
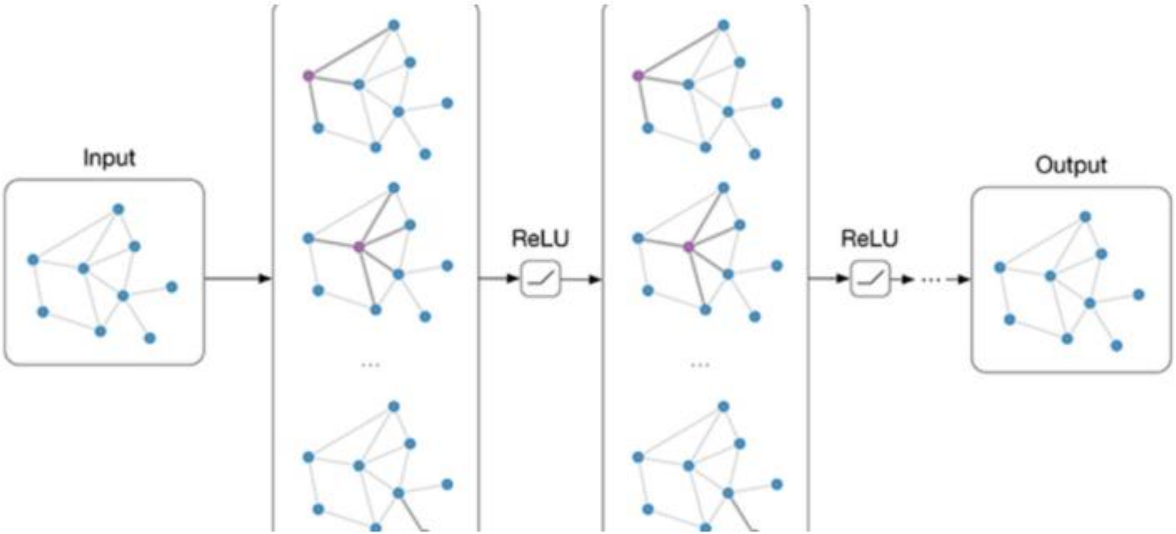


Figure 1: A motivating example of learning dynamic context graphs for event forecasting. Given a raw input data X (1d) and its target value y, existing models (1a) focus on semantic representations; New graph models (1b) ignore the temporal evolution of graphs in a sequence of inputs; Dynamic context graph models (1c) encode both temporal information and semantic embeddings in event modeling.

We create multiple word relation graphs represented by a sequence of adjacency matrices [At−k, ...,At−1], where Ak ∈ Rn×n for each time step (day).

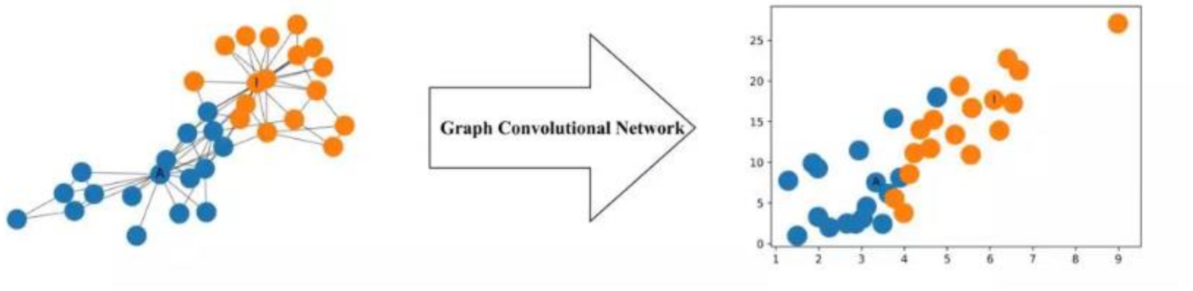
What is GNN

- graph convolutional networks (GCN/GNNs), which nicely integrate local vertex features and graph topology in the convolutional layers



Why from static to dynamic GNN

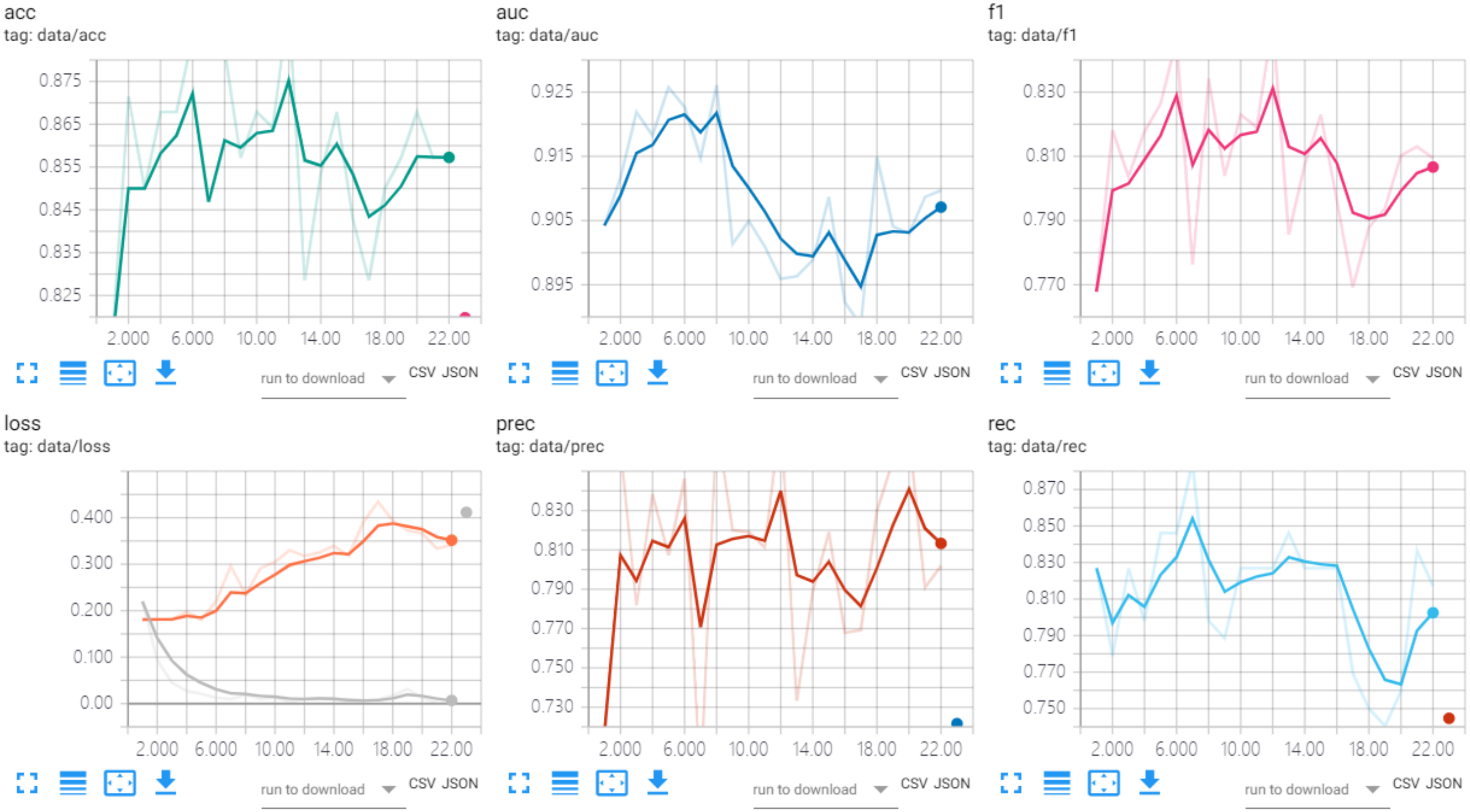
- While existing models use natural language processing techniques do utilize the contextual information within a specific period, but ignore the dynamic connections across the reported events, losing deeper insights like the trends in the short term.



How powerful are GNNs

- Previous deep neural networks like CNNs or RNNs mainly focus on regular Euclidean data (e.g., images and text), which can not be directly applied in non-Euclidean data, like graphs.

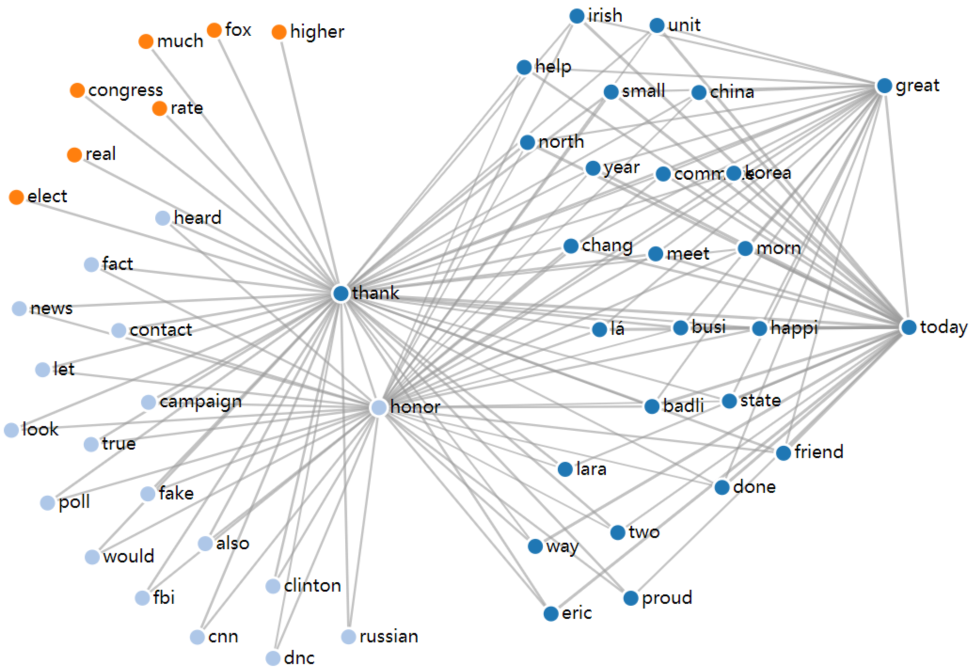
Implementation Results of DynamicGCN KDD 2019



Part 2: Data Preprocessing

- Dataset
    - Reddit top news (Reddit)
        - Crawled from Reddit WorldNews Channel. Ranked by Reddit users' votes. The top 25 each day from 2008-06-08 to 2016-07-01.
    - Trump's twitter (Trump)
        - Trump’s words deeply affect the stock trend in recent years. We crawl twitters of Trump from 2009 to 2018.
    - News about Apple (Apple)
        - Stock value and news from The New York Times API about Apple.

Visualizing and Structuring hidden word graph



“Thank you Brock – it is my honor. We (@FEMA) have never had the support that we have had from thisPresident. @FEMABrock. – Sat Sep 15 20:41:45 +0000 2018

Data Preprocessing

- Dataset
- Data Cleaning
  - Input requirement
    - m days of time series input to predict the stock price of the next day
    - nxn dimension adjacent matrices for each day, n is the number of all the words in m days
  - Setting for each dataset
    - Reddit: we set m=7. n use all the words from 7*25 articles. We predict whether the DJIA goes up or down.
    - Trump: we set m=7. n use all the words from twitter in 7 days. We predict whether the DJIA goes up or down.
    - Apple: we set m=2. n use all the words from 2*7 days of news summaries. We predict whether the AAPL goes up or down.

Data Preprocessing

- Dataset
- Data Cleaning
- Build adjacent matrices
  - Document-based point-wise mutual information (PMI)

$$PMI_t(i,j) = \log \frac{d(i,j)}{d(i)d(j)/D}$$

    d(i,j): total number of articles containing word i,j at time t.
    d(i): the total number of articles containing i.

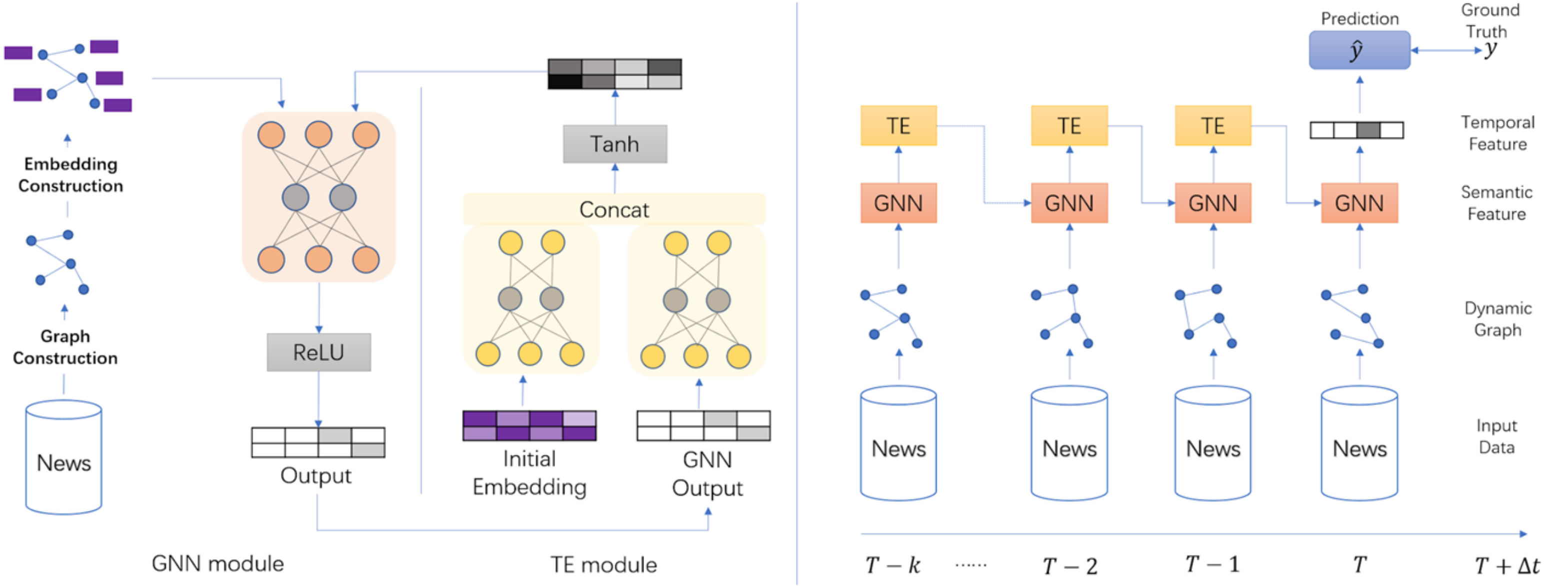
  - Generate elements for adjacent matrix

$$A_t[i,j] = \begin{cases} PMI_t(i,j) & PMI_t(i,j) > 0 \\ 0 & otherwise \end{cases}$$

Data Preprocessing

- Dataset
- Data Cleaning
- Build adjacent matrices
- Additional process
    - Stop word
        - Words that appears too often gives no information: “a”, ”the”.
    - Word stemming
        - Word with the same meaning or root.
        - Transfer them to the initial root word: “work”, ”works”, ”worked”

Model - DynamicGNN framework



Model - Graph Neural Network module (GNN module)

$$H_{t+1} = g\left(\hat{A}_t \tilde{H}_t W^{(t)} + b^{(t)}\right)$$ GCN Layer.

$$H_{t+1} = MLP_t\left(\hat{A}_t \tilde{H}_t\right)$$ GIN Layer.

Notations:

$\hat{A}_t \in \mathbb{R}^{n \times n}$ the normalized symmetric adjacency matrix at time step t.

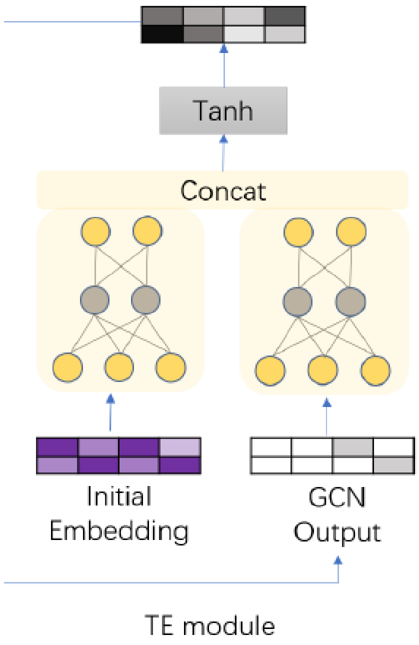
$W^{(t)} \in \mathbb{R}^{F(t) \times F(t+1)}$

$b^{(t)} \in \mathbb{R}^{F(t+1)}$ model parameters of the GCN layer at time step t .

$\tilde{H}_t$ the temporal encoded (TE) embeddings calculated from the last TE layer.

MLP_t the multi-layer perceptron of the GIN layer at time step t .

Model - Temporal Encoding module (TE module)



$$H_p^{(t)} = H_t W_p^{(t)} + b_p^{(t)}$$

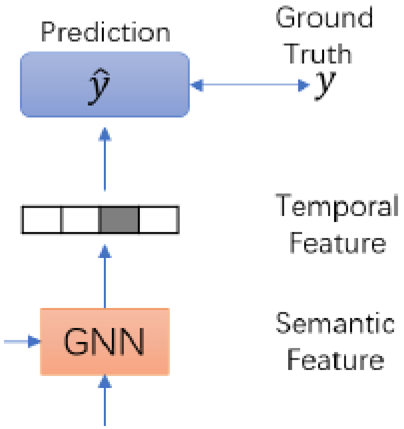
$$H_e^{(t)} = H_0 W_e^{(t)} + b_e^{(t)}$$

$$\tilde{H}_t = \tanh\left(\left[H_p^{(t)} \| H_e^{(t)}\right]\right)$$

$$W_p^{(t)} \in \mathbb{R}^{F^{(t)} \times \alpha}, W_e^{(t)} \in \mathbb{R}^{F \times \left(F^{(t)} - \alpha\right)}, \text{ and } 0 \leq \alpha \leq F^{(t)}$$

Model - Masked Nonlinear Transformation Layer

to map the final output vector to the prediction of the task.



$$z_T = \text{zero\_}padding\left(H_T^\top\right)$$

$$\hat{y} = \sigma\left(z_T w_m^\top + b_m\right)$$

Model - Optimization

Compare the prediction value with the ground truth

Optimize the **binary cross entropy loss**:

$$\mathcal{L} = -\sum y \ln \hat{y}$$

Evaluation

Based on PyTorch, Intel(R) Core(TM) i9-7900X CPU 3.30GHz, 64GB RAM and 2 Nvidia GeForceGTX 1080 Ti GPUs.

Table 1: Performance comparison on test set.

	Reddit				Trump				Apple			
	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.
Random Guess	0.525	0.4884	0.506	0.4836	**0.5714**	0.5569	0.5641	0.5278	0.525	0.6176	0.5676	0.5362
Naive Bayes	**0.5602**	0.4977	0.5271	0.5164	0.5374	0.5	0.518	0.4895	0.6666	0.1765	0.2791	0.5507
DynamicGCN	0.5498	**0.8465**	**0.6667**	**0.5416**	0.5374	**0.9742**	**0.6927**	**0.5347**	**0.6875**	**0.8049**	**0.7416**	**0.6667**
DynamicGIN	0.5587	0.6419	0.5974	0.5315	0.5365	0.9484	0.6853	0.5312	0.6	0.75	0.6667	0.6087

Evaluation

What are the dominant factors?

Table 2: Word Frequency Range affects the performance of two models

Word Frequency Range	DynamicGCN	DynamicGIN
[2-20]	**0.6667**	0.5507
[2-30]	0.5362	**0.6087**
[2-40]	**0.5942**	0.5072
[2-50]	0.5362	0.5362

Conclusion

- We conduct a series of experiments to apply DynamicGNN to stock trend prediction.
- The results show both DynamicGCN and DynamicGIN are effective compared to machine learning method given limited data.

As far as we know, adopting dynamic graph embedding into finance industries with twitter data have great potential.