# FLAG: Stock Movement Prediction via Fusing Logic and Semantic Graphs of Financial News

Angela Li[1], Jiduan Liu[2,3], Yuyong Li[1], Fan Meng[1 †]
[1]*Department of Information Management, Peking University*
[2]*Wangxuan Institute of Computer Technology, Peking University*
[3]*Center for Data Science, AAIS, Peking University*
{ypliaq, liujiduan, liyuyong, mengfan}@pku.edu.cn

2022 IEEE International Conference on Data Mining Workshops (ICDMW) | 979-8-3503-4609-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDMW58026.2022.00034

***Abstract*—Financial news is widely used in stock movement prediction, and the critical point is to extract valuable information from news. Some previous works utilize graph-based approaches to represent news in a topological structure and construct relationships between nodes by semantic similarity. However, most of these models ignore depicting contextual information and logical structures in the news. In this work, we propose FLAG (Network with Fusion of Logic and semAntic Graphs), a novel model that captures both the logical structure and the latent semantic connection in the news for stock movement prediction. In FLAG, we construct two graphs for each news item, a logical graph based on continuation and coordination relations and a semantic graph based on semantic encoding. Then, we fuse the two graphs to represent news. Moreover, we combine news representations with global market information and historical prices for final prediction. FLAG achieves the state-of-the-art accuracy on the dataset we collected.**

***Index Terms*—stock movement prediction, graph neural network, logic graph, semantic graph**

## I. Introduction

Stock prediction has become one of the core applications in financial data mining, as it can lead to huge profits. However, due to the high stochasticity of the stock market, the exact price of a stock is difficult to predict [1]. Many researchers focus on the task of stock movement prediction, i.e., whether the stock prices will rise or fall at a future moment rather than exact stock prices.

Historical stock price is the primary information source for stock movement prediction. With the rapid development of natural language processing (NLP), many approaches try to introduce textual data, mostly public news and social media comments [2]. The key to utilizing textual data is to extract valuable features as representations. With the achievements of Graph Neural Network(GNN) in some applications [3], researchers have applied graphs to represent the topology of textual data and further analyzed them by GNN for downstream tasks.

Most previous works use sentences as nodes and semantic similarities based relationships as edges in the graph [4], [5]. However, using semantic similarities as relationships makes the following limitations. First, these approaches lack the integration of contextual information. Especially for news, where the description of a complete event is often composed of consecutive sentences, using semantic similarities as relationships cannot take into account the integration of contextual information in some cases. Second, the logical relationships between nodes are ignored. For example, Fig 1 shows a news snippet that reports the inflation in Thailand. It can be seen that clause 6 is juxtaposed with clause 4 and clause 5, and they both form a causal relationship with clause 7, which cannot be captured by previous works. Third, the previous approaches cannot capture long-range dependencies in graphs. Many methods fuse information only for 1-hop and 2-hop nodes adjacent to the node. However, there are many implicit relationships between sentences and clauses in news. For instance, clause 1 and clause 7 in Fig 1 are both direct statements about inflation in Thailand, although they are far apart in terms of positional relationships.

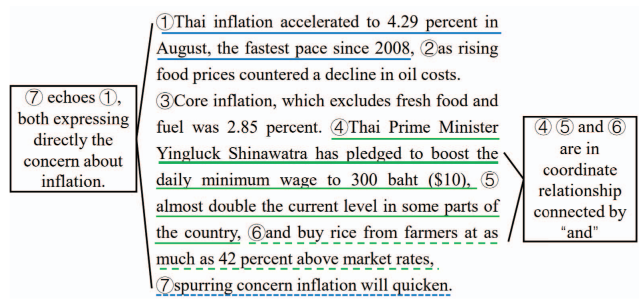


Fig. 1. A news snippet about inflation in Thailand. We mark clauses with logical relations and implicit relations in green and blue respectively. Our model will try to depict these two relationships for better graph representation.

Aiming to tackle the above-mentioned limitations in terms of *contextual information*, *logical structure* and *implicit relation*, we propose FLAG, an end-to-end framework that constructs news graphs with logical and semantic relationships and apply it to the stock movement prediction task. The main ideas of FLAG are as follows. First, FLAG generates graphs to capture logical structures in news by extracting the coordinate and continuation relations between clauses. Second, FLAG builds semantic graphs by semantic similarities to explore the implicit connections between nodes. Third, FLAG adopts GNN and the attention mechanism to compress graphs into

† Corresponding authors: Fan Meng (mengfan@pku.edu.cn).

2375-9259/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDMW58026.2022.00034

vectors. Finally, FLAG combines global market information and historical prices of stocks with obtained news embeddings for final stock prediction.

The contributions of this paper can be summarized as follows:

- We propose a novel graph-based model that fuses logical graph and semantic graph to obtain a better representation of text.
- We propose a logical graph construction algorithm to depict logical relations in text.
- We conduct experiments on stock movement prediction tasks, and achieve the state-of-the-art performance on the dataset we collected.

## II. Related Work

We introduce related works in two aspects: *stock prediction* and *graph representation*.

### A. Stock Prediction

We categorize the works into a) those using technical analysis, b) those using fundamental analysis, and c) those based on graph models. The technical analysis takes historical prices as features to predict stock movements while the fundamental analysis uses other qualitative and quantitative features in addition to prices. Various models [6]–[10] using historical prices are based on long short-term memory units (LSTM) [11] and attention mechanism [12], the former is a representative neural network designed for sequential data, and the latter is commonly used for compressing the hidden representation with adaptive weights.

In addition to technical analysis, there are approaches in fundamental analysis that use external data, mostly textual data. Some researchers have performed sentimental analysis to extract meaningful features from unstructured textual data by sentiment dictionaries [13]–[16]. On the other hand, some approaches extract deep features to acquire more latent information of textual data, addressing the limitation of sentiment analysis that can not extract structured information data [17]–[19].

The application of graph-based approaches for stock movement prediction has increased rapidly in recent years. Existing works usually use stocks as nodes and build edges using external information, such as sector-industry relations and Wiki company-based relations [20], [21], stock description [22], investment facts [23], shareholdings and affiliations [24].

### B. Graph representation

We mainly divide the models about graph representation into two groups: a)word-level graph construction and b)sentence-level graph construction. Word-level approaches convert texts to graph-of words [25], [26]. Specifically, they use words in the text as nodes and calculate word co-occurrence statistics as edges. These methods are effective in capturing long-range semantics but face high computational complexity.

To address the computational complexity limitation, some models have proposed using sentence clusters as nodes and semantic similarities for relations representations [4], [5], [27], [28]. The main limitation of such approaches is that they ignore the logical structure of text, which further hinders the extraction of deeper relationships. Furthermore, taking sentence units means that the number of nodes is drastically reduced while coarser-grained edges are depicted, which may cause the problem of simple structure and sparse edges.

## III. Method

We propose FLAG, a novel method that learns the logical structures and semantic relationships in news and can be used in stock prediction task.

### A. Overview

We provide an overview of the challenges to be addressed and the model components. The following are three challenges in constructing structural graphs for stock movement prediction:

(1) **Depicting contextual information in graphs.** Contextual information is an important part of natural language processing. Using semantic similarities as relationships cannot fully capture the contextual information. *How can we incorporate contextual information into graphs?*

(2) **Building graphs with logical structures.** Graph-based approaches have been widely used for text representation in recent years. Most previous works add edges between nodes with a high level of semantic similarity while ignoring the logical relationships between nodes. *How can we build graphs to depict logical relationships between nodes?*

(3) **Capturing long-range implicit relationships in graphs.** Many methods only aggregate information of nearby nodes, which are effective in learning some information sources, such as citation networks and recommender systems. However, news has more complex relationships, with implicit relationships between distant nodes. *How can we explore implicit relationships between nodes?*

To address the three challenges, we build a novel framework, FLAG, the detail information depicted in Fig 2, which consists of four main components:

**(1) Logical graph construction(Section B).** We utilize clause-level information as nodes and capture the coordinate relations and continuation relations between nodes to incorporate contextual information and logical structures into graphs.

**(2) Semantic graph construction(Section C).** We map nodes to the latent space by semantic encoding and explore implicit relationships by measuring distances between nodes in the latent space.

**(3) Graph representation(Section D).** We use GCN to aggregate node information and attention mechanism for compressing graph information. The attention maps naturally incorporate information about nodes with adaptive weights.

**(4) Stock movement prediction(Section E).** We combine global market information and individual stock information by gate mechanism to make the global information more dynamically integrated into the stock information.

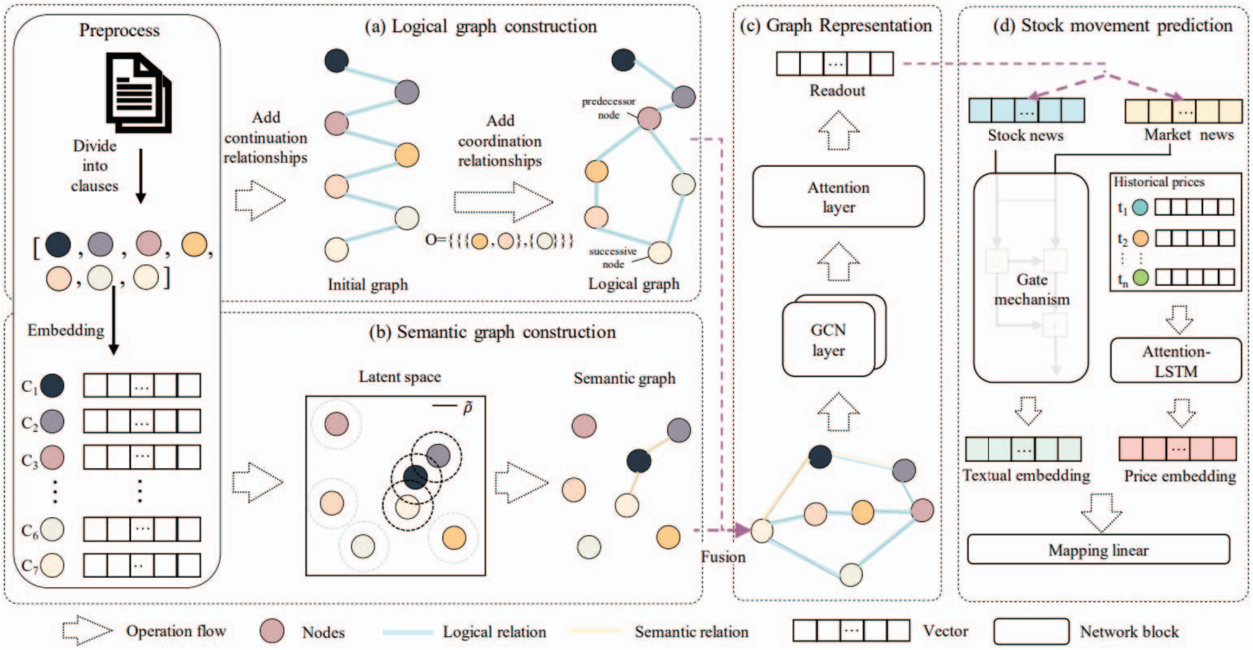


Fig. 2. Overview of FLAG. which can be divided into four steps: (a) Logical graph construction; (b) Semantic graph construction; (c) Graph representation; (d) Stock movement prediction.

### *B. Logical Graph Construction*

We build a clause-level graph to capture the inner clause sequential information and the logical relationships between clauses. Formally, Let $G = (V, E)$ denote a graph, where each node $v \in V$ has a feature vector $x_v \in X$ and each edge $e \in E$ connects two nodes. FLAG obtains $V$,$X$ and $E$ in $G$ as follows.

**Nodes.** We use clauses instead of sentences as nodes due to the fact that financial news is written document with complex semantic structures sentences. There are some differences between a sentence and a clause. A sentence is a group of words that begin with a capital letter and end with a period, question mark, or exclamation mark, while a clause only needs a complete subject and predicate. We choose clauses as nodes because we can depict the relationships between nodes as much as possible while keeping the complete semantics of the nodes. For specific practice, in the case of Chinese text, we use punctuation marks to divide the clauses, such as commas and semicolons. We assume that the news $N = \{c_1, \cdots, c_n\}$ is composed of $n$ clauses and $c_i$ represents the $i$th clause in the news.

**Node features.** To capture sequential information in a clause, we propose to use Sentence-BERT (SBERT) [29] as the node encoder. We feed each clause into SBERT and get the initial clause embeddings as node features $X \in \mathbb{R}^{n \times d}$,where $n$ is the number of nodes and $d$ is the length of embedding vectors.

**Edges.** In building edges, we try to balance carefully crafted rules and maintainability. If we obtain the logical graph with more crafted rules, it is hard to maintain and reproduce, although it has a rigorous structure. Thus, we only considered the *continuation relation* and *coordinate relation*, commonly used to depict the relationships between sentences or clauses.

First, we construct the continuation relationship between the clauses. Since news texts have coherent semantics, there is an implicit continuation relationship between adjacent clauses. To describe the implicit continuation relationship between nodes as much as possible and improve the recall of the continuation relationship, we add edges between two adjacent clauses. The initial adjacency matrix $A^{(0)} \in \mathbb{R}^{n \times n}$ is a Boolean matrix and each element can be defined as:

$$A^{(0)}_{i,j} = \begin{cases} 1 & \text{if} \quad |i-j| = 1 \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

Besides, this way of building the continuation relation makes it possible to notice the interaction of contextual information in the subsequent graph processing.

Then, we add edges to the initial graph that depict the coordinate relation. We divide it into two steps: a) candidate node extraction, and b) coordinate relationship construction.

For the first step, we use two ways to extract candidate nodes: 1) punctuation, some punctuation marks are naturally useful for expressing coordinate relation, such as semicolon in Chinese. 2) Conjunction word templates. Some words are often used to combine two clauses with a coordinate relation, such as and, too, as well as, etc. We assume that $O = \{o_1, \cdots, o_m\}$ is composed of $m$ groups of coordinate relations in the graph. $o_i = \{\widetilde{c}_1, \cdots, \widetilde{c}_v\}$ is the $i$th group that contains $v$ sets of clause clusters where each set is coordinated with others. $\widetilde{c}_j$ denotes the $j$th clause cluster comprising one or more clauses. For instance, in the news snippet in Fig 1, we extract conjunction word template "and" at the start of clause 6, and we can build the coordinate relation between clause 4,5 and clause 6, i.e. $O = \{o_1\}$ and $o_1 = \{\{c_4, c_5\}, \{c_6\}\}$.

In the second step, for each $o_i \in O$, we first remove the edges between clause clusters in $o_i$. Then, each clause cluster in $o_i$ should share the same predecessor and successive nodes as they are in a parallel relationship. The predecessor node is defined as the adjacent node of the first clause in $o_i$, and the index of the adjacent node needs to be less than the index of the first clause. The successive node is defined as the adjacent node of the last clause in $o_i$, and the index of the adjacent node needs to be more than the index of the first clause. We add edges between the predecessor node and the first clause of each clause cluster in $o_i$. Similarly, we add edges between the successive node and the last clause of each clause cluster in $o_i$. The whole process is illustrated in Fig 2(a). The adjacency matrix $A^{(0)}_{i,j}$ can be updated as $A'^{(0)}_{i,j}$ after the coordinate relation is established.

### C. Semantic graph Construction

To address the limitation that some graphs lack attention to distant nodes, we expect to construct graphs that pay enough attention to nodes that are distant from each other but lie implicit connections underneath texts. Inspired by geometric aggregation scheme [30], we weigh the implicit connection of nodes by calculating the distances they are mapped in the latent space. The specific method consists of the following three parts:

**Mapping to latent space.** To explore the implicit connections between nodes, we first need a mapping function $f$ to map the node $v$ into a vector $z_v \in \mathbb{R}^d$. We construct logical relationships in Section III-B, so we aim to depict the implicit connections in another aspect, such as the semantic connection of the nodes. Thus, we choose the same embedding approach as in Section III-B, i.e., we use SBERT to map the node information to the latent continuous space. This mapping function reduces unnecessary computation while expressing semantics, and the nodes with similar semantics should be closer in the latent continuous space.

**Depicting implicit connection.** We first use Euclidean distances to measure the distance between nodes, a shorter distance indicates a higher correlation. We set a threshold $\rho$ that the nodes have a more significant implicit connection if their distances are less than $\rho$. However, the quality and style of news vary drastically. The proportions of connections in news items are not balanced. Thus, we set another parameter $\beta$, to control the number of implicit semantic relations in each graph. We update the threshold $\widetilde{\rho}$ as follows:

$$\widetilde{\rho} = min(\rho, P_\beta), \tag{2}$$

where $\beta \in [0, 100]$, and $P_\beta$ is $\beta$-th percentile of $D \in \mathbb{R}^{n\times n}$, where $D_{i,j}$ denotes the distance between $i$th node and $j$th node in latent space.

Finally, the adjacency matrix $A^{(1)} \in \mathbb{R}^{n\times n}$ is a Boolean matrix to depict the semantic relation and each element can be defined as:

$$A^{(1)}_{i,j} = \begin{cases} 1 & \text{if} \quad d_{i,j} \le \widetilde{\rho} \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

In addition, we find the semantic graph complements the logical graph. Extracting the coordinate relations is not easy for some short news, which leads to constructing long-range graphs with only continuation relationships, equivalent to the typical sequence structure. Mining the implicit semantic relationships can address the limitation.

**Fusing logical and semantic graph.** The final step in our graph construction is to incorporate the logic graph and the semantic graph together. $A'^{(0)}$ is the adjacency matrix of the logical graph in the news depicting the logical relations while $A^{(1)}$ is the adjacency matrix of the semantic graph in the news depicting the implicit semantic relationships. They share the same nodes $V$ and node features $X$, with differing edges $E$ only. We fuse the logical and semantic graph by taking the union of the edges of the two graphs, and apply function $max()$ to produce the final adjacency matrix:

$$A_{i,j} = max(A'^{(0)}_{i,j}, A^{(1)}_{i,j}). \tag{4}$$

If node $i$ and node $j$ have an edge in either of the two graphs, they have an edge in the fusion graph. We show the whole process of the graph construction in Algorithm 1.

### D. Graph Representation

We need to encode graphs for subsequent stock movement prediction after generating them. In this section, we adopt GNN and attention mechanism for the purpose.

**Updating node features.** For each graph $G = (V, E)$, we specify a two-layer GCN [31] to incorporate each node feature with its 1-hop and 2-hop neighbors features. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the nodes features of $G$, where $x_i \in \mathbb{R}^d$ denote the $i$th node features as well as the input to the GCN layer, $n$ is the number of nodes, and $d$ is the number of features in each node. Specifically, we update node features at layer $l$ as follows:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}), \tag{5}$$

where $\widetilde{A}$ is the adjacency matrix of the graph G with added self-connections, and $W^{(l)\in\mathbb{R}^{d_l\times d_{l+1}}}$ is a layer-specific learnable weight matrix. $\sigma$ denotes an activation function. $H^{(l)} \in \mathbb{R}^{N\times d_l}$ is the matrix of activations in the $l$ th layer and $d_l$ is the number of features in each node at layer $l$, $H^{(0)} = X$. We also apply the dropout after the GCN learning, which are not shown in Eq 5 for simplicity.

**Generating graph-level representations.** After the nodes in each graph are sufficiently updated, we aggregate them into a graph-level representation for the downstream task. We apply an attention mechanism to compress the graph into a vector with adaptive weights so that we can reduce the impact of irrelevant information in the news which may degrade

**Algorithm 1** Logical Graph Construction

$N = \{s_1, \cdots, s_n\}$: the news composed of $n$ sentences.
$s_i$: the $i$th sentence in $N$.
$s_i = \{c_{i1}, \cdots, c_{im}\}$: the sentence $i$ composed of $m$ clauses.
$O = \{o_1, \cdots, o_k\}$: the graph composed of $k$ groups of coordinate relations.
$o_i = \{\widetilde{c}_1, \cdots, \widetilde{c}_v\}$: the $i$th group that contains $v$ sets of clause clusters where each set is coordinated with others.
$A^{(0)}_{i,j}$: the adjacent matrix between clauses with only continuation relationship
$A^{(1)}_{i,j}$: the adjacent matrix with both continuation relationships and coordinate relationships
Initialization: $A^{(0)}_{i,j} = A^{(1)}_{i,j} = [0]$
**for** each news $N$ **do**
  $--$Use commas and semicolons to split $N$ into sentences $s_i$ and clauses $c_{ij}$:
  **for** each punctuation $p_i$ in $N$ **do**
    **if** $p_i$ is commas or semicolons **then**
      split $N$ by $p_i$
    **end if**
  **end for**
  $--$Add continuation relationship between adjacent clauses:
  **for** i,j in $A^{(0)}_{i,j}$ **do**
    $A^{(0)}_{i,j} = 1$ if $|i - j| = 1$ else $A^{(0)}_{i,j} = 0$
  **end for**
  $--$Use punctuation and Conjunction word templates to get $O, o_i, \widetilde{c}_i$:
  **for** each sentence $s_m$ in $N$ **do**
    **for** each clause $c_i$ in sentence $s_m$ **do**
      **if** the punctuation between $c_i$ and $c_j$ is semicolon or Conjunction word is in the word templates **then**
        $O$.append($\{\{c_i, \cdots, c_{j-1}\}, \{c_j, \cdots c_f\}\}$)
        where $c_f$ is the last node in $s_m$.
      **end if**
    **end for**
  **end for**
  **for** each group $i$ in $O$ **do**
    **for** each clause set $j$ in $O_i$ **do**
      remove edges between $c_{j1}$ and $c_{\widetilde{s}}$, where $j$ is the $j$th clause set in $O_i$ and $c_{\widetilde{s}}$ is the previous node of $c_{j1}$.
      add edges between $c_{j1}$ and $p$, where $p$ is the previous node of $c_{start}$, and $c_{start}$ is the first node of $O_i$
      remove edges between $c_{je}$ and $c_{\widetilde{e}}$, where $j$ is the $j$th clause set in $O_i$, $e$ is the last index of clause set $j$, and $c_{\widetilde{e}}$ is the next node of $c_{je}$
      add edges between $c_{je}$ and node $e$, where $e$ is the next node of $c_{end}$, and $c_{end}$ is the last node of $O_i$
    **end for**
  **end for**
**end for**
**Output:** $A^{(1)}_{i,j}$

the prediction accuracy. Specifically, we aggregate the node features as:

$$\widetilde{\alpha}_i = {u^{(a)}}^\top v_i, \tag{6}$$

$$a = \sum_{i=1}^{n} \alpha_i v_i, \quad \alpha_i = \frac{exp(\widetilde{\alpha}_i)}{\sum_{j=1}^{n} exp(\widetilde{\alpha}_j)}, \tag{7}$$

where $u^{(a)} \in \mathbb{R}^{d'}$ is the parameter to be learned. $v_i$ is the $i$th node features updated by GCN. $a$ is the aggregated representation of the overall node features.

### *E. Stock Movement Prediction*

The last step is to feed the graph representations of news items to the final predictor. Meanwhile, we incorporate global market information and historical prices of stocks in this section to make our model more compatible with the task of stock movement prediction. We categorize this section into a) aggregating global market information, b)incorporating historical prices, and c)final prediction.

**Aggregating global market information.** Stock movement is not only determined by the individual stock trend; the global trend also plays an important role. We apply the gate structure proposed in LSTM [11] to filter information about the global movement of the market and combine it with news information. First, we generate a vector $a_g$ of market index (e.g., CSI 100 for the Chinese market) news, and the way of generating vector is the same with individual stock news. Then, we aggregate global news vector $a_g$ and the stock news vector $a_s$ by gate structure:

$$f = \sigma(W^{(g)}[a_g || a_s] + b^{(g)}), \tag{8}$$

$$\widetilde{a} = f \cdot a_g + a_s, \tag{9}$$

where $w^{(g)} \in \mathbb{R}^{d' \times 2d'}$ and $b^{(g)} \in \mathbb{R}^{d'}$ are weight matrix and weight vector, respectively, and $\sigma$ represents the sigmoid function. $\widetilde{a}$ is a vector that combines information about individual stock news and index news.

**Incorporating historical prices** We adapt the attention LSTM [32] to summarize the sequential historical prices of each stock in each day into a vector. First, we refer to adv-LSTM [9] to generate a feature vector $z_{ut} \in \mathbb{R}^e$ to extract price information of stock $u$ at day $t$ and we give the matrix $Z_{ut} = [z_{u(t-k+1)}, \ldots, z_{ut}] \in \mathbb{R}^{e \times k}$ which represents the sequential features in the lag of past $k$ time-steps. Then, we utilize LSTM to get the hidden states $\{h_1, \ldots, h_k\}$ and apply the attention mechanism to compress the hidden states of all LSTM cells:

$$\widetilde{h} = \sum_{i=1}^{k} \hat{\alpha}_i h_i, \quad \hat{\alpha}_i = \frac{exp(h_i^\top h_k)}{\sum_{j=1}^{k} exp(h_j^\top h_k)}. \tag{10}$$

The vector $\widetilde{h}$ is the final representation of historical price features.

**Final prediction.** We lastly concatenate the news representation $\widetilde{a}$ with the price representation $\widetilde{h}$ into the final latent representation and use a linear layer to produce the final predictions as:

$$\hat{y} = W^{(y)}[\widetilde{a} || \widetilde{h}] + b^{(y)}. \tag{11}$$

The training of our model is done to minimize the objective function:

$$\sum_{u} L(y_u, \hat{y_u}) + \lambda(\|\Theta\|_F^2). \tag{12}$$

The first item is the cross-entropy loss between its predictions and labels which is widely used in classification tasks. The second item is the L2 regularization of model parameters to avoid overfitting. The cross-entropy loss is computed as:

$$L(y_u, \hat{y_u}) = y_u log\hat{y_u} + (1 - y_u)log(1 - \hat{y_u}) \tag{13}$$

## IV. Experiments

We detail our experimental setup, results, and case study in this section.

### A. Experimental Setup

We present the dataset, competitors, evaluation metrics, and implementation details for our experiments. Our experiments were done in a workstation with NVIDIA GTX 2080 Ti.

**Dataset.** To gauge the validity of FLAG, we use a new dataset collected from the Hongkong stock market. The dataset contains the stock prices and news of the companies listed on the Hang Seng Index (HSI) from 01/01/2019 to 31/10/2020. Since the companies on the HSI list are updated every three months, we selected 64 companies that remained on the list between 06/12/2021 and 01/03/2022.

Our dataset has two main components: a news dataset and a historical price dataset. Both of them are obtained from Uqer[1]. Inspired by StockNet [18], We also remove the samples with the movement percents between -0.5% and 0.55% . The rest of samples with the movement percents $\leq -0.5\%$ and $\geq 0.55\%$are labeled with 1 and 0, respectively. Finally, we use the data from 01/01/2019 to 30/06/2020 as the training, containing 109309 news. The data from 01/07/2020 to 31/08/2020 which contains 23841 news are for development. The data between 01/09/2020 and 31/10/2020 which contains 17651 news are for test.

**Features.** For textual data, we use Sentence transformer [29] to encode node information into node embeddings. Specifically, we select the model *paraphrase-MiniLM-L6-v2* to generate a 384-D vector for each node. For historical prices, we refer to adv-LSTM [9] that use 11 temporal features to describe the trend of stock in one day.

**Evaluation.** we measure the model performance by two metrics: the accuracy (ACC) and the Matthews correlation coefficient (MCC). ACC is the most popular metric for stock movement prediction, which measures the upward or downward differences in the predicted trends compared to the actual changes. MCC is a balanced metric to address the limitation of exhibiting bias, which can be used even if the classes are of very different sizes.

**Competitors.** We compare FLAG with five baselines in different genres. The baselines are described as follows:

- **RAND** is a naive predictor that randomly predicts rise or fall of a stock.
- **ALSTM** [32] is a simplest model that combines temporal representations with attention mechanism for prediction.
- **DTML** [10] is the previous state-of-the-art model using only price signals that learns the correlations between stocks in an end-to-end way for stock prediction.
- **HCAN** [19] is a state-of-the-art model using news title and content for stock prediction with hierarchical attention.
- **TLGNN** [4] is a graph based approach which uses semantic similarity to construct text-level graph for classification.

**Implementation details.** We implement FLAG with PyTorch, and all our variants are trained with a batch size of 1024. We search the hyperparameters as follows: we set the distance threshold $\rho$ and the proportion threshold $\beta$ in latent space to [1, 2, 5] and [10, 20, 50], respectively. In textual embedding, we set the output sizes of two-layer GCN $d^1$ and $d^2$ to 128 and 32, respectively. For historical prices, we set the time lag $k$ as 5 and the hidden layer as 32. In addition, we use the learning rate in [0.001, 0.002], the dropout rate to 0.3, the weight decay of 0.04, and the number of epochs to 100. Moreover, We use each news item as a sample for model training. For model evaluation, we use a voting mechanism for each stock's news set to determine the daily rise and fall.

For competitors, we implement ALSTM and DTML using the default settings in their public implementations. We implement HCAN with the same encoder of the model paraphrase-MiniLM-L6-v2 to get the embeddings of news titles and news content. For TLGNN, we use their approach to get graph representation by semantic similarity, and the rest part is the same as FLAG.

### B. Prediction Accuracy

We show the best performance of the baselines and our model in our new collected dataset in Table I. Clearly, our model achieves state-of-art accuracy in our dataset with both ACC and MCC enhancement.

As shown in Table I, the models that incorporate news information (HCAN, TLGNN, FLAG) generally perform better than the models that use only historical prices for prediction (ALSTM, DTML), with at least 1.82% improvement on ACC. Moreover, the performance in MCC is even more dramatically enhanced, with at least a 0.0711 improvement. This proves that external information related to stock may be helpful for stock prediction.

Compared with TLGNN and FLAG, HCAN only utilizes the attention mechanism to filter out the valuable information in the news and ignores the interaction between sentences. Therefore, the model performs 0.52% and 3.52% worse than TLGNN and FLAG, respectively. TLGNN implements interaction between sentences by using a graph-based approach. Specifically, TLGNN uses semantic similarity to construct graphs that allow semantically related sentences to interact. However, similar to HAN, there is a lack of depiction of structured news events, which hinders the extraction of complete news information.

[1] https://uqer.datayes.com/

TABLE I
PERFORMANCE OF BASELINES AND FLAG VARIATIONS IN ACCURACY AND MCC.

| Baseline models | ACC | MCC | FLAG variations | ACC | MCC |
|---|---|---|---|---|---|
| RAND | 50.85 | 0.0011 | FLAG-LG-SG-GI | 53.87 | 0.0345 |
| ALSTM [32] | 53.52 | 0.0035 | FLAG-GI | 57.56 | 0.1588 |
| DTML [10] | 54.28 | 0.0216 | FLAG-LG | 59.10 | 0.1701 |
| HCAN [19] | 56.10 | 0.0927 | FLAG-SG | 59.27 | 0.1712 |
| TLGNN [4] | 56.62 | 0.1094 | FLAG | **59.62** | **0.1769** |

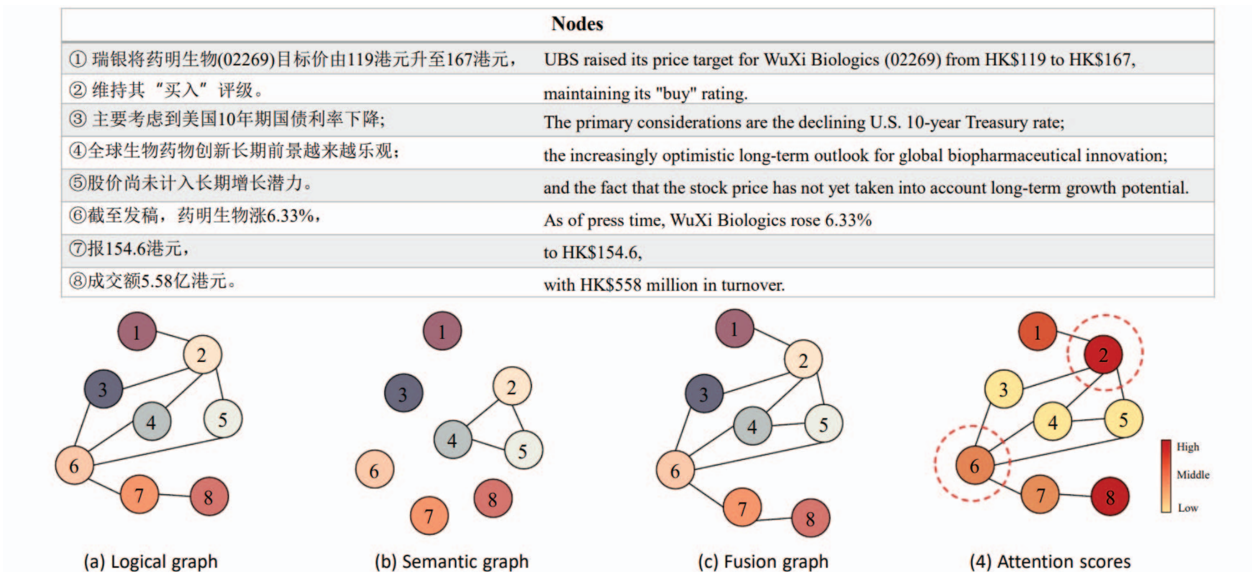


Fig. 3. The attention scores between nodes, generated by FLAG in Eq 7 for the news of WuXi Biologicals on July 13, 2020. The figure illustrate that the attention scores are related to the logical structure.

FLAG takes a further step to capture the logical structure information in the news by constructing a logical graph of news, efficiently depicting the logical relationships between clauses. In addition, we depict the continuation relationship in our model to combine the contextual information. Overall, FLAG makes better news representations than other models, which is why it achieves the highest ACC of 59.62% and the highest MCC of 0.1769 in our experiments.

### C. Ablation Study

To illustrate the effects of every component of FLAG, we compare the accuracy of FLAG and its variants where each of the three modules is removed in Table I:

- **FLAG-LG-SG-GI:** FLAG using only technical analysis.
- **FLAG-LG:** FLAG without the logical graph construction (LG).
- **FLAG-SG:** FLAG without the semantic graph construction (SG).
- **FLAG-GI:** FLAG without the global market information (GI).

The ablation study shows that each part of FLAG improves the accuracy, and the complete FLAG achieves the best performance. Among them, GI has a higher accuracy improvement(1.54%) than the other two parts, as GI provides a new information source depicting the global market trend. LG describes the logical structure of the news, and SG mines the semantic connections between nodes. The three parts work together to make FLAG more comprehensive for news representation.

### D. Case Study

To explain the role of logical structure construction for graph representation specifically, we chose the news of WuXi Biologics on July 13, 2020, as a case study. Fig 3 shows the whole process of constructing the news into a graph. We have two observations from the figure. First, the attention scores of nodes with coordinate relationships (node 3,4,5) are usually lower than other nodes. This is natural, since nodes with coordinate relationships usually elaborate on specific aspects of an event or argument in the news, where the nodes' content focuses on the details. Second, the nodes adjacent or close to those nodes in coordinate relationships (node 2,6) usually have higher attention scores. Because these nodes typically have the role of summarization. And through the graph neural network, these nodes can learn the information of the coordinate relation nodes and aggregate them to contain more valuable information. The case study proves that it is meaningful to consider the logical structure of the news.

## V. CONCLUSION

In this paper, we propose FLAG to better represent news for stock movement prediction. We constructed two graphs for each news item, a logical graph based on continuation and coordinate relations to describe the contextual information and logical structure, and a semantic graph based on semantic encoding to explore the implicit semantic information. We fuse these two graphs to get a new representation of the news. Furthermore, we combine news representations with global market information and historical prices for final prediction. Experiment results show that FLAG achieves the state-of-the-art accuracy on the dataset we collected and improves the quality of news representation. Future work will further consider integrating time series with news representations, for example, predicting stock movement rises and falls jointly from news within a specific period.

## VI. ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China (#71801232, #71932008). And we would like to express our sincere gratitude to all the editors and reviewers.

## REFERENCES


[1] U. Walińska and J. Potoniec, "Urszula walińska at SemEval-2020 task 8: Fusion of text and image features using LSTM and VGG16 for memotion analysis," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 1215–1220. [Online]. Available: https://aclanthology.org/2020.semeval-1.161

[2] M. Hagenau, M. Liebmann, and D. Neumann, "Automated news reading: Stock price prediction based on financial news using context-capturing features," *Decision Support Systems*, vol. 55, no. 3, pp. 685–697, 2013.

[3] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[4] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3444–3450.

[5] H. Zhang, C. Wang, Z. Wang, Z. Duan, B. Chen, M. Zhou, R. Henao, and L. Carin, "Learning hierarchical document graphs from multilevel sentence relations," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[6] H. Li, Y. Shen, and Y. Zhu, "Stock price prediction using attention-based multi-input lstm," in *Asian conference on machine learning*. PMLR, 2018, pp. 454–469.

[7] L.-C. Cheng, Y.-H. Huang, and M.-E. Wu, "Applied attention-based lstm neural networks in stock prediction," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 4716–4718.

[8] J. Wang, T. Sun, B. Liu, Y. Cao, and H. Zhu, "Clvsa: a convolutional lstm based variational sequence-to-sequence model with attention for predicting trends of financial markets," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 3705–3711.

[9] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T.-S. Chua, "Enhancing stock movement prediction with adversarial training," in *IJCAI*, 2019.

[10] J. Yoo, Y. Soun, Y.-c. Park, and U. Kang, "Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2037–2045.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv*, 2017.

[13] R. Ren, D. D. Wu, and T. Liu, "Forecasting stock market movement direction using sentiment analysis and support vector machine," *IEEE Systems Journal*, vol. 13, no. 1, pp. 760–770, 2018.

[14] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, "Stock price prediction using news sentiment analysis," in *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2019, pp. 205–208.

[15] X. Li, P. Wu, and W. Wang, "Incorporating stock prices and news sentiments for stock market prediction: A case of hong kong," *Information Processing & Management*, vol. 57, no. 5, p. 102212, 2020.

[16] Q. Li, J. Tan, J. Wang, and H. Chen, "A multimodal event-driven lstm model for stock prediction using online news," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 10, pp. 3323–3337, 2020.

[17] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Using structured events to predict stock price movement: An empirical investigation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[18] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1970–1979.

[19] Q. Liu, X. Cheng, S. Su, and S. Zhu, "Hierarchical complementary attention network for predicting stock price movements with news," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1603–1606.

[20] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, mar 2019. [Online]. Available: https://doi.org/10.1145/3309547

[21] S. Saha, J. Gao, and R. Gerlach, "Stock ranking prediction using list-wise approach and node embedding technique," *IEEE Access*, vol. 9, pp. 88 981–88 996, 2021.

[22] J. Gao, X. Ying, C. Xu, J. Wang, S. Zhang, and Z. Li, "Graph-based stock recommendation by time-aware relational attention network," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 1, jul 2021. [Online]. Available: https://doi.org/10.1145/3451397

[23] Y. Chen, Z. Wei, and X. Huang, "Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1655–1658.

[24] J. Long, Z. Chen, W. He, T. Wu, and J. Ren, "An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in chinese stock exchange market," *Applied Soft Computing*, vol. 91, p. 106205, 2020.

[25] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1063–1072.

[26] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, S. Y. Philip, and L. He, "Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2505–2519, 2019.

[27] B. Liu, D. Niu, H. Wei, J. Lin, Y. He, K. Lai, and Y. Xu, "Matching article pairs with graphical decomposition and convolutions," *arXiv preprint arXiv:1802.07459*, 2018.

[28] W. Li, J. Xu, Y. He, S. Yan, Y. Wu, and X. Sun, "Coherent comments generation for Chinese articles with a graph-to-sequence model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4843–4852. [Online]. Available: https://aclanthology.org/P19-1479

[29] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: http://arxiv.org/abs/1908.10084

[30] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," *arXiv preprint arXiv:2002.05287*, 2020.

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[32] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.