THE UNIVERSITY
of ADELAIDE

Machine Learning and Natural Language Processing in Stock Prediction

by

Jinan Zou

A thesis submitted for the degree of

Doctor of Philosophy

March 20, 2023

Australian Institute for Machine Learning (AIML)

The University of Adelaide

Contents

List of Tables

List of Figures

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works. I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Jinan Zou March 20, 2023

Acknowledgements

This thesis represents my work in more than three years of dedication at the University of Adelaide, specifically within the Australian Institute for Machine Learning (AIML). First of all, I would like to show my deepest gratitude to my principal supervisor, Prof. Javen Qinfeng Shi, who guided me through the whole project with great patience and wisdom on academic research and how to be an honest and strong person. He is the light that illuminates my way forward. I am so grateful to have him as my supervisor.

I want to thank Dr Lingqiao Liu for his constructive advice and guidance on research, which made me more committed and passionate about doing research. His knowledge and attitude towards research influenced me a lot in my research.

I would also thank Dr Ehsan Abbasnejad, Prof. Qingsen Yan, and Dr Yuankai Qi for their instructive advice with all kinds of difficulties. I am also grateful to all my supportive friends and colleagues at the University of Adelaide.

Finally, I would like to take this opportunity to thank my parents, my grandparents, and my partner. I am grateful for the comfortable and happy life they created to allow me to focus on my research.

Publications

This thesis contains the following works that have been published or prepared to be submitted):

- **Jinan Zou**, Haiyao Cao, Lingqiao Liu, Yuhao Lin, Ehsan Abbasnejad, Javen Qinfeng Shi. Astock: A New Dataset and Automated Stock Trading based on Stock-specific News Analyzing Model. In *Fourth Workshop on Financial Technology and Natural Language Processing*, 2022.

- **Jinan Zou**, Maihao Guo, Yu Tian, Yuhao Lin, Haiyao Cao, Lingqiao Liu, Ehsan Abbasnejad, Javen Qinfeng Shi. Semantic Role Labeling Guided Out-of-distribution Detection. *ACL, Under Review*, 2023.

- **Jinan Zou**, Yanxi Liu, Lingqiao Liu, Yuankai Cui, Javen Qinfeng Shi. Rethinking Document Event Extraction: A Generative Model is All You Need?. *ACL, Under Review*, 2023.

In addition, I have the following papers not included in this thesis:

- **Jinan Zou**, Qingying Zhao, Yang Jiao, Haiyao Cao, Yanxi Liu, Ehsan Abbasnejad, Lingqiao Liu, Javen Qinfeng Shi. Stock Market Prediction via Deep Learning Techniques: A Survey. *ACM Computing Surveys, Under Review*, 2022.

- **Jinan Zou**, Haiyao Cao, Yanxi Liu, Lingqiao Liu, Ehsan Abbasnejad, Javen Qinfeng Shi. UOA at the FinNLP-2022 ERAI Task: Leveraging the Class Label Description for Financial Opinion Mining. In *Fourth Workshop on Financial Technology and Natural Language Processing*, 2022.
- Yuhao Lin, Haiming Xu, Lingqiao Liu, **Jinan Zou**, Javen Qinfeng Shi. Revisiting Image Reconstruction for Semi-supervised Semantic Segmentation.*CVPR, Under Review*, 2023.

Abstract

In this thesis, we first study the two ill-posed natural language processing tasks related to stock prediction, *i.e.* stock movement prediction and financial document-level event extraction. While implementing stock prediction and event extraction, we encountered difficulties that could be resolved by utilizing out-of-distribution detection. Consequently, we presented a new approach for out-of-distribution detection, which is the third focus of this thesis.

First, we systematically build a platform to study the NLP-aided stock auto-trading algorithms. Our platform is characterized by three features: (1) We provide financial news for each specific stock. (2) We provide various stock factors for each stock. (3) We evaluate performance from more financial-relevant metrics. Such a design allows us to develop and evaluate NLP-aided stock auto-trading algorithms in a more realistic setting. We also propose a system to automatically learn a good feature representation from various input information. The key to our algorithm is a method called semantic role labelling Pooling (SRLP), which leverages Semantic Role Labeling (SRL) to create a compact representation of each news paragraph. Based on SRLP, we further incorporate other stock factors to make the stock movement prediction. In addition, we propose a self-supervised learning strategy based on SRLP to enhance the out-of-distribution generalization performance of our system. Through our experimental study, we show that the proposed method achieves better performance and outperforms all strong baselines' annualized rate of return as well as the maximum drawdown in back-testing.

Second, we propose a generative solution for document-level event extraction that takes

into account recent developments in generative event extraction, which have been successful at the sentence level but have not yet been explored for document-level extraction. Our proposed solution includes an encoding scheme to capture entity-to-document level information and a decoding scheme that takes into account all relevant contexts. Extensive experimental results demonstrate that our generative-based solution can perform as well as state-of-the-art methods that use specialized structures for document event extraction. This allows our method to serve as an easy-to-use and strong baseline for future research in this area.

Finally, we propose a new unsupervised OOD detection model that separates, extracts, and learns the semantic role labelling guided fine-grained local feature representation from different sentence arguments and the full sentence using a margin-based contrastive loss. Then we demonstrate the benefit of applying a self-supervised approach to enhance such global-local feature learning by predicting the SRL extracted role. We conduct our experiments and achieve state-of-the-art performance on out-of-distribution benchmarks.

Chapter 1

Introduction

Stock prediction has gained popularity in machine learning with the advancement in technology and social media. Stock prediction based on Natural Language Processing techniques is a promising solution since text information, e.g., tweets or financial announcements, strongly correlate with stock prices. Stock prediction is a complex and challenging task that involves using various data sources and techniques to make predictions about the future movements of a stock's price. The fluctuation of stock prices is influenced by a complex array of factors, including company earnings reports, national policies, influential shareholders, and expert speculations on current events. Therefore, there is a need to utilize specific natural language language techniques for stock market prediction tasks, such as stock movement prediction and financial document-level event extraction. The task of stock movement prediction involves utilizing historical data and financial information to forecast the future movements of a stock's price. On the other hand, financial document-level event extraction is the process of identifying important financial information from documents such as news articles or company announcements.

For stock movement prediction, existing approaches [12, 56, 93, 124, 150, 152, 162] are based on market sentiment analysis and use news to predict the related securities' price movement on the following trading day(s). Despite the limited success in those studies, the current works are still far from realistic for three reasons: Firstly, previous methods ignore the financial factors, which play a key role in practical trading. Secondly, these models are evaluated only on intermediate performance metrics, e.g., stock movement prediction accuracy. It is unclear how well they can support a practical trading system to make sufficient profit. Thirdly, the model often performs well on in-distribution data but fails in out-of-distribution data, which requires high generalization ability.

In recent years, with the rising trend of digitization in the finance domain, Event Extraction has become an increasingly important accelerator to business development. Continuous economic growth has witnessed exploding volumes of digital financial documents, such as financial announcements from listed companies. Such large amounts of announcements call event extraction to assist investors in extracting valuable structured information to sense emerging risks and find profitable opportunities timely in the stock market. Document-level event extraction is a challenging task in NLP because it requires a thorough comprehension of the document and an aggregated ability to assemble arguments across multiple sentences.

In addition, the real world is open and full of unknowns, especially in the stock market, presenting significant challenges for a news analyzing model that must reliably handle diverse inputs. We observed that out-of-distribution(OOD) uncertainty arises when a model sees input that differs from its training data distribution. The prediction may lead to a failed investment decision, which the model should not predict. Moreover, the most benchmark dataset of document-level event extraction was created automatically by the distant supervision approach, which might have a relatively higher annotation error rate. This scenario requires the model to reject inputs that are semantically different from the training distribution, therefore, should not be predicted by the model.

1.1 Motivation

Stemming from the success achieved by the aforementioned modern methods, the overall objective of this thesis is to develop transformer-based methods to solve the stock movement prediction, out-of-distribution detection, and document-level extraction tasks. Though there are many successful applications of deep learning in stock market prediction-related tasks, it brings more difficulties and challenges as well, including:

1. Factor investing is an investment approach that targets quantifiable firm characteristics or factors that explain the differences in stock returns. Stock characteristics that may be included in a factor-based system contain low volatility, value, momentum, asset growth, and profitability. Factor-based strategies may help investors meet particular investment objectives—such as potentially improving returns or reducing risk over the long term. However, precious stock movement prediction datasets and methods ignore the stock factors which play a crucial role in practical trading. The first research question is: ***Would it be helpful to use stock factors for stock forecasting?***

2. In addition to general stock factors, another important indicator is the financial announcements related to the company's performance. The performance of a company will be reflected in the stock price. If we can understand the performance of the company well, it can help to predict the stock. The current method is that the investors look at the earnings report to judge the rise or fall of the stock based on subjective judgment. The third question is: ***Can we automatically extract the key information of the stock in the financial announcements?***

3. The stock market is constantly changing and full of unknowns, presenting significant challenges for stock movement prediction models that must reliably handle diverse inputs such as news which is a common input indicator in stock movement prediction. Out-of-distribution(OOD) uncertainty arises when a model sees input news that differs

from its training data distribution. The prediction for an out-distribution sample may lead to a failed investment decision, which requires accurate ood detection to solve this problem. However, previous ood detection works in NLP identify the ood instance by leveraging a single global feature embedding to represent the sentence, which cannot characterize subtle ood patterns well. The second question is: ***Can we find a better method to characterize subtle ood patterns?***

1.2 Main Contributions and Thesis Outline

1.2.1 Main Contributions

This thesis aims to meet the needs of building realistic stock prediction tasks using NLP and machine learning. In an effort to address some of the challenges, we are devoted to developing algorithms for stock movement prediction, document-level event extraction, and out-of-distribution detection in the following perspectives:

- For the stock movement prediction task, our Astock platform is characterized by three features: (1) We provide financial news for each specific stock. (2) We provide various stock factors for each stock. (3) We evaluate performance from more financial-relevant metrics. Such a design allows us to develop and evaluate NLP-aided stock auto-trading algorithms in a more realistic setting. In addition to designing an evaluation platform and dataset collection, we also made a technical contribution by proposing a self-supervised system to automatically learn a good feature representation from various input information.

- For document-level event extraction task, we explore the current methods for extracting events at the document level, which often involve custom-designed networks and processes. We question whether such extensive efforts are truly necessary for this task. Our research is motivated by recent developments in generative event extraction, which have shown success in sentence-level extraction but have yet to be explored for document-level extraction. To fill this gap, we propose a generative solution for financial document-level event extraction, which is more challenging due to the presence of scattered arguments and multiple events. We introduce an encoding scheme to capture entity-to-document level information and a decoding scheme that makes the generative process aware of all relevant contexts. Our results indicate that using our method, a generative-based solution can perform as well as state-of-the-art methods that use a specialized structure for document event extraction, providing an easy-to-use, strong baseline for future research.

- For the out-of-distribution detection task, we present a novel unsupervised approach for detecting out-of-distribution samples, called Semantic Role Labeling Guided Out-of-distribution Detection (SRLOOD), which leverages semantic role labeling (SRL) to extract and learn fine-grained local feature representations from various sentence arguments, as well as global feature representations of the entire sentence. Our approach employs a margin-based contrastive loss to separate and extract these features. A novel self-supervised approach is also introduced to enhance such global-local feature learning by predicting the SRL extracted role. The resulting model achieves SOTA performance on four OOD benchmarks, indicating the effectiveness of our approach.

1.2.2 Thesis outline

The structure of this thesis is organized as follows.

In Chapter 2, we first review previous state-of-the-art methods for stock movement prediction, OOD detection and event extraction. Also, we provide a detailed literature review on the basics of machine learning methods.

In Chapter 3, we study the problem of NLP-based stock prediction and make two major contributions to this field: (1) We develop a new dataset called AStock, featured by its large number of stocks, stock-relevant news, and availability of various financial factors. (2) We propose a new stock movement prediction system based on two novel techniques.

In Chapter 4, we propose a generative solution for document-level event extraction that takes into account recent developments in generative event extraction. Our proposed solution includes an encoding scheme to capture entity-to-document level information and a decoding scheme that takes into account all relevant contexts.

In Chapter 5, we propose a simple yet effective approach to learn from both global and local fine-grained feature representationto detect OOD instances in NLP.

In Chapter 6, the conclusion and the potential future works are discussed.

1.3 Problem Formulation

In this thesis, we focus on NLP techniques to solve the tasks of (1) stock movement prediction (2) document-level event extraction for the financial announcement (3) out-of-distribution detection on the text data. The purpose is to build a system to predict the stock movement accurately, detect the ood instances that can cause severe semantic shift problems, and automatically extract the financial event records from a whole document.

1.3.1 Stock Movement Prediction

The problem of stock movement prediction can be formulated as a supervised learning problem, where the goal is to predict the stock movement direction of a stock based on its historical data. This can be done using various machine-learning techniques, such as regression or classification algorithms. The input information could be the historical data of the stock, such as its past prices, trading volumes, and other relevant financial indicators such as company announcements or stock factors. We formalize movement based on the difference between the adjusted closing prices of the stock $s \in S$ on trading days d and $d-1$. We can formulate stock movement prediction as a classification problem. Give stock $s \in S$, and historical data for stock s over a lookback window of T days over the day range $[t-T, t-1]$, we define the price movement of stock s from day $t-1$ to t as :

$$Y_t = \begin{cases} 0, & p_c^d < p_c^{d-1} \\ 1, & p_c^d >= p_c^{d-1} \end{cases} \tag{1.1}$$

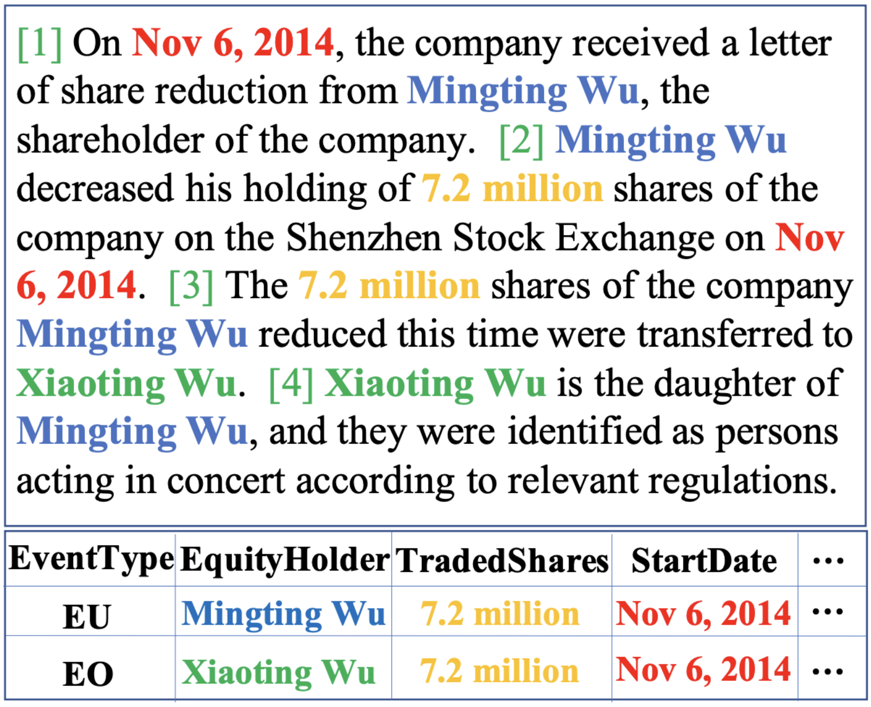
where p_c^d represents the widely used adjusted closing price [1] of a given stock on day t. In this formula 1.1, 1 represents a rising trend in the price, and 0 represents a price downfall trend. In addition, some works might include an extra neutral class representing the little change in price.

1.3.2 Document-level Event Extraction

Event extraction aims to identify the specific type of events and extract the corresponding event arguments from given texts, and event extraction tasks can be divided into sentence-level event extraction and document-level event extraction. Despite successful efforts to extract events within a sentence, many real-world applications such as finance, legislation, or health require DEE, where the event information scatters across multiple sentences in a whole article. Figure 1.1 illustrates an example that one *Equity Overweight*(EO) *Equity Underweight* (EU) event records are extracted from a financial announcement document. Extracting information about the EU event is easier because all the related arguments are in the same sentence. However, the arguments for the EO record appear in different sentences,

[1] https://www.investopedia.com/terms/a/adjusted_closing_price.asp

making it more difficult to identify the event. It is important to consider the relationships between sentences and entities when trying to identify events in a document. In addition, a document may contain multiple related events at the same time, and being able to understand the connections between these events is crucial for successful extraction. It is a major challenge to model the interdependence between correlated events in this task, as demonstrated in the example where the two events are related because they involve the same transaction and have the same start date.



[1] On **Nov 6, 2014**, the company received a letter of share reduction from **Mingting Wu**, the shareholder of the company. [2] **Mingting Wu** decreased his holding of **7.2 million** shares of the company on the Shenzhen Stock Exchange on **Nov 6, 2014**. [3] The **7.2 million** shares of the company **Mingting Wu** reduced this time were transferred to **Xiaoting Wu**. [4] **Xiaoting Wu** is the daughter of **Mingting Wu**, and they were identified as persons acting in concert according to relevant regulations.

EventType	EquityHolder	TradedShares	StartDate	···
EU	Mingting Wu	7.2 million	Nov 6, 2014	···
EO	Xiaoting Wu	7.2 million	Nov 6, 2014	···

Figure 1.1: An example of a document from a Chinese dataset proposed by Zheng et al. [236] that was created to study the document-level event extraction for stock companies' announcements. The document has been translated into English for this illustration. Entity mentions are highlighted. The original document can be found in Appendix A. Due to space constraints, only four sentences and three additional roles for each event type are shown.

In document-level Event Extraction, we will be dealing with three main concepts: entity mentions, event arguments, and event records. An entity mention is a part of the text that refers to a specific entity. An event argument is an entity that plays a specific role in an event. Event roles are predetermined for each event type. An event record is an entry for a particular event type that includes the arguments for the different roles in the event. For simplicity, we will just refer to event records as ”records” in the following sections. Following Zheng et al. [236], given a document composed of sentences $D = \{S_i\}_{i=1}^{D}$ and a sentence containing a sequence of words $s_i = \{w_i\}_{j=1}^{|S_i|}$, the task aims to solve three sub-tasks: (1) **entity extraction:** extracting entities $\varepsilon = \{e_i\}_{i=1}^{|\varepsilon|}$ from a document to serve as argument candidates. An entity may be mentioned multiple times in a document. (2) **event types detection**: The goal is to identify specific types of events that are expressed in the document. (3) **event records extraction**: finding appropriate arguments for the expressed records from entities. It is worth noting that this task does not require identifying event triggers, which reduces the manual effort required for annotation and makes the potential applications of this task more widespread.

1.3.3 Out-of-distribution Detection

Out-of-distribution detection is a process that identifies data samples that come from a different distribution than the one the machine learning model was trained on. The distribution being referred to here is typically the ”label distribution”, which means that the out-of-distribution samples should not have the same labels as the training data. In other words, ood detection should not negatively impact the model’s ability to classify data from the original distribution (called the ”in-distribution” or ID), i.e., $P(Y) \neq P'(Y)$. It’s important to note that the training set typically contains multiple classes, and ood detection should not interfere with the model’s ability to classify these classes correctly.

We formally define the OOD detection task as follows: Given a primary task of natural language classification (such as sentence classification or natural language inference), we aim to develop an auxiliary function, $f(x) : \chi \rightarrow \mathbb{R}$, that assigns an OOD score to an instance x to be classified. The function should return a low score for ID instances where $y \in \gamma_{train}$ train and a high score for OOD instances where y $\notin \gamma_{train}$ y is the underlying label for x and is unknown at inference. During inference, a threshold for the OOD score can be set to filter out most OOD instances.

Chapter 2

Literature Review

In this chapter, I will go through the related works in a detailed literature. As the topics of the thesis are stock prediction on machine learning and NLP. I will first review the conventional methods and deep learning-based methods of the applications including stock movement prediction, out-of-distribution detection and event extraction. Then, I will introduce the basics of most common machine learning methods.

2.1 Conventional Methods Applications

2.1.1 Stock Movement Prediction

Conventional methods in stock movement prediction can be roughly categorised into four groups: (1) Support Vector Machines(SVM), (2) Ensemble Models, (3) Genetic Algorithms and (4) Decision Trees.

- SVM-based. Xie et al. [208] developed a method for predicting changes in stock prices using financial news. They use semantic frames to understand the context of specific sentences and identify the impact of certain companies on the stock price. To do this, they have introduced a new way of organizing information called a tree representation by using support vector machines and tree kernels. Wen et al. [204] proposed a new intelligent trading system that combines stock box theory and support vector machine algorithms to predict oscillation boxes. According to box theory, successful stock buying and selling often occurs when the price breaks out of its original oscillation box into a new one. The system uses two SVM estimators to forecast the upper and lower bounds of the price oscillation box, and then creates a trading strategy based on these forecasts to make trading decisions. Ren et al. [171] integrated sentiment analysis into a machine learning method based on support vector machine and also considered the day-of-week effect in constructing more reliable sentiment indexes. The empirical results showed that the accuracy of forecasting the movement direction of the SSE 50 Index can reach as high as 89.93% with an 18.6% increase after introducing sentiment variables. This model can help investors make more informed decisions and suggests that sentiment may contain valuable information about the fundamental values of assets and can act as a leading indicator of the stock market.

- Ensemble-based. Huang et al. [87] proposed an approach for detecting trading patterns using a biclustering algorithm. They used these patterns to predict market movements using a Naive Bayesian algorithm. They also applied the Adaboost algorithm to improve the accuracy of forecasting. Yang et al. [216] developed a new ensemble prediction model called SRAVoting for predicting stock price movement trends. To create this model, they first identified the most relevant training features using the maximal information coefficient (MIC) and then combined three well-performing classifiers: support vector machine (SVM), random forest (RF), and AdaBoost (AB). They also proposed stock buy and sell strategies for different investing periods (day-span, week, and month). Finally, they tested our model and strategies using Chinese stock price indexes and technical indicators.

- Genetic Algorithms. Hu et al. [81] used genetic algorithms to predict the price movements of 10 stocks listed on the Taiwan Stock Exchange. They compared the performance of their predictions to those made using linear and logistic regression models. They found that the genetic algorithm-based system increased the accuracy of the base models when tested using temporal validation windows. Leitao et al. [108] proposed a method for discovering patterns in financial time series data using a combination of Perceptually Important Points (PIPs), Symbolic Aggregate approximation (SAX) representation, and a Genetic Algorithm (GA). The PIPs and SAX are used to represent the time series data, and the GA is used to generate investment rules and find optimal solutions.

- Decision Trees. Previous works [17, 205, 238] used decision tree (DL) to forecast stock market, which use a subset of randomly selected input variables.

2.1.2 Event Extraction

Event Extraction is a crucial yet challenging task in information extraction research. Event extraction aims to identify the specific type of events and extract the corresponding event arguments from given texts, and event extraction tasks can be divided into sentence-level event extraction and document-level event extraction. This section reviews those conventional methods for Event Extraction, like support vector machine (SVM), maximum entropy (ME).

- **Sentence-level Event Extraction task**, Event Extraction task mainly follows the requirements of ACE event extraction task that requires to identify the event type, argument, and judge the argument role. Ahn [3] proposed a typical pipeline processing model including a nearest neighbor learning algorithm to detect the triggers and a maximum entropy learner to identify the arguments. Chieu and Ng [29] proposed a maximum entropy classifier to information extraction from semi-structured and free text.

- **Document-level Event Extraction task**, Event Extraction aims to identify event types and relevant event argument roles. Compared with sentence-level event extraction, the main difference is that it is no longer requires to identify the triggers words. Li et al. [112] proposed a joint model including trigger identification and event type classification by integer logic programming(ILP) to solve the special characteristics of Chinese event extraction. Specifically, they proposed two trigger detection models: one is based on the conditional random field (CRF) and another one is based on maximum entropy(ME). Li, Ji and Huang [113] proposed to incorporate global features which explicitly capture the dependencies of multiple triggers and arguments to predict triggers and arguments simultaneously for the sentence level event extraction. Venugopal etal. [194] proposed a model to reliably learn and process high-dimensional features though SVMs and encoding their output as low-dimensional features in Markov Logic Networks (MLNs). Some previous works [7, 94, 114, 213] used an information network to represent relations, entities, and events as an information network representation based on structured prediction.

2.1.3 Out-of-distribution Detection

The research on the problem of identifying out-of-distribution data in low-dimensional spaces has been well-studied in various contexts [165]. There are several common techniques that are frequently used for detecting out-of-distribution examples in low-dimensional spaces, including density estimation, nearest neighbor analysis, and clustering [32, 42, 62, 195]. The density estimation approach estimates the probability density of the in-distribution data and considers a test example to be out-of-distribution if it falls in an area of low density. The clustering method uses statistical distance to identify groups of similar examples, and an example is considered out-of-distribution if it is far from its nearest neighbors.

2.1.4 Limitations of Conventional Methods

Although conventional methods have been successful progress in stock movement prediction, OOD detection, and event extraction, the limitations are obvious and inevitable.

- Conventional methods are designed to work in low-dimensional spaces and may not be effective when applied to high-dimensional data such as images.
- Conventional methods may not be robust to adversarial attacks, where an attacker intentionally crafts inputs that are designed to fool the model.
- Many conventional methods rely on assumptions about the data distribution that may not hold in practice, which can limit their effectiveness.
- Difficulty handling noise and ambiguities: Conventional methods may struggle to effectively handle noise and ambiguities in the data, which can be common in real-world scenarios.
- Limited ability to capture complex patterns: Conventional methods may not be able to effectively capture complex linguistic patterns and relationships in text.
- Limited ability to adapt to new domains: Conventional methods may not be able to effectively adapt to new domains or handle shifts in the data distribution.
- Conventional methods often require the creation of hand-crafted features, hich can be time-consuming to develop and may not generalize well to new domains.

2.2 Deep Learning Based Methods Applications

2.2.1 Stock Movement Prediction

As the deep learning models have become advanced, the models in predicting the stock market have gradually changed from the conventional methods to deep learning methods, like Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Graph Neural Network (GNN). Furthermore, the latest works applied Transformer-based models.

Recurrent Neural Network Based Models RNN [174] have been used in stock movement prediction tasks as a way to analyze sequential data and make predictions based on past trends. RNNs are well-suited for this type of task because they have a ”memory” that allows them to incorporate information from previous time steps or input elements. This can be particularly useful in stock prediction tasks because past trends and patterns may have an impact on future stock movements. There are a number of ways that RNNs can be trained and used for stock movement prediction, such as by using historical stock data as input and training the RNN to predict future stock price movements. RNNs have shown promise in stock movement prediction tasks and may be a useful tool for investors and traders. An RNN can be regarded as a recurrent combination of several identical network cells, and the output of each cell is provided as input to the next cell [107]. Each cell contains a set of input, hidden, and output units. Regarding this problem, several variants of RNN have been developed, including LSTM [76], GRU [30], Bidirectional LSTM (Bi-LSTM) [64]. These models have improved the RNN and made gOOD progress in stock market prediction.

Long Short-Term Memory (LSTM). LSTM models have the advantage of long short-term memory in processing text and time series prices, which perform well in stock market prediction. LSTM solves the problem of storing information at longer time intervals based on the gradient method compared with RNN models. Akita et al. [4] proposed a Paragraph Vector method to represent the textual information and used LSTM for the prediction model. In this experiment, ten companies are indicated as ten articles, where the vector of articles is represented as P_t. These companies' prices are represented as $\{c1, c2 \ldots, c10\}$ at a single timestep t concatenate as stock prices vector N_t. The input of LSTM is the combination of P_t

and N_t. LSTM significantly outperforms the baselines, Multi-Layer Perceptron (MLP), Support Vector Regression (SVR), and Simple-RNN, for the opening prices prediction. Ma et al. [144] proposed the News2vec model, in which dense vectors represent news features. They used LSTM with the self-attention mechanism as the predictive model. This text embedding model News2vec contributes to mining the potential relationship between news and its event elements. Nelson, Pereira and Oliveira [156] used the LSTM method to realize stock movement prediction. However, the model inputs are numerical information, including stock prices and volume.

In stock prediction work, processing stock prices as a time series analysis is typical, but few people consider the potential time dependency of the data. Zhao et al. [235] proposed a time-weighted LSTM model with trend retracement, which assigned weights to data based on the temporal nearness towards the data to be predicted. The hybrid models, a combination of LSTM with other models, are proposed to get better prediction performance. Polamuri et al. [166] proposed the Generative Adversarial Network-based Hybrid Prediction Algorithm (GAN-HPA) to implement the Stock-GAN developed from a GAN-based framework. The framework takes different inputs, such as stock datasets and hyperparameters. The algorithm uses linear and non-linear models to extract features from the dataset. The hyperparameters and the pre-processing results are used as inputs to the LSTM, with the output of the generator and raw data being provided to the discriminator. Finally, they applied Bayesian approximation to adjust the parameters and update the prediction results. The main contribution of Wang et al. [203] was dynamically extracting potential representations of financial market trends from transaction data. They proposed a hybrid Convolutional LSTM-based Variational Sequence-to-Sequence model with Attention (CLVSA). The model consists of convolutional LSTM units and the sequence-to-sequence framework with self-attention and inter-attention mechanisms.

Nguyen and Yoon [161] proposed a deep transfer with related stock information (DTRSI) framework involving stock relationships in predicting stock price movement. The LSTM cells are used when pre-training on large-scale data to obtain optimized parameters, and then the base model is fine-tuned using a small amount of target data to obtain the final model. They solved the overfitting problem due to insufficient sample size and considered the relationship between stocks.

The perturbations added in the adversarial training can reasonably simulate the stock price with the stochasticity characteristics, which can improve the effectiveness of the stock movement prediction. In 2019, Feng et al. [60] proposed the attentive LSTM and adversarial training to predict the movement of the stock market. To predict multiple stock market prices simultaneously, Ding and Qin [44] proposed an associated network model consisting of three branches predicting the opening price, the lowest price, and the highest price, respectively.

Chen et al. [19] employed the Bi-LSTM to encode the stock data and financial news representations in their models, the Structured Stock Prediction Model (SSPM) and multi-task Structured Stock Prediction Model (MSSPM). They also involved the trading events representations consisting of the event roles embeddings, which are the subject and object extracted via $Standford\ CoreNLP^2$ [169]. The MSSPM is innovatively jointly trained on the tasks of stock prediction and event extraction, since the authors believed these two tasks had intrinsic relations, and a precise extraction result benefits the stock prediction result. To make time-aware predictions, Sawhney et al. [179] proposed a hierarchical learning method named FAST for ranking stocks based on expected profit. This model used a time-aware LSTM to model the temporal irregularities in news and tweets. The FAST shows the positive effects of factoring the in-text fine-grained temporal irregularities in the simulations on the S&P 500 and China A-shares indexes. Li and Pan [123] believed that the movement of stock prices is influenced by many factors, and they proposed an ensemble deep learning model consisting of LSTM and GRU. For the textual information, they predetermined window size by referring to psychology and economic theories according to the persistence or transience of the news impact.

Gated Recurrent Unit (GRU). GRU is developed from LSTM, and it works well in stock market prediction. This model not only helps alleviate the problem of vanishing gradients but also improves training speed by reducing the number of cells [30]. Inspired by Qin et al. [168], who proposed a dual-stage attention-based RNN for time series prediction, Yang et al. [218] proposed a dual-level attention mechanism, which is based on a gated recurrent units network (GRU) model. The principle of the attention mechanism uses input attention to allocate different weights to different financial news titles according to their contribution to the stock price. They paid attention to explaining the reasons for stock price prediction and avoided errors generated by natural language processing tools that can affect the final result. Similarly, Li, Shen, and Zhu [110] also considered the weights of inputs to avoid the input of useless factors affecting the final result, and they proposed a novel multi-input LSTM (MI-LSTM) model to differentiate between mainstream and auxiliary factors through the attention mechanism and assign different weights to these inputs. To solve the challenge of chaotic news, Hu et al. [82] designed a Hybrid Attention Network (HAN) with a self-paced learning mechanism by considering the credibility and comprehensiveness of financial news. This hybrid attention network consists of two attention layers, news-level attention and temporal attention level bi-directional GRU layer are adopted to encode the temporal sequence of corpus vectors. After back-testing, the annualized rate of return in the simulated transactions has been considerably improved. Some of the work takes the experts' opinions into account. Wang et al. [199] proposed a framework for stock prediction based on high-quality investment opinions from experts. The multi-view fusion network stance detection model called MFN determines texts' upside and downside investment opinions by integrating text features from multiple views and relevant knowledge from the financial domain. A stance aggregation module identifies and aggregates high-quality opinions based on a dynamic expert mining procedure. Finally, a stock prediction module is used to predict the future trend of individual stocks using the input expert opinion indicators from the first two sections. The prediction component first employs a GRU to encode a time-ordered sequence of features. Then a time-aware attention mechanism is used to incorporate hidden states in the stocks dynamically. The real-world dataset validates the model's validity in making investment recommendations and individual stock predictions.

Wang et al. [200] aimed to address the challenges of stock trend forecasting due to non-smooth dynamics and complex market dependencies. They proposed a new Hierarchical Adaptive Temporal-Relational Network (HATR) to describe and predict the evolution of stocks in 2021. The HATR progressively captures short- and long-term transition characteristics from multi-scale local combinations of stock trading series by superimposing expanding causal convolutions and gating paths. This model is similar to their previous work, where they presented a dual attention mechanism with a Hawkes process and target-specific queries, which helps to detect significant time points and scales conditional on individual stock characteristics. In order to uncover latent interdependencies among stocks, they also created a multi-graph interaction module that combines previous domain knowledge with data-driven adaptive learning.

Convolutional Neural Network(CNN) Deep CNNs are broadly examined for effectiveness on both Computer Vision (CV) and NLP tasks [92]. Ding et al. [47] further improved the event embedding method [46] and introduced a CNN for training over the input history event embedding where the pooling layer performs well on extracting the representative event history features. One Universal CNN-based predictor (U-CNN pred) [78] is trained through a layer-wise approach where the subCNNs layers are pre-trained sequentially until the proposed model structure is completed. This model obtained reasonable results and was proved effective due to the shallow model structure having less weight to be learned, thus less likely to reach overfitting.

For exploiting the information underlying industrial relations, some models integrate knowledge graphs with CNN to improve model performance. Deng et al. [39] proposed a Knowledge-Driven Temporal Convolutional Network (KDTCN) for making interpretable stock predictions where the Open IE [58] is used to extract events linked to the knowledge graph. Classic one-dimension convolutional layer has the problem of data leakage where the data on time t may be convoluted from time $t-1$ and $t+1$. TCN overcomes this problem by using causal convolution where the data on time t is convoluted with elements on time t and earlier from the previous layer. The KDTCN model performs well in explaining the abrupt price changes due to the convolutional layer extracting significant features in the price time series.

The combination of CNN and LSTM could further improve time series prediction. Lu et al. [140] proposed a CNN-LSTM model to predict daily stock closing prices where the CNN model extracts features from 10-day historical data time series, and the LSTM model makes the price prediction. In the following research, Lu et al. [141] developed and proposed a CNN-BiLSTM-AM model. This model leverages the attention mechanism to capture the historical influential stock fluctuation on price time series and improve the CNN-based model performance. Wang et al. [202] proposed a CNN-BiLSTM model to make the stock closing price prediction where they added a tanh function to the output gate of the Bi-LSTM and improved the model performance. Mehtab and Sen [149] proposed a univariable convolutional LSTM model for predicting the opening price of Indian stocks. The authors developed the model performance by partitioning a 10-day time series into two 5-day data sequences to enable the convolutional layer to extract more historical data features. Besides, GRU is also introduced recently and proved efficient in processing. In the modelling of multiple stock factors tasks, Zhou, Zhou, and Wang [239] proposed an ensembled stock market prediction model consisting of CNN and Bidirectional GRU, which is feature-selection-based. CNN is responsible for feature extraction, while GRU is responsible for processing the time series data. They used the closing price of the stock market as the model output and all other data as inputs and obtained results with less error than other basic models.

Graph Neural Network (GNN). Matsunaga Suzumura and Takahashi [148] used the knowledge graph to integrate the information of companies and GNN models as the individual stock prediction model. Knowledge graphs can well represent the relationship between entities that represent companies in the direction of stock market predictions. One of the paper's novelties is the backtesting approach, which is achieved by using rolling window analysis. Similarly, Ding et al. [49] developed their previous event-driven works [46, 47] and proposed a Knowledge Graph Neural Tensor Network (NTN) model. This model encodes the entity vectors representing the relationship between entities of extracted events and feeds into the event-embedding learning process. This method alleviates the problem of event embedding suffering the limitation of revealing the syntactic or semantic relationships between two events.

Xu et al. [209] focused on the problem of stock price limit, and they proposed the Hierarchical Graph Neural Network (HGNN) considering the different properties of the market state. They constructed a stock market relationship graph and extracted stock information from multiple market state views such as node view, relation view, and graph view in a hierarchical manner. They achieved high performance in classifying the type of price-limit-hitting stock. Finally, the investment return ratio has been improved by using the proposed model. Some works consider multiple information sources to predict the stock market. Li et al. [122] proposed a GNN-based model for the fusion of multi-source heterogeneous subgraphs for predicting the stock market. The datasets include three kinds of subgraphs representing the relations of the stock market index, the stock market news, and the graphical indicators. The fusion of three subgraphs is finally converted into a fully connected classification layer to make predictions. Ang and Lim [6] used a graph encoding module to propagate multimodal information across companies' relationships. Moreover, an attention module is also proposed for global and local information capturing among inter-company relations and different modalities. The model performs robustly on three forecasting tasks and two applications on real-world datasets.

Graph Convolutional Network (GCN). GCNs are the generalization of CNN to the graph data. GCNs often combine with other deep learning models. Chen and Wei [26] proposed a pipeline prediction model to integrate the relationship among corporations. Furthermore, they proposed a joint prediction model based on the GCN model to form the whole network by integrating more unconnected companies. In the graph, each corporation is abstracted as a node, and each edge connecting two nodes represents the relationship between corporations. The weights of edges stand for the shareholding ratio between two corporations. With the LSTM-based encoder layer, which can encode the historical features of corporations, GCN produces the most impressive performance. Similarly, Li et al. [119] proposed an LSTM Relational Graph Convolutional Network (LSTM-RGCN) model to handle the positive and negative correlation among stocks. The correlation matrix among companies is calculated based on historical market data and shows the connection between companies. The LSTM mechanism added to RGCN layers relieves the over-smoothing problem in predicting overnight stock price movement. A novel Gated temporal convolution is introduced to learn the temporal evolution of stock features.

Most existing models focus on designing sequence models to capture the time dependence between stock prices and information such as news. However, they do not make full use of information from other highly correlated stocks. Yin et al. [222] introduced a graph convolutional network model that combines GCN and GRU to fill this gap. GCN extracts features from each stock price with a high degree of similarity. These extracted feature sequences were fed into the GRU model to capture the time dependence. To consider the relationship between stocks, Feng et al. [61] proposed the Relational Stock Ranking (RSR) framework, which helps predict stock movement in a stock-ranking way. The framework consists of three layers: a sequential embedding layer realized by using LSTM, a relational embedding layer, and a prediction layer. Furthermore, they proposed a temporal graph convolution model to solve the ranking problem. Sawhney et al. [177] proposed the Spatio-Temporal Hypergraph Convolution Network (STHGCN), which is the first hypergraph learning approach. The hypergraph structure simulates the relationship among stocks, and spatial hypergraph convolutions are used.

It is a challenging task to accurately predict the price movements of individual stocks due to the influence of contingent events such as company operating conditions and public opinions. Stock market indices reflect the overall stock price trend of a specific industry or company in the stock market. They are less influenced by variables such as the operating conditions of a single company and have better predictability. Wang et al. [197] used the GCN to fuse the correlation of indicators in stock trend prediction. They proposed the MG-Conv model based on a multi-Graph Convolutional neural network-based that constructs static graphs between indices based on constituent stock data. Moreover, they designed dynamic graphs based on trend correlations between indices with different portfolio strategies and defined multi-graph convolution operations based on both graphs.

Graph Attention Networks (GAT). GAT combines the graph neural network and the attention layers. Complex background noise can adversely affect GNN performance in a large-scale graph. With attention, the GNN model focuses on the graph's most critical nodes to improve the signal-to-noise ratio. Furthermore, the Attention mechanism exploits the interconnectedness of graph nodes, distinguishes the hierarchical connections, and enhances the practical information needed for the task. Kim et al. [95] proposed a hierarchical attention network (HATs) using relational data to predict individual stock prices and market index movements. LSTM and GRU are used as the feature extraction modules for the two tasks, respectively. By aggregating different relation types of data and adding the information to each representation, HATS achieves better results than other existing methods. Sawhney et al. [176] proposed a multipronged attention network for stock forecasting (MAN-SF) by fusing the information of financial data, social media and inter-stock relationships. It captures multimodal signals through hierarchical attention to train a Graph Attention Network (GAT).

A common strategy for predicting the firm trend in terms of its relevant firms is to adopt graph convolution networks (GCNs) with predefined firm relations. However, through a range of firm linkages, the bridging importance of which fluctuates over time, momentum spillovers are transmitted. Cheng and Li [28] proposed an attribute-driven graph attention network (AD-GAT) to capture the attribute-mattered momentum spillovers. This network applies the unmasked attention mechanism for inferring the dynamic firm relation underneath the market signal, which is fusing a tensor-based feature extracting module. The proposed model outperforms GCN, eLSTM [116], and TGC [61] in terms of accuracy and AUC in experiments on the three-year data of the S&P 500.

Transformer Based Models. RNN is better suited for tasks dealing with temporal, sequential data such as financial news, tweets, and stock price time series. However, RNN struggles to process long sequences since the model tends to forget the contents of the distant location or mix the contents of nearby positions. The Transformer avoids recursion by processing sentences using a self-attention mechanism and positional embedding. Hence, the Transformer model has achieved promising results in many stock market prediction tasks presented in the following content. Based on Transformer's excellent capacity to capture long-term dependencies, its based models will be utilized to tackle the problems of temporal dependence. Li et al. [126] suggested a novel Transformer encoder attention (TEA) framework based completely on attention mechanisms to handle time dependency difficulties in financial data and expose hidden information in stock prices related to social media texts. The TEA model employs a feature extractor and a cascade processor architecture. A Transformer encoder, attention mechanism, and normalization technique comprise the feature extractor. To learn the crucial information, the feature extractor effectively gathers aspects from past text and stock prices for five calendar days. Similarly, Zhang et al. [231] introduced the Transformer-based attention network (TEANet) architecture to handle the time-dependent problem utilizing five calendar-day data. The TEANet framework consists of a deep textual feature extractor that uses the Transformer and a concatenation processor to properly incorporate and balance the influence of numerous elements, such as tweets and market prices. Yoo et al. [223] enhanced predicting accuracy by leveraging the connection between numerous equities. For this purpose, they introduced the Data-axis Transformer with Multi-Level Contexts (DTML). It builds asymmetric and dynamic correlations in an end-to-end approach for learning the correlations between stocks and providing the final prediction for all individual stocks.

Many Transformer-based model papers use textual information as the input to capture the sentiment of people in stock-related news media. Financial news sentiment analysis aims to predict market reaction towards the hidden information in texts [221]. Li et al. [115] believed that social sentiment played a leading role in reflecting the public's views on stock trends. A collection of social sentiments and professional opinions was collected from social platforms and financial news articles. The obtained data would then be sent into a tensor Transformer, which would be used for model training to eliminate noise and capture a more intrinsic relationship. Furthermore, the trained model is utilized to investigate the effect and function of social emotions using data from diverse sources. According to Liu et al. [132], existing social media-based stock prediction algorithms covered only individual stock semantics and correlations, but vast social media presented contradicting information. They proposed a Capsule network based on Transformer Encoder (CapTE) to solve this problem which contains a Transformer Encoder to capture deep semantic features and structured relationships among tweets. Yang et al. [219] proposed a Hierarchical, Transformer-based, multi-task (HTML) model for predicting short-term and long-term asset volatility. They also used audio data to make predictions in addition to the common news and reports about finance.

Chen et al. [21] developed the Gated Three-Tower Transformer (GT3) for multivariate stock time series extracting and integration. In order to tackle the problem of limited receptive fields, they created a Shifted Window Tower Encoder (CWTE) for capturing channel-wise features from data embedding. To extract and aggregate multi-scale temporal information, a Shifted Window Tower Encoder (SWTE) combined with multi-temporal aggregation was developed. To acquire sophisticated text features, a vanilla Transformer encoder was used as a Text Tower Encoder (TTE). Meanwhile, the Cross-Tower Attention method was created to assist the model in learning the stock market tendencies and related meanings conveyed by the social media content. The features from CWTE, SWTE, and TTE are finally fused through an adaptive gate layer efficiently and accurately.

Pre-trained Language Model. BERT is a Transformer-based language model predominantly used in pre-trained language models [41]. BERT was pre-trained in two innovative training methods, masked-language modelling (MLM) and next sentence prediction (NSP) [41]. MLM enables BERT to understand relationships between words, while NSP enables BERT to understand long-term dependencies between sentences. The pre-trained model can be fine-tuned to make it more suitable for specific tasks.

Financial news is considered one of the primary sources of stock market information and impacts stock returns [121]. Dong et al. [51] proposed a BERT-LSTM model, where the BERT extracted the stock price direction based on social media news, and the autoregressive LSTM integrated information features as covariates. It can also utilize trends of historical prices to predict the future direction of stock prices. Sonkiya et al. [184] employed the BERT model to perform sentiment analysis on news and headlines about Apple Incorporation. The sentiment scores obtained after sentiment analysis were used as input vectors. The GRU and CNN were used as the generator and discriminator in the GAN to generate data continuously. They can discriminate between true and generated samples on the stock price to achieve the final desired prediction effect. The model's early convergence is optimized by using sentiment scores as input. Colasanto et al. [33] improved stock forecasting by utilizing AlBERTo [167], a Transformer-based model, for Italian social media sentiment analysis. This model calculates the sentiment values of various event news in the market that would affect stocks.

Instead of directly using the sentiment in texts for stock market prediction, some researchers agreed that the news comments would influence the investors' sentiment, thus affecting their estimation of market trends and investment willingness. Li et al. [111] evaluated and classified investor comments on news websites using the BERT pre-training model. To validate the links between investor emotions and stock returns, a cross-sectional regression analysis approach is applied. They employed a two-step cross-sectional regression validation [59] to eliminate the heteroskedasticity problem in the samples and the consistency issue in data. Zhao et al. [234] argued that stock commentary by experts is an essential reference for accurate stock prediction. Therefore, BERT was chosen to translate the comments coming from field experts more comprehensively and accurately, resulting in more reliable stock movement predictions. They also argued that the input to BERT is fixed-length text, which leads to poor performance in long text information exploration. For this reason, they used the sliding window to segment the original text, expand the sample size, reduce over-fitting, and comprehensively capture all the information of the lengthy text. Furthermore, the output features of each layer of the BERT model are extracted, and the ablation strategy will be invoked to extract useful information from these features.

The application of Bert in the stock market is not limited to price prediction or movement prediction. Zhou et al. [243] presented a bi-level BERT-based model for detecting predefined trading events, which is further adapted using a wide variety of financial texts. The low-level model is a multi-label token classifier that identifies the events for each token in each phrase. The high-level model combines the low-level model output with the entire article to determine the likelihOOD of each event occurring. The ultimate trading strategy is based on the recognized time and ticker, utilizing string matching the detected events. Hsu et al. [80] adopted a selective perturbed masking (SPM) approach for aspect-based sentiment analysis. SPM analyzes the value of each word in a sentence and replaces the insignificant word using two replacement strategies without compromising aspect-level polarity to tackle readability and semantic consistency issues. The authors experimented with SPM for stock price and risk change prediction as a real-world scenario for sentiment analysis to further evaluate it in sub-tasks such as aspect term sentiment classification (ATSC) and aspect term extraction (ATE).

2.2.2 Event Extraction

Sentence-level Event Extraction. Traditional sentence-level event extraction methods are challenging to acquire thorough knowledge of features, making it difficult to advance in the task of event extraction that requires an understanding of complex semantic relations. Many recent event extraction efforts have utilized deep learning architectures such as CNNs [24, 232], RNNs[158, 181], GNNs [36, 67, 137], Transformers[133, 220, 236], and other networks[86, 229].

Nguyen et al. [160] proposed a joint framework to capture the inter-dependencies between arguments roles and triggers by using bidirectional recurrent neural networks. Some previous works [157] and [24] both adopted convolutional neural network to reserve more crucial information to help improve performance. Previous works [135, 137] proposed to exploit argument information explicitly by attention mechanisms. Yang and Mitchell [213] decomposed the learning problem into three subtasks: within-event structures, event-event relations, and entity extraction, and then they integrate these learned models into a new model that performs joint inference for triggers, semantic roles, and entities.

Chen et al. [25] proposed to automatically labeling training data via distant supervision for detecting key arguments and trigger words. Liu et al. [133] presents a new way of approaching event extraction by treating it as a machine reading comprehension task. The approach involves generating questions about an event based on its schema and using a BERT-based model to answer those questions in order to extract information about the event. This approach allows for the incorporation of advanced models from the field of machine reading comprehension and helps to address the issue of limited data availability in EE by making use of large datasets from machine reading comprehension task.

Du and Cardie [53] presented a new approach to event extraction that involves treating it as a question-answering task. The goal is to extract the arguments of an event in a single, end-to-end process. Lu et al. [142] introduced TEXT2EVENT, a method for extracting events from text in a single, end-to-end process. It involves using a sequence-to-structure network, a constrained decoding algorithm to incorporate event knowledge during the extraction process, and a curriculum learning algorithm to improve the efficiency of the model. The goal is to extract events in a unified manner.

Yang et al. [220] proposed a method for event extraction that addresses the issue of overlap between roles by separating argument prediction based on roles. Additionally, to address the lack of sufficient training data, they also proposed a technique for creating labelled data through editing prototypes and selecting the highest quality generated samples. Different to Yang et al. [220], Ma et al. [143] does not need the design of heuristics, the proposed model is able to generate representations of event arguments that are aware of the trigger, incorporate syntax through the use of dependency parses, and handle the issue of overlapping roles with a role-specific argument decoder.

Document-level Event Extraction. Document-level Event Extraction (DEE) is the process of identifying and extracting events that occur across an entire document, rather than just within a single sentence. DEE poses two challenges that distinguish it from sentence-level Event Extraction (SEE): (i) Arguments-scattering, where the arguments for a single event may be spread across multiple sentences in the document, making it difficult to extract a complete event record from a single sentence; (ii) Multi-events, where a single document may contain multiple events occurring at the same time, requiring comprehensive modelling of the interdependent relationships between these events.

To date, most works on DEE are focused on deep learning techniques, and research on DEE can be broadly classified into two categories. The first category mainly focuses on extracting the scattering event arguments from a document. Early approaches to document-level argument extraction [52, 54] treated it as a slot-filling problem following the task setting of MUC-4 [1]. Du and Cardie [52] believed that extracting events from a document can be challenging because it requires considering the context of the entire document in order to determine which pieces of text correspond to specific roles in an event. They developed a new type of reader called a multi-granularity reader in order to combine the information learned at different levels of detail including sentence and paragraph levels. Du et al. [54] have taken another look at the difficult problem of extracting role-filling entities from a document and found that there is still room for improvement. They have introduced a new type of model based on transformers that are able to learn a representation of the entire document and understand the relationships between role-filling entities and event roles. This model performs better than other methods on the task and is better able to capture the way that language is used to refer to entities within the document.

In addition, researchers have studied how to identify event arguments in a document, given the event triggers, as part of the Argument-linking task in the RAMS dataset[55, 233]. However, this task assumes that the event triggers are already known, which may not be the case in real-world scenarios. Instead of attempting to directly identify event arguments, some approaches[84, 118, 214] follow a ”detect-then-extract” paradigm. This approach involves first detecting events in a document and then extracting the arguments associated with those events.

Recently, researchers [90, 215, 236] also attempt to conduct DEE task in a trigger-free manner in ChFinAnn [236] which is a large-scale DEE dataset constructed from stock financial documents. Zheng et al. [236] developed a model called Doc2EDAG that can generate a type of graph called an entity-based directed acyclic graph to effectively extract events from a document. In addition, they redesigned the process of labelling events in a document by removing the need for certain trigger words, making it easier to identify events at the document level. Yang et al. [215] proposed an end-to-end model to extract structured events from a document in a parallel manner. To do this, they applied a document-level encoder to understand the context of the document and a multi-level decoder to generate events for the document simultaneously. The model was trained using a matching loss function that helps to optimize the model's overall extraction performance. Huang and Jia [90] leveraged the relationships between entities and sentences within long documents to create an unweighted, undirected graph representation of each document and introduced "Sentence Communities" to represent individual events as subgraphs. The proposed SCDEE method can identify multiple events within a document by using graph attention networks to detect sentence communities and address overlapping roles by predicting the roles of arguments.

2.2.3 Out-of-distribution Detection

In recent years, out-of-distribution detection methods based on deep learning models have been proposed. The literature related to OOD on deep learning methods can be categorised into the following themes: post-hoc detection methods [70, 89, 106, 128, 136, 185, 201], confidence enhancement methods [13, 20, 68, 72, 164, 187, 224], and density-based methods [100, 102, 180, 207, 244].

In 2017, Hendrycks and Gimpel [70] proposed the first baseline system for OOD detection task; they noticed that the maximum probability predicted by the softmax function for in-distribution (ID) examples could be greater than the maximum probability predicted for out-of-distribution (OOD) examples. This observation can be used as a simple baseline for detecting OOD data. Liang et al. [127] proposed ODIN, a technique for out-of-distribution detection that does not require modifying a pre-trained neural network. The method is based on the idea that applying temperature scaling and adding small modifications to the input can differentiate the softmax scores of in-distribution and out-of-distribution images, enabling more accurate detection. Liu et al. [136] proposed a framework for out-of-distribution detection that uses energy scores. The approach demonstrates that energy scores are more effective at distinguishing in-distribution and out-of-distribution samples compared to the traditional method of using softmax scores. Energy scores are aligned with the probability density of the input data and are less prone to overconfidence compared to softmax confidence scores.

Within this framework, energy can be used as a scoring function for any pre-trained neural classifier, or as a trainable cost function to specifically shape the energy surface for OOD detection. Sun et al. [185] proposed ReAct, a method for reducing overconfidence in machine learning models when they are applied to out-of-distribution data. The approach is based on a novel analysis of the internal activations of neural networks, which showed distinctive patterns for out-of-distribution data. ReAct can be effectively applied to various network architectures and used with different OOD detection scores. Huang et al. [89] introduced GradNorm, a method for detecting out-of-distribution inputs by using information from the gradient space. GradNorm uses the vector norm of gradients calculated from the KL divergence between the softmax output and a uniform probability distribution. The approach is based on the observation that the magnitude of gradients is typically larger for in-distribution data compared to out-of-distribution data, which makes them useful for OOD detection.

Most of the work is based on computer vision tasks. Despite the importance, few attempts have been made the problem of detecting OOD data in NLP tasks. This also inspires me to explore more on this important task. Tan et al. [186] proposed a Prototypical Network that is resistant to out-of-distribution data for the task of zero-shot out-of-distribution detection and few-shot in-distribution classification. The approach outperforms state-of-the-art methods on the zero-shot out-of-distribution detection task, while also achieving competitive results on the in-distribution classification task, as demonstrated by evaluations of real-world datasets. Zhou et al. [240] proposed an unsupervised method for out-of-distribution detection on the NLP tasks. The approach involves fine-tuning Transformers using a contrastive loss, which improves the compactness of the learned representations and makes it easier to differentiate out-of-distribution instances from in-distribution ones. The Mahalanobis distance is then used to accurately identify out-of-distribution instances in the model's penultimate layer. Zeng et al. [227] proposed a supervised learning approach that used a contrastive objective to minimize the variance within classes by bringing together in-domain samples of the same class and maximize the variance between classes by separating samples from different classes. Additionally, they used an adversarial augmentation technique to generate diverse views of a sample in the latent space. Zhan et al. [228] proposed a method for training a classifier to identify out-of-scope intent in an end-to-end manner by simulating the test scenario during training. This approach does not make any assumptions about the data distribution and does not require any additional post-processing or threshold setting. To train the classifier, they created a set of pseudo outliers by generating synthetic outliers using self-supervision on in-distribution data and sampling out-of-scope sentences from publicly available open-domain datasets. The pseudo outliers are used to train a discriminative classifier that can be directly applied to and perform well on the test task.

2.3 Basics of Machine Learning Methods

2.3.1 Conventional Methods

Genetic Algorithms Genetic Algorithms (GA) are a way of solving problems that involve finding the best solution by simulating natural evolution [77]. They work by generating a group of initial solutions (called a population), and then combining, changing, and altering these solutions in each iteration. The solutions that perform the best, as measured by a fitness function or objective function, are selected to be used in the next iteration. This process is repeated until an optimal solution is found.

Support Vector Machines Support Vector Machines (SVMs) are a type of algorithm that can be used for classification and prediction tasks. They work by finding a boundary or hyperplane that maximizes the distance between two sets of data. The SVM algorithm can also use a technique called the kernel trick to find these hyperplanes in higher-dimensional spaces. SVMs are a popular choice for linear separation tasks because they often have good performance and require few parameters to set [35].

Ensemble Models Ensemble techniques include Boosting, Bagging, and Stacking. Boosting techniques are meta-ensemble algorithms that alter the distribution of training data to improve the performance of their models. They give more weight to training examples that were not classified well by previous models. Bagging methods, on the other hand, do not change the distribution of data and the most well-known Bagging algorithm is the Random Forest, which trains a number of decision trees using a subset of randomly selected input variables. Stacking techniques allow for the combination of multiple already trained models, regardless of their type, whereas Boosting and Bagging mainly use decision trees [15].

Decision Tree A decision tree is a machine learning model that is used for classification and regression [154]. It works by making decisions based on the value of an input feature and proceeds down the tree until a prediction or classification is reached. Decision trees are easy to understand and can handle both numerical and categorical data. They are efficient at handling large amounts of data and can learn from data with missing or incomplete values. They are also resistant to noise and outliers in the data, making them robust to changes in the data distribution.

2.3.2 Transformer

The transformer model is a neural network architecture that has been successful in natural language processing tasks. It was first presented in a 2017 paper [192] and has since been widely used. One advantage of the transformer model is that it can process input sequences faster than other types of neural networks because it can do so in parallel. It also uses self-attention mechanisms to determine the importance of different input elements when making predictions. The transformer model has performed exceptionally well on many natural language processing benchmarks and is frequently used in industry. It is an encoder-decoder model that is composed of several components, including:

- **Encoder and Decoder:** The encoder takes in an input sequence and produces a context representation of it. This representation is then passed to the decoder, which uses it to generate the output sequence.

- **Multi-head attention:** The Transformer employs self-attention mechanisms, which allow it to selectively focus on specific parts of the input sequence at different times rather than processing the entire sequence in a predetermined order. These mechanisms are implemented using multi-headed attention, which enables the model to simultaneously attend to multiple parts of the input.

- **Position-wise feed-forward network:** The Transformer includes a position-wise feed-forward network that is applied to each position of the input sequence independently. This network is made up of two linear transformations with a ReLU activation function in between them.

- **Residual connections and layer normalization:** The Transformer utilizes residual connections and layer normalization to improve the stability of the training process and the model's generalization capabilitie

2.3.3 Other Deep Learning Methods

RNN Recurrent neural networks (RNNs) [174] are a type of neural network that are effective at analyzing sequential data, such as text, speech, and time series. They have a ”memory” that allows them to consider past information when making predictions or classifications. This is implemented through hidden states that are passed through the network and updated at each step. RNNs can be trained using various algorithms, including backpropagation through time and truncated backpropagation through time. They have been applied to a range of tasks, including language translation, language modeling, and speech recognition.

LSTM Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is able to capture long-term dependencies in sequential data. LSTMs were introduced by Hochreiter and Schmidhuber [76] in 1997 in their paper ”Long Short-Term Memory”. LSTMs are able to remember information for longer periods of time by using gates to control the flow of information into and out of the hidden states of the network. This allows LSTMs to effectively learn from long sequences of data and make predictions based on patterns that may span multiple time steps. LSTMs have been applied to a wide range of tasks, including language modeling, machine translation, and time series prediction.

GRU GRUs (Gated Recurrent Units) [31] are a type of RNN (recurrent neural network) that was first introduced in 2014. They are similar to LSTMs (long short-term memory networks) in their ability to capture long-term dependencies in sequential data, but they have a simpler structure and fewer parameters, which makes them faster and easier to train. GRUs are able to capture long-term dependencies in sequential data by using gates to control the flow of information into and out of the hidden states of the network. These gates allow the GRU to selectively decide which information to retain and which to discard, making it possible for the network to learn from long sequences of data and make predictions based on patterns that may span multiple time steps.

CNN Convolutional neural networks (CNNs) [104] are a type of neural network architecture that is commonly used for image and video classification tasks. They are able to learn hierarchical representations of the input data, which is useful for capturing the structure and spatial relationships in images. Some common types of layers in a CNN include a convolutional layers, pooling layers, activation layers and fully-connected layers. CNNs operate by applying filters to the input data to extract features such as edges, corners, and textures. These features are then transformed using activation functions, which allows the network to learn more complex patterns. The output of a CNN is a prediction or classification based on the patterns learned from the input data.

GNN Graph Neural Networks (GNNs) is a type of neural network that operates on graph-structured data, which consist of nodes (entities) and edges (relationships) between them. GNNs aim to understand the relationships between the nodes and use this information to make predictions or perform other tasks. They operate by sending messages between nodes and updating the node representations based on this message passing. GNNs have seen a lot of recent attention and have been applied to various fields in natural language processing.

Some notable works on GNNs include the Graph Convolutional Network (GCN) [99] and the Graph Attention Network [193].

2.4 Conclusion

In conclusion, this chapter provides a comprehensive review of related works in the field of stock prediction using machine learning and natural language processing. The chapter includes an overview of conventional and deep learning-based methods for predicting stock movement, detecting out-of-distribution data, and extracting relevant event information. The chapter also introduces the fundamental concepts of the most common machine learning techniques used in this field. Overall, this chapter serves as a valuable resource for anyone interested in the intersection of machine learning, NLP, and stock prediction. It lays the groundwork for further research and development in this exciting and rapidly evolving field.

Statement of Authorship

Title of Paper	Astock: A New Dataset and Automated Stock Trading based on Stock-specific News Analyzing Model
Publication Status	□ Published ☑ Accepted for Publication □ Submitted for Publication □ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Published in the Fourth Workshop on Financial Technology and Natural Language Processing

Principal Author

Name of Principal Author (Candidate)	Jinan Zou
Contribution to the Paper	Proposed ideas, experiments and wrote paper
Overall percentage (%)	70%
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.
Signature	Date 05/01/2023

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Haiyao Cao
Contribution to the Paper	Discussion and revise the paper.
Signature	Date 05/01/2023

Name of Co-Author	Lingqiao Liu
Contribution to the Paper	Discussion and revise the paper.
Signature	Date 23/01/2023

Please cut and paste additional co-author panels here as required.

Name of Co-Author	Yuhao Lin		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	06/01/2023

Name of Co-Author	Ehsan Abbasnejad		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	24/01/2023

Name of Co-Author	Javen Shi		
Contribution to the Paper	Discussion and revise the paper.		
Signature		Date	27/01/2023

Chapter 3

Astock: An Automated Stock Trading Based on Stock Movement Prediction Analyzing Model

Natural Language Processing (NLP) demonstrates a great potential to support financial decision-making by analyzing the text from social media or news outlets. NLP-aided stock auto-trading algorithms are a type of algorithm that uses natural language processing (NLP) techniques to analyze financial data, such as news articles and social media posts, to make predictions about stock movements. These algorithms can be used to automate the trading process by making buy and sell decisions based on the predictions made by the NLP model. NLP techniques such as sentiment analysis, event extraction, and machine learning can be used to process and understand large volumes of financial data, identify patterns and trends, and make predictions about future stock movements.

In this work, we build a platform to study the NLP-aided stock auto-trading algorithms systematically. In contrast to the previous work, our platform is characterized by three features: (1) We provide financial news for each specific stock. (2) We provide various stock factors for each stock. (3) We evaluate performance from more financial-relevant metrics.

Such a design allows us to develop and evaluate NLP-aided stock auto-trading algorithms in a more realistic setting. In addition to designing an evaluation platform and dataset collection, we also made a technical contribution by proposing a system to automatically learn a good feature representation from various input information. The key to our algorithm is the method called semantic role labelling Pooling (SRLP), which leverages Semantic Role Labeling (SRL) to create a compact representation of each news paragraph. Based on SRLP, we further incorporate other stock factors to make the stock movement prediction. In addition, we propose a self-supervised learning strategy based on SRLP to enhance the out-of-distribution generalization performance of our system. Through our experimental study, we show that the proposed method achieves better performance and outperforms all the baselines' annualized rate of return as well as the maximum drawdown of the CSI300 index and XIN9 index on real trading. Our Astock dataset and code are available at https://github.com/JinanZou/Astock.

3.1 Introduction

Stock prediction has been an attractive task for a long time, and it is still challenging since the stochasticity of the market and behaviour patterns of participators are fluctuating and elusive. Stock forecasting based on Natural Language Processing (NLP) techniques is a promising solution since text information, e.g., Twitter, financial news etc., is strongly correlated with stock prices. However, the NLP-based stock forecasting research is still scattered without unified definitions, benchmark datasets, and clear articulations of the tasks, which severely hinders the progress of this field.

Existing approaches are usually based on market sentiment analysis [28, 178, 211]. Nguyen, Shirai and Velci [159] build a model to predict stock price movement using the sentiment from social media to forecast the stock price. In addition, some previous studies[47, 198, 211, 230] utilized the information of the price and text attempted to predict the future

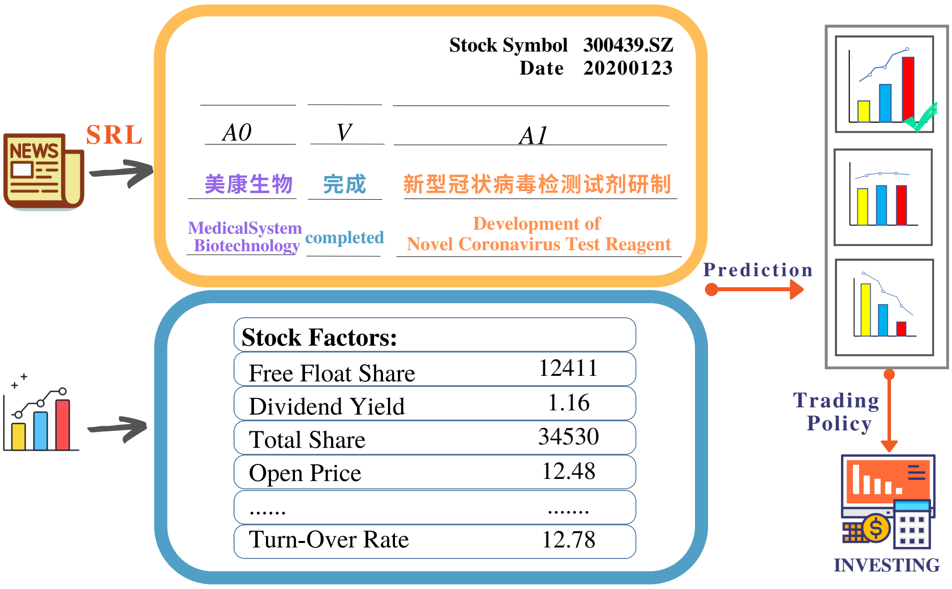


Figure 3.1: Overview of the automated stock trading system.

stock price based on that. Moreover, some news-based approaches [125, 199, 230] utilize news to predict the related securities' price on the following trading day(s). Ding et al. [47] extract events from news articles, calculate event embeddings and use a deep convolutional neural network to predict the direction of stock movements. Despite the limited success in those studies, the existing works are still far from realistic for two reasons: Firstly, previous methods ignore the financial factors, including some fundamentals [1], which plays a key role in practical trading. Secondly, these models are evaluated only on intermediate performance metrics, e.g., stock movement prediction accuracy. It is unclear how well they can support a practical trading system to make sufficient profit.

To address the problems above, we construct a China A-shares market dataset with news and stock factors called Astock. Specifically, we annotate all occurrences of the three trading actions (long, preserve, short) in 40,963 news originating from Tushare [2] with an official

[1] https://www.investopedia.com/articles/fundamental/03/022603.asp

[2] http://tushare.org

license, which describes the major financial events. The dataset also includes various stock factors to build a more realistic system. Based on Astock, we establish a semantic role labelling pooling (SRLP) method to build a compact representation for stock-specific news and predict the stock movement. This work also explores how to leverage the self-supervised method better to upgrade the SRLP method, which achieves outstanding performance for classification and high domain generalization ability.

In experiments, we further propose a realistic trading platform that outperforms all the baseline's average returns and Sharpe Ratios over the CSI300 index and XIN9 index from January 2021 to November 2021. Specifically, unlike other trading systems, we design a dynamic transaction strategy for each trading movement detected by our model as shown in Figure 3.1. Then, we analyze the profitability of the proposed strategy in real trading. The primary contributions of this work can be summarized as follows:

- We construct a brand new Chinese stock prediction task dataset with stock-specific news and stock factors.

- Our proposed SRLP characterizes the key attributes of financial events, which is convenient for incorporating other stock factors and further creating a self-supervised module on top of the SRLP method. Our self-supervised SRLP method obtains competitive stock movement prediction and out-of-distribution (OOD) generalization results.

- We further evaluate algorithm performance on real-world trading from more financial-relevant metrics. By conducting extensive experimental studies, we show that our self-supervised SRLP achieves remarkable performance on these metrics. Furthermore, we observe that the proposed trading strategies work well in practice.

3.2 Related Work

3.2.1 Text-based Stock Prediction

Stock trend prediction has attracted many research efforts, given its critical role in decision-making in stock investment. In general, traditional approaches mainly include technical and fundamental analysis. The technical analysis relies on historical time-series data, such as price and volume. The main goal of technical analysis is to discover trading patterns that can be generalized into future predictions. Previous methods studied with recurrent neural networks (RNN), especially Long Short-Term Memory (LSTM) networks, have been employed for a long time in stock prediction [4, 48, 169]. One limitation of the technical analysis is that it is incapable of predicting the changing markets beyond the price data. On the contrary, fundamental approaches seek information from outside of historical market data, such as the financial environment, business principles, and geopolitics. In recent years, the use of text-based information, especially news and social media, has significantly improved the performance of stock prediction tasks [28, 198, 211]. These methods usually rely on text-based features and sentiment analysis to forecast stock movements [47, 65, 82, 211]. Researchers have utilized a deep neural network to analyze financial news articles to predict stock trends [47, 242]. Xu and Cohen [211] present a novel deep generative model that jointly exploits text and price signals to predict the sixth day's stock movement using the Twitter data and price data of the previous five days. However, the impact of news on stock price is timely, and it lasts for a short period. Therefore, it is unreasonable to predict the trend of the sixth day by the data of the previous five days. To deal with this problem, we collect minute-level data at the time of news release, which can predict the stock movement more accurately. Xu et al. [210] propose a relational event-driven framework (REST) to forecast the stock trend by constructing a stock graph and propagating the effect of event information from related stocks. These approaches assume that the real-trading distribution is the same as the training distribution, which is not realistic as it is difficult to generalize to future trading. By contrast, our self-supervised SRL approach pays closer attention to the quality and comprehensiveness of the news, which not only achieves more competitive stock movement prediction performance compared to previous stock movement baselines and state-of-the-art text classification baselines but also could help with out-of-distribution generalization on the realistic trading platform.

3.2.2 Semantic Role Labeling and Self-Supervised Learning Approach

Semantic Role Labeling (SRL) is defined as the task that automatically answers the question ”Who did What, to Whom, Where, When, and How?” [147]. Semantic role labelling (SRL) aims to disclose the predicate-argument structure of a given sentence, which could provide a clear overlay that uncovers the underlying semantics of text [34]. Such shallow semantic structures have been shown highly useful for a wide range of downstream tasks in natural language processing, such as information extraction [214], machine translation [131] and question answering [146]. However, previous stock movement prediction methods [47, 65, 82, 211] adopted the word or sentence level representation to predict the stock movement. However, due to the lack of abstract information on the news, these approaches can overfit the training data and fail to distinguish the key features of the news. To deal with this problem, we utilized the SRL's characteristics for extracting a clear overlay that uncovers the underlying semantics of news.

Recently, self-supervised learning has become a very popular technique in the training stage of NLP, which generates labels without any human intervention and learns common language representations. Some researchers [91, 212, 237] have proven that self-supervised learning strengthens the generalization ability of models as it improves the performance in many tasks. For instance, masked language model [40] is widely adopted in pre-trained language models, and can be considered as token-level self-supervised learning. We design a self-supervised method based on our SRL approach to predict stock movement. Moreover, the corresponding results will trigger the auto-trading actions on a strategy algorithm, which achieves better out-of-distribution performance and competitive returns compared with the previous approaches [82, 211].

3.3 Data Creation

Table 3.1: The comparison between Astock and other existing widely-used stock movement prediction dataset.

Dataset	Num of Stock	Text Source	Price-level	Stock Factors
DMFT's dataset [230] EMNLP 17'	50	✗	Daily	✗
StockNet's dataset [211] ACL 18'	88	Twitter	✗	✗
Dingxia's dataset [45] EMNLP 14'	500	News	Daily	✗
Trade the event 's dataset [242] ACL 21'	✗	News	✗	✗
Ours	3680	Stock News	Minute-level when news published Daily-level for all the stocks	✓

The stock movement prediction task aims to explore a realistic method to predict stock movement with comprehensive and reasonable information in the China stock market. To this end, it is important to have minute-level price information in the dataset and we are motivated to collect one.

3.3.1 Standard of news and stock factors collection

There are two main components in our dataset: News and stock factors for the China stock market. In terms of news data, there are 40,963 pieces of listed company news, including company announcements and company-related news from July 2018 to November 2021. The news data are split into two parts: the In-distribution split and the out-of-distribution split. The in-distribution split is from July 2018 to December 2020 for training and testing where the training set occupies by 80%, and the validation set and test set occupy 10% respectively. The out-of-distribution split is selected from January 2021 to November 2021, which is used for OOD generalization testing. Every piece of news includes its published time and a corresponding news summary. Factor investing is an investment approach that involves targeting quantifiable firm characteristics or factors that can explain the differences in stock returns. Factor-based strategies may help investors meet particular investment objectives—such as potentially improving returns or reducing risk over the long term. Our Astock dataset covers the 24 stock factors on each stock of the China A-shares including Dividend yield, Total share, Circulated share, Free Float share, Market Capitalization, Price-earning ratio, PE for Trailing Twelve Months, Price/book value ratio, Price-to-sales Ratio, Price to Sales ratio, Circulate Market Capitalization, Open price, High price, Low price, Close price, Previous close price, Price change, Percentage of change, Volume, Amount, Turn over rate, Turn over rate for circulated Market Capitalization, Volume ratio. Furthermore, We compare Astock with several widely used stock movement prediction datasets in Table 3.1. The value is reflected in the following aspects: (1) Astock provides financial news for each specific stock over the entire China A-shares market. (2) Astock provides various stock factors for each stock. (3) Astock provides minute-level historical prices for the news.

3.3.2 Task Formulation

We divide the automated trading system into two tasks: stock movement classification and simulated trading.

Text-based stock movement classification

The goal of the stock movement classification task is to classify the effects of the input information. We measure the impact of each piece of company news by the stock return rate. In this paper, the news is annotated by the stock return rate r , and three cases are considered in our annotation: outperforming, neutral, and underperforming as shown in Equation 3.1. We further model the stock movement by classifying it into three categories. The ground truth for those categories can be derived from r. Specifically, we follow the following rules to categorize the data into three classes after ranking all the news by r, which aims to find the most strong signal of the stock movement and to reduce the disturbance of noises compared to dividing the data evenly. After the domain experts gave us the advice and the experiments with different thresholds were conducted, we set 20% as the threshold where the tunable parameters a, b, c, and d are 20, 40, 60, and 20, respectively.

$$\text{label=}\begin{cases}\text{outperforming} & \text{if } r \text{ ranked top a\%}\\ \text{neutral} & \text{if } r \text{ ranked top b\%-c\%}\\ \text{underperforming} & \text{if } r \text{ ranked bottom d\%}\end{cases} \tag{3.1}$$

where r is the return rate of the news. We randomly select 80% of the in-distribution dataset as the training set, and the other 20% is split evenly into validation and test sets.

Simulated Trading

Stock movement prediction accuracy may not necessarily translate to profitability of an auto-trading system. To further investigate how the stock movement prediction can benefit for the actual trading practice, we employ a practical trading strategy based on the stock movement prediction results and evaluate various metrics for the trading actions. The trading strategy details can be found on our GitHub page.

3.4 Methodology

This section describes the technical contribution of this work: a novel system for stock movement prediction. Our system consists of two major components: semantic role labelling pooling method and self-supervised learning based on SRLP, we will elaborate on those two parts.

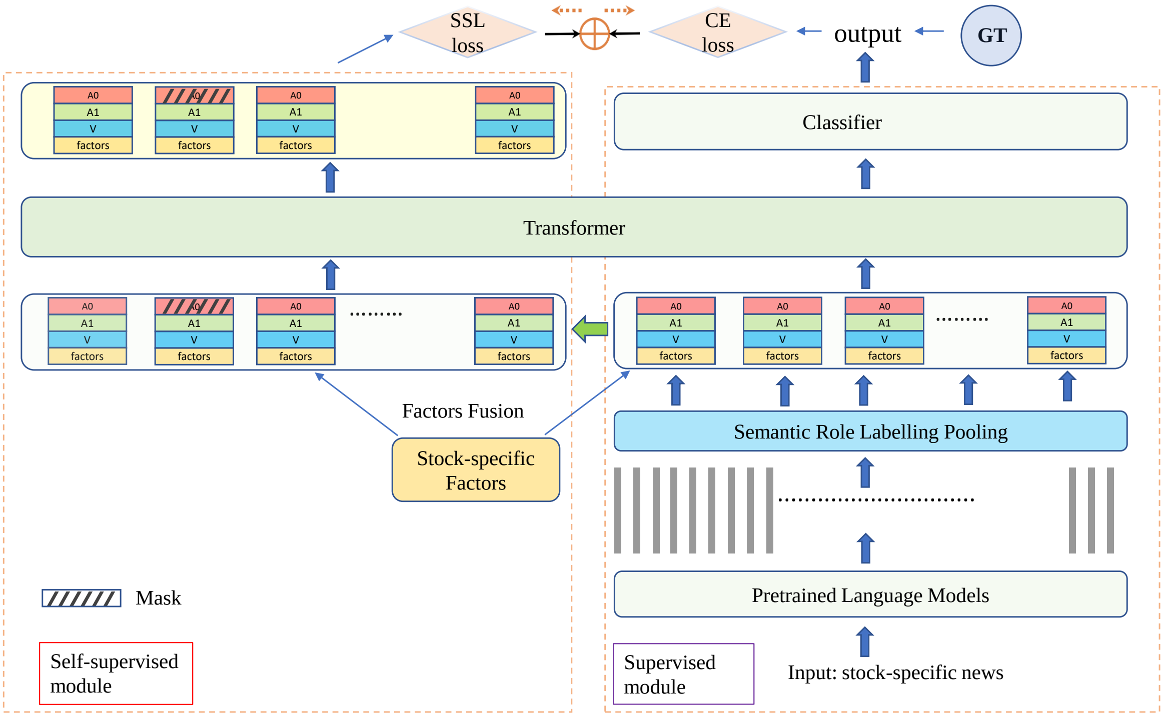


Figure 3.2: Overall framework of our approach, including a domain-adapted pre-trained model (RoBERTa WWM Ext), Semantic Roles Pooling, transformer layer, self-supervised module (left part), and the supervised module (right part). The green arrow represents a duplicate for the SRLP. The final result is generated from the stock movement classifier, and the total loss is obtained from the self-supervised SRLP part and supervised stock movement classification part.

3.4.1 Semantic Role Labelling Pooling

In this work, we propose to leverage the off-the-shelf semantic role labelling, i.e., Propbank [98], to pool the output embeddings of a pre-trained language model to construct an alternative representation. The rationale is that the semantic roles in Propbank, i.e., verb (V), proto-agent (A0), and proto-patient (A1), are general-purposed and are also strongly associated with the event arguments. We show an example of semantic role labelling for financial news in Figure 3.3.

More specifically, we first use the Language Technology Platform (LTP) [18] to auto-

Original text: 科锐国际股东Career HK减持公司股份199万股,占公司总股本的1.103%。

Translation: Career HK, a shareholder of Kerui international, reduced 1.99 million shares of the company, accounting for 1.103% of the total share capital of the company

A0	V	A1
科锐国际股东Career HK	减持	公司股份199万股
Career HK, a shareholder of Kerui international	reduced	1.99 million shares

Figure 3.3: A Semantic role labelling example for a piece of news.

matically mark the semantic roles from the sentences of an entire piece of news and then select V, A0, and A1 to represent the roles for each sentence. Secondly, for each sentence, we process with a pre-trained language model to obtain a sequence of output embeddings $\{\mathrm{s}_1, \mathrm{s}_2, \cdots, \mathrm{s}_n\}$, where s_n is the sentence length. We use $\mathcal{V}$, $\mathcal{A}_0$ and $\mathcal{A}_1$ to denote the indices of tokens corresponding to the V, A0, A1 components. At last, we perform pooling for embeddings with their indices falling into $\mathcal{V}$, $\mathcal{A}_0$ and $\mathcal{A}_1$. We call this scheme Semantic Role Labelling Pooling SRLP in short. Taking A0 as an example, the SRLP feature for A0 is

$$\mathbf{e}_{A0} = \frac{1}{|\mathcal{A}_0|} \sum_{i \in \mathcal{A}_0} \mathrm{s}_i \tag{3.2}$$

For a sentence with N sets of V, A0 and A1, we concatenate $\mathbf{e}_{A0}$,$\mathbf{e}_{A1}$,$\mathbf{e}_{V}$ of each sentence and the financial factor $\mathbf{F}$ of the stock-of-interest into a data matrix:

$$\mathbf{E} = \begin{bmatrix} \mathbf{e}_V^1 & \dots & \mathbf{e}_V^t & \dots & \mathbf{e}_V^N \\ \mathbf{e}_{A0}^1 & \dots & \mathbf{e}_{A0}^t & \dots & \mathbf{e}_{A0}^N \\ \mathbf{e}_{A1}^1 & \dots & \mathbf{e}_{A1}^t & \dots & \mathbf{e}_{A1}^N \\ \mathbf{F} & \dots & \mathbf{F} & \dots & \mathbf{F}, \end{bmatrix} \tag{3.3}$$

where $\mathbf{E}$ is output of the above process. Each column of $\mathbf{E}$, denoted as $\mathbf{e}_j$, is the concatenation of $\mathbf{e}^j_V, \mathbf{e}^j_{A0}, \mathbf{e}^j_{A1}$ and $\mathbf{F}$. Each column in Equation 3.3 corresponds to one triplet. $\mathbf{E}$ is then processed by a Transformer encoder in the same way as the standard text classification to generate the stock movement prediction.

3.4.2 Self-Supervised Learning based on SRLP

Besides standard supervised training loss for stock movement classification, in this work, we further propose to use a self-supervised training task as an auxiliary task to train the network. For stock movement prediction, good generalization is highly desirable since the training data is usually sampled from a period different from the test period.

A significant problem in practice is to ensure that our model generalizes to scenarios different from the training set. We further create a self-supervised learning method on top of the SRLP. Recent studies [73, 151] have shown that incorporating a self-supervised learning task along with the supervised training task could lead to better generalization. As shown in Figure 5.1, the self-supervised task is defined as predicting the position of one randomly masked SRL role from all the roles of SRL in a piece of news. Intuitively, the self-supervised learning task should be designed to encourage the favourable properties of features. In this work, we propose to randomly mask one pooled embedding, i.e., $\mathbf{e}^j_V$, $\mathbf{e}^j_{A0}$ or $\mathbf{e}^j_{A1}$, from a randomly selected sentence, and then ask the network to identify the masked embedding from a pool of candidate embeddings. Such a cloze-style task encourages the network to perform reasoning over other unmasked cues to work out the missing item. We hypothesize that such a reasoning capability is beneficial for understanding the financial news and thus helps stock movement prediction.

Formally, we randomly select a $\mathbf{e}_j$ from $\mathbf{E}$ and then select one element from $\mathbf{e}_j$ = { $\mathbf{e}^j_V$, $\mathbf{e}^j_{A0}$, $\mathbf{e}^j_{A1}$}, after that we replace the selected element with an all-zero vector, indicating a ”mask” operation. Taking masked V at the t-th sentence as an example, we denote the $\mathbf{E}$ after this mask operation as $\mathbf{E}'$.

$$\mathbf{E}' = \begin{bmatrix} \mathbf{e}_V^1 & ... & \mathbf{M} & ... & \mathbf{e}_V^N \\ \mathbf{e}_{A0}^1 & ... & \mathbf{e}_{A0}^t & ... & \mathbf{e}_{A0}^N \\ \mathbf{e}_{A1}^1 & ... & \mathbf{e}_{A1}^t & ... & \mathbf{e}_{A1}^N \\ \mathbf{F} & ... & \mathbf{F} & ... & \mathbf{F} \end{bmatrix} \tag{3.4}$$

Then we feed $\mathbf{E}'$ into the transformer to obtain a query vector sequence $\mathbf{q} \in \mathbb{R}^d$

$$\mathbf{q} = Transformer(\mathbf{E}')[:, t] \tag{3.5}$$

where $[:, t]$ means extracting the t-th column of the vector sequences calculated by the transformer. The unmasked SRLP-V features (or SRLP-A0, SRLP-A1 features, depending on which type of SRLP feature is chosen) is also sent to an encoder to calculate candidate key vectors: Formally, $\mathbf{K} \in \mathbf{R}^{d \times N}$ is defined as:

$$\mathbf{K} = [f_V(\mathbf{e}_V^1), \cdots, f_v(\mathbf{e}_V^t), \cdots, f_V(\mathbf{e}_V^N)] \tag{3.6}$$

where f_V is an encoder specified for encoding V-type SRLP feature. Then the query vector is compared against each column vector in $\mathbf{K}$ and is expected to have the highest matching score at the t-th location. This process could be implemented via matrix multiplication and the softmax operation:

$$P_{SSL} = \text{Softmax}(\mathbf{qK}) \tag{3.7}$$

and we hope the highest probability entry in Eq. 3.7 is at the t-th dimension. This requirement could be enforced via cross-entropy loss. Finally, the training loss for the models is

$$\mathcal{L} = \alpha \mathcal{L}_{CLS} + (1 - \alpha)\mathcal{L}_{SSL} \tag{3.8}$$

where $\mathcal{L}_{CLS}$ is the cross-entropy loss for the text classification and $\mathcal{L}_{SSL}$ is the cross-entropy loss on the self-supervised learning prediction P_{SSL}. α here is a trade-off parameter. We set α as 0.8, and additional information and specific hyperparameters and the impact of this running parameter can be found in Appendix B.3.

3.5 Experiments

In this section, we conduct experiments to evaluate the performance of the proposed model. We first introduce our experimental setting. Then, we present the main results by comparing our self-supervised SRLP against various previous approaches. We conduct experiments on two different splits of our dataset for each model: In-distribution split and out-distribution split. We also feed the prediction result of our method to the proposed trading strategy to analyze the profitability through a back-test on real-world stock data.

3.5.1 Implementation Details

In our proposed model, we set the number of heads and layers of the transformer blocker to 1 and 2, respectively. The weights of L_{SSL} and L_{CE} are set to 0.8 (α) and 0.2, and we also conduct the comparison study by choosing different alpha in Figure B.2. We use AdamW as the optimizer and train the model for 20 epochs with a batch size of 16 and a fixed learning rate of 1e-5. In addition, two epochs of linear warming up are used.

3.5.2 Evaluation metrics

We apply the accuracy, F1 score, recall and precision as the stock movement prediction evaluation metrics. We also design a trading strategy based on news movement detection responses to back-test our model. For more details, please check Appendix B.1. The following are the evaluation metrics for our experiments.

Table 3.2: The performance comparison(%) of in-distribution evaluation on our scheme and others to demonstrate the effectiveness of our self-supervised SRL method. ✔ indicates that the model adopted this Semantic role's pooling information. **-** indicates that the method does not adopt this semantic role's pooling. ✗ indicates that the semantic role's pooling is masked.

Model	Resource	SRLP			Accuracy	F1 Score	Recall	Precision
		A0	V	A1				
StockNet [211]	News	**-**	**-**	**-**	46.72	44.44	46.68	47.65
HAN Stock [82]	News	**-**	**-**	**-**	57.35	56.61	57.20	58.41
Bert Chinese [40]	News	**-**	**-**	**-**	59.11	58.99	59.20	59.07
ERNIE-SKEP [188]	News	**-**	**-**	**-**	60.66	60.66	60.59	61.85
XLNET Chinese [37]	News	**-**	**-**	**-**	61.14	61.19	61.09	61.60
RoBERTa WWM Ext [37]	News	**-**	**-**	**-**	61.34	61.48	61.32	61.97
	News + Factors	**-**	**-**	**-**	62.49	62.54	62.51	62.59
SRLP	News	✔	✔	✔	61.76	61.69	61.62	61.87
	News + Factors	✔	✔	✔	64.79	64.85	64.79	65.26
Self-supervised SRLP	News	✗	✔	✗	61.07	61.11	61.11	61.11
	News	✗	✔	✔	62.36	62.32	62.43	62.64
	News	✔	✔	✗	62.42	62.46	62.44	62.62
	News	✗	✗	✔	62.15	62.15	62.15	62.59
	News	✔	✗	✗	61.34	61.23	61.46	61.30
	News	✔	✗	✔	62.97	63.05	62.93	63.47
Self-supervised SRLP with Factors	News + Factors	✗	✔	✗	64.59	64.62	64.63	64.65
	News + Factors	✗	✔	✔	66.82	66.81	66.90	66.82
	News + Factors	✔	✔	✗	65.54	65.53	65.62	65.50
	News + Factors	✗	✗	✔	65.34	65.21	65.43	65.43
	News + Factors	✔	✗	✗	65.27	65.35	65.24	65.77
	News + Factors	✔	✗	✔	**66.89**	**66.92**	**66.95**	**66.92**

Annualized Rate of Return is the geometric average of annual returns of each year over the investment period. It measures over a period, either longer or shorter than one year, annualized for comparison with a one-year return.

$$Return_{Annual} = (Return_{Total})^{\frac{N_{Annual}}{N_{Total}}} \tag{3.9}$$

Where N_{Annual} is the number of trading days in a year, N_{Total} is the number of the days in the statistical period.

Maximum Drawdown is the maximum observed loss from a peak value to a trough value of a portfolio before a new peak is attained. Maximum Drawdown is an indicator of downside risk over a specified period.

$$Maximum\ Drawdown = \frac{Through\ Value - Peak\ Value}{Peak\ Value} \tag{3.10}$$

Sharpe Ratio is used to help investors understand the return of an investment compared to its risk. It is defined as: $Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$. Where R_p is the return rate. R_f is the risk-free rate. σ_p is the standard deviation of the return rate.

3.5.3 Compared Methods

We re-implement the previous stock movement prediction models as baselines, including StockNet [211], HAN Stock [82] . We also construct baselines by formulating the stock movement prediction problem as text classification and use four strong pre-trained language models as backbones. In addition, we also compare the CSI300 index, XIN9 index [3] against the proposed method when analyzing the profitability of the proposed system. The description of those baselines is elaborated as follows.

StockNet StockNet [211] is a stock temporally-dependent movement prediction model which also uses Twitter data and price information. Since the publication frequency of news is much lower than that of tweets regarding a listed company. We only use historical news data updated on a daily basis, which is different from the StockNet method, which makes each prediction upon the Twitter data collected in five-day periods. Furthermore, the binary stock movement prediction is replaced with the three-classes prediction for our Astock dataset.

HAN Stock Hybrid Attention Networks (HAN) Stock [82] is a stock trend prediction model

[3] Equivalent to the Standard and Poor's 500 (S&P 500) or the Dow Jones Industrial Average (DJIA) in the US stock market

based on a sequence of recent related news. There are three classes for the prediction task, which are the same as ours. Because both news data and Twitter data are of the same text type and are close in terms of text length, they are swappable when being fed to the model. Specifically, since the module of the word vectors is for English text, and we replace the module with the pre-trained word vectors from Chinese financial news [117] to make the model adaptive to Chinese text.

XLNet-base-Chinese XLNet-base-Chinese [37] is a pre-trained language model learned via a novel objective generalized permutation and auto-regressive language method. The model is pre-trained from Chinese Wikipedia extended data, exhibiting excellent performance on various downstream Chinese language tasks.

SKEP SKEP [188] is a Sentiment Knowledge Enhanced pre-trained language model, which is designed for sentiment analysis. SKEP conducts sentiment masking and constructs three sentiment knowledge prediction objectives to embed sentiment information at the word, polarity, and aspect levels into the model. We employ this model since the stock movement prediction is highly related to the sentiment analysis of news.

RoBERTa WWM Ext RoBERTa WWM Ext [37] is a RoBERTa-based[138] pre-trained language model which was pre-trained on Chinese Corpus. It masks whole Chinese words instead of individual Chinese Characters (A Chinese word spans multiple characters without a space in between).

Bert Chinese Bert [40] is a famous NLP pre-trained language model, which stands for the Bidirectional Encoder Representations from Transformers. Bert Chinese is a Chinese version of BERT, which was pretrained on Chinese Wikipedia.

For the above four pre-trained language models, we extract sentence embedding from the

[CLS] token and attach a three-way classifier to predict the stock movements.

CSI300 index & **XIN9 index** We employ the CSI300 index and XIN9 index as the baselines of the market performance. These two indexes represent the capitalization-weighted stock market index, which are designed to replicate the performance of the top 300 stocks or 50 stocks traded on the Shanghai Stock Exchange and the Shenzhen Stock Exchange.

3.5.4 Stock Movement Evaluation

We first compare different methods on the task of stock movement prediction. We conduct experiments on two different splits of our dataset: In-distribution split and out-distribution split. In the in-distribution split, both training and testing data are sampled from the same period while the out-distribution split uses data from different periods to construct the training and testing data. In out-distribution split, we construct a new training/testing split by using the data from July 2018 to December 2020 as training data and the data from January 2021 to November 2021 for the testing data.

In distribution evaluation The results are shown in Table 3.2. From the results, we make the following observations:

- If only text information is used, the proposed SRLP approach achieves state-of-the-art performance. Interestingly, we find that SRLP achieves superior performance when further combining the stock factors. It outperforms RoBERTa WWM (News+Factors) by more than 2%. We postulate that this is because the compact representation in SRLP makes the incorporation of stock factors easier. Note that the proposed way of incorporating stock factors (see Section B.2) does not only introduce extra modalities for the stock movement prediction but also could make the text analysis module adaptive to the stock factors. This could be useful to model the scenario, like the effect of a similar event could result in a different impact on the stock movement for a different type of company.

- The proposed self-supervised SRLP can further boost the performance of SRLP. In the best setting of self-supervised SRLP, i.e., with V being masked, self-supervised SRLP achieves more than 1% improvement over SRLP. The improvement is even larger when the stock factors are provided, showing more than 2% improvement over SRLP (News + Factors), achieving 66.89% prediction accuracy. This validates the effectiveness of the proposed self-supervised learning approach. Interestingly, we observe that masking A0 and A1 usually will not bring improvement. This is in contrast to the case of masking V. Note that V encodes the type of an event, and the argument is encoded by A0 or A1. It seems that predicting the type of events is a more effective self-supervised learning task than working on the argument.

Out-distribution evaluation

In the experiments above, the training data and testing data are sampled from the same period. Thus the distributions of training data and testing data are similar. For real-world applications, the stock movement prediction model is applied to future data unseen at the training time. Hence, it is critical to evaluate the model in such an out-of-distribution setting.

Table 3.3: The comparison (%) of the out-distribution evaluation on stock movement classification with StockNet, RoBERTa-WWM Ext, HAN Stock method and our method from 1/1/2021 to 12/11/2021.

Model	Accuracy	F1 Score	Recall	Precision
StockNet[211]	44.35	42.52	45.42	45.82
HAN Stock[82]	53.41	53.33	53.69	54.53
RoBERTa WWM Ext[37]	60.15	60.08	59.89	60.78
Self-supervised SRLP(V masked)+Factors	**64.09**	**63.95**	**63.90**	**64.43**

We first conduct an evaluation on the stock movement prediction task, and the results are shown in Table 3.3. From the results, we can see that the proposed method is still comparably competitive over other baselines. We further show the performance in each quarter of the test data in Table 5. We can see that the proposed method works well consistently across the entire testing period. Our full model ”(self-supervised SRLP) + Factor” works particularly well in the first and third quarters, leading to over 2% improvement.

Table 3.4: The comparison (%) of the out-distribution evaluation with StockNet, HAN Stock, RoBERTa-WWM Ext and our method in each quarter from 1/1/2021 to 12/11/2021.

Model	Quarter 1th	Quarter 2nd	Quarter 3rd	Quarter 4th
StockNet[211]	42.84	46.98	43.60	45.60
HAN Stock[82]	54.95	58.37	50.89	51.19
RoBERTa WWM Ext[37]	62.62	65.00	56.50	54.87
Self-supervised SRLP(V masked)+Factors	**65.84**	**66.87**	**61.59**	**60.88**

3.5.5 Profitability Test in Real-world

Table 3.5: The comparison of profitability test on Maximum Drawdown(%), Annualized Rate of Return(%), and Sharpe Ratio Rate(%) with strong baselines, XIN9, CSI300 and our proposed method from 1/1/2021 to 12/11/2021.

Model	Maximum Drawdown↓	Annualized Rate of Return↑	Sharpe Ratio↑
XIN9	-15.85	-15.38	-32.01
CSI300	-14.40	-9.34	-32.99
StockNet[211]	-7.40	-22.42	-177.65
HAN Stock[82]	-7.38	-13.50	-55.84
RoBERTa WWM Ext[37]	-3.83	1.35	-16.31
Self-supervised SRLP(V masked) with Factors	**-3.60**	**13.85**	**40.93**

In this section, we discuss the possible profitability of the proposed strategy in real-world trading. We utilize the trading strategy described in Appendix B.1 to conduct trading simulation (backtesting) on stock data from January 2021 to November 2021 and evaluate on various metrics. (1) In Table 3.5, we show that our self-supervised SRLP model achieves a remarkable annualized rate of return of 13.85%, which surpasses the previous baselines and market index XIN9 and CSI300. The resulting baseline HAN Stock [82] and Stock-

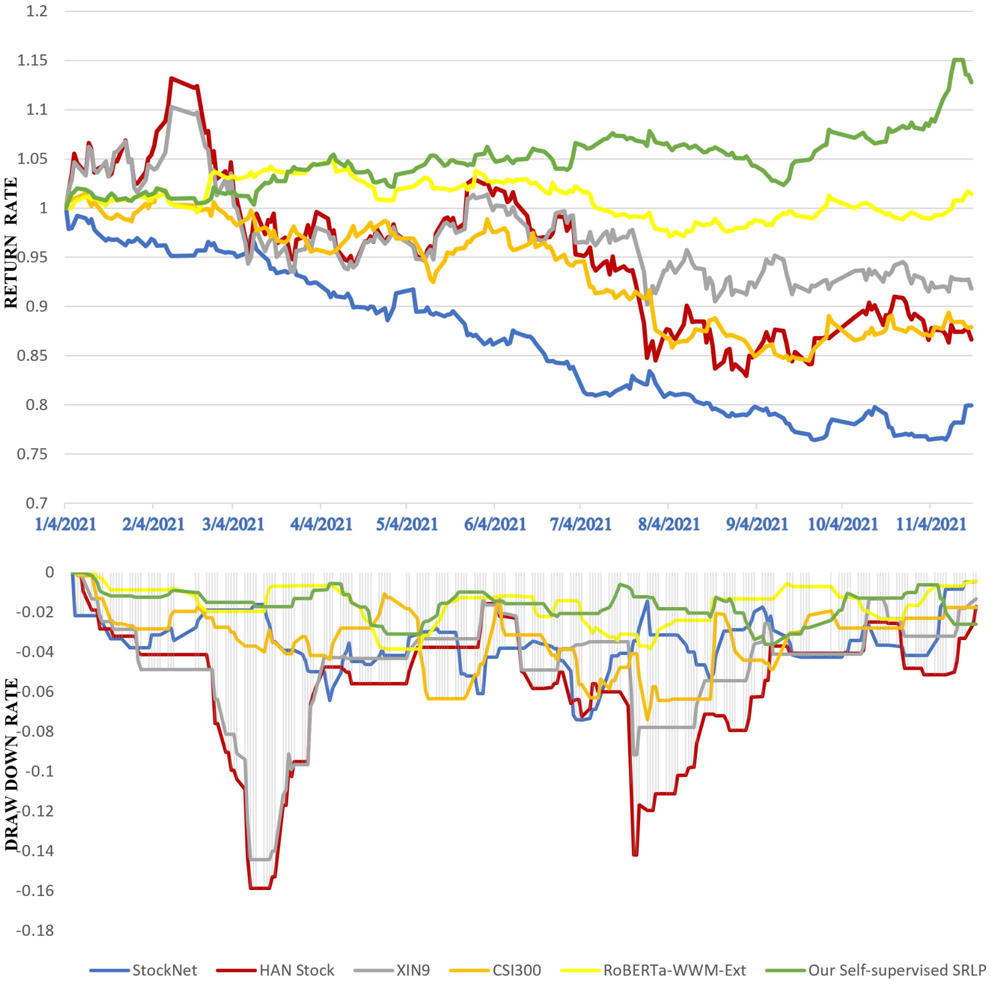


Figure 3.4: The comparison for the real trading performance on Return Rate, Draw Down Rate with CSI300 index, XIN9, Roberta WWM Ext, HAN Stock, StockNet and our proposed method from 1/1/2021 to 12/11/2021

Net [211] achieve an annualized rate of return of -13.5% and -22.42% respectively, and the market XIN9 index and CSI300 were overall declining in 2021, which obtains -15.38 % and -9.34% respectively. In addition, our self-supervised learning method also obtains the lowest Maximum Drawdown of -3.6% and the highest Sharpe Ratio of 40.93%, which significantly outperforms the previous SRLP methods and indicates that our self-supervised could successfully achieve higher expected returns while remaining relatively less risky. (2) Since the investors not only consider the profitability but also consider the risk of the worst-case scenario, we compare the return rates and Draw Down value between the baselines and our proposed model as shown in Figure 3.4. In February 2021, Xin9 increased significantly, achieving a 10% increase. Our model does not show such a high return at the beginning. However, in March 2021, XIN9 began to accelerate its decline, and the Maximum DrawDown value fell by nearly 16%. By contrast, our self-supervised SRLP has generally maintained a profitable trend while remaining low-risk. In addition, our proposed method outperforms the previous study [82] in drawdown value and the return rate. From the above observations, we may conclude that the self-supervised modules of SRLP can lead to better profitability with lower risk on an automated stock trading platform.

3.6 Conclusion

In this paper, we study the problem of NLP-based stock prediction and make two major contributions to this field: (1) We develop a new dataset called AStock, featured by its large number of stocks, stock-relevant news, and availability of various financial factors. (2) We propose a new stock movement prediction system based on two novel techniques. One leverage Propbank-style semantic role labelling results to create compact news representation. Building on top of this representation, another technique is a customized self-supervised learning training strategy for improving generalization performance. We demonstrate that the proposed method achieves superior performance over other baselines through extensive experiments in both in-distribution and out-distribution settings. Also, by feeding our prediction to a practical trading strategy, our method achieves outstanding profitability in backtesting.

Statement of Authorship

Title of Paper	Rethinking Document Event Extraction: A Generative Model is All You Need?
Publication Status	□ Published □ Accepted for Publication ☑ Submitted for Publication □ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Under Review of ACL 2023.

Principal Author

Name of Principal Author (Candidate)	Jinan Zou		
Contribution to the Paper	Proposed ideas, conducted experiments and wrote paper.		
Overall percentage (%)	80%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	23/01/2023

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Yanxi Liu		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	23/01/2023

Name of Co-Author	Yuankai QI		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	23/01/2023

Please cut and paste additional co-author panels here as required.

Name of Co-Author	Lingqiao Liu		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	23/01/2023

Name of Co-Author	Javen Shi		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	27/01/2023

Chapter 4

Financial Document Event Extraction: A Generative Model is All You Need

The finance industry has become increasingly digitized, the volume of digital financial documents, such as financial announcements from publicly traded companies, has grown exponentially, making it difficult for investors to find valuable information. In Chapter 3, we developed a platform to systematically study NLP-aided stock auto-trading algorithms. However, Astock platform lacks the integration of stock company's announcement and is unable to identify specific events that can have a significant impact on stock prices. Document-level event extraction can help extract structured information that can assist investors in identifying emerging risks and capitalizing on profitable opportunities in the stock market. This method can extract specific information from financial documents, such as press releases, earnings reports, and filing statements. The extracted information can include details about a company's financial performance, management changes, and other significant events that could potentially impact the stock price, thus helping investors make more informed decisions and predictions about the future.

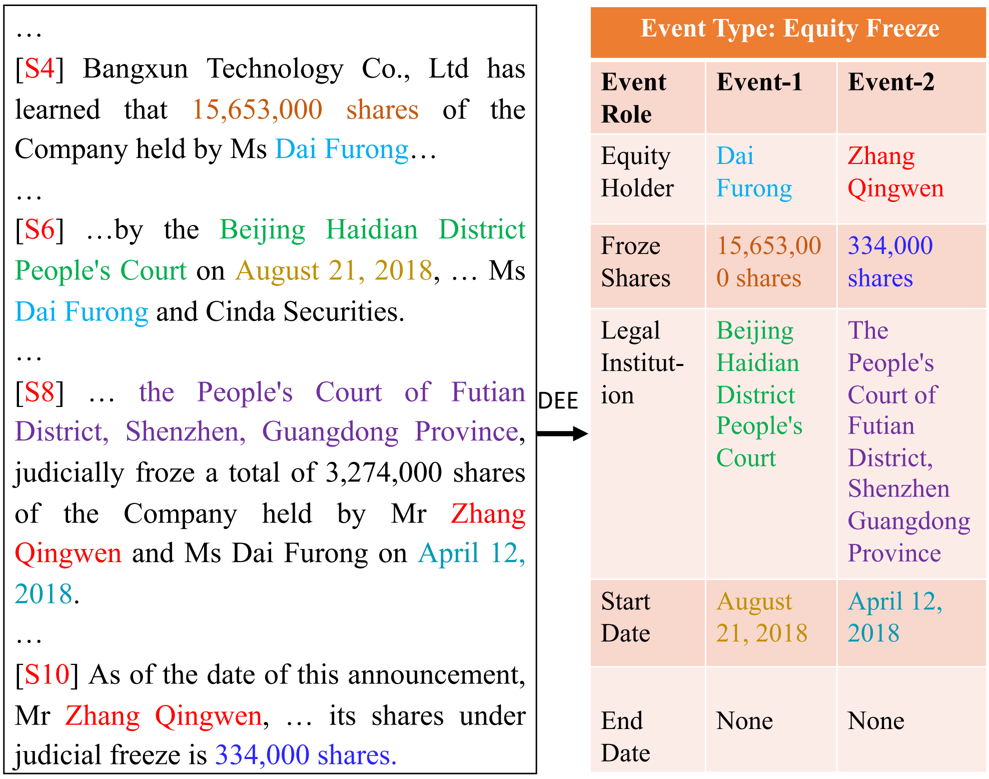
Therefore, based on Chapter 3, we will continue to investigate the NLP techniques to extract information from financial documents for improving the stock market prediction. This

Chapter explores the current methods for extracting events at the document level, which often involve custom-designed networks and processes. We question whether such extensive efforts are truly necessary for this task. Our research is motivated by recent developments in generative event extraction, which have shown success in sentence-level extraction but have yet to be explored for document-level extraction. To fill this gap, we propose a generative solution for financial document-level event extraction, which is more challenging due to the presence of scattered arguments and multiple events. We introduce an encoding scheme to capture entity-to-document level information and a decoding scheme that makes the generative process aware of all relevant contexts. Our results indicate that using our method, a generative-based solution can perform as well as state-of-the-art methods that use a specialized structure for document event extraction, providing an easy-to-use, strong baseline for future research. This document-level event extraction for stock announcements process also can help investors and traders stay informed about the companies they are interested in, and make more informed investment decisions.

4.1 Introduction

Event Extraction (EE) is a crucial task in Information Extraction (IE) that involves identifying the specific type of events and extracting the relevant event arguments. While most existing methods focus on extracting events within a single sentence, real-world scenarios often require extracting events that span multiple sentences and may be described across multiple sentences within a document. This poses a significant challenge to current methods.

Two challenges that are faced when performing document-level event extraction (DEE) as opposed to sentence-level event extraction (SEE) are: (1) Arguments-scattering: In DEE, the arguments of a single event may be spread out over multiple sentences in a document, making it difficult to extract a complete event record from a single sentence. For example,



…

[S4] Bangxun Technology Co., Ltd has learned that 15,653,000 shares of the Company held by Ms Dai Furong…

…

[S6] …by the Beijing Haidian District People's Court on August 21, 2018, … Ms Dai Furong and Cinda Securities.

…

[S8] … the People's Court of Futian District, Shenzhen, Guangdong Province, judicially froze a total of 3,274,000 shares of the Company held by Mr Zhang Qingwen and Ms Dai Furong on April 12, 2018.

…

[S10] As of the date of this announcement, Mr Zhang Qingwen, … its shares under judicial freeze is 334,000 shares.

Event Type: Equity Freeze

Event Role	**Event-1**	**Event-2**
Equity Holder	Dai Furong	Zhang Qingwen
Froze Shares	15,653,000 shares	334,000 shares
Legal Institut-ion	Beijing Haidian District People's Court	The People's Court of Futian District, Shenzhen Guangdong Province
Start Date	August 21, 2018	April 12, 2018
End Date	None	None

Figure 4.1: A simplified DEE example of the event type Equity Freeze with five related argument roles: *Equity Holder, Froze Shares, Legal Institution, Start Date* and *End Date.*

as shown in Figure 4.1, the arguments of Event-1 are distributed in different sentences (S4 and S6). Identifying such events without taking into account the global interactions among sentences and entity mentions can be challenging. (2) Multi-events: A single document may contain multiple event records, and different event records may share the same arguments, making it necessary to have a comprehensive understanding of the interdependence between events. For example, as shown in Figure 4.1, there are two events (Event-1 and Event-2) in a document with the Equity Freeze event type. The multi-event problem requires DEE methods to accurately identify the number of events present in a document and properly assign arguments to each event.

Previous models for document-level event extraction often rely on custom-designed network architecture or decoding processes. For example, the Doc2EDAG model [236] uses a directed acyclic graph to combine arguments, which can be both computationally and memory-consuming. Similarly, the DE-PPN model [215] utilizes a multi-layered decoder to generate possible events simultaneously. Recently, there have been developments in using a text generation process for event extraction, which simplifies the event extraction problem [79, 85, 142, 163]. However, it is yet to be explored if this approach is effective for extracting events at the document level.

This paper aims to fill this gap by investigating the effectiveness of using a generative approach for extracting events at the document level. This is a challenging task due to the issues of scattered arguments and multiple events. To tackle this problem, we propose an encoding scheme to capture information from entities to the whole document and a decoding scheme to ensure that the generative process is aware of relevant contexts. Specifically, our encoding scheme converts sentence embeddings and the embeddings of entities detected in each sentence into a sequence of context-aware embeddings for the decoder. Our decoding scheme predicts the event type, the number of events per event type, and the event arguments for each event record one by one. The previously predicted output is then used as context for future extraction, creating a recurrent decoding process. Our findings suggest that our generative event extraction scheme is as effective as existing approaches in the literature, enabling our method to serve as an easy-to-use strong baseline for future research.

4.2 Preliminaries

Given a document composed of N^s sentences $D = \{S_i\}_{i=1}^{N^s}$ and each sentence S_i containing N_i^w words $S_i = \{w_j\}_{j=1}^{N_i^w}$, the document-level event extraction (DEE) task aims to extract all events presented in the document, where an event belongs to one of several predefined event types.

As shown in Figure 4.1, each type of event has a predefined template, consisting of

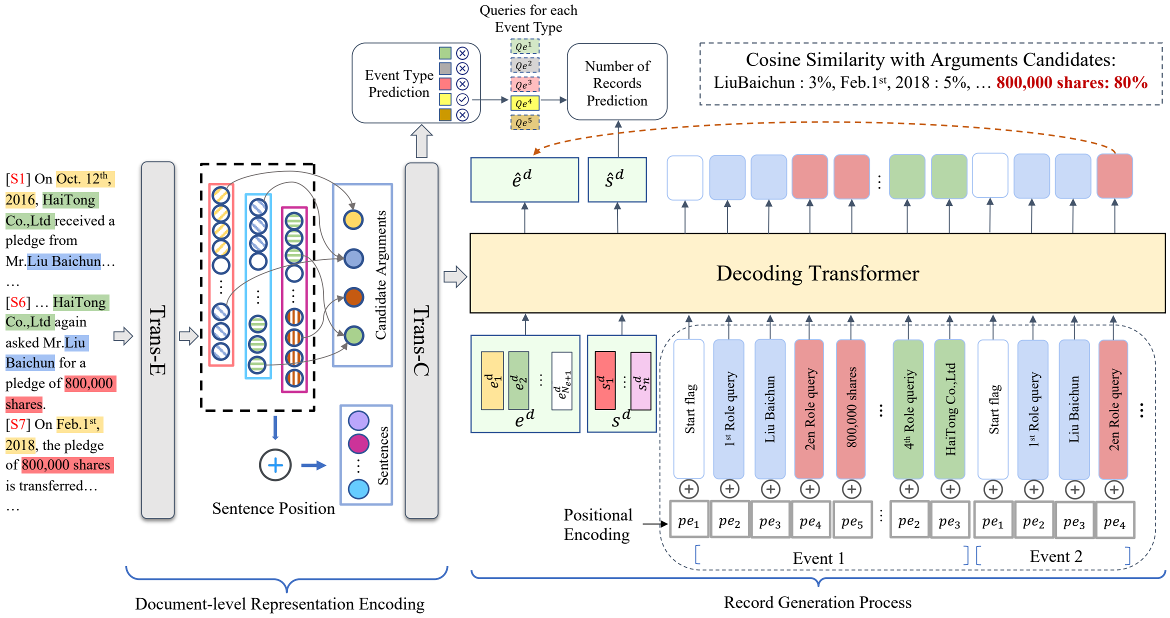


Figure 4.2: The overall architecture of our decoding transformer model. Given a document, we first extract entities (e.g., “HaiTong Co.,Ltd”, “Liu Baichun”, “800,000 shares”), which serves as candidates for event arguments (Section 4.3.1). Then we predict event types that appeared in the document via learning global-awareness representations of entities and sentences (Section 4.3.2 and 4.3.3). Last, we predict the number of events for each event type and leverage a decoding transformer to generate the necessary information for event extraction (Section 4.3.4).

several **event roles**, such as “Equity Holder” and “Legal Institution”. Each event role is instantiated by an entity, and this entity is termed as **event argument**. For example, the entity “Dai Furong” is the event argument for the event role “Equity Holder” in Event-1.

The goal of the DEE task is to detect appeared event types in the input document and assign proper entities to their event roles.

4.3 Methodology

The main pipeline of our proposed model is illustrated in Figure 4.2. We begin by extracting entities, such as “Liu Baichun”, “Oct. 12th”, and “800,000 shares” from the input document to serve as potential event arguments. Next, we predict the event types present in the document by learning global-awareness representations of entities and sentences. Lastly, we predict the number of events for each type and determine the appropriate entities for each event role. Unlike previous methods that require multiple specialized modules to achieve these tasks, we make the first attempt to use the plain transformer with our innovative encoding and decoding scheme to fully utilize its capabilities. We detail each component in the following sections.

4.3.1 Entity Recognition

Given a sentence $S_i = \{w_j\}_{j=1}^{N_i^w} \in D$, we first encode S_i into a sequence of vectors using a Transformer named Trans-E:

$$\left\{h_1, ..., h_{N_i^w}\right\} = \text{Trans-E}(\left\{w_1, ..., w_{N_i^w}\right\}). \tag{4.1}$$

We then perform max-pooling on $\left\{h_1, ..., h_{N_i^w}\right\}$ to obtain sentence embedding h_i^s. Further, we perform entity recognition as a sequence tagging task with BIO (Begin, Inside, Other) schema using $\left\{h_1, ..., h_{N_i^w}\right\}$ as the input like Zheng et al. [236]. Since an entity mention often spans multiple tokens, we use max-pooling on the embeddings of the corresponding tokens.

4.3.2 Context-Aware Encoding of Entities and Sentences

To effectively handle scattered arguments, it is crucial to make the sentence and entity representations be aware of its context within the document. To this end, we propose to format entities and sentences as a sequence and encode them with another transformer named Trans-C. Formally, we have

$$e^d, s^d = \text{Trans-C}([h_1^e, \cdots, h_{N^e}^e, h_1^s, \cdots, h_{N^s}^s]) \tag{4.2}$$

where h_i^e is the representation of i-th entity; h_i^s is the representation of i-th sentence. We add sentence position embeddings to the sentence representations, indicating their location in the document before feeding them into Trans-C. Entities from the same sentence will share the same position embedding as the sentence. After the document-level encoding stage, we obtain entity and sentence embeddings with document-level context awareness, denoted as $e^d = [e_1^d, ..., e_{N_e}^d]$ and $s^d = [s_1^d, ..., s_{N_s}^d]$, respectively.

4.3.3 Event Type Prediction

Before extracting event arguments, we first estimate the event types that appear in the input document. We formulate this task as a multi-label classification problem, i.e., the prediction will be a multi-hot vector with an active dimension indicating the presence of one event type.

To achieve this, we make the prediction from the sentence representation s^d by using a multi-head self-attention layer:

$$S = \mathrm{MultiHead}(Q, s^d, s^d) \in \mathbb{R}^{N^{type} \times d}, \tag{4.3}$$

$$\hat{Y} = \mathrm{sigmoid}(S^\top W_t) \in \mathbb{R}^{N^{type}}, \tag{4.4}$$

where $Q \in \mathbb{R}^{N^{type} \times d}$ is a learnable query token, which works as the [CLS] token in BERT [40]. $W_t \in \mathbb{R}^d$ is the classifier. N^{type} represents the number of event types predefined by the dataset. The MultiHead operation denotes the standard Multi-Head Attention (MHA) mechanism with a query, key, and value in Eq. (4.3).

This module is trained by an event-type detection loss $\mathcal{L}_{type}$ with golden label $Y \in \mathbb{R}^{N^{type}}$:

$$\begin{aligned}\mathcal{L}_{type} = -\sum_{i=1}^{N^{type}} (Y_i = 1)\mathrm{log}P(\hat{Y}_i) \\ +(Y_i = 0)\mathrm{log}P(1 - \hat{Y}_i).\end{aligned} \tag{4.5}$$

Event types with predicted confidence larger than 0.5 are viewed as presented in the input document. We denote the presented event types as $\mathcal{E}_T = \{E_1, \cdots, E_n\}$.

4.3.4 Event Extraction

Unlike previous works, which use multiple specialized modules to predict possible events, we propose to use just one plain transformer (named decoding transformer or Trans-D) to achieve the event extraction. To accurately predict events in a document, several challenges must be addressed: (1) how many events are presented? (2) how to handle the multi-mentioned of the same entity? Furthermore, (3) how to avoid predicting repeat events?

To address these issues, we propose a generative decoding scheme that utilizes a set of learnable queries and makes the decoding process context-aware and recurrent, as illustrated in the right panel of Figure 4.3.

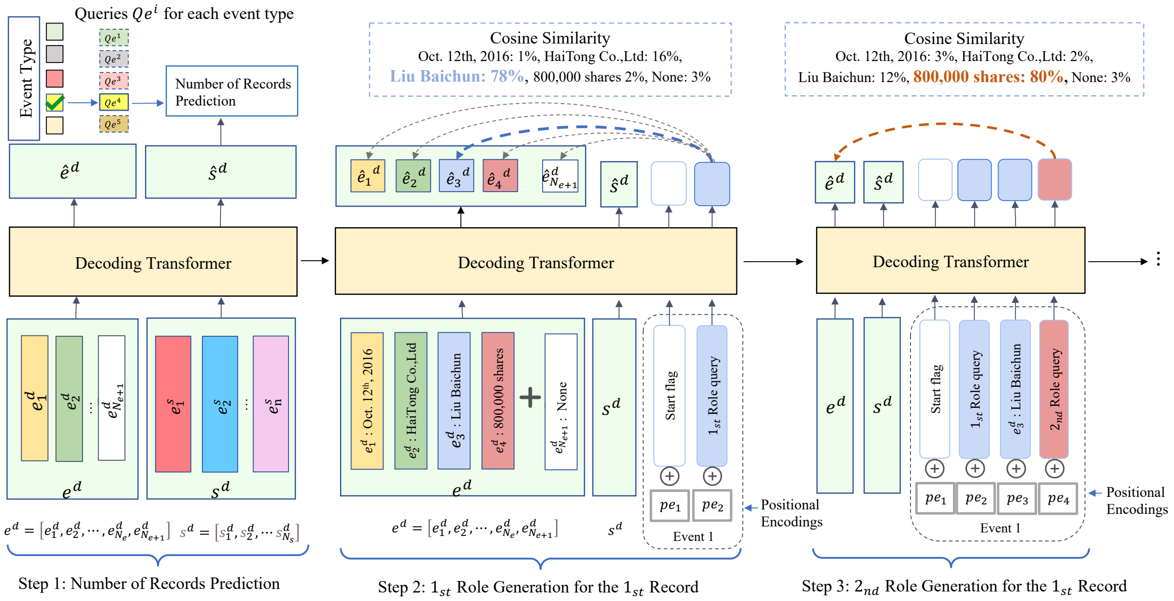


Figure 4.3: Illustration of generating the first event record.

For the input document D, with the predicted event types $\mathcal{E}_T$ at hand (see Section 4.3.3), we first estimate the number of event records of each event type as shown in the left panel of

Figure 4.3.

Estimate the number of records per event type. The calculation starts by first letting the generator transformer process $\{e^d, s^d\}$ to obtain further refined embeddings $\hat{e}^d, \hat{s}^d$:

$$\hat{e}^d, \hat{s}^d = \text{Trans-D}(\{e^d, s^d\}). \tag{4.6}$$

Then we will use a similar procedure as in Section 4.3.3 to estimate the number of records for each event type. Specifically, for each event type E_i, we will learn a specific query embedding $Q_e^i \in \mathbb{R}^d$, and let it go through an MHA with $\hat{s}^d$. The output embedding corresponding to Q_e^i, Q_e^i and then use a classifier W_c (shared across event types) to estimate the number of records:

$$\begin{aligned} \hat{Q}_e^i &= \text{MultiHead}(Q_e^i, \hat{s}^d) \\ P_c &= softmax(W_c^\top \hat{Q}_e^i), \end{aligned} \tag{4.7}$$

where P_c is the posterior probability of the number of the record. We set the dimension of P_c to the maximal number of records in the dataset.

This module can be trained by a cross-entropy loss, denoted as $\mathcal{L}_{num}$, with the ground-truth record number.

Event argument extraction. Next, the decoding transformer will progressively extract event arguments in a recurrent manner as shown in the middle and right of Figure 4.3. Specially, we construct an input sequence by appending an embedding, fixed or learnable, at each time step. Formally, this process can be described as:

$$\begin{aligned} o_t &= \text{Trans-D}(\{e_d, s_d, z_1, \cdots, z_t\}) \\ o_{t+1} &= \text{Trans-D}(\{e_d, s_d, z_1, \cdots, z_t, z_{t+1}\}), \end{aligned} \tag{4.8}$$

where e_d, s_d are embedding sequences obtained from Eq. (4.2). z_t is the input embedding appended at time step t. Unlike the standard generative model, where z_t is obtained from o_t, z_t in our design varies according to a predefined order. In particular, we sequentially decode event records and use multiple consecutive embeddings for a given event. In other words, the input can be seen as $\{e_d, s_d, z_1^1, \cdots, z_1^2, z_2^2, \cdots, z_{t+1}^m\}$ with z_j^i denotes the j-th input embedding for the i-th event. o_t is the output embedding at time step t. Depending on the type of z_t, o_t can be ignored, used to generate output, or used to calculate the next step input embedding.

Below we elaborate on the design of input embeddings for one event record extraction.

(1) [StartFlag]: for each event record, we start with a learnable embedding [StartFlag]. This flag indicates the start of an event record, and the output embedding for [StartFlag] will not be used.

(2) [Event Role Embeddings]: As predefined by the event extraction task [236], each record is composed of $N_{E_i}^r$ event roles. A learnable embedding is assigned and used as the input for each event role. The output embedding corresponding to [Event Role Embeddings] will be compared against all output embeddings of entity embeddings e_d (denoted as $\{e_i^r\}$ hereafter) by the cosine similarity. The entity with the highest similarity score will be chosen as the event argument.

(3) Previously extracted event argument: After extracting one event argument, we will use its corresponding entity embedding as the input embedding for the next time step. In this way, we can inform the decoding transformer of what has been detected from the document.

Thus, for a given event record with $N_{E_i}^r$ event roles, we will create an input sequence by interleaving event role embedding, and its extracted event augment entity embedding. This decoding step is demonstrated in Figure 4.3. Also, we add positional encodings to the input embeddings to make the decoding transformer be aware of the input order:

$$\begin{aligned} pe_{(pos,2i)} &= sin(pos/10000^{2i/d_{model}}) \\ pe_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{model}}). \end{aligned} \tag{4.9}$$

The positional encodings have the same dimension d_{model} as the embeddings, where the pos is the position, and i is the dimension.

Avoid repeated records. To prevent the decoding transformer predicts repeat or highly-similar records, we design a simple yet effective rule-based strategy: (1) We choose the entity with the highest similarity score with the output embedding e_i^r if this entity has not filled the corresponding roles in the previous predicted records. (2) If the highest similarity score entity has been assigned to a previous record, we check if we have other entities that the similarity score is higher than a predefined threshold θ. If yes, we choose the entity with the largest similarity among these entities; otherwise, the entity with the largest similarity will be chosen, although it has been used in other records.

4.3.5 Optimization

During training, the total loss is the sum of losses from four sub-tasks as follows:

$$\begin{aligned}\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{ner} + \lambda_2 \mathcal{L}_{type} + \lambda_3 \mathcal{L}_{num} \\ + \lambda_4 \mathcal{L}_{argu},\end{aligned} \tag{4.10}$$

where $\mathcal{L}_{ner}$, $\mathcal{L}_{type}$, $\mathcal{L}_{num}$, and $\mathcal{L}_{argu}$ are cross-entropy loss function for entity recognition, event type prediction, number of records prediction, and event argument extraction, respectively. λ_1, λ_2, λ_3 and λ_4 are hyper-parameters.

4.4 Experiments and Analysis

4.4.1 Experimental Setup

Dataset. We evaluate our method on the public dataset ChFinAnn collected by Zheng et al. [236]. This is a large-scale DEE dataset that includes a total of 32,040 documents and covers five different types of financial events related to the stock market, such as Equity Freeze (EF), Equity Repurchase (ER), Equity Underweight (EU), Equity Overweight (EO), and Equity Pledge (EP). We follow the standard split of the dataset, 25,632/3,204/3,204 documents for training/dev/test. The dataset presents a challenge due to the complexity: an average of 20 sentences per document and 912 tokens per document. Additionally, event records typically involve around six sentences, and 29% of the documents contain multiple events.

Models		**EF**			**ER**			**EU**			**EO**			**EP**	
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
DCFEE-O	66.0	41.6	51.1	84.5	81.8	83.1	62.7	35.4	45.3	51.4	42.6	46.6	64.3	63.6	63.9
DCFEE-M	51.8	40.7	45.6	83.7	78.0	80.8	49.5	39.9	44.2	42.5	47.5	44.9	59.8	66.4	62.9
GreedyDec	79.5	46.8	58.9	83.3	74.9	78.9	68.7	40.8	51.2	69.7	40.6	51.3	*85.7*	48.7	62.1
Doc2EDAG	77.1	64.5	70.2	*91.3*	83.6	87.3	*80.2*	65.0	71.8	*82.1*	69.0	*75.0*	80.0	**74.8**	*77.3*
DE-PPN-1	77.8	55.8	64.9	75.6	76.4	76.0	76.4	63.7	69.4	77.1	54.3	63.7	85.5	43.0	57.2
DE-PPN	78.2	**69.4**	**73.5**	89.3	85.6	87.4	69.7	**79.9**	**74.4**	**87.0**	*71.3*	**75.8**	83.8	*73.7*	**78.4**
Ours-1	**84.3**	51.6	64.0	**93.6**	*87.1*	*90.2*	**81.7**	57.8	67.7	79.0	55.9	65.5	**88.5**	50.1	64.0
Ours	*80.2*	*65.9*	*72.4*	90.3	**90.5**	**90.4**	76.1	*69.4*	*72.6*	74.6	**72.0**	73.3	79.3	70.7	74.8

Table 4.1: Overall event-level precision (P), recall (R) and F1-score (F1) on the ChFinAnn dataset. The top two results are highlighted in red bold and blue italic fonts, respectively.

Evaluation Metrics. For fair comparisons, we adopt the same evaluation standard used in Doc2EDAG [236], and DE-PPN [215]. Specifically, for each predicted event, the most similar ground truth is chosen without replacement to calculate the Precision(P), Recall(R), and F1-score.

Implementation Details. We take a document as input and set the maximum number of sentences and maximum sentence length as 64 and 128, respectively. We set the maximum record number as 20. We employ the basic Transformer [191], featuring 768 hidden units and 8 attention heads in each layer. During training, we used the Adam [97] optimizer with a learning rate of 1e-4 for 100 epochs. We set $\lambda_1 = 0.1$, $\lambda_2 = 0.4$, λ_3 =1, and $\lambda_4 = 1$ for the loss function. Additional information and specific hyperparameters can be found in the Appendix D.1.

Models	EF		ER		EU		EO		EP		Avg.		
	S.	M.	S.	M.	S.	M.	S.	M.	S.	M.	S.	M.	S.&M.
DCFEE-O	56.0	46.5	86.7	54.1	48.5	41.2	47.7	45.2	68.4	61.1	61.5	49.6	58.0
DCFEE	48.4	43.1	83.8	53.4	48.1	39.6	47.1	42.0	67.0	60.0	58.9	47.7	55.7
GreedyDec	75.9	40.8	81.7	49.8	62.2	34.6	65.7	29.4	88.5	42.3	74.8	39.4	60.5
Doc2EDAG	80.0	61.3	89.4	68.4	77.4	64.6	79.4	*69.5*	85.5	*72.5*	82.3	67.3	76.3
DE-PPN-1	*82.4*	46.3	78.3	53.9	**82.2**	45.6	78.1	39.3	82.8	38.5	80.7	44.7	66.2
DE-PPN	82.1	*63.5*	89.1	*70.5*	79.7	**66.7**	*80.6*	**69.6**	88.0	**73.2**	83.9	**68.7**	*77.9*
Ours-1	**83.1**	44.0	**93.5**	56.8	*81.6*	47.0	**80.9**	42.3	**91.4**	43.1	**86.1**	46.7	71.5
Ours	81.2	**64.8**	*92.6*	**73.6**	78.4	*65.4*	80.3	65.5	*89.6*	66.2	*84.4*	*67.1*	**78.3**

Table 4.2: F1-score for all event types and the averaged ones (Avg.) on single-event (S.) and multi-event (M.) sets. The top two results are highlighted in red bold and blue italic fonts, respectively.

4.4.2 Results

We compare our method with the following state-of-the-art baselines: **DCFEE** Yang et al. [214] introduced the DCFEE method for key-event detection, which uses the arguments from key-event mentions and surrounding sentences to fill an event table. There are two versions of the DCFEE: **DCFEE-O**, which extracts only one event, and **DCFEE-M**, which extracts multiple events from a document. **Doc2EDAG** Zheng et al. [236] proposed an end-to-end model for DEE that employs a transformer encoder to obtain sentence and entity embeddings and approaches DEE by directly filling event tables with entity-based path expending. **Greedy-Dec** is a variant of Doc2EDAG, which generates only one record greedily. **DE-PPN** Yang et al. [215] proposed a document-level encoder and a multi-granularity decoder to extract events in parallel with document-aware representations. There are two versions of the DE-PPN: **DE-PPN-1**, which only generates one event, and **DE-PPN**, which generates multiple events from a document.

Overall Performance. Table 4.1 shows results on the test set under each event type. Overall, our framework performs favourably compared to state-of-the-art methods. Specifically, our method achieves the best F1-score for event type ER, which renders a large margin (3% absolute improvement) to the state-of-the-art method DE-PPN. On other event types, our method ranks 2nd with a slight performance drop. As we use a plain transformer as a decoder without pretraining on a large corpus, these results show that our method can serve as a strong baseline for future exploration.

Single-Event vs. Multi-Event. To get a closer look at our method, we evaluate single-event and multi-event documents separately. The results are presented in Table 4.2. Our method (Ours-1) achieves the best F1-score on 4 out of 5 event types on single-event documents. Moreover, on multi-event documents, our method performs the best on type EF and ER, ranks 2nd on type EU, and performs badly on type EP (about 7% left behind). This shows one possible direction to improve the performance in the future.

Performance w.r.t. the number of sentences in documents. In general, longer articles are usually more complex and contain more irrelevant information, which makes the event extraction task more difficult. To comprehensively reflect the performance of models, we also report the results when handling documents containing a different number of sentences. To this end, we first count the total number of sentences in each document in the test set. Then, we compute the frequency of the number of sentences that appeared in the test split. The statistics are shown in Figure 4.4. We find that documents with 6 to 10 sentences make up 3%, those with 11 to 15 sentences make up 33%, those with 16 to 20 sentences makeup 23%, and those with more than 20 sentences make up 33%. Next, we compute the F1-score under these splits, and the results are shown in Figure 4.5.

We observe that our model performs best when documents have less than 20 sentences compared with state-of-the-art methods DE-PPN and Doc2EDAG. When documents have more than 20 sentences, DE-PPN performs much better than ours. Considering this dataset has 41% documents with more than 20 sentences, a model would benefit significantly if it is good at handling long documents.

4.4.3 Ablation Studies

To verify the effectiveness of key designs in our framework, we carry out the following ablation studies: (1) *-NumRecord:* replacing the prediction of the number of records with

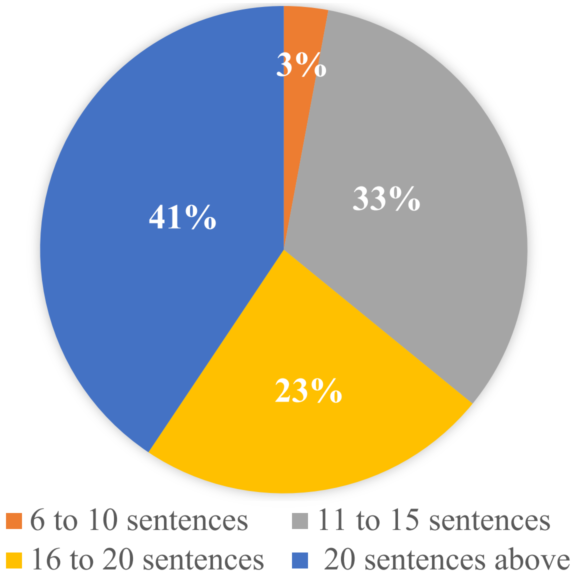


Figure 4.4: The percentage of documents with different numbers of sentences in the test split.

a binary prediction to determine whether there is a next record. We concatenate a learnable embedding [isNext] to the output embedding of previously generated events and feed them into the Transformer model. Then, we use the output embedding of [isNext] to conduct a binary classification. We keep generating events until the classifier predicts there is no further event. (2) *-RuleStrategy:* replacing the rule-based strategy with choosing the largest similarity for each event argument prediction. (3) *-ArgumentEmbedding:* removing the already predicted argument embeddings from the input of the decoding transformer. (4) *-DecodingScheme:* replacing the recurrent scheme with the traditional standard transformer scheme, namely generates all event arguments in parallel.

The results are shown in Table 4.3. It shows that: (1) The prediction of the number of records plays a critical role, enabling our decoding transformer to generate a more precise number of events, resulting in an average improvement of +4.7% F1-score. (2) The rule-based approach contributes +3.5% on average to the final performance via reducing repetitive or highly similar records. (3) The already-predicted argument embeddings in the

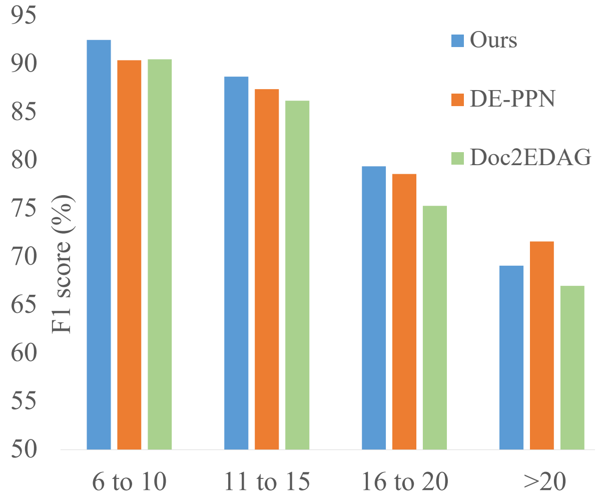


Figure 4.5: Performance of our method against two state-of-the-art methods w.r.t. different document lengths.

Model	**EF**	**ER**	**EU**	**EO**	**EP**	**Avg.**
Ours	72.4	90.4	72.6	73.3	74.8	76.7
-NumRecord	-6.3	-5.8	-4.6	-6.8	-5.1	-4.7
-RuleStrategy	-5.2	-0.2	-1.8	-4.3	-5.8	-3.5
-ArgumentEmbedding	-3.4	-2	-1.2	-4	-1.9	-2.5
-DecodingScheme	-13	-3.7	-9.3	-11.7	-9.3	-9.4

Table 4.3: Ablation study: F1-score of variants of our model under each event type and the average performance.

input of the decoding transformer also play an important role in accurate event extraction, suggesting awareness of historical context leading to better performance (+2.5% F1-score on average). (4) The recurrent decoding scheme contributes the most, which brings an average improvement of 9.4%. This indicates the importance of historical context, namely the already-generated arguments and records.

4.5 Related Work

Event extraction involves identifying specific types of events and extracting the associated event arguments from the text. This task can be further divided into sentence-level and document-level event extraction.

Sentence-level Event Extraction Previous efforts in event extraction (EE) mainly focus on sentence-level extraction using the benchmark dataset ACE2005 [5]. Most of them cast event extraction as a classification problem, using global features to capture dependencies among local classifiers and applying joint inference [113, 129, 160, 196, 213, 220]. On the other hand, event extraction is viewed as an extractive machine learning comprehension task [53, 109, 133], where models are trained to identify relevant answers to a variety of framing questions for each event component. Most recently, a few generation-based event extraction models [79, 85, 142, 163] have been proposed, which generate all arguments and their roles as a way to convert text into a structured form. While these sentence-level methods perform well in extracting events within a sentence, they may struggle in real-world scenarios where document-level texts contain multiple sentences with scattered multiple events.

Document-level Event Extraction Many real-world applications such as finance, legislation, or health require DEE, where the event information scatters across the whole article. Existing DEE works could be generally divided into two groups. The first group mainly focuses on extracting scattering event arguments in the document [52, 54, 55, 233]. However, these methods assume that the event type or triggers are given in advance, which may not be applicable in some real-world applications. The second group of DEE researches follow the detect-then-extraction paradigm similar to SEE to extract events from the document [83, 118, 214]. Researchers attempt to conduct DEE in a more realistic trigger-free on the ChFinAnn dataset [236], where event types are directly predicted based on the document semantics. In Zheng et al. [236] a transformer-based end-to-end model is proposed to solve the DEE problem by filling the event tables with an entity-based directed acyclic graph. Yang et al. [215] proposed an encoder-decoder model to generate events in parallel with document-aware representations from a document. Despite progress, these methods rely on highly custom-designed network architecture or decoding processes.

4.6 Conclusion

In this work, we propose a generative solution for document-level event extraction that takes into account recent developments in generative event extraction, which have been successful at the sentence level but have not yet been explored for document-level extraction. Our proposed solution includes an encoding scheme to capture entity-to-document level information and a decoding scheme that takes into account all relevant contexts. Extensive experimental results demonstrate that our generative-based solution is able to perform as well as state-of-the-art methods that use specialized structures for document event extraction. This allows our method to serve as an easy-to-use and strong baseline for future research in this area.

Statement of Authorship

Title of Paper	Semantic Role Labeling Guided Out-of-distribution Detection
Publication Status	□ Published □ Accepted for Publication ☑ Submitted for Publication □ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Under Review of ACL 2023.

Principal Author

Name of Principal Author (Candidate)	Jinan Zou		
Contribution to the Paper	Proposed ideas, conducted experiments and wrote paper.		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	04/01/2023

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Maihao Guo		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	08/01/2023

Name of Co-Author	Yu Tian		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	08/01/2023

Please cut and paste additional co-author panels here as required.

Name of Co-Author	Yuhao Lin		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	06/01/2023

Name of Co-Author	Haiyao Cao		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	05/01/2023

Name of Co-Author	Lingqiao Liu		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	23/01/2023

Name of Co-Author	Ehsan Abbasnejad		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	24/1/2023

Name of Co-Author	Javen Shi		
Contribution to the Paper	Discussion and revise the paper		
Signature		Date	27/01/2023

Chapter 5

Semantic Role Labeling Guided Out-of-distribution Detection

In Chapters 3 and 4, while conducting research on stock movement prediction and document-level event extraction, we identified several issues that require out-of-distribution detection for a realistic approach to stock market prediction. The reasons for this include:

- The stock market is constantly changing and new information is always being released, making it necessary to identify input data that is not representative of the training data and prevent the model from making predictions on data it is not familiar with. This can improve the overall accuracy of the model's predictions by reducing errors caused by unseen data.

- Financial documents, such as earnings reports and press releases, can vary greatly in format and content, making it difficult for a model to extract relevant information. It's essential to identify which parts of a document may be more challenging for the model to extract events from, allowing the model to focus on the most relevant parts and make more accurate predictions.

- Stock market prediction models are often required to make predictions on unseen data and require a high level of generalization ability in the model to handle unseen data and make accurate predictions.

- The current mainstream datasets for financial document-level event extraction, such as ChFinAnn [1] , are labelled using a distant supervision approach. However, this approach may not always be representative of the real-world data the model will encounter. OOD detection is important in distant supervision as it helps to improve the reliability and robustness of the models used.

Hence, incorporating OOD detection in our research will make our system more efficient, stable, and reliable. From this Chapter, we aim at another important yet challenging NLP task: OOD detection. The Out-of-distribution detection in Natural Language Processing aims to determine whether an instance is OOD. Previous methods commonly adopted an overall representation for a sentence when detecting an OOD instance, which assumes that the sentence-level representation could represent the complete information of a sentence. Identifying unexpected domain-shifted instances in natural language processing is crucial in real-world applications. Previous works identify the OOD instance by leveraging a single global feature embedding to represent the sentence, which cannot characterize subtle OOD patterns well. Another major challenge current OOD methods face is learning effective low-dimensional sentence representations to identify the hard OOD instances that are semantically similar to the ID data. In this paper, we propose a new unsupervised OOD detection method, namely Semantic Role Labeling Guided Out-of-distribution Detection (SRLOOD), that separates, extracts, and learns the semantic role labelling (SRL) guided fine-grained local feature representations from different arguments of a sentence and the global feature representations of the full sentence using a margin-based contrastive loss. A novel self-supervised approach is also introduced to enhance such global-local feature learning by predicting the SRL extracted role. The resulting model achieves SOTA performance on four OOD benchmarks, indicating the effectiveness of our approach.

[1] https://github.com/dolphin-zs/Doc2EDAG/blob/master/Data.zip

5.1 Introduction

Recent advances in natural language classification have shown tremendous improvements in various natural language processing tasks. Natural language classification is usually formulated as a close-set problem, where training and testing samples are from the same domain/distribution. Despite the accurate predictions on the inlier close-set classes, the classifier often fails to properly identify out-of-distribution (OOD) instances from other unknown/unexpected domains that deviate from the close-set training distribution, making it barely applicable to real-world scenarios. Tackling such failure cases is crucial to real-world safety-critical NLP applications. For instance, OOD instances can be represented by unknown sentences from different domains or distributions, such as semantically shifted sentences that can be incorrectly predicted as a part of the inlier classes, leading to potential impairment to user trust [8].

Despite the importance, little literature has addressed the problem of OOD detection in NLP. One proposed method is to train a model to increase the inter-class discrepancy of ID classes and tends to depend on classification uncertainty or latent embedding distance to detect OOD instances [240]. The high classification uncertainty association with OOD instances (i.e., max softmax or energy) is intuitive, but it obtains a few caveats. One of the major issues is that classification uncertainty happens when samples are close to classification decision boundaries. However, there is no guarantee that all OOD instances will be close to classification boundaries (i.e., subtle OOD samples may share similar semantic features to ID data), leading to a subpar performance in detecting OOD samples. Moreover, complicated inlier sentences containing more outlier components, such as punctuation and discourse fillers, tend to fall close to the decision boundary, which can incorrectly lead to high classification uncertainty. Latent embedding-based approaches rely on the assumption that the OOD instance resides outside a bounded or unbounded latent hyperspace constructed by the ID feature distributions [16, 74, 170, 186, 240]. However, it is challenging to define such a latent hyperspace to encode all possible ID features, significantly affected by many outlier components from a sentence, and the aforementioned subtle OOD issue still exists.

In this paper, we propose a new OOD detection method designed for NLP tasks, namely Semantic Role Labeling Guided Out-of-distribution Detection (SRLOOD), simultaneously extracting, separating, and learning both global and SRL-guided local fine-grained feature representations through a margin-based contrastive loss and self-supervision. In particular, our contributions can be summarised into three folds: (1) we propose SRLOOD that learns fine-grained low-dimensional representations by increasing the inter-class discrepancies between the concatenation of the global and SRL-guided local features of different ID classes. Our proposed SRLOOD aims to effectively eliminate the outlier phrases (e.g., punctuation and discourse fillers) and extract key local semantic components (e.g., verbs and arguments) from a sentence to better characterize subtle OOD samples; (2) a novel self-supervised pretext task is also proposed to strengthen the relations between different local arguments, further facilitating the optimization of SRL-guided local features; and (3) a Transformer block is introduced to resemble some of the SRL-guided features from a sentence, so our model is enforced to learn discriminative representations through such strong perturbations for the better discriminability of subtle semantic features. Extensive experiments on four different OOD benchmarks show that our resulting model achieves the best performance on four different scoring functions.

5.2 Related Work

Out-of-distribution detection Machine learning aims to design models that can learn generalizable knowledge from training data. The success of machine learning models lies in the assumption that training and test data share the same distribution. However, in many real-world tasks, it is unknown whether the training and test data share the same distribution. This potential distribution gap is known as OOD and can be a major issue, with the performance of classical ML models often deteriorating. To handle the OOD issue, OOD detection aims to detect whether test data is from the training distribution. Based on the availability of OOD data, recent methods can be categorized into classification methods, density-based methods and distance-based methods [217]. Classification methods often formulate the OOD task as a one-class classification problem, then use appropriate methods to solve it [27, 43, 71, 105, 153, 173, 189]. Hendrycks and Gimpel [69] proposed a softmax prediction probability baseline for error and out-of-distribution detection across several architectures and numerous datasets. Density-based methods [2, 16, 101, 244] in OOD detection explicitly model the in-distribution with some probabilistic models and flag test data in low-density regions as OOD. Zong et al. [245] utilizes a deep autoencoder to generate a low-dimensional representation and reconstruction error for each input data point, which is further fed into a Gaussian Mixture Model for anomaly detection. The main idea of distance-based methods is that the testing OOD samples should be relatively far away from the centroids of in-distribution classes [23, 105, 190, 226]. Previous methods primarily studied for computer vision [50, 88, 130, 225, 241] and OOD detection has been overlooked in NLP. Only few works recently that adapted the solutions designed for images into the text to leverage the features representation of an entire sentence for detecting the OOD case. For example, Zhou et al. [240] adapted a contrastive OOD detection from computer vision using a pre-trained Transformer to improve the compact news of representations and evaluate the trained classifier on the four text datasets.

In contrast, we propose a self-supervised SRL method to learn fine-grained feature representations of text data and shows that is a surprisingly effective approach for OOD detection.

Semantic Role Labeling Semantic role labelling(SRL) leads to the advancement of many NLP tasks and applications due to the clear detection of augments regarding predicates. For example, Sarzynska-Wawer et al. [175] proposed a BERT-based model incorporating semantic role labelling, which significantly improves the text understanding ability of the model.Chen et al. [22] used the verb-specific semantic role, a variant of semantic role labelling, for the controllable image captioning, which is a task about image description. Conditioned on the semantic role representation. More recently, Ross et al. [172] proposed a Tailor model for the sequence-to-sequence task, which gained a great improvement in measuring the reliance on syntactic heuristics.

Self-supervised Learning Self-supervised learning method and has been soaring and achieving big success in representative learning because of the powerful generalization ability. BERT (Pre-training of deep bidirectional Transformers for language understanding) proposed by Devlin et al. [40] are fine-tuned for many downstream tasks. as a result, BERT has become a milestone of not only NLP but also the development of self-supervised learning. Baevski et al. [9] built a platform based on a self-supervised method for either speech, text or computer vision. In the work of Hendrycks et al. [75], it is shown that the self-supervised method drastically improves the OOD detection performance on the difficult near-domain outliers. Self-supervised learning methods tackle the OOD in two aspects: (1) the enhancement of feature quality can improve OOD performance; (2) some well-designed surrogate tasks can help reveal the anomalies from OOD samples[217].

5.3 Methodology

Generally, the OOD instances can be defined as instances (x, y) sampled from an underlying distribution other than the training in-distribution $P(\mathcal{X}_{train}, \mathcal{Y}_{train})$, where $\mathcal{X}_{train}$ and $\mathcal{Y}_{train}$ are the training corpus and training label set. Specifically, an instance $(\boldsymbol{x}, y)$ is primarily deemed OOD if $y \notin \mathcal{Y}_{train}$ to be consistent with previous works [69, 71, 74, 240]. Following the previous work [240], we formally define the OOD detection task. Given the main task of natural language classification, the OOD detection task is the binary classification of each instance $\boldsymbol{x}$ as either ID or OOD, judged by its OOD score computed with scoring function $f(x) \rightarrow \mathbb{R}$. A lower OOD score value indicates ID where $y \in \mathcal{Y}_{train}$ and a higher OOD score value indicates OOD where $y \notin \mathcal{Y}_{train}$ (y is the underlying label for $\boldsymbol{x}$ and is unknown at inference).

The key idea of our proposed model, SRLOOD, is extracting and learning the SRL-guided fine-grained local representation. Building on top of this representation, a novel supervised approach is introduced to enhance such local argument representation.

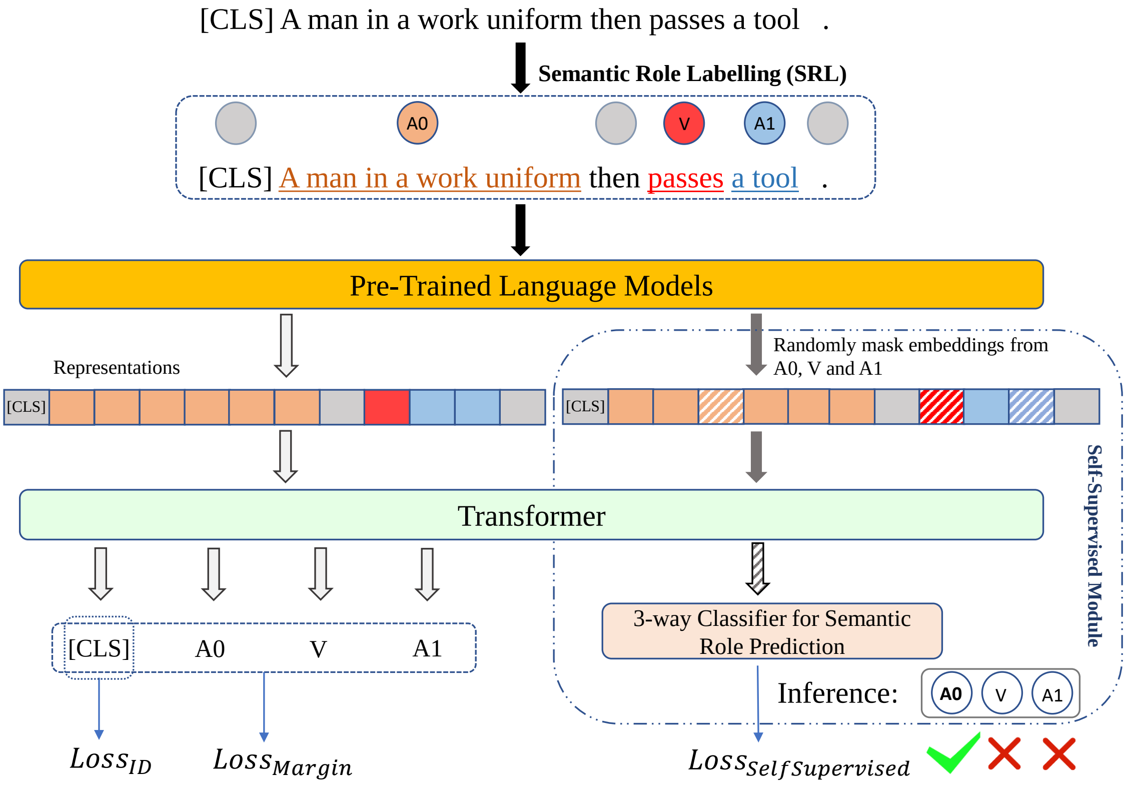

[CLS] A man in a work uniform then passes a tool .
Semantic Role Labelling (SRL)
A0
V
A1
[CLS] A man in a work uniform then passes a tool .
Pre-Trained Language Models
Representations
[CLS]
Randomly mask embeddings from A0, V and A1
[CLS]
Transformer
Self-Supervised Module
3-way Classifier for Semantic Role Prediction
[CLS]
A0
V
A1
Inference:
A0
V
A1
$Loss_{ID}$
$Loss_{Margin}$
$Loss_{SelfSupervised}$
✓ ✗ ✗

Figure 5.1: Model architecture of our framework.

Our model could attend to each sequence at a much finer, granular level in comparison with only using the global [CLS][240]. We innovate a self-supervised task based on a masking mechanism that randomly mask a proportion of A0, V, A1, where A0, V, A1 are the sets of all Argument-0's, Verbs, and Argument-1's in a sequence respectively, and let a trinary classifier infer A0, V, or A1 providing an averaged masked embedding representing either A0,V, or A1 in this training iteration (forward propagation) for each sequence in the batch. Unlike the pre-trained RoBERTa language model, the Transformer Encoder is initialized from scratch. The self-supervised task exerts pressure on the Transformer Encoder in order to let it well comprehend the in-domain (ID data) in just a few training epochs, and generate better representations of each token than using RoBERTa alone. Our framework consists of (1) semantic role labelling, (2) an SRL-guided self-supervised module, and (3) OOD detection with OOD scoring functions as illustrated in Figure 5.1.

5.3.1 Semantic Role Labeling

The task of SRL is to determine the underlying predictive argument structure of a sentence and to provide representations that can answer the basic questions about the meaning of the sentence, including who did what to whom [155]. Therefore the SRL primarily extracts the essential features and passingly filters out outlier phrases (e.g., punctuation and discourse fillers). We leverage off-the-shelf SRL-BERT [182] to label each token sequence in a batch with Propbank [98] semantic roles proto-agent, verb, and proto-patient, then labelled tokens are recorded into sets A0, V, and A1 respectively, as illustrated in Figure 5.1. Each token sequence is fed to the pre-trained language model, whose output is fed to the Transformer block. We compute the mean of A0, V, A1 embeddings $\boldsymbol{\mu}_{A0}$, $\boldsymbol{\mu}_{V}$, and $\boldsymbol{\mu}_{A1}$ pooled from the Transformer's output for fine-grained feature representations.

5.3.2 Self Supervised Learning based on SRL

Our proposed SRLOOD framework uses SRL to extract and learn key local semantic features and use self-supervision to further strengthen such fine-grained local representations. We introduce strong perturbation by randomly masking a certain percentage of SRL-extracted local representations for better generalization on detecting hard OOD instances. Guided by SRL, strong perturbation is independently exerted on the pre-trained language model representations of A0, V, and A1 according to a generated and recorded supervising ground truth label for each sequence. The perturbed embeddings are input to the Transformer encoder.

Subsequently, we compute the mean embeddings of A0, V, or A1 from the Transformer's output and use them for an auxiliary three-way classification task. This task aims to improve the feature discriminability by predicting the semantic role of a given embedding, computing the mean embeddings, and inputting one of them to a classifier for semantic role prediction according to its self-supervising label. To this end, our framework consists of the pre-trained language model, the Transformer head, the SRL-guided pooling, and the 3-way self-supervised classifier. This framework is optimized by the loss functions introduced in the next section.

5.3.3 Loss Functions

Our model extracts the [CLS] embedding, the averaged embeddings of all A0, V, and A1 respectively from the firstly unmasked Transformer output, and propagates the concatenation of the four embeddings through a 2-layer MLP, and uses their output to compute the Margin loss. The MLP for [CLS] embedding propagates through an additional fully connected layer and outputs its logits for the ID sequence classification task, and generates an ID loss.

We adopt the margin-based contrastive loss that drives the model to encode tokens in the same ID class with adjacent SRL-guided comprehensive representation measured by L2 distances:

$$\begin{aligned}\ell_{\text{margin}} = \frac{1}{md} & \left[\sum_{i=1}^{m} \frac{1}{|P(i)|} \sum_{p \in P(i)} \|\boldsymbol{h}_i - \boldsymbol{h}_p\|^2 \right. \\ & \left. + \sum_{i=1}^{m} \frac{1}{|N(i)|} \sum_{n \in N(i)} \left(\xi - \|\boldsymbol{h}_i - \boldsymbol{h}_n\|^2\right)_+ \right],\end{aligned} \tag{5.1}$$

where $P(i)$ is the subset of training data with the same class as instance i, $N(i)$ is the subset of training data with different class labels from instance i, m is the number of instances in the entire training set. The d is the dimensionality of the comprehensive representation of a sequence

$$\boldsymbol{h} = Concat(\boldsymbol{h}_{[CLS]}; \boldsymbol{\mu}_{A0}; \boldsymbol{\mu}_{V}; \boldsymbol{\mu}_{A0}). \tag{5.2}$$

where $\boldsymbol{h}_{[CLS]}$ is the [CLS] embedding, $\boldsymbol{\mu}_{A0}$, $\boldsymbol{\mu}_{V}$, $\boldsymbol{\mu}_{A1}$ are the mean embeddings pooled from the Transformer's output according to A0, V, A1 respectively. The Margin Loss will give rise to clusters in the latent space of $\boldsymbol{h}$. Combined with cross-entropy losses ℓ_{ID} and ℓ_{SSL} from the ID sequence classification task and the self-supervised task, respectively. The total loss is their weighted sum with hyper-parameters α_1, α_2 and α_3:

$$\ell_{total} = \alpha_1 \ell_{ID} + \alpha_2 \ell_{Margin} + \alpha_3 \ell_{SSL}, \tag{5.3}$$

5.3.4 Scoring Functions

During OOD inference, we extract the local key components/features using SRL-BERT [182] based on a previously fine-tuned language model backbone. We compute the mean embeddings $\boldsymbol{\mu}_{A0}$, $\boldsymbol{\mu}_{V}$, and $\boldsymbol{\mu}_{A1}$ for local feature representations. The [CLS] embedding $\boldsymbol{h}_{[CLS]}$ is used for global feature representations. The global and local representations are then concatenated together to produce the final feature vector to represent a sentence:

$$\boldsymbol{h} = Concat(\boldsymbol{h}_{[CLS]}; \boldsymbol{\mu}_{A0}; \boldsymbol{\mu}_{V}; \boldsymbol{\mu}_{A0}). \tag{5.4}$$

For a fair comparison, we use the same OOD scoring functions as [240]. For the validation set $\mathcal{D}^{val} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{D}^{val}|}$, we computed the Mahalanobis distance based on the class mean embedding

$$\boldsymbol{\mu}_c = \mathbb{E}_{y_i = c}\left[\boldsymbol{h}_i\right], c \in C, \tag{5.5}$$

and its covariance:

$$\boldsymbol{\Sigma} = \mathbb{E}\left[\left(\boldsymbol{h}_i - \boldsymbol{\mu}_{y_i}\right)\left(\boldsymbol{h}_i - \boldsymbol{\mu}_{y_i}\right)^{\intercal}\right], \tag{5.6}$$

where C is the number of classes. The OOD score is then defined as the minimum Mahalanobis distance among the C ID classes given an instance $\mathbf{x}$ during inference

$$S = -\min_{c=1}^{C}(\boldsymbol{h} - \boldsymbol{\mu}_c)^{\intercal}\boldsymbol{\Sigma}^{\dagger}(\boldsymbol{h} - \boldsymbol{\mu}_c), \tag{5.7}$$

where $\boldsymbol{\Sigma}^{\dagger}$ denotes the pseudo-inverse of the covariance matrix $\boldsymbol{\Sigma}$. Such a distance considers both the global sentence features and the SRL-guided local features, enabling better performance on OOD detection.

For cosine similarity, we compute the maximum cosine similarity of the concatenated feature representation $\boldsymbol{h}$ to instance features of the validation set

$$\mathcal{H}^{val} = \{(\boldsymbol{h}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{H}^{val}|}. \tag{5.8}$$

The OOD score is computed as

$$s = -\max_{i=1}^{|\mathcal{H}^{val}|} \cos\left(\boldsymbol{h}, \boldsymbol{h}_i\right). \tag{5.9}$$

5.4 Experiments

5.4.1 Datasets

We evaluate our method for multiple NLP datasets, following the same criteria for choosing ID and OOD settings in the previous work [240]. The ID datasets correspond to three natural language classification tasks, including sentiment analysis, topic classification, and question classification: SST2 [183]; IMDB [145], 20 Newsgroup dataset [103], and TREC-10 dataset [120]. For the OOD dataset, any pair of datasets mentioned above for different tasks can be regarded as OOD to each other. Following the previous work [240], we also employ additional datasets solely as the OOD data (See Appendix D.3).

We utilize alternative datasets to ID data for three natural language classification tasks, these are:

Sentiment Analysis. For this task, we have included two datasets, *SST2* and *IMDB*, which are both used for sentiment analysis. The sentences in these datasets are labelled as either positive or negative. The SST2 dataset comes with pre-defined train, validation, and test splits, while for IMDB, we have randomly chosen 10% of the training instances as the validation set. It's important to note that both datasets belong to the same task and are not considered out-of-distribution to each other.

Topic Classification. We use *20 Newsgroup*, a dataset that is used for topic classification which contains 20 different categories. The entire dataset has been randomly divided into three sets: 80% for training, 10% for validation, and the remaining 10% for testing.

Question Classification. *TREC-10* is a dataset that categorizes questions according to the type of answer they are seeking. We used the coarse version of the dataset, which has 6 categories. From the training set, we have randomly selected 10% of the instances as the validation set.

The three tasks above can be considered OOD to each other when using different datasets. Additionally, four more datasets were selected specifically as OOD data: the combination of premises and hypotheses from the *RTE* [10, 38, 63, 66] and *MNLI*[206], which are *NLI* datasets, and the English side of the *WMT16*[14] and *Multi20K*[57] Machine Translation datasets. The test splits of these datasets were used as OOD instances during testing. For MNLI, both the matched and mismatched test sets were used. The test set for *Multi20K* was the union of the flickr 2016 English, mscoco 2017 English, and flickr 2018 English test sets. These datasets were not used as ID data for several reasons: 1) *WMT16* and *Multi30K* are MT datasets and not applicable to natural language classification problems, 2) It is challenging to determine OOD instances for NLI datasets because they are labelled with comprehensive relationships such as entailment/non-entailment for *RTE* and entailment/neutral/contradiction for MNLI.

5.4.2 Evaluation Metrics

We adopted the same two metrics [240] commonly used for measuring OOD detection performance in machine learning research [69, 105]: **AUROC** and **FAR95**. For an evaluation metric, we used (1) the area under the receiver operating characteristic (AUROC) and (2) the False Alarm Rate (FAR 95).

For AUROC, it is calculated as the area under the ROC curve. A ROC curve shows the trade-off between a true positive rate (TPR) and a false positive rate (FPR) across different decision thresholds. ***High*** AUROC values are interpreted as a stronger ability of OOD.

For FAR 95, it is the probability that an in-distribution example raises a false alarm, assuming that 95% of all out-of-distribution examples are detected. ***Lower*** FAR 95 values indicate better OOD performance.

5.4.3 Experiments Setting

We conducted all experiments based on the same codebase and used the same $\mathrm{RoBERTa}_{\mathrm{LARGE}}$ from previous work [240]. In our framework, the RoBERTa-Large pre-trained by [139] is fine-tuned. It has 24 layers and 16 attention heads. The warm-up ratio for the learning rate is 0.06. The hyper parameters $\alpha_1 = 1$, $\alpha_2 = 3$, $\alpha_3 = 1$, batch size is 12, the optimal masking ratio for the self-supervised module is 0.3. The model is trained for 10 epochs. The full hyper-parameters we used in this paper are in the Appendix D.1 in detail. During the evaluation, when one of SST2, IMDB, TREC-10, and 20 Newsgroup datasets is chosen as ID, the remaining datasets serve as OOD datasets. Followed with [240], SST2 and IMDB are deemed as binary sentiment classification tasks and once SST2 was ID, the IMDB would not be included in the OOD datasets, and vice versa.

5.4.4 Compared Methods

We compare our method with the SOTA methods as follows: [240] proposed to fine-tune the Transformers with a contrastive loss. There are three variants of the model: RoBERTa without contrastive loss, RoBERTa with supervised contrastive loss, and RoBERTa with margin-based loss. We compare our method with three baselines: OOD detection using probabilities from softmax distributions w/o $\mathcal{L}_{Cont}$-MSP [69], fine-tuning the Transformers with supervised contrastive loss w/ $\mathcal{L}_{SCl}$, and with margin-based loss w/ $\mathcal{L}_{margin}$ [240].

AUROC↑ /FAR95↓		**Avg**	**SST2**	**IMDB**	**TREC-10**	**20NG**
w/o $\mathcal{L}_{Cont}$	MSP	94.1/35.0	88.9/61.3	94.7/40.6	98.1/7.6	94.6/30.5
	Energy	94.0/34.7	87.7/63.2	93.9/49.5	98.0/10.4	96.5/15.8
	Maha	98.5/7.3	96.9/18.3	99.8/0.7	99.0/2.7	98.3/7.3
	Cosine	98.2/9.7	96.2/23.6	99.4/2.1	99.2/2.3	97.8/10.7
w/ $\mathcal{L}_{SCl}$	$\mathcal{L}_{scl}$+MSP	90.4/46.3	89.7/59.9	93.5/48.6	90.2/36.4	88.1/39.2
	$\mathcal{L}_{scl}$+Energy	90.5/43.5	88.5/64.7	92.8/50.4	90.3/32.2	90.2/26.8
	$\mathcal{L}_{scl}$+Maha	98.3/10.5	96.4/26.6	99.6/2.0	99.2/1.9	97.9/11.6
	$\mathcal{L}_{scl}$+Cosine	97.7/13.0	95.9/28.2	99.2/4.2	99.0/2.4	96.8/17.0
w/ $\mathcal{L}_{margin}$	$\mathcal{L}_{margin}$+MSP	93.0/33.7	89.7/49.2	93.9/46.3	97.6/6.5	90.9/32.6
	$\mathcal{L}_{margin}$+Energy	93.9/31.0	89.6/48.8	93.4/52.1	98.4/4.6	94.1/18.6
	$\mathcal{L}_{margin}$+Maha	99.5/1.7	99.9/0.6	100/0	99.3/0.4	98.9/6.0
	$\mathcal{L}_{margin}$+Cosine	99.0/3.8	99.6/1.7	99.9/0.2	99.0/1.5	97.4/11.8
Ours	MSP	*94.8/24.7*	*90.8/46.4*	*97.0/18.3*	*98.6/2.5*	92.9/31.4
	Energy	*95.7/20.7*	*90.4/45.5*	*97.0/19.9*	*98.5/3.2*	*96.9/14.0*
	Maha	*99.6/0.8*	99.4/2.2	99.5/0.7	*99.9/0*	*99.1/0.8*
	Cosine	*99.0/3.5*	98.7/6.5	98.7/4.8	*99.5/0.4*	*98.9/2.3*

Table 5.1: OOD Detection performance (in %) of RoBERTa$_{\text{LARGE}}$ trained on the four different ID datasets. Due to space limits, for each of our training ID datasets, we report the macro average of AUROC and FAR95 on all OOD datasets (See Appendix for full results). Following the standard [240], the results of our SRL-guided Self Supervision method achieving SOTA on both evaluation metrics are in blue italic fonts.

5.4.5 Main Results

As shown in Table 5.1, our SRLOOD achieves the best results across multiple OOD benchmarks when compared with three different SOTA methods. Particularly, our model achieves the State-of-the-art mean AUC and FAR performance with MSP, energy, Cosine, and Mahalanobis distance as scoring functions. Without fine-tuning the classification logits, our model obtains significant improvements to the previous SOTA approaches. This suggests that our SRLOOD is able to learn effective global and local fine-grained representations, enabling better generalisation for both ID and OOD classification. Furthermore, our model achieves comparable mean detection performance with Mahalanobis and Cosine OOD score when against the previous method [240].

5.4.6 Ablation Studies

In Table 5.2, we justify the effectiveness of different proposed components on IMDB and TREC benchmarks. Note that the baseline method refers to the model trained using ℓ_{margin} without SRLOOD, transformer, and Self-supervised learning modules. The results demonstrate that each module contributes significant improvements in terms of AUROC and FAR95 on both benchmarks, indicating the effectiveness of all proposed components.

Baseline	**SRL**	**SSL**	**IMDB** (AUROC↑/FAR95↓)				**TREC** (AUROC↑/FAR95↓)			
			MSP	Energy	Maha	Cosine	MSP	Energy	Maha	Cosine
✓			93.9/46.3	93.4/52.1	**1/0**	**99.9/0.2**	97.6/6.5	98.1/4.6	99.3/0.7	99.0/1.5
✓	✓		95.6/25.4	95.6/25.5	99.7/0.4	99.6/1.3	97.6/6.9	97.6/5.1	99.2/0.6	99.3/0.7
✓	✓	✓	**97.0/18.3**	**97.0/19.9**	**99.9/0**	98.7/4.8	**98.6/2.5**	**98.2/3.2**	**99.9/0**	**99.5/0.4**

Table 5.2: Ablation study of our method on IMDB and TREC

5.5 Conclusion

In this paper, we propose a simple yet effective approach called Semantic Role Labeling Guided Out-of-distribution Detection (SRLOOD), which learns from both global and SRL-guided local fine-grained feature representation to detect OOD instances in NLP. The model jointly optimizes both global and local representations using a margin-based contrastive loss and self-supervised loss. The resulting model is able to effectively extract the key semantic roles and eliminate outliers from a sentence to detect subtle OOD samples effectively. The resulting model shows SOTA performance on four different OOD benchmarks with four different OOD scoring functions, indicating the effectiveness of our proposed SRLOOD framework.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this thesis, we have explored the natural language processing techniques in stock prediction and event extraction in the finance domain. We have studied two ill-posed tasks related to stock prediction: stock movement prediction and financial document-level event extraction. Challenges were encountered during the implementation of these tasks, which were addressed by utilising out-of-distribution detection. As a result, a new approach for out-of-distribution detection is presented as the third task of this thesis.

Regarding stock movement prediction in Chapter 3, we have proposed a platform that allows for a more realistic evaluation of NLP-aided stock auto-trading algorithms. This platform, characterised by the provision of financial news and various stock factors for each stock, as well as the use of financial-relevant metrics, has allowed us to develop and evaluate the proposed method based on semantic role labelling pooling (SRLP) and a self-supervised learning strategy. The primary contributions of this work can be summarised as follows: 1) We construct a brand new stock prediction task dataset with stock-specific news and stock factors. 2) Our proposed SRLP characterises the key attributes of financial events, which is convenient for incorporating the crucial stock factors and further creating a self-supervised module on top of the SRLP method. Our self-supervised SRLP method obtains competitive stock movement prediction and out-of-distribution (OOD) generalisation results. 3) We further evaluate algorithm performance on real-world trading from more financial-relevant metrics. By conducting extensive experimental studies, we show that our self-supervised SRLP achieves remarkable performance on these metrics. Furthermore, we observe that the proposed trading strategies work well in practice. Our experimental study shows that the proposed method achieves better performance and outperforms all strong baselines in terms of an annualised rate of return and maximum drawdown in back-testing.

We then proposed a generative solution for document-level event extraction in Chapter 4, which is a challenging task in natural language processing due to the need for a thorough comprehension of the document and an aggregated ability to assemble arguments across multiple sentences. In order to address the problem at hand, we propose a novel approach for event extraction that utilises an encoding scheme to incorporate information from entities into the entire document and a decoding scheme that considers relevant contexts. Our proposed solution includes an encoding scheme to capture entity-to-document level information and a decoding scheme that takes into account all relevant contexts. Our encoding scheme converts sentence embeddings and embeddings of entities detected in each sentence into a sequence of context-aware embeddings for the decoder. The decoding scheme then predicts the event type, the number of events per event type, and the event arguments for each event record, using previously predicted output as context for future extraction in a recurrent process. Our results indicate that our generative event extraction scheme is comparable to existing methods in the literature, making it a useful and easy-to-use strong baseline for future research. Extensive experimental results demonstrate that our generative-based solution is able to perform as well as state-of-the-art methods and assist investors in extracting valuable structured information from financial documents.

Finally, we presented a new approach for out-of-distribution detection, which is the third focus of this thesis in Chapter 5. We propose a simple yet effective approach called Semantic Role Labeling Guided Out-of-distribution Detection (SRLOOD), which learns from both global and SRL-guided local fine-grained feature representation to detect OOD instances in NLP. The model jointly optimises both global and local representations using a margin-based contrastive loss and self-supervised loss. Particularly, semantic role labelling (SRL) [98] was originally developed to assign their semantic roles to words or phrases from a sentence, which is significantly useful for eliminating the outlier phrases (e.g., punctuation and discourse fillers) from a sentence and extract the key semantic features (verb and arguments). Inspired by this, our novel self-supervised learning randomly masks those SRL extracted verbs and arguments and utilises a transformer to learn those masked embedding with a 3-way auxiliary classification loss. The transformer block aims to resemble some of the key components from sentences while the classification loss tries to predict its affiliated verbs and arguments, so our model is enforced to learn discriminative key ID representations by this SRL-based self-supervised learning. Finally, we adopt a margin-based contrastive loss [240] to increase the inter-class discrepancies between SRL-extracted features of different ID classes, which enables a more effective and efficient strategy for optimising the ID hyperspace. The resulting model can effectively extract the key semantic roles and eliminate outliers from a sentence to detect subtle OOD samples effectively. The resulting model shows SOTA performance on four different OOD benchmarks with four different OOD scoring functions.

Overall, our research has highlighted the potential of natural language processing techniques in stock prediction and event extraction in the finance domain and has provided new approaches to address the challenges encountered in these tasks.

6.2 Future Work

We have presented some solutions to the issues mentioned earlier, however, these methods are not without limitations and there are still some unresolved issues.

6.2.1 Stock Movement Prediction

Current deep-learning models for stock prediction are primarily trained on static, homogenous datasets that cannot adapt or evolve over time. Continual learning is a technique that enables a model to learn multiple tasks consecutively while retaining information from previous tasks, even when the data from those tasks is no longer available. This allows neural networks to continuously build knowledge in different stock prediction tasks and overcome the problem of catastrophic forgetting. To date, no continual learning models have been specifically designed for stock market prediction. The dynamic nature of the stock market requires models to autonomously acquire new skills, adapt to new situations, and perform new tasks. Therefore, continuing learning might be useful in improving the performance of stock movement prediction in real life.

Besides, we have implemented the task of document-level event extraction in Chapter 4. However, our Astock platform currently lacks the integration of document-level event extraction. Incorporating the information extracted from financial-related documents can be a crucial aspect of future developments. Specifically, event extraction can help to identify specific events, such as company announcements, product launches, and financial results, that can have a significant impact on stock prices. It can also help to identify patterns and trends in the data that can provide insight into the market sentiment, which can be a key indicator of stock price movements. Additionally, event extraction can help to extract named entities, such as companies and individuals, that can provide a more nuanced understanding of the factors that are driving stock prices. These kinds of information can be used as input features for Astock platform, which can help to improve its accuracy and performance.

6.2.2 Document-level Event Extraction

Our Document-level Event Extraction Framework is a generative-based model that recurrently generates events. This approach is specifically designed to extract events from unstructured text data at the document level, by generating events in a sequence, which allows it to capture the temporal dependencies between events. However, the recurrent nature of the model also makes it computationally intensive, which can slow down the event extraction process. This limitation is a potential focus for future research and there are several ways to address it.

One approach to improve the efficiency of the model is to optimise the architecture of the model to reduce computational complexity. For example, using more efficient neural network architectures or utilising parallel computing can help speed up the event extraction process. Another approach is to employ techniques such as transfer learning or pre-training, which can reduce the amount of data required for the model to extract events and decrease the training time.

Furthermore, future research could also aim at developing more advanced generative models that can handle the complexity of unstructured text data. For instance, utilising deep reinforcement learning or GANs (Generative Adversarial Networks) could enhance the model's ability to capture underlying patterns and dynamics of the text data, thereby extracting more accurate events.

In summary, our Document-level Event Extraction Framework is a promising approach for extracting events from unstructured text data. However, the recurrent nature of the model can be a limitation. Future research could focus on addressing this limitation by optimising the architecture, using more efficient techniques, and developing more advanced generative models.

6.2.3 Out-of-distribution Detection

We have achieved state-of-the-art performance on AUROC and FAR95 based on the current dataset, and there is limited space for improvement. AUROC is already very high, mostly over 95% and close to 100%. Therefore, this setting for OOD detection can be considered solved. However, one of the major challenges in this field is that a new proper benchmark dataset is needed to evaluate OOD detection methods and compare different approaches. The current dataset may have reached its limitations in terms of providing a diverse range of OOD examples and real-world scenarios, thus a new benchmark dataset is crucial to continue advancing the field. To address this challenge, our future work will focus on redesigning a proper benchmark dataset to implement OOD detection. This dataset should be representative of real-world scenarios and should include a diverse set of OOD examples that can challenge the model's generalisation capabilities. Additionally, it should be designed to be easily accessible and reusable for the research community. This can enable the comparison of different OOD detection methods on a common benchmark and accelerate the progress of the field. In addition, it is possible to adapt some uncertainty-based methods for OOD, such as the spectral-normalised Neural Gaussian Process proposed by Liu et al. [134].

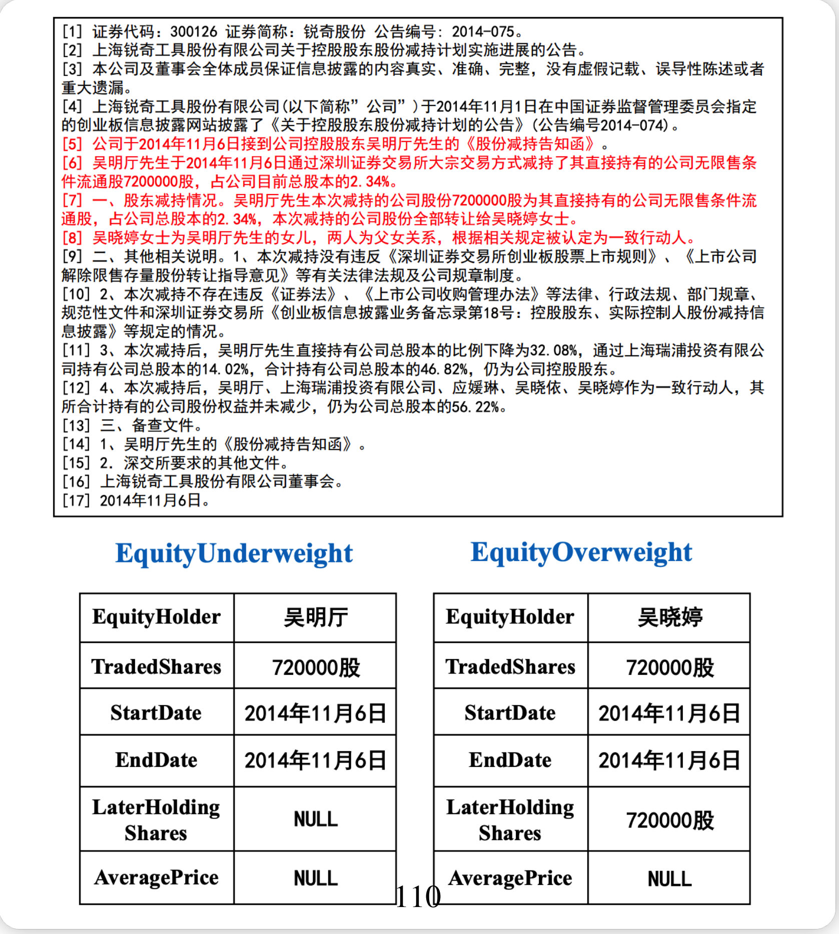
Another important direction for future research is to investigate how to use causal inference to improve the performance of OOD detection. Causal inference is a powerful tool that can be used to understand the underlying causes of OOD examples and to identify the factors that contribute to the model's failure. By leveraging this information, OOD detection methods can be improved to better handle OOD examples and improve the robustness of the model.

In summary, the current biggest problem in OOD detection in NLP is the lack of a proper benchmark dataset. Our future work will focus on redesigning a proper benchmark dataset that can be used to evaluate OOD detection methods and to compare different approaches. Additionally, we will investigate how to use causal inference to improve the performance of OOD detection and to better understand the underlying causes of OOD examples.

Appendix A

Intro (Chapter 1) Appendix

A.1 The original complete document.



[1] 证券代码：300126 证券简称：锐奇股份 公告编号：2014-075。
[2] 上海锐奇工具股份有限公司关于控股股东股份减持计划实施进展的公告。
[3] 本公司及董事会全体成员保证信息披露的内容真实、准确、完整，没有虚假记载、误导性陈述或者重大遗漏。
[4] 上海锐奇工具股份有限公司(以下简称”公司”)于2014年11月1日在中国证券监督管理委员会指定的创业板信息披露网站披露了《关于控股股东股份减持计划的公告》(公告编号2014-074)。
[5] 公司于2014年11月6日接到公司控股股东吴明厅先生的《股份减持告知函》。
[6] 吴明厅先生于2014年11月6日通过深圳证券交易所大宗交易方式减持了其直接持有的公司无限售条件流通股7200000股，占公司目前总股本的2. 34%。
[7] 一、股东减持情况。吴明厅先生本次减持的公司股份7200000股为其直接持有的公司无限售条件流通股，占公司总股本的2. 34%，本次减持的公司股份全部转让给吴晓婷女士。
[8] 吴晓婷女士为吴明厅先生的女儿，两人为父女关系，根据相关规定被认定为一致行动人。
[9] 二、其他相关说明。1、本次减持没有违反《深圳证券交易所创业板股票上市规则》、《上市公司解除限售存量股份转让指导意见》等有关法律法规及公司规章制度。
[10] 2、本次减持不存在违反《证券法》、《上市公司收购管理办法》等法律、行政法规、部门规章、规范性文件和深圳证券交易所《创业板信息披露业务备忘录第18号：控股股东、实际控制人股份减持信息披露》等规定的情况。
[11] 3、本次减持后，吴明厅先生直接持有公司总股本的比例下降为32. 08%，通过上海瑞浦投资有限公司持有公司总股本的14. 02%，合计持有公司总股本的46. 82%，仍为公司控股股东。
[12] 4、本次减持后，吴明厅、上海瑞浦投资有限公司、应媛琳、吴晓依、吴晓婷作为一致行动人，其所合计持有的公司股份权益并未减少，仍为公司总股本的56. 22%。
[13] 三、备查文件。
[14] 1、吴明厅先生的《股份减持告知函》。
[15] 2. 深交所要求的其他文件。
[16] 上海锐奇工具股份有限公司董事会。
[17] 2014年11月6日。

EquityUnderweight

EquityHolder	吴明厅
TradedShares	720000股
StartDate	2014年11月6日
EndDate	2014年11月6日
LaterHolding Shares	NULL
AveragePrice	NULL

EquityOverweight

EquityHolder	吴晓婷
TradedShares	720000股
StartDate	2014年11月6日
EndDate	2014年11月6日
LaterHolding Shares	720000股
AveragePrice	NULL

Figure A.1: The original complete document in Figure 1.1.

Appendix B

Astock (Chapter 3) Appendix

B.1 Trading Strategy

We design a dynamic trading strategy based on news responses to back-test our models. Generally, we define three types of actions: short, long, and preserve. The stocks from China A-shares can be shortened by short selling, and we assume that it applies to all stocks. When a piece of news is published, an action is automatically triggered according to the result predicted by the models. Different types of the transaction have different methods to calculate the return rate. For a ”long” transaction, we define its return as $\frac{P_{sell}-P_{buy}}{P_{buy}}\%$. For a ”short” transaction, we define its return as $\frac{P_{sell}-P_{buy}}{P_{sell}}\%$.(P stands for the price.)

Our strategy considers three types of actions: short, preserve, and long. When news is published, the corresponding stock price will surge or plummet abnormally. This strategy is based on the assumption that the stock price will return to the normal interval after surging or plummeting. The trading actions will be triggered according to the result predicted by our model. Specifically, if a news piece's predicted result is a downtrend, the long action will be triggered, and if the result is an uptrend, the short action will be triggered. Otherwise, the preserve action will be triggered. The position will be closed the day after the next trading day. To make the strategy flexible to deal with the different situations, a dynamic weight is introduced to control the position [1] of each transaction under different actions, especially for the actions of long and short. The weight is determined by the sum of the latest three days' return rates from the same type of action. The position for action a_i is P_i defined as Equation B.1 and the process is shown in the Figure B.1:

$$P_i = \frac{p_i C^{\sum_{t=0}^{3} r_{ij}}}{N_i \sum_{i \in A} C^{\sum_{j=0}^{3} r_{ij}}} \tag{B.1}$$

Where r_{ij} is the average return rate of the j-th latest day for a type of action a_i, $a_i \in \{a_1, a_2, a_3\}$. N_i is the number of transactions for an action in a trading day. C is the hyper-parameter to adjust the flexibility, p is the prediction probability of a piece of news. Figure B.1 describes the progress of the trading transaction from an action triggered by the news to the transaction finalized. Since taking preserve action cannot make a profit, we introduce a hypothetical return rate to calculate the weight of preserve. Specifically, if the t-th latest day return rates of long action and short action are both negative, we assume that the return rate of the preserve action is the opposite average of short return and long return. Otherwise, the return rate of preserve action is set to 0.

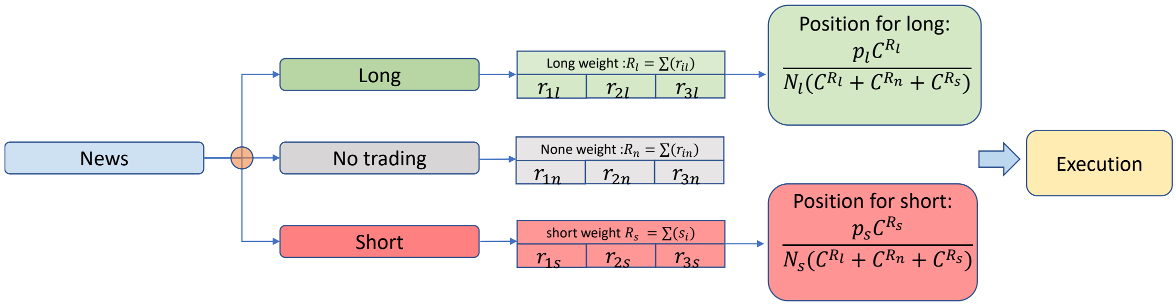

Position for long:
$\frac{p_l C^{R_l}}{N_l(C^{R_l} + C^{R_n} + C^{R_s})}$
Long
Long weight :$R_l = \sum(r_{il})$
r_{1l}
r_{2l}
r_{3l}
News
No trading
None weight :$R_n = \sum(r_{in})$
r_{1n}
r_{2n}
r_{3n}
Execution
Position for short:
$\frac{p_s C^{R_s}}{N_s(C^{R_l} + C^{R_n} + C^{R_s})}$
Short
short weight $R_s = \sum(s_i)$
r_{1s}
r_{2s}
r_{3s}

Figure B.1: The weight calculation process of each transaction.

For each transaction, the commission fee is set to 0.13%, and the position is opened by the price when the news is published. The position will be closed if a loss achieves -10%.

[1] The position is financial terminology representing the ratio of the transaction out of the whole asset.

B.2 Stock Factors

The stock factors include: Dividend yield, Total share, Circulated share, Free Float share, Market Capitalization, Price-earning ratio(PE), PE for Trailing Twelve Months(TTM), Price/book value ratio, Price-to-sales Ratio, Price to Sales ratio (TTM), Circulate Market Capitalization, Open price, High price,Low price, Close price , Previous close price, Price change, Percentage of change, Volume, Amount, Turn over rate, Turn over rate for circulated Market Capitalization, Volume ratio.

B.3 The study of using different alpha value

This section shows the result of the test accuracy regarding different hyper-parameter α. When α equals 0.8, our model yields the best prediction accuracy. As shown in the Figure B.2.

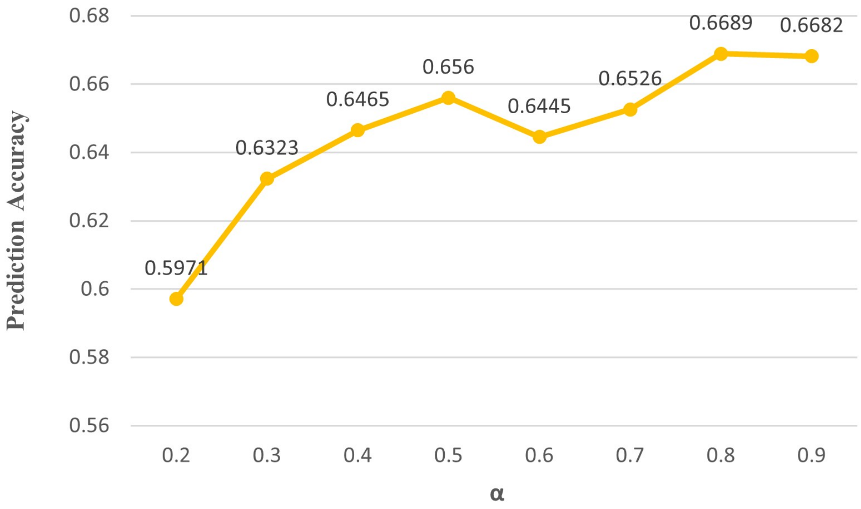


Figure B.2: The effect of the alpha on the accuracy performance.

Appendix C

Document-level Event Extraction (Chapter 4) Appendix

C.1 More Experimental Details

The detailed hyperparameter is shown in Table C.1. The value of the hyperparameters we finally adopted are in bold.

Hyper-parameter	**Value**
Maximum number of records	20
Embedding size	768
Hidden size	768
Tagging scheme	BIO
Layers of Trans-E	4
Layers of Trans-C	4
Layers of Decoding Transformer	2, 4, **6**, 8
Optimizer	AdamW
Learning rate for encoder	$1e^{-4}$
Batch size	8, 12, **16**
$\{\lambda_1\}$	0.1
$\{\lambda_2\}$	0.4
$\{\lambda_3\}$	1
$\{\lambda_4\}$	1
Dropout	0.1
Training epoch	100

Table C.1: The hyper-parameter setting.

Appendix D

SRLOOD (Chapter 5) Appendix

D.1 More Experimental Details

The Transformer encoder has 3 layers and 16 attention heads. The weights $\alpha_1 = 1$, $\alpha_2 = 3$, $\alpha_3 = 1$. The warm-up ratio for learning rate is 0.06. The batch size is 12. We use AdamW [96] to optimized our model, and a learning rate of $1e - 5$ and weighted decay 0.01. We pick the masking probability that optimize the average OOD detection performance, 30% for SST2 and IMDB, and 50% for TREC-10 and 20NG, guided by Figure D.1. The model is trained for 10 epochs with runtime ranging from 5 hours to 10 hours on one Tesla V100 GPU. We further discuss the performance of taking different masking probabilities in D. Please note that we manually select all hyper-parameters based on the AUC and FAR performance on testing sets. The total number of model parameters is 392M. All the hyper-parameters are tuned on the development sets.

D.2 Experiments with different masking probabilities

The figure shows three different masking probabilities: 0.3, 0.5, and 0.7 with a higher value indicating a stronger perturbation to the key feature representations, and we adopt the optimal perturbation for each ID task, judged by both the average AUROC and FAR95 over all four OOD scores as shown in Figure D.1 .

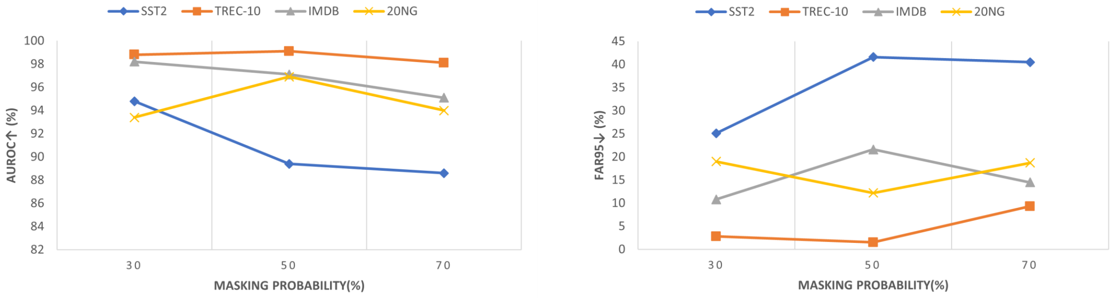


Figure D.1: Performance on different masking probabilities on four benchmark datasets.

D.3 Full Results

Following [240], we also employ for additional datasets solely as the OOD data: RTE [11, 38, 63, 66] , MNLI [206], WMT16 [14] and Multi30K [57]. We show the full OOD detection performance of ID datasets on OOD datasets in Table D.1 and Table D.2 .

AUROC	**SST2**				**IMDB**				**TREC-10**				**20NG**			
	MSP	Energy	Maha	Cosine	MSP	Energy	Maha	Cosine	MSP	Energy	Maha	Cosine	MSP	Energy	Maha	Cosine
SST2	-	-	-	-	-	-	-	-	**97.8**/96.2	**97.3**/96.6	**99.8**/98.4	**99.1**/97.8	**96.5**/96.3	**99.0**/98.1	99.2/99.5	**99.0**/99.0
IMDB	-	-	-	-	-	-	-	-	**99.5**/99.3	92.0/99.7	**99.7**/99.6	**99.6**/99.3	93.6/94.5	96.7/96.9	**99.7**/99.0	**98.8**/98.4
TREC-10	**96.0**/95.1	**96.1**/94.9	**99.8**/99.5	**99.6**/99.0	**98.8**/93.8	**98.9**/93.3	99.9/100	99.8/100	-	-	-	-	**99.3**/88.0	**99.9**/92.4	99.3/99.6	**99.5**/96.5
20NG	**96.8**/95.2	**97.0**/95.0	**100**/100	99.9/100	**96.5**/95.4	**96.6**/95.3	99.8/100	98.0/99.9	**99.6**/99.2	99.7/99.8	**100**/99.8	**100**/99.7	-	-	-	-
MNLI	**83.0**/82.8	**82.8**/82.7	98.4/99.8	96.6/99.5	**95.7**/92.4	**95.6**/91.7	99.8/100	96.9/99.9	**98.0**/96.6	**98.0**/97.6	**99.8**/99.2	**99.0**/98.8	**92.1**/91.0	**96.8**/94.2	**98.9**/98.4	**99.1**/97.2
RTE	**89.4**/87.4	**88.2**/87.5	99.9/100	99.5/99.9	**96.1**/92.9	**96.0**/92.1	99.9/100	98.4/99.9	**98.7**/96.6	**98.6**/98.1	**99.9**/99.6	**99.6**/99.2	**85.5**/84.5	**92.1**/88.7	**98.7**/98.2	**98.5**/95.6
WMT16	**85.5**/83.9	**84.3**/84.0	98.9/99.9	97.3/99.4	**96.9**/92.9	**96.8**/92.2	99.9/100	99.8/99.9	**97.8**/97.1	97.8/98.0	**99.9**/99.4	**99.5**/99.1	**91.9**/88.3	**96.8**/92.5	**99.0**/98.5	**98.8**/96.7
Multi30K	**94.2**/93.5	**93.7**/93.6	99.5/100	99.3/99.9	**98.1**/95.9	**98.3**/95.7	99.9/100	99.8/100	**99.1**/97.9	**99.1**/98.9	100/99.5	**99.9**/99.3	91.6/93.7	**96.9**/96.0	99.0/99.1	**98.6**/98.3
Avg	**90.8**/89.7	**90.4**/89.6	99.4/99.9	98.7/99.6	**97.0**/93.9	**97.0**/93.4	99.9/100	98.7/99.9	**98.6**/97.6	**98.5**/98.4	**99.9**/99.3	**99.5**/99.0	**92.9**/90.9	**96.9**/94.1	**99.1**/98.9	**98.9**/97.4

Table D.1: OOD detection AUROC (%) of ours and w/ $\mathcal{L}_{margin}$ [240].

FAR95	**SST2**				**IMDB**				**TREC-10**				**20NG**			
	MSP	Energy	Maha	Cosine	MSP	Energy	Maha	Cosine	MSP	Energy	Maha	Cosine	MSP	Energy	Maha	Cosine
SST2	-	-	-	-	-	-	-	-	**5.3**/11.9	**6.8**/10.4	**0**/1.6	**0.4**/6.9	20.5/13.7	**4.3**/5.3	**0**/1.2	**0.1**/12.6
IMDB	-	-	-	-	-	-	-	-	**0**/0.5	**0**/0.2	**0**/0	**0**/0	31.6/23.6	14.5/11.4	**0.5**/4.7	**3.5**/7.4
TREC-10	**23.2**/35.3	**23.0**/35.0	**0**/2.4	**0**/4.3	**0.6**/50.0	**0.8**/54.0	**0**/0	**0**/0	-	-	-	-	**3.5**/37.2	**0**/13.8	**0**/1.4	**0**/4.4
20NG	**15.7**/36.4	**13.7**/36.3	**0**/0	**0**/0	**22.7**/37.8	**24.6**/33.1	**0**/0	6.3/0	**0**/0.6	**0**/0.2	**0**/0	**0**/0	-	-	-	-
MNLI	68.7/64.6	67.8/64.3	7.8/0.4	22.0/2.6	**32.6**/52.2	**34.8**/83.8	**0**/0.1	14.9/0.9	**3.9**/9.6	**4.5**/6.7	**0**/0.7	**1.5**/1.9	38.1/37.4	**16.9**/24.7	**0.1**/9.6	**0.3**/16.7
RTE	58.5/58.3	**57.5**/57.7	**0**/0	0.9/0.3	**29.3**/52.9	**32.6**/54.3	**0**/0	5.9/0.3	**2.8**/9.8	**3.8**/6.2	**0**/0.1	**0.2**/0.5	**51.3**/52.9	**30.6**/35.4	**2.6**/11.1	**4.5**/24.2
WMT16	68.3/64.3	67.0/64.1	4.8/0.5	15.3/3.0	**18.9**/53.7	**21.2**/55.7	**0**/0	2/0.4	**5.3**/7.9	7.2/5.7	**0**/0.5	**0.6**/1.3	**37.3**/45.3	**15.8**/27.8	**2.0**/7.5	**4.3**/18.7
Multi30K	44.0/36.3	43.8/35.4	0.2/0	1/0.3	**5.6**/30.9	**5.7**/31.9	**0**/0	**0**/0	**0.3**/5.3	**0.3**/2.6	**0**/0	**0**/0.2	37.9/27.8	16.2/12.0	**0.7**/6.9	**3.9**/8.7
Avg	**46.4**/49.2	**45.5**/48.8	2.2/0.6	6.5/1.7	46.3/18.3	**19.9**/52.1	**0**/0	4.8/0.2	**2.5**/6.5	**3.2**/4.6	**0**/0.4	**0.4**/1.5	**31.4**/32.6	**14.0**/18.6	**0.8**/6.0	**2.3**/11.8

Table D.2: OOD detection FAR95 (%) of ours and w/ $\mathcal{L}_{margin}$ [240].

Bibliography

[1] Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992, 1992. URL https://aclanthology.org/M92-1000. 31

[2] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara. Latent space autoregression for novelty detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 481–490, 2019. 91

[3] D. Ahn. The stages of event extraction. In Proceedings of the Workshop on Annotating and Reasoning about Time and Events, pages 1–8, Sydney, Australia, July 2006. Association for Computational Linguistics. URL https://aclanthology.org/W06-0901. 13

[4] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara. Deep learning for stock prediction using numerical and textual information. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pages 1–6. IEEE, 2016. 16, 45

[5] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research, 6:1817–1853, Dec. 2005. ISSN 1532-4435. 83

[6] G. Ang and E.-P. Lim. Guided attention multimodal multitask financial forecasting

with inter-company relationships and global and local news. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6313–6326, 2022. 22

[7] J. Araki and T. Mitamura. Joint event trigger identification and event coreference resolution with structured perceptron. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2074–2080, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1247. URL https://aclanthology.org/D15-1247. 14

[8] U. Arora, W. Huang, and H. He. Types of out-of-distribution texts and how to detect them. arXiv preprint arXiv:2109.06827, 2021. 89

[9] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:2202.03555, 2022. 92

[10] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The sixth pascal recognizing textual entailment challenge. In Text Analysis Conference, 2009. 98

[11] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The fifth pascal recognizing textual entailment challenge. In TAC, 2009. 118

[12] B. Bhatt and S. Das. Stock prediction using twitter sentiment analysis. In 2016 International Conference on Computing, Communication and Automation (ICCCA), pages 1108–1113. IEEE, 2016. 1

[13] J. Bitterwolf, A. Meinke, and M. Hein. Certifiably adversarially robust detection of out-of-distribution data. Advances in Neural Information Processing Systems, 33: 16085–16095, 2020. 32

[14] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, et al. Findings of the 2016 conference on machine translation. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pages 131–198, 2016. 98, 118

[15] O. Bustos and A. Pomares-Quimbaya. Stock market movement forecast: A systematic review. Expert Systems with Applications, 156:113464, 2020. 35

[16] S. Cao and Z. Zhang. Deep hybrid models for out-of-distribution detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4733–4743, 2022. 89, 91

[17] P. Chakraborty, U. S. Pria, M. R. A. H. Rony, and M. A. Majumdar. Predicting stock movement using sentiment analysis of twitter feed. In 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), pages 1–6. IEEE, 2017. 13

[18] W. Che, Y. Feng, L. Qin, and T. Liu. N-ltp: A open-source neural chinese language technology platform with pretrained models. arXiv preprint arXiv:2009.11616, 2020. 50

[19] D. Chen, Y. Zou, K. Harimoto, R. Bao, X. Ren, and X. Sun. Incorporating fine-grained events in stock movement prediction. arXiv preprint arXiv:1910.05078, 2019. 18

[20] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha. Atom: Robustifying out-of-distribution detection using outlier mining. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 430–445. Springer, 2021. 32

[21] J. Chen, T. Chen, M. Shen, Y. Shi, D. Wang, and X. Zhang. Gated three-tower transformer for text-driven stock market prediction. Multimedia Tools and Applications, 2022. doi: 10.1007/s11042-022-11908-1. 26

[22] L. Chen, Z. Jiang, J. Xiao, and W. Liu. Human-like controllable image captioning with verb-specific semantic roles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16846–16856, 2021. 91

[23] X. Chen, X. Lan, F. Sun, and N. Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. In European Conference on Computer Vision, pages 572–588. Springer, 2020. 91

[24] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 167–176, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1017. URL https://aclanthology.org/P15-1017. 29

[25] Y. Chen, S. Liu, X. Zhang, K. Liu, and J. Zhao. Automatically labeled data generation for large scale event extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 409–419, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1038. URL https://aclanthology.org/P17-1038. 29

[26] 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, pages 1655–1658, 2018. 23

[27] Y. Chen, Y. Tian, G. Pang, and G. Carneiro. Deep one-class classification via interpolated gaussian descriptor. arXiv preprint arXiv:2101.10043, 2021. 91

[28] R. Cheng and Q. Li. Modeling the momentum spillover effect for stock prediction via

attribute-driven graph attention networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 55–62, 2021. 25, 42, 45

[29] H. L. Chieu and H. T. Ng. A maximum entropy approach to information extraction from semi-structured and free text. Aaai/iaai, 2002:786–791, 2002. 13

[30] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014. 16, 19

[31] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014. 37

[32] C. Chow. On optimum recognition error and reject tradeoff. IEEE Transactions on information theory, 16(1):41–46, 1970. 14

[33] F. Colasanto, L. Grilli, D. Santoro, and G. Villani. Albertino for stock price prediction: a gibbs sampling approach. Information Sciences, 597: 341–357, 2022. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2022.03.051. URL https://www.sciencedirect.com/science/article/pii/S002002552200264X. 27

[34] S. Conia, R. Orlando, F. Brignone, F. Cecconi, and R. Navigli. Invero-xl: Making cross-lingual semantic role labeling accessible with intelligible verbs and roles. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), Punta Cana, Dominican Republic, 2021. 46

[35] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995. 34

[36] S. Cui, B. Yu, T. Liu, Z. Zhang, X. Wang, and J. Shi. Edge-enhanced graph convolution networks for event detection with syntactic relation. arXiv preprint arXiv:2002.10757, 2020. 29

[37] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu. Revisiting pre-trained models for Chinese natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 657–668, Online, Nov. 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.findings-emnlp.58. 55, 57, 59, 60

[38] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In Machine learning challenges workshop, pages 177–190. Springer, 2005. 98, 118

[39] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, and H. Chen. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In Companion Proceedings of The 2019 World Wide Web Conference, pages 678–685, 2019. 20

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 46, 55, 57, 73, 92

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423. 27

[42] L. Devroye, L. Györfi, and G. Lugosi. A probabilistic theory of pattern recognition, volume 31. Springer Science & Business Media, 2013. 14

[43] A. R. Dhamija, M. Günther, and T. Boult. Reducing network agnostophobia. Advances in Neural Information Processing Systems, 31, 2018. 91

[44] G. Ding and L. Qin. Study on the prediction of stock price based on the associated network model of lstm. International Journal of Machine Learning and Cybernetics, 11(6):1307–1317, 2020. 18

[45] 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), pages 1415–1425, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1148. URL https://aclanthology.org/D14-1148. 47

[46] 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), pages 1415–1425, 2014. 20, 22

[47] X. Ding, Y. Zhang, T. Liu, and J. Duan. Deep learning for event-driven stock prediction. In Twenty-fourth international joint conference on artificial intelligence, 2015. 20, 22, 42, 43, 45, 46

[48] X. Ding, Y. Zhang, T. Liu, and J. Duan. Knowledge-driven event embedding for stock prediction. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2133–2142, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1201. 45

[49] X. Ding, Y. Zhang, T. Liu, and J. Duan. Knowledge-driven event embedding for stock prediction. In Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers, pages 2133–2142, 2016. 22

[50] X. Dong, J. Guo, A. Li, W.-T. Ting, C. Liu, and H. Kung. Neural mean discrepancy for efficient out-of-distribution detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 19217–19227, 2022. 91

[51] Y. Dong, D. Yan, A. I. Almudaifer, S. Yan, Z. Jiang, and Y. Zhou. Belt: A pipeline for stock price prediction using news. In 2020 IEEE International Conference on Big Data (Big Data), pages 1137–1146, 2020. doi: 10.1109/BigData50022.2020.9378345. 27

[52] X. Du and C. Cardie. Document-level event role filler extraction using multi-granularity contextualized encoding. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8010–8020, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.714. URL https://aclanthology.org/2020.acl-main.714. 30, 31, 83

[53] X. Du and C. Cardie. Event extraction by answering (almost) natural questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 671–683, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.49. URL https://aclanthology.org/2020.emnlp-main.49. 30, 83

[54] X. Du, A. Rush, and C. Cardie. GRIT: Generative role-filler transformers for document-level event entity extraction. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 634–644, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.52. URL https://aclanthology.org/2021.eacl-main.52. 30, 31, 83

[55] S. Ebner, P. Xia, R. Culkin, K. Rawlins, and B. Van Durme. Multi-sentence argument linking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8057–8077, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.718. URL https://aclanthology.org/2020.acl-main.718. 31, 83

[56] S. Ebrahimi, H. Naderifar, and S. Khadivi. Predicting stock prices using sentiment analysis of twitter data. In 2016 23rd Iranian Conference on Biomedical Engineering and Computing (ICBMEC), pages 77–82. IEEE, 2016. 1

[57] D. Elliott, S. Frank, K. Sima'an, and L. Specia. Multi30K: Multilingual English-German image descriptions. In Proceedings of the 5th Workshop on Vision and Language, pages 70–74, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-3210. URL https://aclanthology.org/W16-3210. 98, 118

[58] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. Communications of the ACM, 51(12):68–74, 2008. 21

[59] E. Fama and J. D. MacBeth. Risk, return, and equilibrium: Empirical tests. Journal of Political Economy, 81(3):607–36, 1973. URL https://EconPapers.repec.org/RePEc:ucp:jpolec:v:81:y:1973:i:3:p:607-36. 28

[60] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T.-S. Chua. Enhancing stock movement prediction with adversarial training. arXiv preprint arXiv:1810.09936, 2018. 18

[61] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua. Temporal relational ranking for stock prediction. ACM Transactions on Information Systems (TOIS), 37(2):1–30, 2019. 23, 25

[62] A. Ghoting, S. Parthasarathy, and M. E. Otey. Fast mining of distance-based outliers in

high-dimensional datasets. Data Mining and Knowledge Discovery, 16(3):349–364, 2008. 14

[63] D. Giampiccolo, B. Magnini, I. Dagan, and W. B. Dolan. The third pascal recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing, pages 1–9, 2007. 98, 118

[64] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., volume 4, pages 2047–2052. IEEE, 2005. 16

[65] M. Hagenau, M. Liebmann, and D. Neumann. Automated news reading: Stock price prediction based on financial news using context-capturing features. Decision Support Systems, 55(3):685–697, 2013. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2013.02.006. URL https://www.sciencedirect.com/science/article/pii/S0167923613000651. 45, 46

[66] R. B. Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The second pascal recognising textual entailment challenge. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, volume 7, 2006. 98, 118

[67] Y. Hei, R. Yang, H. Peng, L. Wang, X. Xu, J. Liu, H. Liu, J. Xu, and L. Sun. Hawk: Rapid android malware detection through heterogeneous graph attention networks. IEEE Transactions on Neural Networks and Learning Systems, 2021. 29

[68] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 41–50, 2019. 32

[69] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. Proceedings of International Conference on Learning Representations, 2017. 91, 92, 99, 100

[70] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. Proceedings of International Conference on Learning Representations, 2017. 32

[71] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606, 2018. 91, 92

[72] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. Proceedings of the International Conference on Learning Representations, 2019. 32

[73] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. Advances in Neural Information Processing Systems (NeurIPS), 2019. 52

[74] D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song. Pretrained transformers improve out-of-distribution robustness. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2744–2751, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.244. URL https://aclanthology.org/2020.acl-main.244. 89, 92

[75] D. Hendrycks et al. Using self-supervised learning can improve model robustness and uncertainty. arXiv preprint arXiv:1906.12340, 2019. 92

[76] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9 (8):1735–1780, 1997. 16, 36

[77] J. H. Holland. A simple genetic algorithm. Complex systems, 9(2):121–129, 1975. 34

[78] E. Hoseinzade, S. Haratizadeh, and A. Khoeini. U-cnnpred: A universal cnn-based predictor for stock markets. arXiv preprint arXiv:1911.12540, 2019. 20

[79] I.-H. Hsu, K.-H. Huang, E. Boschee, S. Miller, P. Natarajan, K.-W. Chang, and N. Peng. Degree: A data-efficient generative event extraction model. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2022. 70, 83

[80] T.-W. Hsu, C.-C. Chen, H.-H. Huang, and H.-H. Chen. Semantics-preserved data augmentation for aspect-based sentiment analysis. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4417–4422, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.362. URL https://aclanthology.org/2021.emnlp-main.362. 28

[81] H. Hu, L. Tang, S. Zhang, and H. Wang. Predicting the direction of stock markets using optimized neural networks with google trends. Neurocomputing, 285:188–195, 2018. 12

[82] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In Proceedings of the eleventh ACM international conference on web search and data mining, pages 261–269, 2018. 19, 45, 46, 47, 55, 56, 59, 60, 62

[83] K.-H. Huang and N. Peng. Document-level event extraction with efficient end-to-end learning of cross-event dependencies, 2020. URL https://arxiv.org/abs/2010.12787. 83

[84] K.-H. Huang and N. Peng. Document-level event extraction with efficient end-to-end learning of cross-event dependencies. arXiv preprint arXiv:2010.12787, 2020. 31

[85] K.-H. Huang, S. Tang, and N. Peng. Document-level entity-based extraction as template generation. In The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 2021. Association for Computational Linguistics. 70, 83

[86] P. Huang, X. Zhao, R. Takanobu, Z. Tan, and W. Xiao. Joint event extraction with hierarchical policy network. In Proceedings of the 28th International Conference on Computational Linguistics, pages 2653–2664, 2020. 29

[87] Q. Huang, Z. Kong, Y. Li, J. Yang, and X. Li. Discovery of trading points based on bayesian modeling of trading rules. World Wide Web, 21(6):1473–1490, 2018. 12

[88] R. Huang and Y. Li. Mos: Towards scaling out-of-distribution detection for large semantic space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8710–8719, June 2021. 91

[89] R. Huang, A. Geng, and Y. Li. On the importance of gradients for detecting distributional shifts in the wild. Advances in Neural Information Processing Systems, 34: 677–689, 2021. 32, 33

[90] Y. Huang and W. Jia. Exploring sentence community for document-level event extraction. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 340–351, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.32. URL https://aclanthology.org/2021.findings-emnlp.32. 31, 32

[91] J. Im, M. Kim, H. Lee, H. Cho, and S. Chung. Self-supervised multimodal opinion summarization. In Proceedings of the 59th Annual Meeting of the Association for

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 388–403, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.33. URL https://aclanthology.org/2021.acl-long.33. 46

[92] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. Data mining and knowledge discovery, 33 (4):917–963, 2019. 20

[93] V. N. Iyengar, U. M. Dholakia, and R. Singh. Predicting stock price movements with twitter sentiment analysis. In 2014 IEEE International Conference on Big Data (Big Data), pages 72–79. IEEE, 2014. 1

[94] A. Judea and M. Strube. Incremental global event extraction. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2279–2289, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1215. 14

[95] R. Kim, C. H. So, M. Jeong, S. Lee, J. Kim, and J. Kang. Hats: A hierarchical graph attention network for stock movement prediction. arXiv preprint arXiv:1908.07999, 2019. 24

[96] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. 117

[97] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL http://arxiv.org/abs/1412.6980. 78

[98] P. Kingsbury and M. Palmer. Propbank: the next level of treebank. In Proceedings of Treebanks and lexical Theories, volume 3. Citeseer, 2003. 50, 94, 105

[99] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016. 38

[100] P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 20578–20589. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/ecb9fe2fbb99c31f567e9823e884dbec-Paper.pdf. 32

[101] P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. Advances in neural information processing systems, 33:20578–20589, 2020. 91

[102] I. Kobyzev, S. J. Prince, and M. A. Brubaker. Normalizing flows: An introduction and review of current methods. IEEE transactions on pattern analysis and machine intelligence, 43(11):3964–3979, 2020. 32

[103] K. Lang. Newsweeder: Learning to filter netnews. In Machine Learning Proceedings 1995, pages 331–339. Elsevier, 1995. 97

[104] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989. 37

[105] K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=ryiAv2xAZ. 91, 99

[106] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in neural information processing systems, 31, 2018. 32

[107] C. Lei. Rnn. In Deep Learning and Practice with MindSpore, pages 83–93. Springer, 2021. 16

[108] J. Leitao, R. F. Neves, and N. Horta. Combining rules between pips and sax to identify patterns in financial markets. Expert Systems with Applications, 65:242–254, 2016. 13

[109] F. Li, W. Peng, Y. Chen, Q. Wang, L. Pan, Y. Lyu, and Y. Zhu. Event extraction as multi-turn question answering. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 829–838, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.73. URL https://aclanthology.org/2020.findings-emnlp.73. 83

[110] H. Li, Y. Shen, and Y. Zhu. Stock price prediction using attention-based multi-input lstm. In Asian conference on machine learning, pages 454–469. PMLR, 2018. 19

[111] M. Li, W. Li, F. Wang, X. Jia, and G. Rui. Applying bert to analyze investor sentiment in stock market. Neural Computing and Applications, 33(10):4663–4676, 2020. doi: 10.1007/s00521-020-05411-7. 28

[112] P. Li, Q. Zhu, H. Diao, and G. Zhou. Joint modeling of trigger identification and event type determination in Chinese event extraction. In Proceedings of COLING 2012, pages 1635–1652, Mumbai, India, Dec. 2012. The COLING 2012 Organizing Committee. URL https://aclanthology.org/C12-1100. 14

[113] Q. Li, H. Ji, and L. Huang. Joint event extraction via structured prediction with global features. In Proceedings of the 51st Annual Meeting of the Association

for Computational Linguistics (Volume 1: Long Papers), pages 73–82, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1008. 14, 83

[114] Q. Li, H. Ji, Y. Hong, and S. Li. Constructing information networks using one single model. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1846–1851, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1198. URL https://aclanthology.org/D14-1198. 14

[115] Q. Li, J. Wang, F. Wang, P. Li, L. Liu, and Y. Chen. The role of social sentiment in stock markets: a view from joint effects of multiple information sources. Multimedia Tools and Applications, 76(10):12315–12345, 2017. 26

[116] 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, 33(10):3323–3337, 2021. doi: 10.1109/TKDE.2020.2968894. 25

[117] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du. Analogical reasoning on chinese morphological and semantic relations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 138–143. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-2023. 57

[118] S. Li, H. Ji, and J. Han. Document-level event argument extraction by conditional generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 894–908, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.69. URL https://aclanthology.org/2021.naacl-main.69. 31, 83

[119] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su. Modeling the stock relation with graph network for overnight stock movement prediction. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pages 4541–4547, 2021. 23

[120] X. Li and D. Roth. Learning question classifiers. In COLING 2002: The 19th International Conference on Computational Linguistics, 2002. 97

[121] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng. News impact on stock price return via sentiment analysis. Knowledge-Based Systems, 69: 14–23, 2014. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2014.04.022. URL https://www.sciencedirect.com/science/article/pii/S0950705114001440. 27

[122] X. Li, J. Wang, J. Tan, S. Ji, and H. Jia. A graph neural network-based stock forecasting method utilizing multi-source heterogeneous data fusion. Multimedia Tools and Applications, pages 1–23, 2022. 22

[123] Y. Li and Y. Pan. A novel ensemble deep learning model for stock prediction based on stock prices and news. International Journal of Data Science and Analytics, 13(2): 139–149, 2022. 18

[124] Y. Li, J. Wu, W. Wang, and X. Zhu. Predicting stock prices using sentiment analysis of financial news articles. In 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 1, pages 28–35. IEEE, 2014. 1

[125] Y. Li, H. Bu, J. Li, and J. Wu. The role of text-extracted investor sentiment in chinese stock price prediction with the enhancement of deep learning. International Journal of Forecasting, 36(4):1541–1562, 2020. 43

[126] Y. Li, S. Lv, X. Liu, and Q. Zhang. Incorporating transformers and attention networks for stock movement prediction. Complexity, 2022, 2022. 25

[127] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690, 2017. 32

[128] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=H1VGkIxRZ. 32

[129] Y. Lin, H. Ji, F. Huang, and L. Wu. A joint neural model for information extraction with global features. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7999–8009, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.713. URL https://aclanthology.org/2020.acl-main.713. 83

[130] Z. Lin, S. D. Roy, and Y. Li. Mood: Multi-level out-of-distribution detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15313–15323, June 2021. 91

[131] D. Liu and D. Gildea. Semantic role features for machine translation. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 716–724, 2010. 46

[132] J. Liu, H. Lin, X. Liu, B. Xu, Y. Ren, Y. Diao, and L. Yang. Transformer-based capsule network for stock movement prediction. In Proceedings of the First Workshop on Financial Technology and Natural Language Processing, pages 66–73, Macao, China, Aug. 2019. URL https://aclanthology.org/W19-5511. 26

[133] J. Liu, Y. Chen, K. Liu, W. Bi, and X. Liu. Event extraction as machine reading comprehension. In Proceedings of the 2020 Conference on Empirical Methods in

Natural Language Processing (EMNLP), pages 1641–1651, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.128. URL https://aclanthology.org/2020.emnlp-main.128. 29, 83

[134] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 7498–7512. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/543e83748234f7cbab21aa0ade66565f-Paper.pdf. 108

[135] S. Liu, Y. Chen, K. Liu, and J. Zhao. Exploiting argument information to improve event detection via supervised attention mechanisms. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1789–1798, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1164. URL https://aclanthology.org/P17-1164. 29

[136] W. Liu, X. Wang, J. D. Owens, and Y. Li. Energy-based out-of-distribution detection. arXiv preprint arXiv:2010.03759, 2020. 32

[137] X. Liu, Z. Luo, and H. Huang. Jointly multiple events extraction via attention-based graph information aggregation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1247–1256, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1156. URL https://aclanthology.org/D18-1156. 29

[138] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer,

and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019. 57

[139] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS. 99

[140] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang. A cnn-lstm-based model to forecast stock prices. Complexity, 2020, 2020. 21

[141] W. Lu, J. Li, J. Wang, and L. Qin. A cnn-bilstm-am method for stock price prediction. Neural Computing and Applications, 33(10):4741–4753, 2021. 21

[142] Y. Lu, H. Lin, J. Xu, X. Han, J. Tang, A. Li, L. Sun, M. Liao, and S. Chen. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2795–2806, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.217. URL https://aclanthology.org/2021.acl-long.217. 30, 70, 83

[143] J. Ma, S. Wang, R. Anubhai, M. Ballesteros, and Y. Al-Onaizan. Resource-enhanced neural model for event argument extraction. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3554–3559, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.318. URL https://aclanthology.org/2020.findings-emnlp.318. 30

[144] Y. Ma, L. Zong, Y. Yang, and J. Su. News2vec: News network embedding with subnode information. 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), pages 4843–4852, 2019. 17

[145] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015. 97

[146] U. Maqsud, S. Arnold, M. Hülfenhaus, and A. Akbik. Nerdle: Topic-specific question answering using wikia seeds. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations, pages 81–85, 2014. 46

[147] L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson. Semantic role labeling: an introduction to the special issue, 2008. 46

[148] D. Matsunaga, T. Suzumura, and T. Takahashi. Exploring graph neural networks for stock market predictions with rolling window analysis. arXiv preprint arXiv:1909.10660, 2019. 21

[149] S. Mehtab and J. Sen. Stock price prediction using cnn and lstm-based deep learning models. In 2020 International Conference on Decision Aid Sciences and Application (DASA), pages 447–453. IEEE, 2020. 21

[150] S. Mittal and A. Goel. Stock price prediction using sentiment analysis. International Journal of Computer Applications, 55(10):14–19, 2012. 1

[151] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang. Self-supervised learning for generalizable out-of-distribution detection. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 5216–5223, 2020. 52

[152] F. Morstatter, J. Pfeffer, and H. Liu. Financial forecasting with sentiment analysis: A literature review. arXiv preprint arXiv:1607.01450, 2016. 1

[153] P. Morteza and Y. Li. Provable guarantees for understanding out-of-distribution detection. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 8, 2022. 91

[154] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown. An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6):275–285, 2004. 35

[155] L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson. Semantic Role Labeling: An Introduction to the Special Issue. Computational Linguistics, 34(2):145–159, 06 2008. ISSN 0891-2017. doi: 10.1162/coli.2008.34.2.145. URL https://doi.org/10.1162/coli.2008.34.2.145. 94

[156] D. M. Nelson, A. C. Pereira, and R. A. De Oliveira. Stock market's price movement prediction with lstm neural networks. In 2017 International joint conference on neural networks (IJCNN), pages 1419–1426. IEEE, 2017. 17

[157] T. H. Nguyen and R. Grishman. Event detection and domain adaptation with convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 365–371, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2060. URL https://aclanthology.org/P15-2060. 29

[158] T. H. Nguyen and R. Grishman. Modeling skip-grams for event detection with convolutional neural networks. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 886–891, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1085. URL https://aclanthology.org/D16-1085. 29

[159] T. H. Nguyen, K. Shirai, and J. Velcin. Sentiment analysis on social media for stock movement prediction. Expert Syst. Appl., 42:9603–9611, 2015. 42

[160] T. H. Nguyen, K. Cho, and R. Grishman. Joint event extraction via recurrent neural networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 300–309, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1034. URL https://aclanthology.org/N16-1034. 29, 83

[161] T.-T. Nguyen and S. Yoon. A novel approach to short-term stock price movement prediction using transfer learning. Applied Sciences, 9(22):4745, 2019. 17

[162] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. Using twitter for predictive stock market analysis. In 2010 AAAI Spring Symposium Series, 2010. 1

[163] G. Paolini, B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. dos Santos, B. Xiang, and S. Soatto. Structured prediction as translation between augmented natural languages. In 9th International Conference on Learning Representations, ICLR 2021, 2021. 70, 83

[164] A.-A. Papadopoulos, M. R. Rajati, N. Shaikh, and J. Wang. Outlier exposure with confidence control for out-of-distribution detection. Neurocomputing, 441:138–150, 2021. 32

[165] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. Signal processing, 99:215–249, 2014. 14

[166] S. R. Polamuri, K. Srinivas, and A. K. Mohan. Multi-model generative adversarial network hybrid prediction algorithm (mmgan-hpa) for stock market prices prediction. Journal of King Saud University-Computer and Information Sciences, 2021. 17

[167] M. Polignano, P. Basile, M. Degemmis, G. Semeraro, and V. Basile. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In CLiC-it, 2019. 27

[168] 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. 19

[169] A. M. Rather, A. Agarwal, and V. Sastry. Recurrent neural network and a hybrid model for prediction of stock returns. Expert Systems with Applications, 42(6):3234–3241, 2015. 18, 45

[170] M. Rawat, R. Hebbalaguppe, and L. Vig. Pnpood : Out-of-distribution detection for text classification via plug andplay data augmentation. CoRR, abs/2111.00506, 2021. URL https://arxiv.org/abs/2111.00506. 89

[171] R. Ren, D. D. Wu, and T. Liu. Forecasting stock market movement direction using sentiment analysis and support vector machine. IEEE Systems Journal, 13(1):760–770, 2019. doi: 10.1109/JSYST.2018.2794462. 12

[172] A. Ross, T. Wu, H. Peng, M. Peters, and M. Gardner. Tailor: Generating and perturbing text with semantic controls. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3194–3213, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.228. URL https://aclanthology.org/2022.acl-long.228. 92

[173] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In International conference on machine learning, pages 4393–4402, 2018. 91

[174] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. nature, 323(6088):533–536, 1986. 16, 36

[175] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek. Detecting formal thought disorder by deep contextualized word representations. Psychiatry Research, 304:114135, 2021. 91

[176] R. Sawhney, S. Agarwal, A. Wadhwa, and R. Shah. Deep attentive learning for stock movement prediction from social media text and company correlations. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8415–8426, 2020. 24

[177] R. Sawhney, S. Agarwal, A. Wadhwa, and R. R. Shah. Spatiotemporal hypergraph convolution network for stock movement forecasting. In 2020 IEEE International Conference on Data Mining (ICDM), pages 482–491. IEEE, 2020. 23

[178] R. Sawhney, M. Goyal, P. Goel, P. Mathur, and R. Shah. Multimodal multi-speaker merger & acquisition financial modeling: A new task, dataset, and neural baselines. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6751–6762, 2021. 42

[179] R. Sawhney, A. Wadhwa, S. Agarwal, and R. Shah. Fast: Financial news and tweet based time aware network for stock trading. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 2164–2175, 2021. 18

[180] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=SyxIWpVYvr. 32

[181] L. Sha, F. Qian, B. Chang, and Z. Sui. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018. 29

[182] P. Shi and J. Lin. Simple BERT models for relation extraction and semantic role labeling. CoRR, abs/1904.05255, 2019. URL http://arxiv.org/abs/1904.05255. 94, 96

[183] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642, 2013. 97

[184] P. Sonkiya, V. Bajpai, and A. Bansal. Stock price prediction using bert and gan. arXiv preprint arXiv:2107.09055, 2021. 27

[185] Y. Sun, C. Guo, and Y. Li. React: Out-of-distribution detection with rectified activations. Advances in Neural Information Processing Systems, 34:144–157, 2021. 32, 33

[186] M. Tan, Y. Yu, H. Wang, D. Wang, S. Potdar, S. Chang, and M. Yu. Out-of-domain detection for low-resource text classification tasks. 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), pages 3566–3572, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1364. URL https://aclanthology.org/D19-1364. 33, 89

[187] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. Advances in Neural Information Processing Systems, 32, 2019. 32

[188] H. Tian, C. Gao, X. Xiao, H. Liu, B. He, H. Wu, H. Wang, and f. wu. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4067–4076, Online, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.374. 55, 57

[189] Y. Tian, Y. Liu, G. Pang, F. Liu, Y. Chen, and G. Carneiro. Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes. arXiv preprint arXiv:2111.12264, 2021. 91

[190] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal. Uncertainty estimation using a single deep deterministic neural network. In International conference on machine learning, pages 9690–9700. PMLR, 2020. 91

[191] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017. 78

[192] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. 35

[193] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. arXiv preprint arXiv:1710.10903, 2018. 38

[194] D. Venugopal, C. Chen, V. Gogate, and V. Ng. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In Proceedings of the 2014 Conference on Empirical Methods in Natural

Language Processing (EMNLP), pages 831–843, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1090. URL https://aclanthology.org/D14-1090. 14

[195] P. Vincent and Y. Bengio. Manifold parzen windows. Advances in neural information processing systems, 15, 2002. 14

[196] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi. Entity, relation, and event extraction with contextualized span representations. 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), pages 5784–5789, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1585. URL https://aclanthology.org/D19-1585. 83

[197] C. Wang, H. Liang, B. Wang, X. Cui, and Y. Xu. Mg-conv: A spatiotemporal multi-graph convolutional neural network for stock market index trend prediction. Computers and Electrical Engineering, 103:108285, 2022. ISSN 0045-7906. doi: https://doi.org/10.1016/j.compeleceng.2022.108285. URL https://www.sciencedirect.com/science/article/pii/S0045790622005134. 24

[198] G. Wang, L. Cao, H. Zhao, Q. Liu, and E. Chen. Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty. AAAI21, pages 1–9, 2021. 42, 45

[199] H. Wang, T. Wang, and Y. Li. Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 971–978, 2020. 19, 43

[200] H. Wang, S. Li, T. Wang, and J. Zheng. Hierarchical adaptive temporal-relational modeling for stock trend prediction. In IJCAI, pages 3691–3698, 2021. 20

[201] H. Wang, W. Liu, A. Bocchieri, and Y. Li. Can multi-label classification networks know what they don't know? Advances in Neural Information Processing Systems, 34:29074–29087, 2021. 32

[202] H. Wang, J. Wang, L. Cao, Y. Li, Q. Sun, and J. Wang. A stock closing price prediction model based on cnn-bislstm. Complexity, 2021, 2021. 21

[203] 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. arXiv preprint arXiv:2104.04041, 2021. 17

[204] Q. Wen, Z. Yang, Y. Song, and P. Jia. Automatic stock decision support system based on box theory and svm algorithm. Expert systems with Applications, 37(2):1015–1022, 2010. 12

[205] B. Weng, M. A. Ahmed, and F. M. Megahed. Stock market one-day ahead movement prediction using disparate data sources. Expert Systems with Applications, 79:153–163, 2017. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2017.02.041. URL https://www.sciencedirect.com/science/article/pii/S0957417417301331. 13

[206] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101. 98, 118

[207] Z. Xiao, Q. Yan, and Y. Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. Advances in neural information processing systems, 33:20685–20696, 2020. 32

[208] B. Xie, R. Passonneau, L. Wu, and G. G. Creamer. Semantic frames to predict stock price movement. In Proceedings of the 51st annual meeting of the association for computational linguistics, pages 873–883, 2013. 11

[209] C. Xu, H. Huang, X. Ying, J. Gao, Z. Li, P. Zhang, J. Xiao, J. Zhang, and J. Luo. Hgnn: Hierarchical graph neural network for predicting the classification of price-limit-hitting stocks. Information Sciences, 2022. 22

[210] W. Xu, W. Liu, C. Xu, J. Bian, J. Yin, and T.-Y. Liu. Rest: Relational event-driven stock trend forecasting. In Proceedings of the Web Conference 2021, pages 1–10, 2021. 45

[211] 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), pages 1970–1979, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1183. URL https://aclanthology.org/P18-1183. 42, 45, 46, 47, 55, 56, 59, 60, 62

[212] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 5065–5075, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.393. URL https://aclanthology.org/2021.acl-long.393. 46

[213] B. Yang and T. M. Mitchell. Joint extraction of events and entities within a document context. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 289–299, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1033. URL https://aclanthology.org/N16-1033. 14, 29, 83

[214] H. Yang, Y. Chen, K. Liu, Y. Xiao, and J. Zhao. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In Proceedings of ACL 2018, System Demonstrations, pages 50–55, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-4009. URL https://aclanthology.org/P18-4009. 31, 46, 79, 83

[215] H. Yang, D. Sui, Y. Chen, K. Liu, J. Zhao, and T. Wang. Document-level event extraction via parallel prediction networks. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6298–6308, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.492. URL https://aclanthology.org/2021.acl-long.492. 31, 70, 78, 79, 83

[216] J. Yang, R. Rao, P. Hong, and P. Ding. Ensemble model for stock price movement trend prediction on different investing periods. In 2016 12th International Conference on Computational Intelligence and Security (CIS), pages 358–361, 2016. doi: 10.1109/CIS.2016.0087. 12

[217] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334, 2021. 91, 92

[218] L. Yang, Z. Zhang, S. Xiong, L. Wei, J. Ng, L. Xu, and R. Dong. Explainable text-

driven neural network for stock prediction. In 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), pages 441–445. IEEE, 2018. 19

[219] L. Yang, T. L. J. Ng, B. Smyth, and R. Dong. HTML: Hierarchical Transformer-Based Multi-Task Learning for Volatility Prediction, page 441–451. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450370233. URL https://doi.org/10.1145/3366423.3380128. 26

[220] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li. Exploring pre-trained language models for event extraction and generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5284–5294, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1522. URL https://aclanthology.org/P19-1522. 29, 30, 83

[221] Y. Yang, M. C. S. UY, and A. Huang. Finbert: A pretrained language model for financial communications, 2020. 26

[222] X. Yin, D. Yan, A. Almudaifer, S. Yan, and Y. Zhou. Forecasting stock prices using stock correlation graph: A graph convolutional network approach. In 2021 International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2021. doi: 10.1109/IJCNN52387.2021.9533510. 23

[223] 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 &amp; Data Mining, KDD '21, page 2037–2045, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467297. URL https://doi.org/10.1145/3447548.3467297. 26

[224] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6023–6032, 2019. 32

[225] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah. Out-of-distribution detection using union of 1-dimensional subspaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9452–9461, June 2021. 91

[226] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah. Out-of-distribution detection using union of 1-dimensional subspaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9452–9461, 2021. 91

[227] Z. Zeng, K. He, Y. Yan, Z. Liu, Y. Wu, H. Xu, H. Jiang, and W. Xu. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 870–878, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.110. URL https://aclanthology.org/2021.acl-short.110. 33

[228] L.-M. Zhan, H. Liang, B. Liu, L. Fan, X.-M. Wu, and A. Y. Lam. Out-of-scope intent detection with self-supervision and discriminative training. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3521–3532, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.273. URL https://aclanthology.org/2021.acl-long.273. 34

[229] J. Zhang, Y. Qin, Y. Zhang, M. Liu, and D. Ji. Extracting entities and events as a single task using a transition-based neural model. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pages 5422–5428. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/753. URL https://doi.org/10.24963/ijcai.2019/753. 29

[230] L. Zhang, C. Aggarwal, and G.-J. Qi. Stock price prediction via discovering multi-frequency trading patterns. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pages 2141–2149, 2017. 42, 43, 47

[231] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang, and P. Liu. Transformer-based attention network for stock movement prediction. Expert Systems with Applications, 202:117239, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.117239. URL https://www.sciencedirect.com/science/article/pii/S0957417422006170. 25

[232] Z. Zhang, W. Xu, and Q. Chen. Joint event extraction based on skip-window convolutional neural networks. In Natural Language Understanding and Intelligent Applications, pages 324–334. Springer, 2016. 29

[233] Z. Zhang, X. Kong, Z. Liu, X. Ma, and E. Hovy. A two-step approach for implicit event argument detection. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7479–7485, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.667. URL https://aclanthology.org/2020.acl-main.667. 31, 83

[234] F. Zhao, X. Li, Y. Gao, Y. Li, Z. Feng, and C. Zhang. Multi-layer features ablation of bert model and its application in stock trend prediction. Expert Systems with

Applications, 207:117958, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.117958. URL https://www.sciencedirect.com/science/article/pii/S0957417422011939. 28

[235] Z. Zhao, R. Rao, S. Tu, and J. Shi. Time-weighted lstm model with redefined labeling for stock trend prediction. In 2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI), pages 1210–1217. IEEE, 2017. 17

[236] S. Zheng, W. Cao, W. Xu, and J. Bian. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. 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), pages 337–346, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1032. URL https://aclanthology.org/D19-1032. xi, 8, 9, 29, 31, 70, 72, 76, 77, 78, 79, 83

[237] Y. Zheng, Z. Tan, M. Zhang, M. Maimaiti, H. Luan, M. Sun, Q. Liu, and Y. Liu. Self-supervised quality estimation for machine translation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3322–3334, 2021. 46

[238] P.-Y. Zhou, K. C. C. Chan, and C. X. Ou. Corporate communication network and stock price movements: Insights from data mining. IEEE Transactions on Computational Social Systems, 5(2):391–402, 2018. doi: 10.1109/TCSS.2018.2812703. 13

[239] Q. Zhou, C. Zhou, and X. Wang. Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection. Plos one, 17(2): e0262501, 2022. 21

[240] W. Zhou, F. Liu, and M. Chen. Contrastive out-of-distribution detection for pretrained transformers. In Proceedings of the 2021 Conference on Empirical Methods in Natural

Language Processing, pages 1100–1111, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.84. URL https://aclanthology.org/2021.emnlp-main.84. x, 33, 89, 91, 92, 93, 96, 97, 99, 100, 101, 105, 118, 119, 120

[241] Y. Zhou. Rethinking reconstruction autoencoder-based out-of-distribution detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7379–7387, 2022. 91

[242] Z. Zhou, L. Ma, and H. Liu. Trade the event: Corporate events detection for news-based event-driven trading. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2114–2124, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.186. URL https://aclanthology.org/2021.findings-acl.186. 45, 47

[243] Z. Zhou, L. Ma, and H. Liu. Trade the event: Corporate events detection for news-based event-driven trading. arXiv preprint arXiv:2105.12825, 2021. 28

[244] E. Zisselman and A. Tamar. Deep residual flow for out of distribution detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13994–14003, 2020. 32, 91

[245] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In International conference on learning representations, 2018. 91