Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
Special Track on AI in FinTech

F-HMTC: Detecting Financial Events for Investment Decisions Based on Neural Hierarchical Multi-Label Text Classification

Xin Liang[1,2], **Dawei Cheng**[3,4], **Fangzhou Yang**[4],
Yifeng Luo[1,2*], **Weining Qian**[1,2], **Aoying Zhou**[1,2]
[1]Shanghai Engineering Research Center on Big Data Management System, Shanghai, China
[2]School of Data Science and Engineering, East China Normal University, Shanghai, China
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[4]SeekData Inc., Shanghai, China
xliang@stu.ecnu.edu.cn, dawei.cheng@sjtu.edu.cn, fangzhou.yang@seek-data.com,
{yfluo, wnqian, ayzhou}@dase.ecnu.edu.cn

Abstract

The share prices of listed companies in the stock trading market are prone to be influenced by various events. Performing event detection could help people to timely identify investment risks and opportunities accompanying these events. The financial events inherently present hierarchical structures, which could be represented as tree-structured schemes in real-life applications, and detecting events could be modeled as a hierarchical multi-label text classification problem, where an event is designated to a tree node with a sequence of hierarchical event category labels. Conventional hierarchical multi-label text classification methods usually ignore the hierarchical relationships existing in the event classification scheme, and treat the hierarchical labels associated with an event as uniform labels, where correct or wrong label predictions are assigned with equal rewards or penalties. In this paper, we propose a neural hierarchical multi-label text classification method, namely F-HMTC, for a financial application scenario with massive event category labels. F-HMTC learns the latent features based on bidirectional encoder representations from transformers, and directly maps them to hierarchical labels with a delicate hierarchy-based loss layer. We conduct extensive experiments on a private financial dataset with elaborately-annotated labels, and F-HMTC consistently outperforms state-of-art baselines by substantial margins. We will release both the source codes and dataset on a public repository [1].

1 Introduction

Stock trading is an important kind of financial activity concerning investing and financing, and a lot of people and institutions are involved as investors in the stock trading market, buying and selling stock shares to gain margin profits. The share prices of listed companies are generally supported by several key drivers concerned with these companies, such as sales volumes, quarterly/annual revenues, gross profit margins, net incomes and earnings per share, etc. While these key drivers are usually influenced by various factors, such as politics, policies and macroeconomy, and changes of these factors would eventually result in changes of key drivers of supporting share prices. The status changes of a factor that influences the key drivers are usually presented as formal documents, such as news articles and policy statements, and the status changes of factors are deemed as events that would eventually affect the share prices of listed companies.

Since the share prices of a listed company may experience frequent ups and downs, investors in the stock trading markets need to face routine investment risks and opportunities. Since implementing event detection could help them to identify the investment risks and opportunities as early as possible, people are focusing on analyzing the documents related with events, so as to perceive the status changes of factors, and get prepared to avoid deleterious risks and seize beneficial opportunities, aiming to gain as many margin profits as possible. With detected events, people often perform backtesting to verify the connections between the events and the variation trends of stock share prices, and thus derive investment rules or tactics from these established connections. With the derived investment rules or tactics, people could leverage event detection to predict the variation trends of stock share prices, assuming that similar events would exert similar impacts on the variation trends of stock share prices. Based on the detected events, people could make their investment decisions, and some devoted to quantitative investment may even directly buy or sell stock shares, via following the investment rules programmed in their automated trading systems.

It is thus crucial to implement accurate and reliable event detection mechanisms for these event-driven investment applications, since they rely on detected events to make investment decisions. In actual application scenarios, the financial event scheme is often presented and organized with hierarchies, where a hierarchy denotes a sequence of event categories, and lower hierarchies denote finer event cate-

*Corresponding Author
[1]https://github.com/finint/F-HMTC

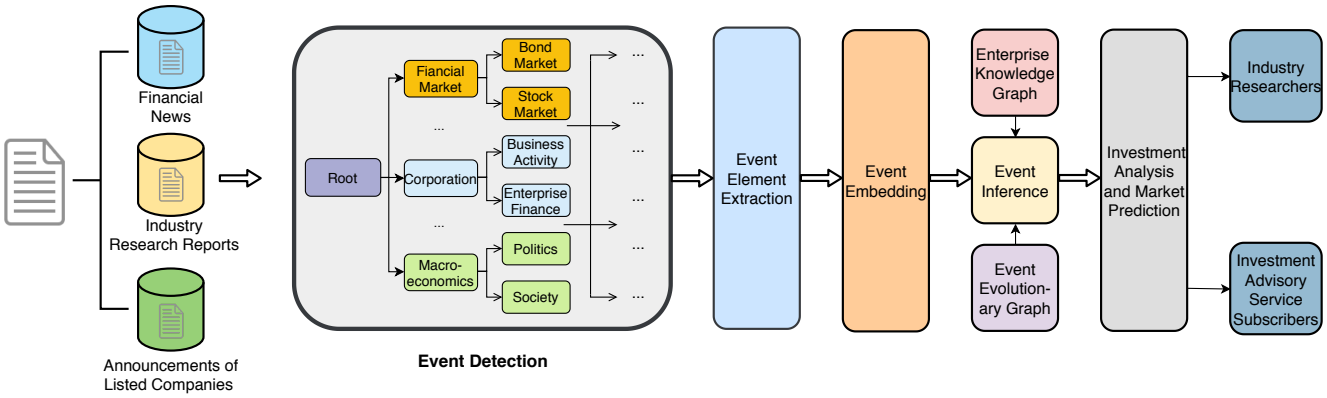

Figure 1: An investment analysis and market prediction framework based on event detection.

gories. As events could be classified with various hierarchies of categories, an event could be assigned with multiple category labels, indicating the hierarchical categories it should be classified with. Thus the financial event detection problem could be modeled as a hierarchical multi-label text classification (HMTC) problem[Wu *et al.*, 2016], where the whole event scheme could be denoted as a category tree, and an event instance can be associated with multiple hierarchical categories[Triguero and Vens, 2016] and represented as an unbranched path covering the contiguous hierarchical categories within the event scheme, from a top hierarchy node to a bottom hierarchy node[Sun and Lim, 2001].

Most HMTC methods assume that categories are independent, and dismiss the structure dependencies existing between category hierarchies, while others utilizing ensemble-based methods to simulate the hierarchical structures are computationally inefficient when dealing with large-scale numbers of categories. Besides, equal weights are given to all categories in these HMTC methods for false-positive penalties, while the parent, brother and child category labels associated with a category label should actually be treated discriminatively in real applications, since they denote different hierarchical levels and thus contain various amounts of information for predicting the label. Although [Wu *et al.*, 2019] considered hierarchical dependency information, they did not distinguish the impacts of different dependency relationships. Sometimes it is more acceptable to predict a label as its ground-truth parent hierarchy label than to predict it as some sibling label, since the parent label is a super hierarchical category of the children labels, and sibling labels within the same hierarchy are mutually exclusive.

In this paper, we propose an event detection mechanism based on neural hierarchical multi-label text classification. We employ bidirectional encoder representations from Transformers to learn deep representations for documents, and then embed the tree-structured event category scheme into a neural prediction layer. Moreover, we propose an industrial evaluation metric, and employ the metric for loss computation, where losses of wrong predictions for labels across different hierarchies are computed discriminatively. We finally conduct extensive experiments on our industrial dataset to evaluate our model, and we achieve remarkable performance improvements, compared with the state-of-art baseline methods. Our main contributions include:

1. We model financial event detection as a hierarchical multi-label text classification problem, and propose a neural event detection model, namely F-HMTC, for our real-life industrial application scenario.
2. We propose an industrial evaluation metric, and leverage the metric for loss evaluation with discriminative hierarchical prediction losses.
3. We conduct extensive experiments on our private financial dataset with elaborately-annotated labels, and we will release the dataset for public research.

2 Application Background

In this paper, we leverage event detection to help people make informed investment decisions in the application scenario illustrated in Fig.1. Here financial documents including news reports, industry research reports and announcements of listed companies, are collected for event detection based on a tree-structured event scheme, where a document may be associated with multiple event paths. Then the detected events are used for event inference with the extracted event elements and their embedding representations. We implement event inference via combining two kinds of knowledge graphs, namely the enterprise knowledge graph and the event evolutionary graph. The enterprise knowledge graph describes the relationships of entities related to all listed companies, including their major shareholders, senior managers, suppliers, partners and competitor companies, etc. and is mainly used to infer which companies shall be influenced by a detected event. The event evolutionary graph describes the connections between various events and their consequences, and is mainly used to infer how the detected event would influence the key drivers and ultimately the stock prices of the listed companies. The final investment analysis and market prediction results could be provided to industry researchers and investment advisory service subscribers, and they can make their own investment decisions based on the provided results.

In this application scenario, the hierarchical event scheme is constructed as an event category tree [Sun and Lim, 2001], where a tree node denotes an event category, and a descen-

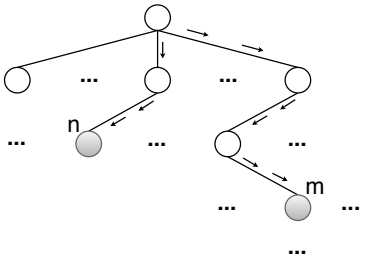

Figure 2: Two event paths associated with a document.

dant of the node denotes a finer-granularity event category. Here a document can be associated with multiple event paths of various depths in the event category tree, where an event path denotes a sequence of contiguous event hierarchies. As we could deduce the complete event path with the bottom-most node contained in an event path, given the readily-built event category tree, we thus could just leverage the bottom-most node to denote the whole event path, and the event hierarchies could just be represented with the finest-granularity event contained in the event hierarchies. For example, a document is associated with two event paths as Fig.2 shows, and they could just be denoted by node m and node n respectively. Denoting the event hierarchies with the finest-granularity events could reduce the workload of annotating training dataset, where a document instance is annotated with the finest-granularity event categories.

3 Methodology

The architecture of our model for hierarchical multi-label text classification is illustrated in Fig.3. The overall neural networks are presented in the right part, which mainly consists of an encoding network and a label prediction network. The encoding network contains 12 encoding layers, and each layer is composed of multiple basic units, each of which is a stack of six identical encoder modules as shown in the left part. The label prediction network is a shallow feedforward neural network.We mainly illustrate the encoding network and model optimization objective in rest of this section.

3.1 Encoding Network

For each document, the encoding network extracts the high-level features with bidirectional Transformer[Devlin *et al.*, 2018] encoders, from the document's various input embeddings. For a document $x = \{c_1, c_2, \cdots, c_m\}$, where c_i denotes a character included, we represent the document into an embedding matrix $\mathrm{M} \in \mathbb{R}^{|\mathrm{V}| \times d}$, where V denotes the fixed-sized character vocabulary and d denotes the dimension size of the character embeddings. The overall embedding, denoted as $\mathrm{E}_{\mathrm{c_i}}$, for a character is the concatenation of the character's token and position embeddings. The pre-trained Word2Vec [Mikolov *et al.*, 2013] Chinese character embeddings are leveraged as the token embeddings, and a learned positional embedding matrix $\mathrm{M}_p \in \mathbb{R}^{s \times d}$ is leveraged as position embeddings, where s denotes the length of character sequence supported in the model.

The embedding matrix of a document is fed to each of the Transformer encoders of the first encoding layer for feature extraction, and then the output vectors of these Transformer encoders are fed to each of the Transformer encoders of the second encoding layer. Each encoding layer is fully-connected to the next encoding layer in the encoding network, and the vectors of encoded hidden states output by the last encoding layer are fed to the prediction network for classification label prediction.

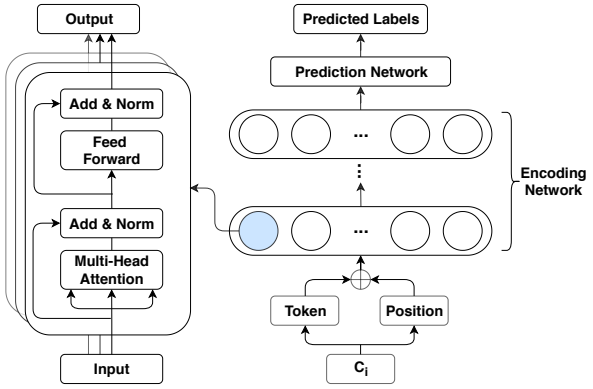

Figure 3: The architecture of F-HMTC.

A basic encoder module contained in Transformer encoders implements the multi-head attention defined as:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Con}(\mathrm{head_i})W^O \tag{1}$$

where $\mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ $(i \in [1, h])$, h denotes the number of parallel attention layers, $\mathrm{Con}(\mathrm{head_i})$ denotes the concatenation of h heads, and parameter matrices $W^O \in \mathbb{R}^{hd \times d_{model}}$, $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model} \times d}$. Because all of the queries, keys and values come from the same input, $\mathrm{E_{c_i}}$, their dimension sizes are all set to d. Besides the attention sub-layers, our encoder module includes a position-wise feed-forward network, consisting of two linear transformations with ReLU activations.

3.2 Optimization Objective

We define two vectors for a document instance d_i, namely a target vector y_i and a prediction vector $\hat{y}_i$. The target vector is a multi-hot 0-1 vector of N dimensions, with bits corresponding to the document's finest-granularity event categories set to 1, and other bits set to 0, where N is the overall number of hierarchical event categories. The prediction vector $\hat{y}_i$ is an n-dimensional vector of real values ranging from 0 to 1, and a component of $\hat{y}_i$ denotes the probability that corresponding event category is assumed to be the document's finest-granularity event category, and the event categories corresponding to components that exceed some predefined threshold are predicted as the document's event categories. For document d_i, we then define $dist(y_{i,t}, \hat{y}_i)$ to denote the distance between the t-th component of document's target vector and its prediction vector, just as:

$$dist(y_{i,t}, \hat{y}_i) = \frac{1}{N} \sum_{j}^{N} \alpha_{tj} (y_{i,t} - \hat{y}_{i,j})^2 \tag{2}$$

where α_{tj} is the penalty coefficient associated with the distance between the target hierarchy and predicted hierarchy.

Supposing target event category corresponding to $y_{i,t}$ falls in the l-th hierarchy of the event category tree and the predicted event category corresponding to $\hat{y}_{i,j}$ falls in the k-th layer, the penalty coefficient α_{tj} is defined as:

$$\alpha_{tj} = \begin{cases} 1 & if\ Label(\hat{y}_{i,j}) = Label(y_{i,t}) \\ \alpha_p^{l-k} \times \alpha_a & if\ Label(\hat{y}_{i,j}) \in Anc(Label(y_{i,t})) \\ \alpha_p^{l-k} \times \alpha_d & if\ Label(\hat{y}_{i,j}) \in Des(Label(y_{i,t})) \\ \alpha_o & otherwise \end{cases}$$

where $Label(\hat{y}_{i,j})$ denotes the label corresponding to $\hat{y}_{i,j}$, $Anc(c)$ and $Des(c)$ respectively denote the ancestor labels and descendant labels of label c. We define the hierarchical multi-label distance (HMD) as the prediction loss for d_i as:

$$\mathrm{HMD}(d_i) = \frac{1}{N} \sum_{t}^{N} dist(y_{i,t}, \hat{y}_i) \tag{3}$$

As we often face the label imbalance problem in the hierarchical classification task, where finer-granularity event categories are annotated with fewer label instances, we introduce the recursive regularization[Gopal and Yang, 2013; Peng *et al.*, 2018] to improve the model performance with such fewer finer-granularity event labels. As parent and child event categories possess hierarchical relationships, considering hierarchical dependencies between labels could encourage the parameters of child event categories to approach that of the parent event category, which could be learned with more label instances. Formally, we denote total parameter matrix of the final fully-connected prediction network as $\mathbf{W} = \{w_{l_i} : l_i \in \mathrm{L}\}$, where w_{l_i} denotes the parameter vector of event category l_i, l_i^j refers to a child event category of l_i and L denotes the whole label set. Then the recursive regularization is defined as:

$$\lambda(\mathbf{W}) = \sum_{l_i \in \mathrm{L}} \sum_{l_i^j} \frac{1}{2} \left\| w_{l_i} - w_{l_i^j} \right\|^2 \tag{4}$$

The model's overall optimization objective is defined as:

$$\mathcal{L} = \sum_{i=1}^{Z} \mathrm{HMD}(d_i) + C\lambda(\mathbf{W}) \tag{5}$$

where d_1,..., d_Z denote all the training documents, and C is the weight decay hyper-parameter. The MSE (i.e. Mean Square Error) optimization objective function is a special case of the objective function we propose, when the penalty coefficient matrix is the identity matrix and $C = 0$.

4 Experimental Evaluations

4.1 Experimental Settings

We invited senior professionals to sort out a hierarchical event scheme, which contains 98 event categories spreading across the event category nodes of the constructed event category tree of depth 7, based on our application scenario illustrated in Sec.2, and had our personnel with industrial research expertise annotate a Chinese financial dataset (CN-Fin) to evaluate our model's performance. We totally collected more than $500,000$ news documents from some major Chinese financial websites, and have financial labors annotate these documents with the defined event hierarchies, and finally chose $35,000$ news documents with high-quality event labels as the CN-Fin dataset. The dataset statistics are presented in Table.2.

In our model implementation, we set the hidden size to 768, self-attention head size to 12 and dropout rate to 0.1, learning rate to 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L2 weight decay to 0.001. Via data driven analysis and cross validation, we set α_a to 0.0002, α_d to 0.0003, α_o to 0.01 and α_p to 1.1, with $\alpha_o > \alpha_d^6 > \alpha_a^6$. For recursive regularization, we set regularization weight decay C to 0.00005, as suggested in [Peng *et al.*, 2018].

We compare the proposed method with the following state-of-the-art text classification methods:

- **TextCNN** [Kim, 2014]: which leverages a convolutional neural network for sentence-level classification.
- **FastText**[Joulin *et al.*, 2017]: which applies n-gram features to capture some partial information about the local word order for text classification.
- **HAN**[Yang *et al.*, 2016]: which leverages the Hierarchical Attention Network to capture basic insights about document structure to classify documents.
- **Transformer**[Vaswani *et al.*, 2017]: which leverages global dependencies with attention mechanisms for text classification.
- **Star-Transformer**[Guo *et al.*, 2019], which is a lightweight alternative of Transformer with a star-shaped topology to reduce classification complexity.
- **HSVM**[Cai and Hofmann, 2004], which is a classic hierarchical classification method.

The baseline models, except HSVM, are implemented with the Tencent open-source toolkit[2] for neural hierarchical multi-label text classification, and we implement HSVM according to the method proposed in [Cai and Hofmann, 2004]. For documents contained in the CN-Fin dataset, we perform word segmentation with Jieba[3], and then train 300-dimensional character embeddings on the dataset according to the skip-gram Word2Vec [Mikolov *et al.*, 2013] model, with the window size set to five, and the trained character embeddings are leveraged to train all models as initial input.

4.2 Evaluation Metrics

We benchmark the model's performance with three kinds of metrics, namely the accuracy metric, the micro-averaged hPRF metrics (hPrecision, hRecall and hF-score) and the HMDScore we define based on our actual application scenario. The PRF (precision, recall and F-score) metrics are commonly used for evaluating classification performance. We do not leverage the PRF metrics, as they are not suitable for hierarchical text classification tasks, where wrong classification predictions could not be clearly discriminated with these metrics. hPRF metrics are proposed by [Kiritchenko

[2]https://github.com/Tencent/NeuralNLP-NeuralClassifier

[3]https://github.com/fxsjy/jieba

Models	Exact Match	Parent	Grandparent	Child	Grandchild	Others	HMDScore
TextCNN	45.97%	8.17 %	2.41 %	7.58 %	1.86 %	34.01%	0.5945
FastText	50.01%	8.59 %	1.06 %	7.00 %	1.67 %	31.67%	0.6113
HAN	51.18%	10.21%	1.83 %	7.99%	1.84 %	26.95%	0.6276
Transformer	50.48%	10.01%	2.62 %	7.26%	1.83 %	27.80%	0.6158
Star-Transformer	44.09%	8.24 %	1.90 %	6.98%	1.76 %	37.03%	0.5902
TextCNN-HR	46.69%	7.68 %	1.76 %	7.05%	1.65 %	35.17%	0.6250
FastText-HR	51.35%	8.34 %	0.92 %	6.40%	1.48 %	31.52%	0.6367
HAN-HR	53.62%	9.64 %	1.85 %	6.97%	1.56 %	26.37%	0.6440
Transformer-HR	51.89%	9.37 %	2.31 %	6.77%	1.58 %	28.06%	0.6395
Star-Transformer-HR	45.28%	8.38 %	1.77 %	5.51%	1.39 %	37.67%	0.6174
HSVM	37.13%	7.00 %	0.76 %	5.06%	1.43 %	48.61%	0.4804
F-HMTC	**57.42%***	7.39 %	1.16 %	6.25%	1.39%	26.38%	**0.7102***

Table 1: Hierarchical distributions of prediction results and HMDSCores achieved by various models. Models with 'HR' denote that recursive regularization is used for these models. "Exact Match" denotes the ratio of prediction labels exactly matching the target labels, and "Parent", "Grandparent", "Child", and "Grandchild" respectively denote the ratio of prediction labels being parent, grandparent, child and grandchild labels of the target labels, while "Others" denotes the ratio of other prediction labels.

CN-Fin	Train	Test
#Vocabulary	268,200	268,200
#Documents	31,306	4,104
Average #s	16.4	17.8
Max #s	706	593
Average #w	168.9	415.6
Max #w	60,614	57,534

Table 2: Dataset statistics. #s denotes the number of sentences, and #w denotes the number of words.

et al., 2005] for evaluating hierarchical text classification models, where hierarchical classification predictions are discriminatively considered. Assuming document d_i with target label set C_i is predicted with the lable set $C_i^{'}$, we extend C_i and $C_i^{'}$ with the corresponding ancestor labels, and get: $\hat{C}_i = \{\bigcup_{c_k \in C_i} Anc(c_k)\}$, $\hat{C}_i^{'} = \{\bigcup_{c_k \in C_i^{'}} Anc(c_k)\}$, where c_k is an event category label. Then the micro-averaged hP and hR are defined as:

$$\mathrm{hP} = \frac{\sum_i \left|\hat{C}_i \bigcap \hat{C}_i^{'}\right|}{\sum_i \left|\hat{C}_i^{'}\right|} \qquad \mathrm{hR} = \frac{\sum_i \left|\hat{C}_i \bigcap \hat{C}_i^{'}\right|}{\sum_i \left|\hat{C}_i\right|} \tag{6}$$

hF-score is defined as:

$$\text{hF-score} = \frac{2 \cdot \mathrm{hP} \cdot \mathrm{hR}}{\mathrm{hP} + \mathrm{hR}} \tag{7}$$

In our application scenario, we deem a document's target labels and the corresponding parent labels as acceptable prediction results, and tend to exactly measure the hierarchical differences between the target labels and prediction labels. We define the hierarchical multi-label distance score (HMDScore) for performance evaluation, based on the hierarchical multi-label distance (HMD) aforementioned, just as:

$$\mathrm{HMDScore} = \frac{1}{Z}\sum_{i=1}^{Z}(1 - \tanh(\mathrm{HMD}(d_i))) \tag{8}$$

Models	hPresion	hRecall	hF-score
TextCNN	0.5529	0.5642	0.5585
FastText	0.5278	0.6397	0.5784
HAN	0.5518	0.6593	0.6008
Transformer	0.6139	0.5780	0.5954
Star-Transformer	0.5462	0.5568	0.5515
TextCNN-HR	0.5630	0.5754	0.5691
FastText-HR	0.5607	0.6048	0.5819
HAN-HR	0.5728	0.6495	0.6087
Transformer-HR	0.5602	0.6391	0.5971
Star-Transformer-HR	0.5301	0.5385	0.5343
HSVM	0.3328	0.8250	0.4743
F-HMTC	0.8009	0.6023	**0.6876***

Table 3: hPRF performance achieved by various models.

4.3 Experimental Results and Analysis

We benchmark the performance of our model, and compare it with the baseline models with and without recursive regularization. The experimental results of evaluating prediction accuracy and HMDScore are presented in Table.1, and experimental results of evaluating hPRF performance are presented in Table.3. From Table.1 and Table.3, we can see that our model outperforms all baseline models with or without recursive regularization. Among all the baseline models, HAN achieves the best performance with all the three kinds of evaluation metrics, while the performance of our model, namely F-HMTC, leads that of HAN by substantial margins, especially on HMDScore and hF-score. As far as prediction accuracy is concerned, F-HMTC yields the most "Exact Match" prediction results than all baseline models, and leads TextCNN by over 10% of "Exact Match" prediction labels, and leads HAN without recursive regularization by over 6.2% of "Exact Match" prediction labels.

As target labels' parent labels are deemed as acceptable prediction results, the overall accuracy of acceptable predic-

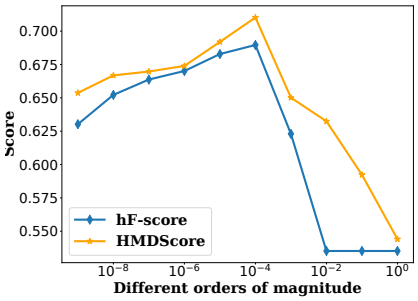

Figure 4: F-HMTC's performance on hF-score and HMDScore affected by the magnitude of hyper-parameters.

tions achieved by F-HMTC only leads HAN without recursive regularization by 3.4%, and only leads HAN with recursive regularization by about 1.5%. However, F-HMTC could provide more accurate finer-granularity event detection results for our application scenario with its leading accuracy of "Exact Match" predictions. The reason why F-HMTC's performance leads all baseline models is straight-forward, as leveraging hierarchical dependency information helps to make better label predictions. One thing worth noting is that introducing recursive regularization could improve the performance of baseline models, this also validates our assumption that hierarchical dependency information could improve the performance of hierarchical text classification models.

We finally explore how F-HMTC's performance is influenced by the α series of hyper-parameters, associated with hierarchical dependencies, via evaluating its hF-scores and HMDScores when α_a is set with various orders of magnitude, where the orders of magnitude of other α parameters are set accordingly, and the results are presented in Figure.4. We can see that the $\boldsymbol{\alpha}$ series of parameters play an important role in controlling the intensity of the penalty in the model. As the order of magnitude becomes larger, the model's performance will get better until the order of magnitude exceeds 10^{-4}, when the model's performance begins to decline. This verifies our assumption that hierarchical dependency information is indeed supplementary to the semantic information hidden within the document texts, while having hierarchical dependency information dominate the overall semantic information will degrade the model's performance, when α_a is set above some critical value.

5 Related Work

Text classification is one of the fundamental tasks of natural language processing. Traditional methods are mostly based on feature engineering and feature selection to obtain good features for text classification [Aggarwal and Zhai, 2012]. Support Vector Machines and Naive Bayes have already been used on text classification tasks for a long time [Joachims, 1998; McCallum *et al.*, 1998]. Dimensionality reduction methods like Latent Dirichlet Allocation [Blei *et al.*, 2003] has been used to reduce the feature space which can be better than bag-of-word with few features. As leveraging label dependencies can greatly improve the performance of multi-label hierarchical classification[Sun and Lim, 2001; Cai and Hofmann, 2004], [Wu *et al.*, 2016] proposes ML-Forest to learn an ensemble of hierarchical multi-label classifiers to reveal the intrinsic label dependencies.

Deep learning has been adopted for text classification, and pre-trained language models, such as Word2Vec [Mikolov *et al.*, 2013], FastText [Joulin *et al.*, 2017] and BERT [Devlin *et al.*, 2018], are leveraged in text classification models to improve documents' representations. While [Rousseau *et al.*, 2015; Peng *et al.*, 2018] use co-occurrence matrix to convert texts to graphs and apply graph mining algorithms to get frequent sub-graphs to define new features for text classification. CNN has also been applied for text classification using kernels to capture the documents' latent semantic information [Kim, 2014], and [Conneau *et al.*, 2016] designs VDCNN to operate directly at the character level with smaller convolutional kernels. [Liu *et al.*, 2016] proposes an RNN-based model for long document classification, which jointly learns across multiple related tasks via multi-task learning. In addition, [Wang, 2018] proposes a disconnected recurrent neural network to restrict the hidden states of each step to cover the words near the current position, by limiting the distance of information flow in RNN.

Models based on attention mechanisms, such as hierarchical attentions [Yang *et al.*, 2016] and self-attentions [Vaswani *et al.*, 2017], have been introduced for text classification recently. To reduce the dependencies of Transformers on large-scale training datasets, due to its heavy structure with fully-connected attention connections, [Guo *et al.*, 2019] proposes the Star-Transformer by replacing the fully-connected structure with a star-shaped topology. All of these methods could be further improved for HMTC tasks with massive numbers of hierarchical categories, because they often assume that categories are independent and thus dismiss the hierarchical dependencies existing between various categories.

6 Conclusions

In this paper, we model the financial event detection as a hierarchical multi-label text classification problem, and propose the F-HMTC event detection model for our industrial application scenario. F-HMTC leverages bidirectional encoders from Transformers for document representations, and implements a carefully-designed hierarchy-based loss layer for model optimization. The extensive experiments demonstrate that F-HMTC can achieve better classification performance, compared with baseline text classification models. We can conclude that the distance function we introduce with penalty coefficients can provide more insights for the hierarchical multi-label text classification problem.

Acknowledgments

This work was supported by the National Key R&D Program of China (2018YFC0831904) and the Joint Research Program of SeekData Inc.[4] and ECNU.

[4]www.seek-data.com

References

[Aggarwal and Zhai, 2012] Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[Cai and Hofmann, 2004] Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87. ACM, 2004.

[Conneau *et al.*, 2016] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Gopal and Yang, 2013] Siddharth Gopal and Yiming Yang. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 257–265. ACM, 2013.

[Guo *et al.*, 2019] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. 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 1315–1325, 2019.

[Joachims, 1998] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[Joulin *et al.*, 2017] Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[Kiritchenko *et al.*, 2005] Svetlana Kiritchenko, Stan Matwin, and A Fazel Famili. Functional annotation of genes using hierarchical text categorization. In *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology (held at ISMB-05*. Citeseer, 2005.

[Liu *et al.*, 2016] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

[McCallum *et al.*, 1998] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Peng *et al.*, 2018] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, pages 1063–1072. International World Wide Web Conferences Steering Committee, 2018.

[Rousseau *et al.*, 2015] François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. Text categorization as a graph classification problem. 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 1702–1712, 2015.

[Sun and Lim, 2001] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Proceedings of IEEE International Conference on Data Mining*, pages 521–528. IEEE, 2001.

[Triguero and Vens, 2016] Isaac Triguero and Celine Vens. Labelling strategies for hierarchical multi-label classification techniques. *Pattern Recognition*, 56:170–183, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[Wang, 2018] Baoxin Wang. Disconnected recurrent neural networks for text categorization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2311–2320, 2018.

[Wu *et al.*, 2016] Qingyao Wu, Mingkui Tan, Hengjie Song, Jian Chen, and Michael K Ng. Ml-forest: a multi-label tree ensemble method for multi-label classification. *IEEE transactions on knowledge and data engineering*, 28(10):2665–2680, 2016.

[Wu *et al.*, 2019] Cinna Wu, Mark Tygert, and Yann LeCun. A hierarchical loss and its problems when classifying non-hierarchically. *PLoS ONE*, 14(12):e0226222–e0226222, 2019.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.