Exploiting Network Structures to Improve Semantic Representation for the Financial Domain

Chao Feng[1] , **Shi-jie Wei**[2]
[1]University of Zurich
[2]Harbin Institute of Technology
chao.feng2@uzh.ch, 2190400620@stu.hit.edu.cn

Abstract

This paper presents the participation of the Mini-True team in the FinSim-3 shared task on learning semantic similarities for the financial domain in English language. Our approach combines contextual embeddings learned by transformer-based language models with network structures embeddings extracted on external knowledge sources, to create more meaningful representations of financial domain entities and terms. For this, two BERT based language models and a knowledge graph embedding model are used. Besides, we propose a voting function to joint three basic models for the final inference. Experimental results show that the model with the knowledge graph embeddings has achieved a superior result than these models with only contextual embeddings. Nevertheless, we also observe that our voting function brings an extra benefit to the final system.

1 Introduction

Semantic representation and meaning representation have attracted the attention of linguists and philosophers for a long time [Dowty, 2012; Chierchia and McConnell-Ginet, 2000]. Human could share knowledge, experiences and thoughts through languages, i.e. words could be mapped to the concept space [Vigliocco and Vinson, 2007]. With the developing of Natural Language Processing technologies, words can be represented with dense vectors, which makes this mapping more promising [Maarouf *et al.*, 2020]. However, early studies have pointed that there is no one-to-one mapping between words meaning and concepts [Murphy, 2004; Vigliocco and Vinson, 2007].

An alternative is to map concepts to their hypernyms, which are the generic terms conceive the meaning of the concepts [Maarouf *et al.*, 2020]. FinSim-3 is such a shared task of hypernym categorization which focuses on financial domain. Financial entities have strong domain characteristics, which makes hypernym classification more difficult. It also means that this classification tightly coupled with semantic understanding. Therefore, we need a model which can fully mine semantic information as well as domain characteristics, to automatically classify financial concepts to their hypernyms.

Our work is done as part of the FinSim-3 (The 3rd Shared Task on Learning Semantic Similarities for the Financial Domain), which is a classification task for financial domain text data in English. We aim to combine sentence level embeddings learned by deep-learning models like BERT [Devlin *et al.*, 2018] with knowledge graph embedding model, to create a multi-label classification system. To solve the domain specific problem, we leverage various features of language models and knowledge graph embeddings model, and implement our system in four separate stages:

1. In order to solve the domain specific problem, we use two different pretrained language models in produce the sentence level embeddings, including the original English version BERT [Devlin *et al.*, 2018], and the financial text pretrained language model FinBERT [Araci, 2019].
2. To exploit the network structure of the concepts, we create a knowledge graph with external knowledge sources. Besides, a knowledge graph embedding algorithm has been applied to leverage the concepts structure information.
3. Three different models are built upon these different features and inference the output label independently.
4. We use a simple voting mechanism to integrate these three basic models, and to get the final label.

The paper is organized as follows. We introduce the previous works related to our method in Section 2, and present the description of our method in Section 3. Then we group the experimental results in Section 4 before discussing the perspectives and concluding in Section 5.

2 Related work

As an important part of semantic relation extraction, hyponym-hypernym relationship identification has attracted the interest of NLP researchers for a long time.

Traditionally, technologies of hyponym-hypernym detection make use of knowledge bases (e.g. WordNet, WikiData, etc.) to extract hypernymy relations [Bordea *et al.*, 2015]. However, recent approaches are more relied on data-driven methods. Text features (e.g. Character Count, Word

Proceedings of the Third Workshop on Financial Technology and Natural Language Processing
(FinNLP@IJCAI 2021), pages 58-62, Online, August 19, 2021.

Count, etc.) are extracted from the raw text. Then, a number of traditional machine learning algorithm could be applied for this problem [Saini, 2020; Anand *et al.*, 2020; Béchara *et al.*, 2015].

Besides, approaches based on end-to-end neural networks offer an another promising alternative. Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) networks are commonly used for Natural Language Processing tasks in earlier works. [Anand *et al.*, 2020] has applied LSTM models to hyponym-hypernym detection. But these approaches are mainly used the context-independent word embeddings or character embeddings. Since the pretrained transformer-based language models have been introduced to the NLP area, context-dependent embeddings are heavily used in different tasks. Researches have adopted BERT model to hyponym-hypernym detection task, and achieved highly scores [Keswani *et al.*, 2020].

3 System Architecture

This section presents the architecture and the description of our system. Firstly, we show how to construct the knowledge graph from the provided extra knowledge sources, and how to train the knowledge graph embeddings. Secondly, three basic models are described in the Subsection 3.2. Lastly, we introduce the integration of our basic models.

3.1 Knowledge Graph Embeddings

As a powerful tool, knowledge graphs are usually used for knowledge representation. Within knowledge graphs, a factual triplet (h, r, t) represents a relation r between a head-term h and a tail-term r. Variety of NLP applications are based on knowledge graph technologies, such as question-answering system [Hao *et al.*, 2017] and recommendation systems [Zhang *et al.*, 2016].

In this FinSim-3 task, the organizers provide a set of prospectuses to be used for embeddings training. On the other hand, we can also use these raw text data to construct a knowledge graph, which means if two-tuples of terms h and t are co-occurrence in one document d, we can group it as a triplet (h, d, t). As shown in Algorithm 1, after traversing all the head and tail elements and articles, we can get a new knowledge graph KG containing all co-occurrence relationships.

Algorithm 1 Construction of Knowledge Graph

Input: Terms set E, documents set D.
Output: A knowledge graph KG.

```
for $h$ in $E$ do
    for $t$ in $E$ do
        for $d$ in $D$ do
            if $d$ contains $h$ and $t$ then
                $KG$.append(($h$, $d$, $t$))
            end if
        end for
    end for
end for
return $KG$
```

Figure 1 visualises the knowledge graph we created. The nodes present the terms of our training data-set. The edges present the co-occurrence relations between the head-term and the tail-term, and the colors of edges represent different documents.

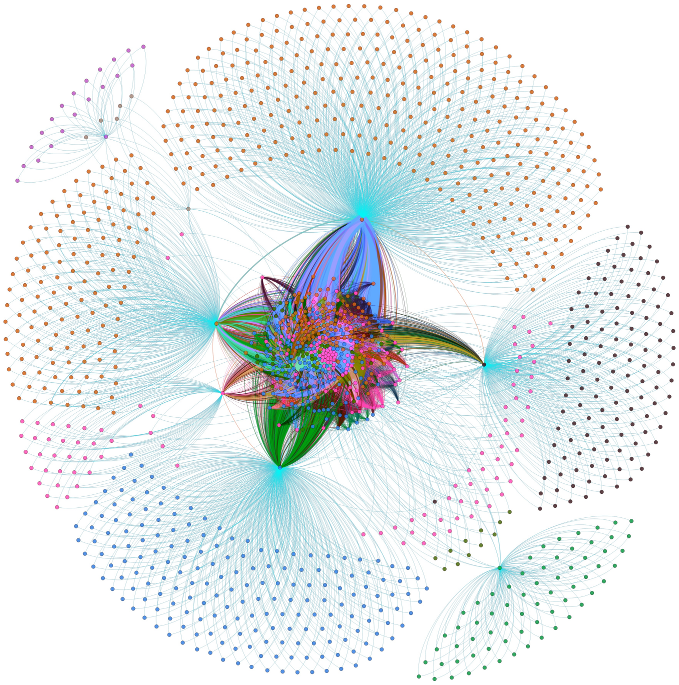

Figure 1: The visualization of knowledge graph. Nodes means the terms, edges means the relations, and the colors present the different relations.

Similar with word embeddings, using low dimensional vectors to represent entities and relationships has been proofed to be effective and efficient [Bordes *et al.*, 2013]. As we have created the knowledge graph KG, the next step is to train the knowledge graph embeddings. We use the RotatE algorithm to train our KG embeddings, which is proposed in [Sun *et al.*, 2019]. Training with 1374 nodes and 1048575 edges, we get our 756-dimensions knowledge graph vectors.

Shown in Figure 2, we use a Principal components analysis (PCA) method to cast these vectors into two-dimensions space, and randomly choose two categories of words to visualise these data, including Alternative Debenture and Bearer bond which belong to Bond category, and Alternative inverestment fund and closed End Fund which belong to Fund category. As we can see, there is a strong cohesion between terms within a same category. This means that our KG embeddings may provide additional information that can be used to help us classify effectively.

3.2 Basic Models

We develop three independent basic models for label prediction by combining language models with simple neural networks. As before mentioned, two language models and one knowledge graph embeddings model are applied in our system, including BERT, FinBERT, and RotatE.

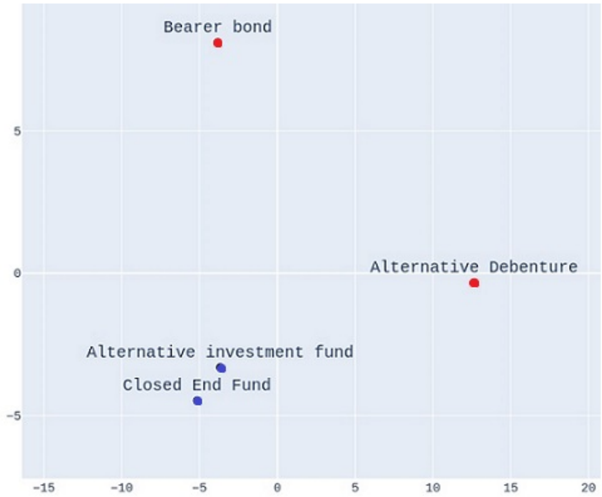

Figure 2: A simple visualization of knowledge graph embeddings after PCA dimensionality reduction .

Basic Model One:
This first basic model connects the language model, i.e. the BERT, with a feed forward network. After tokenization, the text data feeds into the language model, and generate a sentence embedding. After that, embedded data flows to a feed forward network for the final inference I_1.

Basic Model Two:
Research points out that the performance of original BERT model in domain related data is not as good as the domain specific language models [Araci, 2019]. On this count, we replace the BERT by financial version language model, FinBERT. Similar with the first model, sentence level embeddings flow into a inference network to get the final output I_2.

Basic Model Three:
To leverage the network structure information of the input data, we create a knowledge graph embedding model to obtain the network structure features from the extra knowledge source. In this time, we combine the features learned from FinBERT with the KG embeddings, and use a simple classifier network to get the final inference I_3.

3.3 Voting Mechanism

As before mentioned, I_1, I_2, I_3 are three inference values predicted by our basic models, and a simple voting mechanism is applied to count the final output. If two or more models draw the same inference label, this label will be our final prediction. The final output is predicted as:

$$f(x) := \begin{cases} 0 \text{ if } I_1{+}I_2{+}I_3 < 2 \\ 1 \text{ if } I_1{+}I_2{+}I_3 >= 2 \end{cases} \quad (1)$$

The architecture of our model is shown in Figure 3. As we can see, the input text date feeds into each basic model, and then flows to each inference network respectively. Finally, all of this information is combined in the voting function for the prediction.

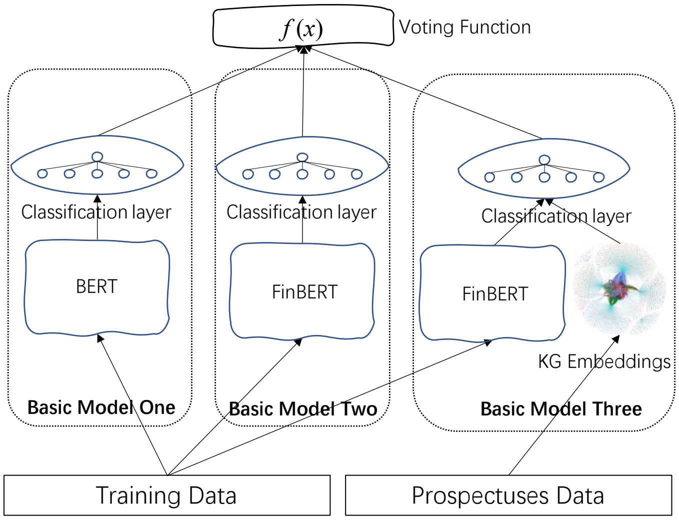

Figure 3: The final model combines three basic models with a voting mechanism

4 Experiments and Results

In this section, we present the description of datasets, the setup of our experiments, and the final results of our experiments. Experimental validation is conducted on the FinSim-3 training and test corpus. The datasets are described in subsection 4.1 followed by the analysis of our final results in subsection 4.2.

4.1 Data Description

In this shared task, the organizers give a list of selected terms from the financial domain. Participants are required to design and implement a classifier to automatically classify these financial terms to the most relevant hypernym concepts.

As shown in Table 1, the train and the test sets are not of equal size. There are 1050 terms in the train set, and 326 terms in the test set.

Corpus	train	test
Number of terms	1050	326

Table 1: Dataset of terms for FinSim-3.

In this task, we need to classify these terms into categories of financial instruments, including ’Bonds’, ’Forward’, ’Funds’, ’Future’, etc. Table 2 illustrates the distribution of the terms in each category. As we can see, the number of terms in each label are not balanced, there are huge gaps among these numbers of different categories. For example, about one-fourth of terms belong to the ’Equity Index’ label, but only eight of them belong to ’Securities restrictions’. This data-imbalance problem makes the task more difficult.

4.2 Results and Discussion

The evaluation of this hypernym identification task is mainly based on Accuracy score and Average Rank score. Our results are gathered in Table 3, which contains the Accuracy and Average Rank scores for the basic models and the final

Corpus	Term category	Number of terms
Train	Bonds	55
	Central Securities Depository	107
	Credit Events	18
	Credit Index	129
	Debt pricing and yields	58
	Equity Index	286
	Forward	9
	Funds	22
	Future	19
	MMIs	17
	Option	24
	Parametric schedules	15
	Regulatory Agency	205
	Securities restrictions	8
	Stock Corporation	25
	Stocks	17
	Swap	36

Table 2: Number of terms in each financial instruments.

systems. M1 - M3 refer to our basic model one - three, and M4 refers to our final model. As we can only submit three results to the organizers, so only the results of M2 - M4 are shown in the table.

Corpus	Model	Accuracy	Average Rank
train	M1(BERT)	0.944	1.168
	M2(FinBERT)	0.971	1.065
	M3(M2 + KG emb)	0.972	1.068
	M4(M3 + Voting)	0.973	1.051
test	M2(FinBERT)	0.825	1.337
	M3(M2 + KG emb)	0.855	1.346
	M4(M3 + Voting)	0.865	1.315

Table 3: Evaluation Results: Accuracy and Average Rank.

Q1. *Is the domain specific language model brings additional benefits to the system?*

The answer is simple yes. As we can see int Table 3, models with FinBERT are better than the original BERT model. With simply replace the BERT with FinBERT, our model gets a 3% extra performance, which strongly support our hypothesis.

Q2. *Is the knowledge graph embeddnigs brings additional benefits to the system?*

Back to the Table 3, our M3, FinBERT with KG embeddings, is better than the model with only FinBERT in both train and test sets. Especially in the test set, KG embeddings improve our model from 82.5% to 85.5%, which is a huge increase.

Q3. *Is the voting function brings additional benefits to the system?*

As we can see, the our final system (with a simple voting mechanism) performs the best Accuracy and Average Rank scores among these four systems. We also care about where are the errors come from for our system. So we draw a confusion matrix of our final model in Figure 4. As we can see, most of the mis-classification terms are incorrectly labeled as 'Bond'. The reason for this error may be the data imbalance problem, so that the weights of each label are unbalanced in our model.

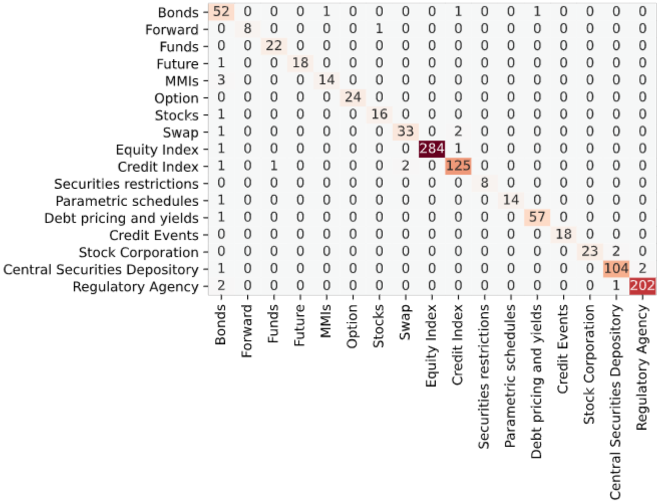

Figure 4: Heat map for the confusion matrix of our final model

5 Conclusions

In this paper, we implement a classification system with combining the language models and a knowledge graph embeddings model. Our system has been evaluated on the FinSim-3 shared task. The evaluation results has shown that our system is domain robust, and the KG embeddings and voting function indeed bring in extra benefits to our final system. Our experiments prove that the pre-trained language model is also highly adaptable to financial domain texts, and the network structure can highly leverage the predictive ability of the multi-label model system.

References

[Anand *et al.*, 2020] Vivek Anand, Yash Agrawal, Aarti Pol, and Vasudeva Varma. FINSIM20 at the FinSim task: Making sense of text in financial domain. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 104–107, Kyoto, Japan, 5 January 2020. -.

[Araci, 2019] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[Béchara *et al.*, 2015] Hanna Béchara, Hernani Costa, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. MiniExperts: An SVM approach for measuring semantic textual similarity. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 96–101, Denver, Colorado, June 2015. Association for Computational Linguistics.

[Bordea *et al.*, 2015] Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. SemEval-2015 task 17: Taxonomy extraction evaluation (TExEval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, Colorado, June 2015. Association for Computational Linguistics.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[Chierchia and McConnell-Ginet, 2000] Gennaro Chierchia and Sally McConnell-Ginet. *Meaning and grammar: An introduction to semantics*. MIT press, 2000.

[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.

[Dowty, 2012] David R Dowty. *Word meaning and Montague grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ*, volume 7. Springer Science & Business Media, 2012.

[Hao *et al.*, 2017] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[Keswani *et al.*, 2020] Vishal Keswani, Sakshi Singh, and Ashutosh Modi. IITK at the FinSim task: Hypernym detection in financial domain via context-free and contextualized word embeddings. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 87–92, Kyoto, Japan, 5 January 2020. -.

[Maarouf *et al.*, 2020] Ismail El Maarouf, Youness Mansar, Virginie Mouilleron, and Dialekti Valsamou-Stanislawski. The FinSim 2020 shared task: Learning semantic representations for the financial domain. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 81–86, Kyoto, Japan, 5 January 2020. -.

[Murphy, 2004] Gregory Murphy. *The big book of concepts*. MIT press, 2004.

[Saini, 2020] Anuj Saini. Anuj at the FinSim task: Anuj@FINSIM¡VLearning semantic representation of financial domain with investopedia. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*, pages 93–97, Kyoto, Japan, 5 January 2020. -.

[Sun *et al.*, 2019] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[Vigliocco and Vinson, 2007] Gabriella Vigliocco and David P Vinson. Semantic representation. *The Oxford handbook of psycholinguistics*, pages 195–215, 2007.

[Zhang *et al.*, 2016] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.