Enriching BERT With Knowledge Graph Embedding For Industry Classification

Shiyue Wang, Youcheng Pan, Zhenran Xu, Baotian Hu(✉), and Xiaolong Wang

Harbin Institute of Technology, Shenzhen, China
{19s151128,youcheng.pan,xuzhenran}@stu.hit.edu.cn,
{hubaotian,xlwangsz}@hit.edu.cn

Abstract. Industry classification for startup companies is meaningful not only to navigate investment strategies but also to find potential competitors. It is essentially a challenging domain-specific text classification task. Due to the lack of such dataset, in this paper, we first construct a dataset for industry classification based on the companies listed on the Chinese National Equities Exchange and Quotations (NEEQ), which consists of $17,604$ annual business reports and their corresponding industry labels. Second, we introduce a novel Knowledge Graph Enriched BERT model (KGEB), which can understand a domain-specific text by enhancing the word representation with external knowledge and can take full use of the local knowledge graph without pre-training. Experimental results show the promising performance of the proposed model and demonstrate its effectiveness for tackling the domain-specific classification task.

Keywords: Industry classification · Knowledge graph · Graph convolutional network

1 Introduction

Industry classification is a primary problem to classify companies into specific industry category according to their primary business sectors, market performances and the major products [1]. It is essential to research in the financial field, as dividing the companies into homogeneous groups could help the academic researchers narrow down the scope of their investigation, identify comparable companies and set performance benchmarks [2]. It also can reflect the industry characteristics of companies and provide investors with market trends.

Unlike A-shares with persistent main business sectors, small-and-medium-sized enterprises (SMEs), especially new startup companies, usually react to the ever-evolving demand of the market by changing their main businesses frequently. For startup companies that aim at publicly trading, classification can help them catch up with the existing A-share companies and find potential competitors. There are already plenty of applications on industry classification on

T. Mantoro et al. (Eds.): ICONIP 2021, CCIS 1517, pp. 709–717, 2021.
https://doi.org/10.1007/978-3-030-92310-5_82

A-share companies like Global Industry Classification Standard (GICS), still, there is a lack of datasets on startup companies for further studies.

As industry classification can be attributed to financial text classification task, common deep neural networks do not perform well on domain-specific tasks. The text classification task is a fundamental task in neural language processing as numerous methods have been proposed, such as TextCNN [6] and BERT [4]. However, the professional terms stand for special meaning which needs an additional explanation when understanding. Recent studies have made attempts to integrate knowledge graphs into basic models. Zhang et al. [15] propose an enhanced language representation model, but the model ignores the relation between entities. W. Liu et al. [10] transform input sentence into a knowledge-rich sentence tree and introduce soft-position and visible matrix. Still, it only concerns relevant triples from the entities present in the sentence, dismissing expanding relations in the knowledge graphs.

Table 1. The annual business reports of one company and its corresponding classification label.

Year: 2015 **Label**: Industrials **Business model**: The abbreviation of the company's security name is "Daocong Technology". The company is mainly engaged in the research and development of traffic safety technology, technical consulting
Year: 2016 **Label**: Information Technology **Business model**: The company's security name is changed to "Gaiya Entertainment". The company has established a new strategic pattern with mobile game development and operation business as the core

For the problems mentioned above, in this work, we focus on solving the integration of word representation and knowledge. As an effort towards it, we first construct a dataset on startup companies for the industry classification task. The dataset contains the annual business reports of companies on NEEQ and their corresponding labels. These companies are typically SMEs, and their classifications could be wavering in years. For instance listed in Table 1, a firm renamed its security from Daocong Technology into Gaiya Entertainment, with the leading business sector changing from transportation to mobile games. Second, We propose a Knowledge Graph Enriched BERT (KGEB) which can load any pre-trained BERT models and be fine-tuned for classification. It makes full use of the structure of the knowledge graphs extracted from texts by entity linking and nodes expanding. Finally, experiments are conducted on the dataset, and results demonstrate that KGEB can get superior performances.

The contribution of this work is threefold: (1) A large dataset is constructed for industry classification based on the companies listed on NEEQ, consisting of companies' descriptions of business models and corresponding labels.

(2) A Knowledge Graph Enriched BERT (KGEB), which can understand domain-specific texts by integrating both word and knowledge representation, is proposed and is demonstrated beneficial. (3) The KGEB obtains the results of 0.9198 on Accuracy and 0.9089 on F1, which outperforms the competitive experiments and demonstrates that the proposed approach can improve the classification quality.

2 NEEQ Industry Classification Dataset

NEEQ is the third national securities' trading venue after the Shanghai Stock Exchange and Shenzhen Stock Exchange. We construct the industry classification dataset based on the NEEQ website, and the process is summarized as follows: 1) we acquire 20,040 descriptions of the business model from 2014 to 2017 from the open-source dataset [1]. 2) For each description of the business model, we acquire the releasing time of the report and check out the investment-based industry classification result which is rightly after the releasing time. 3) By filtering and cleaning repeated descriptions, we obtain the final dataset which consists of 17,604 pairs of descriptions of business models and their industry classification labels. We split the dataset into a training set (80%), a dev set (10%), and a test set (10%). Among the dataset, the maximum of descriptions of business model is 13,308, and the minimum is 38, and the median is 630. On average, each company contributes to 1.79 different business model descriptions, demonstrating the wavering features of startup companies. Table 2 summarizes the preliminary information about the dataset of industry classification. The dataset is freely available at https://github.com/theDyingofLight/neeq_dataset.

Table 2. Overview of the classification dataset NEEQ industry classification.

Label	Name	Train	Dev	Test
0	Materials (MT)	1726	200	213
1	Consumer Discretionary (CD)	1834	242	233
2	Industrials (ID)	4122	488	492
3	Information Technology (IT)	3793	494	490
4	Financials (FN)	198	20	26
5	Telecommunication Services (TS)	322	33	49
6	Consumer Staples (CS)	739	93	107
7	Health Care (HC)	894	115	108
8	Energy (EG)	298	27	29
9	Utilities (UT)	98	7	15
10	Real Estates (RE)	84	6	9
	Total	14108	1725	1771

3 Methodology

The text classification task can be defined as follows. Given a passage denoted as $X = \{x_1, x_2, ..., x_n\}$, n is the length of the passage. In this paper, Chinese tokens are at the character level. The model's target is to predict the classification label Y defined as $Y = argmaxP(Y|X, \theta)$, where θ denotes the model parameters. Our overall approach is depicted in Fig. 1.

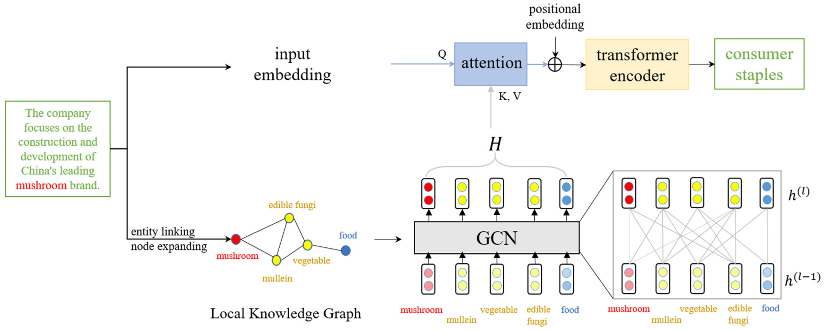

Fig. 1. The overview of our approach. It contains three steps: (1) Build the local knowledge graphs. (2) Transform the graphs into node representation. (3) Combine input passage representation with node representation for classification.

Local Knowledge Graph. Given an input passage, a set of triples can be retrieved from the knowledge base by linking the mentions parsed from the passage to the entities in the knowledge base and expanding relation paths. We define the set of triples as a Local Knowledge Graph. Formally, a Knowledge Base is represented as a $K = (V, E)$, where $V = \{v_j\}$ is the set of vertices and $E = \{e_j\}$ is the set of edges of the vertices, and each triple (head entity, relation, tail entity) in KB is denoted as $\tau = (e_h, v_{hs}, e_s)$. The local knowledge graph is assumed as $G = \{\tau_1, \tau_2, ..., \tau_g\}$, where g is the number of triples. The way to construct the local knowledge graph is as follows. Firstly, for each passage X, we conduct mention parsing to obtain mentions and entity disambiguation to get pairs of entities and nodes from the knowledge base called XLore [13] with the entity linking system XLink [5]. We rank all the candidate nodes by their cosine similarity with the word embedding [9] and select the entities by the threshold larger than 0.4 and top-10 entities if there are more than 10.

Node Representation. After obtaining the local knowledge graph G with g nodes, we feed G into the $L-$layer GCN model [14] for the representation of each node, where we denote $h_i^{(l-1)}$ as the input vector and $h_i^{(l)}$ as the output vector of node i at the $l-$layer. The process of calculation is: $h_i^{(l)} = \sigma(\sum_{j=1}^{n} \widetilde{A}_{ij} W^{(l)} h_j^{(l-1)} / d_i + b^{(l)})$ where $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ represents the matrix sum of adjacency matrix $\boldsymbol{A}$ and identity matrix $\boldsymbol{I}$, $d_i = \sum_{j=1}^{n} \widetilde{A}_{ij}$ is the degree of entity i in the local knowledge graph, $W^{(l)}$ is a trainable linear transformation and σ is a nonlinear function. We initialize the node embedding with the output of a pre-trained model, which takes the whole words in the node as input and outputs a fixed length vector. The output of the GCN last layer is used as the node representation $H = \{h_1^{(L)}, h_2^{(L)}, ..., h_g^{(L)}\}$.

Knowledge Graph Enriched BERT. Knowledge Graph Enriched BERT is proposed to enrich the representation of long passage with node representation from local knowledge graphs. As a multi-layer bidirectional Transformer encoder, BERT maps an input sequence of characters X to a sequence of representations $Z = \{z_1, z_2, ..., z_n\}$. To fuse node representation into the word embedding layer, we utilize attention mechanism to integrate word embedding $W = \{w_1, w_2, ..., w_n\}$ and node representation $H = \{h_1^{(L)}, h_2^{(L)}, ..., h_g^{(L)}\}$ formulated as: $\alpha_t = softmax(H^T W^P w_t)$, $w'_t = H \cdot \alpha_t$ where W^P is the trainable parameters and $W' = \{w'_1, w'_2, ..., w'_n\}$ is the output of the fusion. Then we add a residual connection on the original word embedding to avoid vanishing gradient. We also adopt consistent position embedding and token type embedding with BERT and we sum up three layers of embedding as the output of the embedding layer. The output is then fed into a stack of identical layers which contains a multi-head self-attention mechanism and a position-wise fully connected feed-forward network [12]. We utilize the final hidden vector $z_1 \in \mathbb{R}^H$ corresponding to the first input token ($[CLS]$) to represent the entire sequence. We introduce classification layer weights $W \in \mathbb{R}^{K \times H}$, where K is the number of labels.

We compute a standard classification loss with z_1 and W, and I^* denotes the target category: $O = softmax(z_1 W^T)$, $\mathcal{L} = -log(O(I^*))$

4 Experiments

Our experiments study the proposed model on the NEEQ industry classification dataset, compare the model with existing approaches and analyze the results.

4.1 Experimental Settings

We have five models for comparison. The models are implemented on open source code. (1) GCN [8]: A fundamental GNN model for the classification task. (2) Logistic Regression [11]: A basic linear model for classification. (3) TextCNN [6]: A CNN with one convolution layer on top of word vectors. (4) BERT [4]: A language model pre-trained on a large scale of corpus to obtain deep bidirectional representations and renews the records on many downstream tasks. (5) K-BERT [10]: it enables language representation model with knowledge graphs by first injecting relevant triples into the input sentence and second being fed into the embedding layer, seeing layer and the mask-transformer.

In all our experiments, we initialize the word embedding with parameters of bert-base-chinese with a hidden size of 768 and 12 hidden layers. In the fine-tuning process, we use a batch size of 8, the number of gradient accumulation steps of 8, the learning rate of 5e−5. In the embedding layer, we use bert-as-service equipped with bert-base-chinese to get the initial node embedding whose dimension is 768, the threshold of similarity of entites in XLore is 0.4, and the layer size of GCN is 4. The dropout in GCN layers is 0.5. Models are trained with the Adam optimizer [7]. We select model based on the dev set.

We conduct the automatic evaluation with the following metrics: Accuracy measures the proportion of the number of samples that are correctly predicted to the total samples. F1 is the macro average of Precision and Recall, which measures the correctness of all categories.

4.2 Results and Analysis

Table 3 shows the experimental results against the competitor methods. GCN achieves the worst performance since it only utilizes the local knowledge graphs, missing information from the passages. Logistic Regression focuses much on the statistical information of words and can benefit from the long texts as in our dataset. TextCNN is a CNN model designed for text classification. It can represent text sequences with a deep neural network, but it doesn't model the long passage well and lacks domain-specific knowledge. Likewise, BERT is initialized with the pre-train parameters and has been significantly improved, but it still has problems understanding domain-specific texts. Although the result of K-BERT is worse than our model, it demonstrates the influence of knowledge graphs. Compared with CN-DBpedia [3] K-BERT extracts knowledge triples from, XLore contains 3.6 times the amount of entities, which contributes to the deeper comprehension of domain information.

Table 3. The experimental results (%) on the NEEQ industry classification dataset.

Model	Accuracy	Precision	Recall	F1
GCN	69.70	66.16	61.77	63.20
LR	89.21	91.95	85.99	88.63
TextCNN	89.16	90.98	84.32	87.18
BERT	91.41	91.71	88.46	89.95
K-BERT	90.97	89.26	86.55	87.88
KGEB	**91.98**	**92.55**	**89.45**	**90.89**

Compared with the competitor methods, our model takes advantage of the knowledge. Complementing words with node representation is helpful because it provides additional information and considers the structure of the graphs, making the word embedding select more helpful information from the nodes. The performance achieves absolute improvements by at least +0.5 in Accuracy and +0.94 in F1.

As Table 4 shows, the experimental results on each category support the effectiveness of KGEB. TextCNN performs worst on almost all of the classes. BERT performs better, but in most categories, KGEB achieves the highest Precision,

Table 4. The experimental results (%) on the NEEQ industry classification dataset on each category.

	Precision			Recall			F1		
	TextCNN	BERT	KGEB	TextCNN	BERT	KGEB	TextCNN	BERT	KGEB
MT	86.57	87.91	**88.69**	87.79	88.73	**92.02**	87.18	88.32	**90.32**
CD	83.12	88.16	**90.31**	84.55	86.27	**87.98**	83.83	87.20	**89.13**
ID	89.19	**91.52**	91.03	92.28	94.31	**94.92**	90.71	92.89	**92.94**
IT	90.69	93.69	**94.39**	91.43	**93.88**	92.65	91.06	**93.78**	93.51
FN	100	92.86	**100**	92.31	100	**100**	96.00	96.30	**100**
TS	87.88	**90.48**	84.09	59.18	**77.55**	75.51	70.73	**83.52**	79.57
CS	**94.06**	93.40	93.46	88.79	92.52	**93.46**	91.35	92.96	**93.46**
HC	91.96	93.40	**95.24**	**95.37**	91.67	92.59	93.64	92.52	**93.90**
EG	91.67	85.19	**92.00**	75.86	79.31	79.31	83.02	82.14	**85.19**
UT	100	92.31	**100**	**93.33**	80.00	86.67	**96.55**	85.71	92.86
RE	85.71	**100**	88.89	85.71	88.89	**88.89**	75.00	**94.12**	88.89

Recall and F1 score, demonstrating that the additional knowledge information can not only help distinguish terms and draw attention on domain-specific words but also understand the meaning of decisive words. However, since KGEB is still far from a perfect classifier, we also show that there are descriptions our current model cannot classify well. These labels could be predicted correctly if we had a better entity linking system and node expanding strategies. Taking a text labeled "Telecommunication Service" as an example, the local knowledge graph is constructed based on the entities containing "digital television", "health" and "care" from the text "In addition to retaining the original digital television business, the future will be based on the field of health care" and it is possible to bring noise interference.

4.3 Case Study

Comparing the predicted labels of TextCNN, BERT and KGEB, our model improves the recall of the labels with respect to the baselines. In the cases where all of them predict correctly, the keywords in the sentence can help the models increase the probability of correct labels. In the cases where BERT and TextCNN predict wrong labels, like the sentence "The company is committed to the production and sales of core equipment for water treatment and recycling of domestic sewage", BERT and TextCNN label the description "Industrials", but owing to the additional information about "sewage disposal", KGEB obtains the correct prediction. In some cases, all of the models classify the description with the wrong label. If we removed the noise when conducting entity linking system and node expanding strategies, we could obtain the correct classification results like in the last example.

5 Conclusions

In this paper, we construct a large dataset for industry classification task and propose a novel knowledge enriched BERT which can extract the local knowledge graph from the business sentences and integrate word and knowledge representation. The experimental results outperform the competitor methods and demonstrate the effectiveness of our proposed model. In the future, we will continue to improve this work and extend the method to more applications.

Acknowledgements. We appreciate the insightful feedback from the anonymous reviewers. This work is jointly supported by grants: Natural Science Foundation of China (No. 62006061), Strategic Emerging Industry Development Special Funds of Shenzhen (No. JCYJ20200109113441941) and Stable Support Program for Higher Education Institutions of Shenzhen (No. GXWD20201230155427003-20200824155011001).

References

1. Bai, H., Xing, F.Z., Cambria, E., Huang, W.B.: Business taxonomy construction using concept-level hierarchical clustering. Papers (2019)
2. Bhojraj, S., Lee, C., Oler, D.K.: What's my line? A comparison of industry classification schemes for capital market research. J. Acc. Res. **41**(5), 745–774 (2003)
3. Bo, X., et al.: CN-DBpedia: a never-ending Chinese knowledge extraction system. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (2017)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Zhang, J., Cao, Y., Hou, L., Li, J., Zheng, H.-T.: XLink: an unsupervised bilingual entity linking system. In: Sun, M., Wang, X., Chang, B., Xiong, D. (eds.) CCL/NLP-NABD -2017. LNCS (LNAI), vol. 10565, pp. 172–183. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69005-6_15
6. Kim, Y.: Convolutional neural networks for sentence classification. Eprint Arxiv (2014)
7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
9. Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., Du, X.: Analogical reasoning on Chinese morphological and semantic relations. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Vol. 2: Short Papers (2018)
10. Liu, W., Zhou, P., Zhao, Z., Wang, Z., Wang, P.: K-bert: enabling language representation with knowledge graph. In: Proceedings of the AAAI Conference on Artificial Intelligence (2020)
11. Menard, S.: Logistic regression. American Statistician (2004)
12. Vaswani, A., et al.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008 (2017)
13. Wang, Z., et al.: Xlore: a large-scale english-chinese bilingual knowledge graph. In: Proceedings of the 12th International Semantic Web Conference (2013)

14. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018)
15. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: Ernie: enhanced language representation with informative entities. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)