Construction Research and Applications of Industry Chain Knowledge Graphs

Boyao Zhang[1,2], Zijian Wang[1], Haikuo Zhang[3](✉), Yonghua Zhao[1], Jingqi Sun[1], and Jing Wang[1]

[1] Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
{zhangby,yhzhao}@sccas.cn, {wangzj,jqsun,wangjing}@cnic.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] China Internet Network Information Center, Beijing, China
zhanghaikuo@cnnic.cn

Abstract. Research on listed companies is an important part of stock analysis. This study proposes an automatic construction method of the knowledge graph for the financial industry chain, which can better track and study the relationship between the production and operation of listed companies from the perspective of the graph. To solve the problems of high expert labor costs, late update and maintenance, and unstandardized and label-lacking datasets during the construction of vertical domain knowledge graphs, this study conducts knowledge extraction of unstructured text by integrating two dependency parsing methods, phrase structure trees, and dependency parse trees. Furthermore, this study uses automatically labeled datasets to train a deep learning-based named entity recognition model. This method, which integrates the two abovementioned methods, improves normalization ability and allows for the automatic construction of a financial industry chain knowledge graph.

Keywords: Knowledge graph · Dependency parsing · Named entity recognition · Construction · Phrase structure tree · Dependency parse tree

1 Introduction

The stock market is an important channel for corporate financing and resource allocation because it is a significant part of the capital market. The stock price reflects the value of the listed company, making stock price analysis inseparable from listed company research [1]. The evaluation of a listed company's value is often based on its fundamental dimensions, including its production and operation, the future development prospects of its industry, and the impact of macroeconomic policies. However, listed companies are often related to and influenced by each other. Particularly in the production and operation processes of listed companies, there are often complex relationships between upstream and downstream suppliers, partners, or competitors.

The financial industry has accumulated a large amount of structured and unstructured data, and many high-performance big data processing technologies [2] have made significant contributions to financial security and data analysis [3, 4]. A knowledge graph is

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13368, pp. 120–132, 2022.
https://doi.org/10.1007/978-3-031-10983-6_10

a semantic web that describes the relationship between real objective entities and records attributes of entities and their interrelationships in a "entity-relation-entity" triple set. Knowledge graphs have become increasingly popular in the financial industry in recent years for financial investment [5], fraud detection [6], and *Knowledge Graph-based Question and Answer* (KBQA) [7].

Previous research on the correlation and influence between listed companies has demonstrated that a knowledge graph can not only link the entity knowledge buried in the text in a network, but also perform vectorized mapping of the embedding space for the graph using matrix decomposition [8], random-walk based embedding [9, 10], graph neural networks [11], and other powerful graph embedding technologies. Finally, during the downstream task, the mapped knowledge graph can be used to extract entity attributes and additional information hidden in the graph structure, or it can be integrated with other knowledge graphs.

The present study proposes a method which automatically extract entities such as raw materials and products from the texts describing production and operation activities of listed companies, and then construct a knowledge graph of the company's industry chain. Listed companies usually provide detailed information on their main products and raw materials used for production in their *Initial Public Offering* (IPO) prospectus and annual financial reports. Natural language texts describing production and operation can differ significantly across industries and even between companies within the same industry. Furthermore, many entity names are long and contain complex relationships. Therefore, recognizing financial industry chain entities is more difficult than recognizing named entities in the common field.

The construction of a financial knowledge graph has its own challenges because the required dataset is extracted from descriptive unstructured text in prospectuses and annual financial reports of listed companies, and there is no labeled dataset of enterprise upstream and downstream industry chains. Manual data labeling is almost equivalent to manually constructing a knowledge graph, and the text materials of listed companies are continuously updated. Thus, building and maintaining a knowledge graph of the industry chain is time consuming. Therefore, an automated extraction and labeling method is urgently required. The *Named Entity Recognition* (NER) is the most important technique for building a knowledge graph [12]. Currently, extracting accurate company information from prospectuses and annual financial statements of listed companies using the general domain NER model is difficult. Downstream industry chain entity information urgently requires an automated construction method.

The main objectives of this study are as follows. First, this study proposes combining phrase structure tree [13] and dependency parse tree [14] to analyze the subject-predicate-object structure, sentence components, corresponding parts of speech, and other information from relatively regular sentences to address the lack of labeled data in the financial industry chain. Second, this study combines a set of manual rules to complete the task of extracting financial entities. Third, automatic data labeling is achieved by reverse-mapping the dictionary generated from the financial entities extracted from the entity extraction task into the original text. Manual labeling cost can thus be significantly reduced. Finally, based on the financial text annotation dataset, the present study constructs a NER model for the financial industry chain based on the pre-training model and Bi-LSTM+CRF [15]. The accuracy and recall rate of entity extraction in the financial industry chain have been effectively improved through the integration and optimization of phrase structure tree, dependency parse tree, and the NER model.

This article contains the following sections: in Sect. 2, we introduce related work on construction of knowledge graphs in vertical domains; in Sect. 3, we introduce the method for constructing industry chain knowledge graph, including ontology modeling and knowledge extraction methods; in Sect. 4, experiments are conducted on the knowledge extraction, which proposes the effectiveness of knowledge extraction methods in this paper for construction of industry chain knowledge graph; Sect. 5 is conclusion.

2 Related Work

This section discusses related work on knowledge graph construction, vertical domain knowledge extraction, and dependency parsing. With the rapid advances in computer capabilities [16, 17], network facilities [18, 19], and the new algorithms [20, 21], large amounts of data [22, 23] can be collected and processed quickly, safely [24, 25] and accurately to generate the useful information or knowledge with the aid of machine learning [26, 27] and artificial intelligence [28, 29] techniques.

A *knowledge graph* is a technology that uses a graph model to describe knowledge and model the relationships between everything in the world. Recently, the accumulation of crowdsourced data on the World Wide Web has led to the creation of various knowledge graphs in the general domain, including Freebase [30], DBpedia [31], Yago [32], and Baidu Graph [33]. Unlike general domain knowledge graphs, vertical domain knowledge graphs are usually developed to solve a specific problem in a specific profession, and they have a higher practicability. Typical examples of such knowledge graphs include the e-commerce domain knowledge graph from Alibaba [34], Linked Life Data medical domain knowledge graph [35], and Chinese medicine knowledge graphs [36].

When constructing a vertical domain knowledge graph, domain ontology modeling, knowledge extraction, knowledge fusion, and knowledge storage are all required technologies. Knowledge extraction methods for unstructured text are mainly classified into four categories: extracting knowledge based on expert-created rules, extracting knowledge based on statistical models such as *Hidden Markov Model* (HMM) and *Conditional Random Field* (CRF) [37, 38], extracting knowledge based on dependency parsing combined with domain knowledge [39, 40], and extracting knowledge based on deep learning methods such as pre-trained models and recurrent neural networks [41]. The development of a knowledge extraction method based on manually created rules often requires high labor costs owing to the need to involve a large group of experts and the need for professionals to maintain time and update. However, knowledge extraction methods based on statistical models and deep learning models usually require a large number of labeled datasets in vertical domains to be input.

To solve the problems of high labor costs, late updates, and lack of labeled datasets that occur when developing knowledge graphs in the financial industry chain, this study proposes a knowledge extraction model based on dependency parsing technology. This study uses this foundation to label unstructured texts and use the labeled data as inputs to train a depth-based NER model Finally, the problem of the insufficient generalization ability of a single method is solved by integrating the two models, and the automatic construction of the knowledge graph for the financial industry chain can be achieved.

3 Construction Method of Industry Chain Knowledge Graph

First, the ontology of the knowledge graph in the financial field is constructed in this section using the analytical logic used when researching listed companies. Subsequently, a knowledge extraction method for industry chains based on dependency parsing is proposed. According to the extraction results, this study performed an automatic labeling task and used deep learning to train the NER model. Finally, the generalization ability of knowledge extraction is improved using model fusion, and the method proposed in this study can automatically develop an industry chain knowledge graph in the financial field.

3.1 Ontology Modeling of Knowledge Graphs in Financial Field

Ontology is the foundation and framework of a knowledge graph. An ontology is a semantic model that defines entity types, entity attributes, and associations between entities in a knowledge graph [42]. A knowledge graph in the financial field developed in this study must support the fundamental analysis of listed companies; therefore, the logical structure of the ontology must be consistent with stock research methods. The ontology must include not only listed companies and entities related to them, but also the relationships between entities and the unique attribute information of each entity.

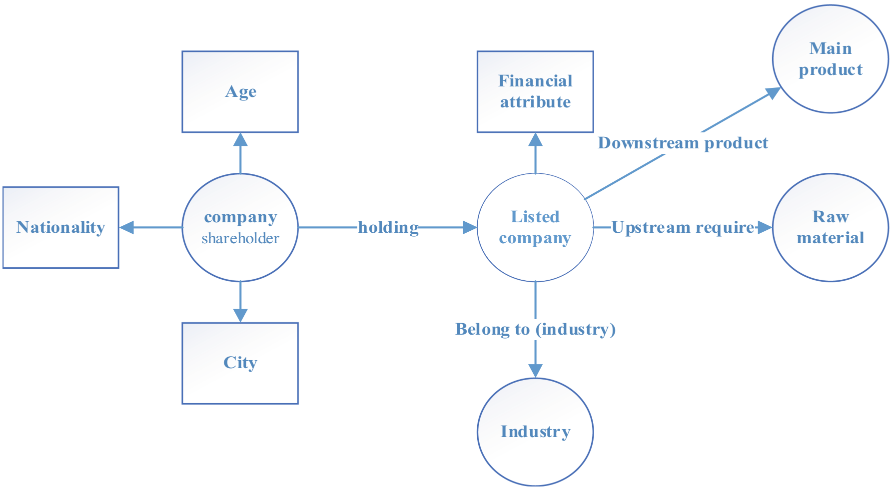

Fig. 1. The ontology of knowledge graphs in the financial field. Circles represent entities, squares represent attributes, and lines with text represent edges. The listed company entity is the core of the knowledge graph, and other industry entities, shareholder entities, etc. are connected to the listed company entities through edges.

The ontology of the financial knowledge graph developed in this study, as shown in Fig. 1, includes the following:

1. Entity types are defined as listed companies, raw materials, main products, industry, and company shareholders.
2. Following the definition of entity types, the relationship types between them are also defined. The relationship type between the listed company entity and the industry entity; for example, is "belong to (industry)," and between the listed company entity and the main product entity is a "downstream product."
3. The attribute information of each entity is also defined; for example, the entity attributes of the company shareholder entity include age, nationality, and city.

A graph definition includes a series of triplets. In this study, we are mainly concerned with extracting $<$ listed company-downstream product-main product $>$ and $<$ listed company-upstream needs-raw materials $>$ knowledge in the form of triples.

As shown in Table 1, the data used in this study to develop an industry chain knowledge graph mainly comes from the IPO prospectus and annual financial statements of listed companies. The specific data used included statements, shareholder data, and tabular and text data describing the main product and raw materials.

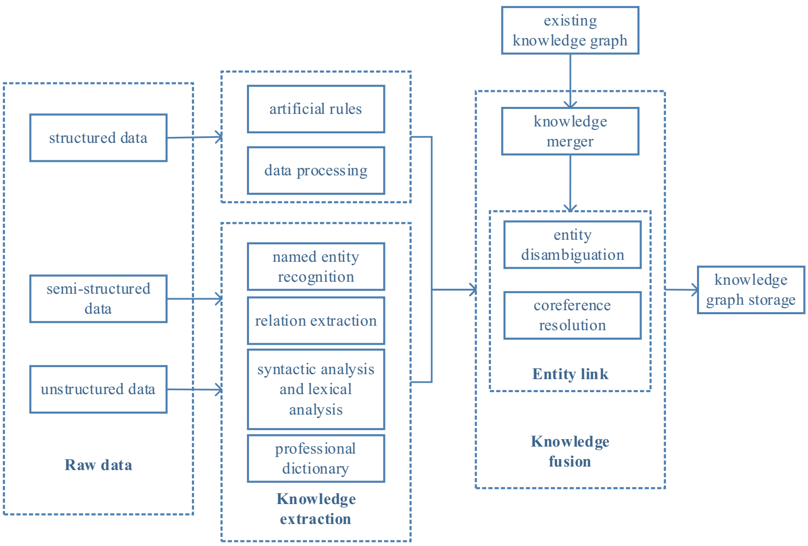

Fig. 2. The framework shows the general steps to construct a knowledge graph. From left to right, it shows a typical process from raw data, knowledge extraction, knowledge fusion to finally constructing a graph and storing it.

Figure 2 depicts the financial knowledge graph construction process. Raw data were divided into three types: structured, semi-structured, and unstructured. Structured data are directly extracted from listed company entities using manual rules and used to supplement the attribute information. Semi-structured data are used to produce shareholder and main product information of listed companies. Unstructured data is used to extract upstream raw material information from the text of the listed company, which describes its production and operation processes. The extracted knowledge is fused with the existing knowledge graph to produce a final knowledge graph.

Table 1. Data sources for building knowledge graph.

Data type	Data content	Knowledge graph
Structured data	Statement data in Annual Financial Statements	Financial attribute of listed company
Semi-structured data	Shareholder data in Annual Financial Statements	Information about shareholders
Semi-structured data	Tabular data describing the downstream main product of listed company in Annual Financial Statements	Main product entity
Unstructured data	Text data describing upstream raw materials of listed company in Annual Financial Statements and prospectus	Raw material entity

3.2 Knowledge Extraction Based on Dependency Parsing

Dependency parsing is one of the most important tasks in natural language processing. It is a prerequisite for an in-depth understanding of natural language. It performs a

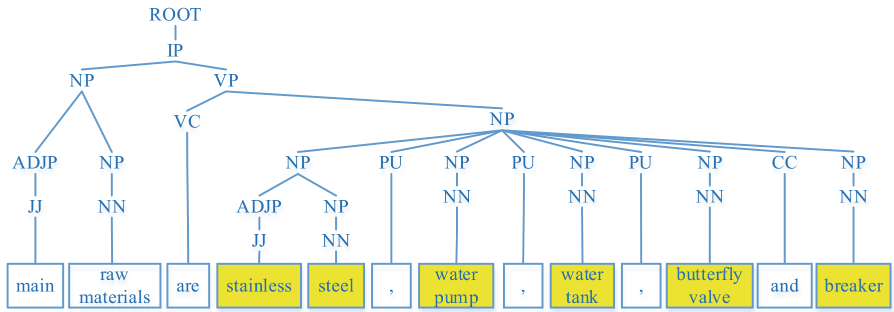

Fig. 3. The example of extracting entities from sentences and building knowledge graphs based on the phrase structure tree method. The sample is a sentence of a listed company describing the raw materials needed for its production. In the typical example of the phrase structure tree, the NP words appearing after VC (marked in bright yellow) are raw material entities. (Color figure online)

structural analysis of natural language, splitting different sentence structures and their dependencies into a tree-like structure logic, which is easy to understand and analyze. Dependency parsing techniques can be divided into two language structure disassembly methods: phrase structure trees and dependency parse trees. This study proposes a raw material entity extraction method based on these two language structure disassembly methods.

The phrase structure tree divides sentence from top to bottom based on *Probabilistic Context-Free Grammar* (PCFG) and recursively decomposes them into *Inflection Phrase* (IP), *Verb Phrase* (VP), *Noun Phrase* (NP), and other structures according to the phrase structure tree, as shown in Fig. 3. In this study, the knowledge extraction task was implemented according to the NP structure division.

The theory of dependency parsing is used to create a dependency parse tree. It generates a dependency syntax tree for a sentence and focuses on the dependencies between words in the sentence, such as the definite relationship, subject-predicate relationship, and nouns juxtaposition, as shown in Fig. 4. In this study, the knowledge extraction task is mainly accomplished by extracting the object and the *coordinate word* (COO) based on dependencies such as the *verb-object* (VOB).

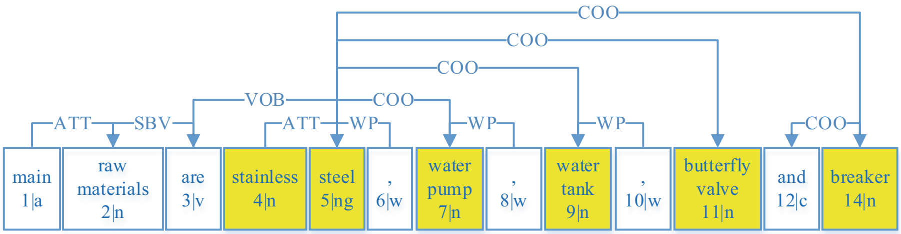

Fig. 4. The example of extracting entities from sentences and building knowledge graph based on the dependency parse tree method. The sample is a sentence of a listed company describing the raw materials needed for its production. In the typical example of the dependency parse tree, the words pointed to by the VOB relation and the words associated with it (by COO relation) are raw material entities (marked in bright yellow). (Color figure online)

In this study, the phrase structure tree and dependency parse tree were combined to extract joint entities. This method not only considers the sentence phrase structure, but also introduces the dependency between words to obtain a better financial entity extraction effect, as shown in Fig. 5.

3.3 Automatic Labeling and Named Entity Recognition

The financial knowledge extraction task can be completed using well-structured text using a dependency parsing-based method. Owing to the diversity of the semantic structure of financial texts, a dependency parsing-based method cannot completely cover all scenarios. Therefore, this method has certain limitations. In this study, the NER model is used to cover other less regular sentences and is combined with dependency parsing to improve generalization ability.

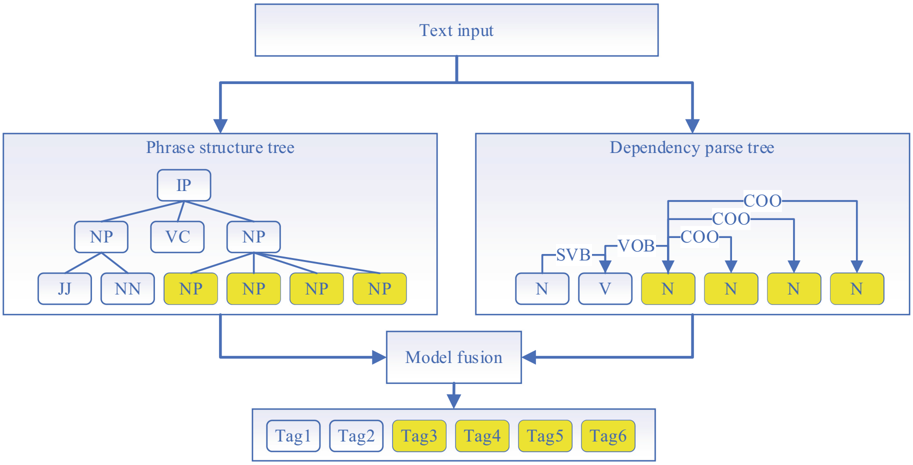

Fig. 5. The framework shows the model fusion process of phase structure trees and the dependency parse trees. The input text describes the raw materials required for production. The output is the union of the results of the knowledge extraction of the two models. The fusion model takes into account both sentence phrase structure and dependency between words.

In this study, the dependency parsing module was used to complete part of the entity extraction tasks. The extracted entity information can be customized into dictionaries for various industries, and then mapped back to the text using manual rule-matching to complete the automatic labeling task. The BIO labeling mode is used for data set labeling, which means that non-entity data are marked with "O," and the first character of the upstream raw material entity is marked as B_PM, and the internal character is marked as I_PM. After the automatic data labeling was completed, to scientifically and reasonably evaluate the performance of the model trained in this study, the labeled data were divided according to a certain proportion, of which 28,000 pieces were divided into training data sets, and the remaining 1000 pieces of data were manually annotated by experts to form the test dataset.

The NER method developed in this study uses the encoder of the pretraining model as a feature extractor. The embedding sequence of the sentence was used as input into the Bi-LSTM model for re-encoding, yielding a better representation of the input features. Bi-LSTM is adopted instead of the other versions of LSTM to achieve the retention of information transfer between the context and the context. Finally, SoftMax function is combined with the CRF to produce the optimal label sequence. The pre-training model and the downstream task model are combined to generate a deep bidirectional language representation using ALBERT as a pre-training model [43]. Bi-LSTM can effectively use the bidirectional data features of the sequences. Owing to the existence of the CRF layer, the model can also use the constraint information between sentence-level tags, as shown in Fig. 6.

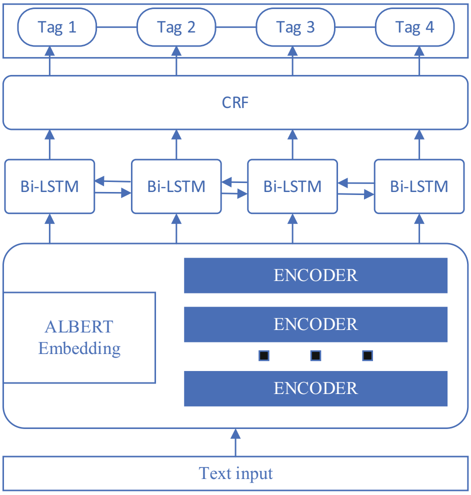

Fig. 6. Named Entity Recognition model. The samples used for training are annotated by the method based on dependency parsing in Sect. 3.2.

4 Experiment

In this section, experiments are conducted on the knowledge extraction model based on the dependency parsing method and the NER model based on an automatic labeling task. Unstructured texts from the IPO prospectuses and annual financial statements of listed companies that describe their production and operations were used as data. Among them, 28,000 samples were classified as training datasets, while the remaining 1,000 samples were manually annotated by stock analysis experts to create test datasets. The experimental results were used to compare model performance, including *Precision*, *Recall* and $F1 - Score$, which are commonly used evaluation metrics in knowledge extraction tasks.

Precision was used for the prediction results, and it indicates how many of the predicted positive samples were true positive. *Precision* is determined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

where *TP* (true positives) is the number of samples that the model predicts positive classes as positive, and *FP* (false positives) is the number of samples that the model predicts negative classes as positive.

Recall is for the original sample, which indicates how many positive examples in the sample were predicted correctly. *Recall* is determined as follows:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

where *FN* (false negatives) is the number of samples that the model predicts a positive class as negative. Because the p *recision* and *recall* indicators sometimes contradict each other, they must be considered comprehensively. The most used method is $F1 - Score$, which is calculated as follows:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

This study uses the Stanford CoreNLP + NLTK technologies [44] and the Baidu artificial intelligence platform [33] to achieve knowledge extraction based on a phrase structure tree and dependency parse tree. This study also proposes a joint extraction method that combines the two models, allowing for simultaneous consideration of the phrase structure of the sentence and dependencies between words. Table 2 shows the specific experimental results, the F1 score of the fusion model is higher than the individual models, which shows that after integrating the phrase structure and dependency parse, the ability to extract knowledge has further improved.

Table 2. Knowledge extraction based on dependency parsing.

	Precision	Recall	F1-socre
Phrase structure tree	0.6335	0.5605	0.5948
Dependency parse tree	0.7844	0.6559	0.7144
Phrase structure tree + Dependency parse tree	0.7917	0.7110	0.7492

The dataset obtained after completing the labeling task based on the dependency parsing module is used to train the NER model of ALBERT + Bi-LSTM + CRF. The parameters for training the entity recognition model: GELU was used as the loss function, the hidden layer size was 768, a 12-head attention mechanism was used, a 12-layer transformer encoding structure was used, which supported a maximum encoding length of 512 bytes, and the hidden layer of Bi-LSTM was 128 neurons. During the training process, the epoch was set to five, and the batch size was set to 128, the initial learning rate was set to 0.001, and the number of training rounds was automatically corrected.

The proposed dependency parsing-based knowledge extraction method can directly extract industrial chain entities from well-structured sentences. The entity extraction task for other less-regular sentences can be better accomplished by developing a NER model. Finally, this study used two models to jointly complete the entity extraction task.

Compared with English NER, Chinese NER is different since it usually involves word segmentation. Two Chinese NER mothods, FLAT [45] and UIE [46], are used for performance comparison. FLAT is Flat-LAttice Transformer, which leverages the span position encoding. UIE is Universal Information Extraction, which uses a unified text-to-structure generation framework. As shown in Table 3, in the entity extraction task of the financial industry chain, FLAT uses more word segmentation information, so it has a high recall score, while the precision descends heavily. Instead, UIE has a more smooth Precision and Recall. Compared with a single model, the fused model in this paper gets a higher F1-Score and significantly improved Recall, and improves generalization and knowledge extraction abilities.

Table 3. Named entity recognition based on deep learning.

	Precision	Recall	F1-socre
Phrase structure tree + Dependency parse tree	0.7917	0.7110	0.7492
NER (ALBERT + Bi-LSTM + CRF)	0.7565	0.6616	0.7059
FLAT	0.5937	0.8311	0.6826
UIE	0.6970	0.7252	0.7110
Phrase structure tree + Dependency parse tree + NER	0.7097	0.8076	0.7555

5 Conclusion

This study proposed a knowledge extraction method based on a dependency parsing that combined phrase structure trees and dependency parse trees to extract industry chain knowledge and perform automated labeling tasks. This study used the obtained dataset to train the NER model based on ALBERT + Bi-LSTM + CRF and fused it with the dependency parsing knowledge extraction model, which improved generalization ability and allowed for the automatic construction of an industry chain knowledge graph. This automated knowledge graph construction technology can assist in automatically tracking and studying associations between listed companies.

Acknowledgements. The authors would like to acknowledge the support of this research by National Key R&D Program of China (no. 2019YFF0301300).

References

1. Fama, E.F., French, K.R.: The cross-section of expected stock returns. J. Finance **47**(2), 427–465 (1992)
2. Lu, R., Jin, X., Zhang, S., Qiu, M., Wu, X.: A study on big knowledge and its engineering issues. IEEE Trans. Knowl. Data Eng. **31**(9), 1630–1644 (2018)
3. Qiu, M., Zhang, L., et al.: Security-aware optimization for ubiquitous computing systems with SEAT graph approach. J. Comp. Sys. Sci. **79**(5), 518–529 (2013)
4. Gai, K., Qiu, M., Thuraisingham, B., Tao, L.: Proactive attribute-based secure data schema for mobile cloud in financial industry. In: EEE 17^{th} HPCC, pp. 1332–1337 (2015)
5. Liu, Y., Zeng, Q., Ordieres Meré, J., Yang, H.: Anticipating stock market of the renowned companies: a knowledge graph approach. In: Complexity 2019 (2019)
6. Wang, J., Guo, Y., Wen, X., Wang, Z., Li, Z., Tang, M.: Improving graph-based label propagation algorithm with group partition for fraud detection. Appl. Intell. **50**(10), 3291–3300 (2020)
7. Cao, Z., Ni, L., Dai, L.: A review of knowledge graph-based question and answer system research and its application in chronic disease diagnosis. Acad. J. Comput. Inf. Sci. **4**(4), 1–11 (2021)
8. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)

9. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
10. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
11. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2008)
12. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigations **30**(1), 3–26 (2007)
13. De Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. Lrec **6**, 449–454 (2006)
14. Nivre, J.: Dependency grammar and dependency parsing. MSI Report **5133**(1959), 1–32 (2005)
15. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
16. Qiu, M., Xue, C., Shao, Z., Sha, E.: Energy minimization with soft real-time and DVS for uniprocessor and multiprocessor embedded systems. In: IEEE DATE Conference, pp. 1–6 (2007)
17. Qiu, M., Khisamutdinov, E., et al.: RNA nanotechnology for computer design and in vivo computation. Philos. Trans. R. Soc. A **371**(2000), 20120310 (2013)
18. Lu, Z., Wang, N., et al.: IoTDeM: an IoT big data-oriented MapReduce performance prediction extended model in multiple edge clouds. JPDC **118**, 316–327 (2018)
19. Liu, M., Zhang, S., et al.: H infinite state estimation for discrete-time chaotic systems based on a unified model. IEEE Trans. Syst. Man Cybern. (B) **42**(4), 1053–1063 (2012)
20. Qiu, L., Gai, K., Qiu, M.: Optimal big data sharing approach for tele-health in cloud computing. In: IEEE SmartCloud, pp. 184–189 (2016)
21. Qiu, M., Chen, Z., Niu, J., Zong, Z., Quan, G., Qin, X., Yang, L.T.: Data allocation for hybrid memory with genetic algorithm. IEEE Trans. Emerg. Top. Comput. **3**(4), 544–555 (2015)
22. Wu, G., Zhang, H., Qiu, M., et al.: A decentralized approach for mining event correlations in distributed system monitoring. JPDC **73**(3), 330–340 (2013)
23. Qiu, M., Cao, D., et al.: Data transfer minimization for financial derivative pricing using Monte Carlo simulation with GPU in 5G. Int. J. Comm. Sys. **29**(16), 2364–2374 (2016)
24. Qiu, M., Gai, K., Xiong, Z.: Privacy-preserving wireless communications using bipartite matching in social big data. FGCS **87**, 772–781 (2018)
25. Shao, Z., Xue, C., et al.: Security protection and checking for embedded system integration against buffer overflow attacks via hardware/software. IEEE TC **55**(4), 443–453 (2006)
26. Qiu, H., Qiu, M., Lu, Z.: Selective encryption on ECG data in body sensor network based on supervised machine learning. Inf. Fusion **55**, 59–67 (2020)
27. Li, C., Qiu, M.: Reinforcement Learning for Cyber-Physical Systems: with Cybersecurity Case Studies. Chapman and Hall/CRC (2019)
28. Li, Y., Song, Y., et al.: Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. IEEE TII **17**(4), 2833–2841 (2020)
29. Qiu, H., Zheng, Q., et al.: Topological graph convolutional network-based urban traffic flow and density prediction. In: IEEE Trans on ITS (2020)
30. Bollacker, K., Cook, R., Tufts, P.: Freebase: a shared database of structured general human knowledge. In: AAAI, vol. 7, pp. 1962–1963 (2007)
31. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
32. Mahdisoltani, F., Biega, J., Suchanek, F.: Yago3: a knowledge base from multilingual wikipedias. In: 7th Biennial Conference on Innovative Data Systems Research. CIDR Conference (2014)

33. Baidu Artificial Intelligence Platform. https://ai.baidu.com/. Accessed 12 Mar 2022
34. Luo, X., Liu, L., Yang, Y., Bo, L., et al.: AliCoCo: Alibaba e-commerce cognitive concept net. In: ACM SIGMOD Conference on Management of Data, pp. 313–327 (2020)
35. Linked Life Data homepage. http://linkedlifedata.com/. Accessed 12 Mar 2022
36. Yu, T., Li, J., Yu, Q., Tian, Y., et al.: Knowledge graph for TCM health preservation: design, construction, and applications. Artif. Intell. Med. **77**, 48–52 (2017)
37. Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: 43rd ACL, pp. 363–370 (2005)
38. Liu, F., Zhao, J., Lv, B., Xu, B., Yu, H.: Product named entity recognition based on hierarchical hidden Markov model. In: 4^{th} SIGHAN Workshop on Chinese Language Processing (2005)
39. McClosky, D., Surdeanu, M., Manning, C.D.: Event extraction as dependency parsing. In: 49th Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 1626–1635 (2011)
40. Zamiralov, A., Sohin, T., Butakov, N.: Knowledge graph mining for realty domain using dependency parsing and QAT models. Procedia Comp. Sci. **193**, 32–41 (2021)
41. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of NAACL-HLT, pp. 260–270 (2016)
42. Brachman, R.J., Schmolze, J. G.: An overview of the KL-ONE knowledge representation system. In: Readings in Artificial Intelligence and Databases, pp. 207–230 (1989)
43. Lan, Z., Chen, M., Goodman, S., et al.: Albert: a lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
44. Stanford-CoreNLP-API-in-NLTK. https://github.com/nltk/nltk/wiki/Stanford-CoreNLP-API-in-NLTK. Accessed 12 Mar 2022
45. Li, X., Yan, H., Qiu, X., Huang, X.: FLAT: Chinese NER using flat-lattice transformer. In: 58th Annual Meeting of the Association for Computational Linguistics, pp. 6836–6842 (2020)
46. Lu, Y., Liu, Q., Dai, D., Xiao, X., et al.: Unified structure generation for universal information extraction. In: 60th Meeting of the Association for Computational Linguistics, pp. 5755–5772 (2022)