A Tutorial and Survey on Fault Knowledge Graph

XiuQing Wang[1] and ShunKun Yang[2(✉)]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] School of Reliability and Systems Engineering, Beihang University, Beijing 100083, China
ysk@buaa.edu.cn

Abstract. Knowledge Graph (KG) is a graph-based data structure that can display the relationship between a large number of semi-structured and unstructured data, and can efficiently and intelligently search for information that users need. KG has been widely used for many fields including finance, medical care, biological, education, journalism, smart search and other industries. With the increase in the application of Knowledge Graphs (KGs) in the field of failure, such as mechanical engineering, trains, power grids, equipment failures, etc. However, the summary of the system of fault KGs is relatively small. Therefore, this article provides a comprehensive tutorial and survey about the recent advances toward the construction of fault KG. Specifically, it will provide an overview of the fault KG and summarize the key techniques for building a KG to guide the construction of the KG in the fault domain. What's more, it introduces some of the open source tools that can be used to build a KG process, enabling researchers and practitioners to quickly get started in this field. In addition, the article discusses the application of fault KG and the difficulties and challenges in constructing fault KG. Finally, the article looks forward to the future development of KG.

Keywords: Fault Knowledge Graph · Key technologies · Tools · Applications

1 Introduction

The Fault Knowledge Graph (KFG) is based on the KG in the field of faults. Now it has been applied to finance [1, 2], medical [3–5], biological [6, 7], agriculture [8, 9], journalism, education, question answering [10–12], and other industries. Because KG can display the relationship between various data types and efficient and intelligent search information. However, fault areas such as mechanical engineering [13], trains [14], power grids [15], power equipment [16], etc. All of this raised the need for building KG, but there are not many KGs that actually build success. Each domain has the same place and different places in building KGs. Therefore, this paper investigates the architecture and key technologies of KGs in other fields, and also investigates the construction process and key technologies of KFG. At the same time, KGs in other fields can be used as a guide to complete the construction of KFG. The appearance of the FKG can well analyze the relationship between various faults, achieve prediction, and promote development.

H. Ning (Ed.): CyberDI 2019/CyberLife 2019, CCIS 1138, pp. 256–271, 2019.
https://doi.org/10.1007/978-981-15-1925-3_19

This paper provides an overview of the FKG, summarizes its build process and key technologies, and summarizes the open source tools and applications that may be used.

The structure of this paper is as follows:

- Section 2 outlines the concept of the FKG and the reasons for its development, some key technologies, as well as the application of faults;
- Section 3 provides the architecture of the general KG and some key technologies, as well as open source tools that may be used to guide the construction of the FKG. These include steps such as data acquisition, information extraction, knowledge fusion, knowledge processing, and knowledge storage;
- Section 4 analyzes the application of the FKG and the challenges and problems to be solved in the establishment process;
- Section 5 summarizes the FKG, and predicts the future development of FKG.

2 Overview

In this section, we will describe the origins of the concept of the FKG and some of the reasons that motivated its development. We will also briefly present scope of some of its application fault devices.

The KG [17] was originally proposed by the Google knowledge graph project to enhance the google search engine and enhance the user experience. Later, the KG was applied to many fields with its advantages. However, the application of the KG to the field of failure is still relatively few. Still, there is still a certain demand for constructing KG in the field of failure. For example, [13–16] all addressed the needs of the field of failure. Therefore, in 2107, Yuan-cheng et al. provided a definition of a formal Fault Knowledge Graph [13] based on the collection of a large number of mechanical engineering fault data. The specific applications and methods used are listed below in tabular form (Table 1).

Table 1. Fault domain application and construction technology

Fault area	Fault application	Technology
Engineering machinery	Fault knowledge question-answering and troubleshooting assistance	Data-driven iterative
Train	Anomaly detection	Building consistency matrices
Power grids	Achieve automatic/semi-automatic disposal of faults	“hidden Markov model” + “punctuation-based segmentation”
Power equipment	Fault diagnosis	Entity relationship extraction + RDF

As can be seen from the above chart, the types of faults includes engineering machinery, train, power grids, power equipment and so on. In [13], Yuan-cheng LU et al. provided construction process of FKG and proposed a data-driven KG iterative automatic construction method. In [16], the author used entity extraction and relation extraction techniques and combined the relevant data of multi-source heterogeneous power equipment to construct a power equipment KG, and to improve the efficiency of power equipment management. In [15], the author used information extraction.

3 Key Technologies

The public construction process of the knowledge graph includes Data Acquisition, Knowledge Acquisition, Knowledge Fusion and Knowledge Processing [18], as shown in below (Fig. 1).

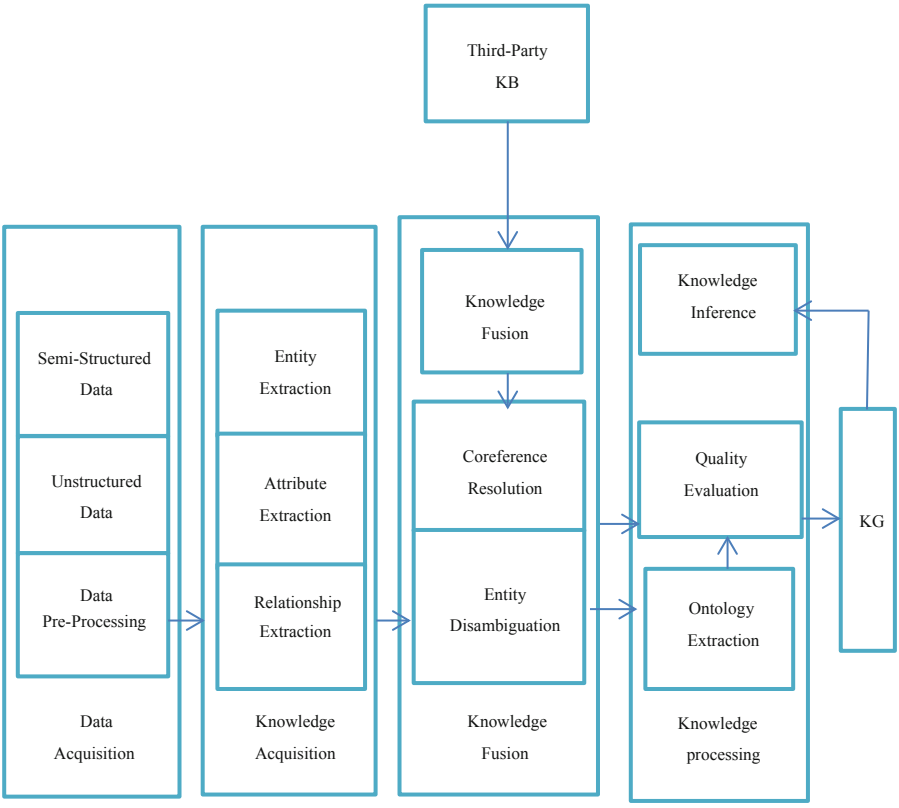

Fig. 1. The knowledge graph construction process

3.1 Data Acquisition

The first problem facing the construction of a KG is the source of the data. There are three types of data sources, which are structured, semi-structured, and unstructured. Structured data is stored in the database and tables [19]. Semi-structured data has a specific structure but is not very strict, such as xml data. Finally, unstructured data has no predefined data model information, such as publications, web pages or social media [20]. Most data sources are obtained by crawling unstructured data (such as an encyclopedia) as a data source.

As mentioned earlier, data sources can get data from web pages or extract them from databases. Then the technology needed to get data from the Internet, namely web crawler technology [21]. It described Mercator, a web crawler that is fully written in Java and can be extended. In [22], the author pointed out that web crawlers are a recursive process, and the user has to add restrictions, such as specifying the maximum number of tags or documents to retrieve, and the time limit. To improve the quality of the data, users can limit the crawl to a specific domain or file format and apply a blacklist of unwanted URLs/domains.

In [23], it provided the technology of Micro-blog Website, including depth control, breadth control and URL controller. URL controller is important. Because we have to consider whether this URL is suitable for crawling is very important for the results obtained, such as too many nodes or poor data quality, etc.

3.2 Information Extraction

KG consists of nodes and their relationships. The nodes include entities, concepts and literals. Entities are real-world individuals. Concepts represent a set of individuals with the same characteristics. Literals are strings indicated specific values of some relations. Information extraction is a kind of automatic entity extraction from semi-structured data and unstructured data [24]. Information extraction includes entity extraction, relationship extraction and attribute extraction [25].

(1) Named Entity Recognition

Named Entity Recognition was presented at the 6th MUC Conference (MUC-6, the Sixth Message Understanding Conferences) in November 1995. The core idea is to identify and classify the proper nouns needed for a given text. NER's method research is divided into three main categories: firstly, dictionary-based and rule-based methods; secondly, traditional machine learning methods, the main conditional random field (CRF) and support vector machine (SVM), Long Short Term Memory Network (LSTM), Bi-directional LSTM (BILSTM), Part-Aware LSTM (PLSTM); thirdly, Deep learning methods such as Convolutional Neural Network (CNN) [73] and Recurrent Neural Network (RNN) [74].

A rule- based approach is to manually build rules and then match the strings of those rules in the text. The most representative of these is the DLCoTrain method proposed by Collins [35]. They proposed two algorithms. One is boosting-like framework. Another described an algorithm that directly optimizes this function. Similar to this, there was also a way to automatically generate rules through Bootstrapping [36].

The machine learning based approach is implemented by a categorical approach. Mainly through the method of serialization annotation, its main models include SVM (Support Vector Machine) [37], ME (Maximum Entropy) [38], HMM (Hidden Markov Model) [39, 40], CRF (Conditional Random Field) [41].

In [85], Thanh Hai Dang et al. proposed a novel named entity model. The model uses conditional random fields and bidirectional long-term short-term memory, and the experiment works well. In [86], Hui Chen et al. proposed a simple but effective CNN-based network, the Gate Correlation Network (GRN). This network is better at capturing context information than CNN. At the same time, Parallel Recurrent Neural Networks and CNN-RNN have promoted the development of NER

(2) Relationship Extraction

The entity relationship extraction task was first proposed at the 7th Message Understanding Conference (MUC) in 1998 [60]. Many research methods for entity relationship extraction include pattern matching [61], dictionary-driven [62] and machine learning methods. At present, the study of entity relationship extraction mainly uses machine learning or deep learning. The machine learning algorithm considers the relationship extraction as a classification problem, constructs the classifier when training the corpus, and finally applies it to the category judgment of the entity relationship.

Machine learning-based entity relationship extraction methods include supervised methods, unsupervised methods, weakly supervised methods, remotely supervised methods, and open domain-oriented methods. There was a supervised method in [63] that could obtain better performance by determining the relationship of entities in a sentence by a given entity and a sentence containing a pair of entities. The disadvantage of this method is that it requires a lot of manual labeling when training data. The unsupervised learning [64] method did not require manual labeling of corpus, but the performance of relational extraction was poor. The method of weak supervised learning was bootstrapping at the earliest [65]. This algorithm is simple and easy to operate, but it uses a lot of statistical assumptions, so it is assumed whether the accuracy of sampling is established. Later, remote supervision appeared, using the alignment of the open knowledge base to automatically mark the corpus, reducing the dependence on manual annotation data, and enhancing cross-domain adaptability. However, its shortcoming is that it will bring a lot of noise to the corpus, so solving the noise has become a problem that scholars are concerned about. If there is a relationship between the two entities in the knowledge base, then the sentence containing the two entities has more or less the relationship. Open-domain-based relationship extraction is not limited to relationship categories and text classification, and does not need to be labeled corpus, which is suitable for processing large-scale network data, but the extracted results require a lot of processing, and there is no objective evaluation standard.

The above method requires the use of the NLP tool, but the NLP itself will have system errors, so when the algorithm is used later, the performance of the algorithm will be degraded. The deep learning method is applied to the relationship extraction, and the advantages of feature extraction and automatic learning are taken. SemEval-2010 task 8 [66] was used as the test standard. Deep learning based algorithms include recurrent neural networks and convolutional neural networks. The RNN model [67] set vectors and matrices, learned the meanings of propositional logic and natural language operators,

but relied on traditional NLP tools to focus on semantic learning, thus reintroducing tool noise. Therefore, Zeng et al. [68] used convolutional neural networks to extract the hierarchical features of sentences and vocabulary for relational extraction, reducing the pre-marking processing of input materials. Then Nguyen et al. [69] added convolution kernels of different sizes to the convolutional layer as filters to extract more features and add position vectors. Lin et al. [70] introduced PCNN (piecewise CNN), which pooled the feature map into three segments for the two physical locations of the pooled layer of the traditional convolutional neural network, while using the attention mechanism to establish the sentence level. Selective attention to the neural model mitigated the problem of mislabeling.

Remote supervisory relationship extraction extends relation extraction to a very large corpus. Neural relation extraction has made great progress in modeling sentences in low-dimensional space, but lacks consideration of simulated entities. Firstly, the context coding on the dependency tree is embedded as a tree-GRU-based sentence entity, and then the relationship between the sentence and the sentence is used to obtain the entity embedding of all sentence sets. Finally, the sentence embedding and the entity embedding are combined to classify the relationship. Better performance extraction [78]. In 2019, XIAOYU GUO proposed a combination of CNN and RNN for relationship classification based on single attention. In 2019, PLSTM-CNN [81] based on remote supervision was used to implement relationship extraction. At the same time, Multi-Gram CNN-Based Self-Attention [82] is also based on remote supervision for relationship classification. In the same year, attention-based Att-RCNN [83] was also used to achieve relationship classification. For the classification of semantic relations, SHEN Y proposed a neural network based ED-LSTM [84] algorithm.

However, the above entity relationship extraction is to extract the entity and the relationship separately, which will generate redundant information, so the joint extraction is proposed. Extract the entities and their relationship types in the same model, realize parameter sharing and synchronization optimization, and reduce the possibility of extracting errors before extraction. Zheng [71] et al. used the underlying expression of shared neural network for joint learning. Li et al. [72] proposed a joint structured extraction method for incremental cluster search algorithm and a constraint method using global features.

3.3 Knowledge Fusion

Information extraction obtains attribute information of entities, relationships, and entities from raw unstructured and semi-structured data. But this information contains a lot of redundancy and error messages. Therefore, it needs to clean up the extracted knowledge items. Knowledge fusion aims to solve errors in network data and errors in knowledge extraction. Knowledge fusion mainly includes entity links and knowledge integration [26].

(1) Entity Linking

Entity link is an operation that connects an entity object extracted from text to the corresponding item in external databases or knowledge bases. Wikipedia is often used as one of the knowledge bases for entity linking. The entity link was to assign the reference item

to the correct entity object through disambiguation processing and co-finding digestion [27]. Tomoaki Urata proposed a disambiguation method for Wikipedia's Weibo entity links [28]. Entity disambiguation has four steps. The first step is to obtain the candidate entity. This requires obtaining a Wikipedia article for the candidate entity from the Wikipedia disambiguation page and preparing for the next entity disambiguation. The second step is to compare the nearest entity in the target entity with the Wikipedia article for each candidate entity and match the entity in Wikipedia. The third step is to calculate the similarity. They use word2vec to get the relevant entity of the nearest entity. The final step is to multiply the results of the second and third steps and extract the Wikipedia article with the highest score as the correct entity.

However, there is a lack of relationship between some entities and entities, so link prediction is required. In some algorithms, Daniel Neil proposed a new model, Graph Convolutional Neural Networks (GCNNs) [77], which improves performance on clear datasets and accommodates noise in KGs pervasive issue in real world applications. What's more, this model is more interpretable. Because it allowed measurement the effect of a particular edge on prediction by adjusting the link weight or completely removing the edge. In 2019, Binling Nie combined latent feature models and graph feature models to propose a model of Text-enhanced KG Embedding (TKGE), which can perform inference over entities, relations and text [80].

(2) Knowledge Integration

Knowledge Integration is to combine knowledge from multiple, distributed, heterogeneous knowledge sources [18]. In the previous entity link, it linked data extracted from semi-structured or unstructured data, but there were structured data packages for external or relational databases. The process of knowledge integration involves concept matching, entity matching, the evaluation of knowledge and the resolution of conflicts [29].

3.4 Knowledge Processing

(1) Ontology Construction

Ontology is a formal language used to construct ontology. It is a descriptive language and a metaphysical form of framework language. Ontologies include individuals, classes, attributes, function terms, constraints, rules, axioms, events, and so on. In the structure of ontology, it contains the relationships between things, the nature of things, the constraints of things and so on. In [30], the author proposed an automatic ontology extension method based on supervised learning and text clustering. This method used the k-means algorithm to separate domain knowledge and guided the creation of the Naïve Bayes classifier training set. First, the candidate set words will be added to the target ontology. At the same time, the noise words will be added to the stop word dictionary, which can automatically expand the ontology. The experimental results show that the expansion effect is very good.

(2) Knowledge Inference

After completing the ontology construction step, the relationship between many nodes and nodes of the KG is still vacant. Therefore, we need to fill in these relationships

through knowledge reasoning to better improve the KG. In [31], Lucas Fonseca Navarro et al. proposed the Graph Rule Learner (GRL), a method for extracting inference rules from the ontology knowledge base graphed to graphs, and explored the combination of link prediction indicators. The input to GRL is a ontology graph, and the output is a list of induced inference rules. GRL uses the link prediction metric of the extra neighbor [32] to rank the possible rules, and it's scalable, using a structure called Graph DB-Tree [33] to store the graphical representation on disk.

In [34], the author proposed a unified ontology reasoning framework to incorporate co-occurrence and subclass relationships, and this framework can also automatically build ontology. The article pointed out that subclass relationships could be obtained from WordNet. It covers almost all common nouns and can find the relationship between words and words. Then get the co-occurrence relationship from the training set. This framework effectively combines the concept of reasoning with superior performance and other methods.

In 2013, BORDES proposed to encode the triplet into a low-dimensional distributed vector, the TransE model. TransE [42] is a translation model, which considers the relationship between the three heads (head, relation, and tail) as a translation from head to tail. By constantly adjusting the relationship between head, relation, and tail, h + r is equal to t as much as possible. But TransE is dealing with the properties of some relational graphing, such as reflexivity. So, in 2014, ZhenWang1, Jianwen Zhang and others proposed TransH [43] model. TransH models the relationship as a relational hyperplane along with the translation operations on it, which can preserve some graphing properties such as reflexivity, one-to-many, many-to-one, many-to-many, etc. when sneaking, and in efficiency It is almost the same as TransE. TransH has a predictive accuracy comparable to that of TransE. Both TransE and TransH assume that entities and relationships are in the same space, but in fact one entity has multiple attributes. Different relationships focus on different attributes of the entity. In other words, some similar entities should be close to each other in space, not similar in space. The middle should be away from each other. So in 2015, Yankai Lin et al. proposed the TransR model [44], Embedding entities and relationships into different spaces and implementing translations in the corresponding spatial relationships. The disadvantage of the TransR model is that the parameters are too many and too complicated. So in the same year Guoliang Ji, et al. proposed the TransD model [45], which not only considers the diversity of relationships, but also considers the entity. The main idea is to use two vectors to represent entities and relationships, one (h, r, t) for entities or relationships, and one for dynamically constructing graphing matrices. The advantage is that there are fewer parameters and no multiplication of matrix vectors, and it is applied to large-scale graphics. The previous representations oversimplified the loss metric and did not have enough power to simulate complex entities and relationships in the knowledge base. Han Xiao proposed the TransA model [46], which replaced the metric function, used the metrics to learn the sneak method, and treated each dimension in the vector differently, improving the representation ability. In 2016, in order to solve the problem of heterogeneity and imbalance, the connection relationship of the entity is complicated and simple, and the number of head and tail of many connection relationships is not equal. Guoliang Ji proposed the TranSpare model [47]. The core idea is that the transfer matrix is replaced by an adaptive matrix, and the sparsity of the adaptive

matrix is determined by the number of pairs of entities, thus preventing under-fitting of complex relationships or over-simplification of simple relationships. Then, in order to solve the problem of multi-relational semantics, the TransG model emerged, and a Bayesian nonparametric Gaussian mixture model was used to generate multiple translation parts for a relationship. This discovered the potential semantic relationship and embeds a triple with the mixture-specific component vector [48]. Finally extended to the KG2E model [49]. The new idea of using the covariance of multidimensional Gaussian distributions to represent the uncertainty of entities and relationships can properly represent its certainty. The above models are all extensions to the TransE model.

3.5 Knowledge Storage

After obtaining the relationship between the entity and the entity, the data is used to form a KG. Usually, the ontology acts as a carrier for KG. Ontology language based on Web ontology includes Resource Description Framework (RDF) [57] and Web Ontology Language (OWL) [58]. RDF can be used to describe resources on the network and their relationship to each other, including resources, attributes, and relationships, in the form of triples. Later, RDFS appeared which mainly added some vocabulary to expand the ability of RDF. OWL is still an extension of RDF's vocabulary. It has strong knowledge representation and knowledge reasoning ability, and adds representations of classes, vocabulary and relationships. So OWL is more expressive and more powerful than RDF.

The other is storage based on a graph database, such as Neo4j [59]. The Neo4j graphics database is based on attribute graphs and focuses on queries and searches. The disadvantage of Neo4j is that it does not support distributed. Although OrientDB and JanusGraph (formerly Titan) support distributed, they are immature. So choosing Neo4j to store KG is better. And it separates the relationship between nodes and nodes. The last set of triplet row vectors is stored in csv format and then imported into the Neo4j database (graphics database), which is also convenient for future queries and optimizations.

4 Tools and Platforms

For the processing of data, in addition to some algorithms, it is also possible to implement information extraction through some open source natural language tool processing packages. Below we introduce the open source toolkits written by Python and Java, and analyze their functions. And the language that is suitable for processing, Chinese or English.

Gensim [50] is an open source Python toolkit that handles raw unstructured text and unsupervised learning of topic vector representations of text. It supports a variety of topic model algorithms including TF-IDF, LSA, LDA, and word2vec, supports streaming training, and provides API interfaces for common tasks such as similarity calculation and information retrieval.

Stanford CoreNLP [51] is an open source toolkit for relation extraction based on Java. Its methods such as supervising, remote monitoring, and neural network can realize the analysis of natural language texts, including part of speech restoration, part-of-speech

tagging, named entity tagging, co-finger decomposition, syntax analysis and dependency analysis.

NLTK [52] is a third-party toolkit based on Python. NLTK is suitable for processing English, but there are some restrictions on handling Chinese. NLTK does not have a Chinese stop word due to the lack of a Chinese corpus. And the stop words of Chinese text cannot be filtered, so Chinese text cannot be segmented. Therefore, NLTK is not suitable for processing Chinese text.

FudanNLP [53] is a Java-based Chinese open source toolkit that contains machine learning algorithms and data sets. It enables information retrieval such as text categorization and news clustering. For Chinese processing, it includes Chinese word segmentation, part of speech tagging, entity name recognition, keyword extraction, dependency syntax analysis and time phrase recognition.

Stanford University has developed a toolkit that supports Chinese processing, namely deepdive [54]. It is used for knowledge extraction, the extraction of triples. It extracts structured relational data from unstructured text through weakly supervised learning. This project has modified the model package for natural language processing to support Chinese and provide Chinese tutorial.

SOFIE [55] is an automated ontology extension system developed by the max planck institute. It can extract ontology-based events from text, implement ontology links, and perform logical reasoning for disambiguation.

OpenCCG [56] is a Java-based open source natural language processing library that implements text grammar based on Mark Steedman's combination of grammatical forms, including syntax analysis and dependency analysis.

OLLIE is a three-tuple extraction component of KnowItAll, a knowledge base developed by the University of Washington. It enables the extraction of relationships based on grammar-dependent trees and can extract relationships over long distances. Reverb is also an open ternary extraction tool developed by the University of Washington. It can extract triples of entity relationships from English sentences. The advantage is that it does not need to specify relationships in advance, and supports information extraction on a network-wide scale. This reduces a lot of manual intervention.

ICTCLAS (NLPIR) is a Chinese language open source tool based on multiple languages, such as Java, C++, C, C#. It is mainly used to deal with Chinese, such as Chinese word segmentation, named entity recognition, part-of-speech tagging, custom user dictionary, new word micro-blog word segmentation, new word discovery and keyword extraction. Visual interface operation and API call provide users with Great convenience.

In summary, most open source toolkits for Chinese natural language processing are implemented in the Java language, while most of the self-language toolkits that handle English are written in the Python language. NLTK is suitable for English processing. FudanNLP, deepdive, ICTCLAS are suitable for Chinese processing. Gensim processes text based on an unsupervised learning algorithm. Stanford CoreNLP processes text based on remote monitoring and neural networks.

5 Conclusion and Futures Challenges

KG is an important branch of artificial intelligence. It simulates the way people think, and carries out efficient knowledge management and knowledge acquisition on data.

This article aims to use crawling technology to crawl some fault data and then generate KG through natural language processing. This allows a clearer understanding of the relationship between faults and faults, and analysis and prediction to prevent failures. Natural language processing mainly uses the entity relationship extraction technology. In the previous introduction, there are many algorithms and tools. The algorithms generally have unsupervised methods, supervised methods, remote monitoring methods and bootstrapping methods. Entity and relationship extraction have first entity extraction and relationship extraction and joint extraction of entities and relationships. Nowadays, although there are more separate extraction algorithms, its extraction is limited by entity extraction. Joint extraction is better than separate extraction, but there are no particularly mature algorithms. The specific algorithms of entity relationship extraction mainly include entity relationship extraction based on deep convolutional neural network, circular convolutional neural Venaero algorithm, CRF-based named entity, semi-supervised learning method combined with word rules and SVM model and open Chinese. A major problem in the extraction of entity relationships is the need for a large number of manual annotations. However, many Chinese entities can also be used to extract relational tools such as NTLK, Sanford CoreNLP, OLLIE, SOFIE, and so on. However, NTLK is not suitable for the processing of Chinese texts. It lacks Chinese corpus and stop words. Most of the tools developed are not very suitable for processing Chinese texts.

The important point is that most of the projects related to KG involve English. However, Chinese is a different language. It is not feasible to convert English KGs into Chinese. Thus, the construction of Chinese KG is very significant. The main difficulty lies in the following three points: (I) quality of data sources, (II) taxonomy derivation and (III) knowledge harvesting [75].

The construction of previous KG was constructed with node-relationship-nodes, but we still lack the unified definition and standard expression of KG. Therefore, in order to make the KG clearer, Yucong Duan clarified the structure of the KG from the aspects of data, information, knowledge and wisdom, and suggest to specify the KG in a gradual manner, including data graphs, infographics, KG and wisdom graph [76].

References

1. Fu, X., Ren, X., Mengshoel, O.J., et al.: Stochastic optimization for market return prediction using financial knowledge graph. In: 2018 IEEE International Conference on Big Knowledge (ICBK). IEEE Computer Society (2018)
2. Liu, Y., Zeng, Q., Yang, H., Carrio, A.: Stock price movement prediction from financial news with deep learning and knowledge graph embedding. In: Yoshida, K., Lee, M. (eds.) PKAW 2018. LNCS (LNAI), vol. 11016, pp. 102–113. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97289-3_8
3. Shen, Y., Yuan, K., Dai, J., et al.: KGDDS: a System for Drug-Drug Similarity Measure in therapeutic substitution based on knowledge graph curation. J. Med. Syst. **43**(4), 43 (2019)
4. Shengtian, S., Zhihao, Y., Lei, W., et al.: SemaTyP: a knowledge graph based literature mining method for drug discovery. BMC Bioinform. **19**(1), 193 (2018)
5. Sang, S., Yang, Z., Liu, X., et al.: GrEDeL: a knowledge graph embedding based method for drug discovery from biomedical literature. IEEE Access **7**, 8404–8415 (2018)
6. Ali, M., Hoyt, C.T., Domingo-Fernandez, D., et al.: BioKEEN: a library for learning and evaluating biological knowledge graph embeddings. BioRxiv, 475202 (2018)

7. Alshahrani, M., Khan, M.A., Maddouri, O., et al.: Neuro-symbolic representation learning on biological knowledge graphs. Bioinformatics **33**(17), 2723–2730 (2017)
8. Xiaoxue, L., Xuesong, B., Longhe, W., et al.: Review and trend analysis of knowledge graphs for crop pest and diseases. IEEE Access **7**, 62251–62264 (2019)
9. Chenglin, Q., Qing, S., Pengzhou, Z., et al.: Cn-makg: China meteorology and agriculture knowledge graph construction based on semi-structured data. In: Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), F, 2018. IEEE (2018)
10. Sawant, U., Garg, S., Chakrabarti, S., et al.: Neural architecture for question answering using a knowledge graph and web corpus. Inf. Retrieval J. **22**(3–4), 324–349 (2019)
11. Shin, S., Jin, X., Jung, J., et al.: Predicate constraints based question answering over knowledge graph. Inf. Process. Manage. **56**(3), 445–462 (2019)
12. Zheng, W., Cheng, H., Yu, J.X., et al.: Interactive natural language question answering over knowledge graphs. Inf. Sci. **481**, 141–159 (2019)
13. Lu, Y.-C., Wen, Y.-J., Xuan, L., et al.: Exploration of the construction and application of knowledge graph in equipment failure. DEStech Transactions on Computer Science and Engineering, (smce) (2017)
14. Qin, Z., Cen, C., Jie, W., et al.: Knowledge-graph based multi-target deep-learning models for train anomaly detection. In: Proceedings of the 2018 International Conference on Intelligent Rail Transportation (ICIRT). IEEE (2018)
15. Shan, X., Zhu, B., Wang, B., et al.: Research on deep learning based dispatching fault disposal robot technology. In: Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2). IEEE (2018)
16. Tang, Y., Liu, T., Liu, G., et al.: Enhancement of power equipment management using knowledge graph. arXiv preprint arXiv:190412242 (2019)
17. Steiner, T., Verborgh, R., Troncy, R., et al.: Adding realtime coverage to the google knowledge graph. In: Proceedings of the 11th International Semantic Web Conference (ISWC 2012). Citeseer (2012)
18. Zheng, M., Ma, Y., Zheng, A., et al.: Constructing method of public opinion knowledge graph with online news comments. In: Proceedings of the 2018 International Conference on Robots & Intelligent System (ICRIS). IEEE (2018)
19. Choudhury, S., Agarwal, K., Purohit, S., et al.: Nous: construction and querying of dynamic knowledge graphs. In: Proceedings of the 2017 IEEE 33rd International Conference on Data Engineering (ICDE). IEEE (2017)
20. Zheng, M., Ma, Y., Zheng, A., et al.: Constructing method of public opinion knowledge graph with online news comments. In: Proceedings of the 2018 International Conference on Robots & Intelligent System (ICRIS). IEEE (2018)
21. Heydon, A., Najork, M.: Mercator: a scalable, extensible web crawler. World Wide Web **2**(4), 219–229 (1999)
22. De Groc, C.: Babouk: focused web crawling for corpus compilation and automatic terminology extraction. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. IEEE (2011)
23. Xia, J., Wan, W., Liu, R., et al.: Distributed web crawling: a framework for crawling of micro-blog data (2015)
24. Cowie, J., Wilks, Y.: Information extraction. Handbook Nat. Lang. Process. **56**, 57 (2000)
25. Lian, H., Qin, Z., He, T., et al.: Knowledge graph construction based on judicial data with social media. In: Proceedings of the 2017 14th Web Information Systems and Applications Conference (WISA). IEEE (2017)
26. Wang, X., Ma, C., Liu, P., et al.: A potential solution for intelligent energy management-knowledge graph. In: Proceedings of the 2018 IEEE International Conference on Energy Internet (ICEI). IEEE (2018)

27. Li, Y., Wang, C., Han, F., et al. Mining evidences for named entity disambiguation. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2013)
28. Urata, T., Maeda, A.: An entity disambiguation approach based on wikipedia for entity linking in microblogs. In: Proceedings of the 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI). IEEE (2017)
29. Wang, X., Ma, C., Liu, P., et al.: A potential solution for intelligent energy management-knowledge graph. In: Proceedings of the 2018 IEEE International Conference on Energy Internet (ICEI). IEEE (2018)
30. Song, Q., Liu, J., Wang, X., et al.: A novel automatic ontology construction method based on web data. In: Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE (2014)
31. Navarro, L.F., Hruschka, E.R., Appel, A.P.: Finding inference rules using graph mining in ontological knowledge bases. In: Proceedings of the 2016 5th Brazilian Conference on Intelligent Systems (BRACIS). IEEE (2016)
32. Appel, A.P., Junior, E.R.H.: Prophet–a link-predictor to learn new rules on NELL. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops. IEEE (2011)
33. Navarro, L.F., Appel, A.P., Junior, E.R.H.: GraphDB – storing large graphs on secondary memory. In: Catania, B., et al. (eds.) New Trends in Databases and Information Systems. AISC, vol. 241, pp. 177–186. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-01863-8_20
34. Tsai, S.-F., Tang, H., Tang, F., et al.: Ontological inference framework with joint ontology construction and learning for image understanding. In: Proceedings of the 2012 IEEE International Conference on Multimedia and Expo. IEEE (2012)
35. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 100–110 (1999)
36. Cucerzan, S., Yarowsky, D.: Language independent named entity recognition combining morphological and contextual evidence. In: Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC, pp. 90–99 (1999)
37. Isozaki, H., Kazawa, H.:[Association for Computational Linguistics the 19th international conference - Taipei, Taiwan (2002.08.24–2002.09.01)] Proceedings of the 19th international conference on Computational linguistics, - - Efficient support vector classifiers for named entity recognition[In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, pp. 1–7 (2002)
38. Borthwick, A.E.: A Maximum Entropy Approach to Named Entity Recognition. New York University, New York (1999)
39. Bikel, D.M., Miller, S., Schwartz, R., et al.: Nymble: a High-Performance Learning Namefinder. Anlp 94–201 (1998)
40. Bikel, D.M.: An algorithm that learns what's in a name. Machine Learning 34 (1999)
41. Mccallum, A., Li, W.: [Association for Computational Linguistics the seventh conference - Edmonton, Canada (2003.05.31-.)] Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, - - Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons, vol. 4, pp. 188–191 (2003)
42. Bordes, A., Usunier, N., Garcia-Duran, A., et al.: Translating embeddings for modeling multi-relational data. In: Proceedings of the Advances in Neural Information Processing Systems (2013)

43. Wang, Z., Zhang, J., Feng, J., et al.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
44. Lin, Y., Liu, Z., Sun, M., et al.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
45. Ji, G., He, S., Xu, L., et al.: Knowledge graph embedding via dynamic mapping matrix. 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) (2015)
46. Xiao, H., Huang, M., Hao, Y., et al.: TransA: An adaptive approach for knowledge graph embedding. arXiv preprint arXiv:150905490 (2015)
47. Ji, G., Liu, K., He S., et al.: Knowledge graph completion with adaptive sparse transfer matrix. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (2016)
48. He, S., Liu, K., Ji, G., et al.: Learning to represent knowledge graphs with gaussian embedding. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM (2015)
49. Xiao, H., Huang, M., Hao, Y., et al.: TransG: a generative mixture model for knowledge graph embedding. arXiv preprint arXiv:150905488 (2015)
50. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer (2010)
51. Manning, C., Surdeanu, M., Bauer, J., et al.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2014)
52. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media Inc., Beijing (2009)
53. Qiu, X., Zhang, Q., Huang, X.: Fudannlp: a toolkit for chinese natural language processing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2013)
54. Zhang, C.: DeepDive: A Data Management System for Automatic Knowledge Base Construction. University of Wisconsin-Madison, Madison (2015)
55. Suchanek, F.M., Sozio, M., Weikum, G.: SOFIE: a self-organizing framework for information extraction. In: Proceedings of the 18th International Conference on World wide web. ACM (2009)
56. Baldridge, J., Chatterjee, S., Palmer, A., et al.: DotCCG and VisCCG: Wiki and programming paradigms for improved grammar engineering with OpenCCG; proceedings of the CSLI Studies in Computational Linguistics Online. Citeseer (2007)
57. Miller, E.: An Introduction to the Resource Description Framework. Bull. Am. Soc. Inf. Sci. Technol. **25**(1), 15–19 (1998)
58. Bechhofer, S.: OWL: web ontology language. Encyclopedia Inf. Sci. Technol. Second Ed. **63**(45), 990–996 (2004)
59. Partner, J., Vukotic, A., Watt, N.: Neo4j in Action. Pearson Schweiz Ag (2014)
60. Chinchor, N., Marsh, E.: Muc-7 information extraction task definition. In: Proceeding of the Seventh Message Understanding Conference (MUC-7), Appendices (1998)
61. Vilain, M., Burger, J., Aberdeen, J.: Proceedings of the 6th Conference on Message Understanding (MUC-6) (1995)
62. Brants, T.: Proceedings of the Sixth Conference on Applied Natural Language Processing (2000)
63. Kambhatla, N.: Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions (2004)

64. Gonzalez, E., Turmo, J.: Unsupervised relation extraction by massive clustering. In: Proceedings of the 2009 Ninth IEEE International Conference on Data Mining. IEEE (2009)
65. Liu, X., Yu, N.: Multi-type web relation extraction based on bootstrapping. In: proceedings of the 2010 WASE International Conference on Information Engineering. IEEE (2010)
66. Hendrickx, I., Kim, S.N., Kozareva, Z., et al.: Semeval-2010 task 8: multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. Association for Computational Linguistics (2009)
67. Socher, R., Huval, B., Manning, C.D., et al.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics (2012)
68. Zeng, D., Liu, K., Lai, S., et al.: Relation classification via convolutional deep neural network (2014)
69. Nguyen, T.H., Grishman, R.: Relation extraction: perspective from convolutional neural networks. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing (2015)
70. Lin, Y., Shen, S., Liu, Z., et al.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2016)
71. Zheng, S., Hao, Y., Lu, D., et al.: Joint entity and relation extraction based on a hybrid neural network. Neurocomputing **257**, 59–66 (2017)
72. Li, Q., Ji, H.: Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers (2014)
73. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: proceedings of the Advances in Neural Information Processing Systems (2012)
74. Goller, C., Kuchler, A.: Learning task-dependent distributed representations by backpropagation through structure. In: Proceedings of International Conference on Neural Networks (ICNN 1996). IEEE (1996)
75. Wang, C., Gao, M., He, X., et al.: Challenges in chinese knowledge graph construction. In: Proceedings of the 2015 31st IEEE International Conference on Data Engineering Workshops. IEEE (2015)
76. Duan, Y., Shao, L., Hu, G., et al.: Specifying architecture of knowledge graph with data graph, information graph, knowledge graph and wisdom graph. In: Proceedings of the 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA). IEEE (2017)
77. Neil, D., Briody, J., Lacoste, A., et al.: Interpretable graph convolutional neural networks for inference on noisy knowledge graphs. arXiv preprint arXiv:181200279 (2018)
78. He, Z., Chen, W., Li, Z., et al.: SEE: syntax-aware entity embedding for neural relation extraction. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (2018)
79. Guo, X., Zhang, H., Yang, H., et al.: A single attention-based combination of CNN and RNN for relation classification. IEEE Access **7**, 12467–12475 (2019)
80. Nie, B., Sun, S.: Knowledge graph embedding via reasoning over entities, relations, and text. Future Gener. Comput. Syst. **91**, 426–433 (2019)
81. Yan, D., Hu, B.: Shared representation generator for relation extraction with Piecewise-LSTM convolutional neural networks. IEEE Access **7**, 31672–31680 (2019)
82. Zhang, C., Cui, C., Gao, S., et al.: Multi-gram CNN-based self-attention model for relation classification. IEEE Access **7**, 5343–5357 (2019)

83. Guo, X., Zhang, H., Yang, H., et al.: A single attention-based combination of CNN and RNN for relation classification. IEEE Access **7**, 12467–12475 (2019)
84. Shen, Y., Sun, J, Jia, P., et al.: Entity-dependent long-short time memory network for semantic relation extraction. In: Proceedings of the 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). IEEE (2019)
85. Le, H.Q., Nguyen, T.M., Vu, S.T., et al.: D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information. Bioinformatics (2018)
86. Chen, H., Lin, Z., Ding, G., et al.: GRN: gated relation network to enhance convolutional neural network for named entity recognition (2019)