This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

A Method about Building Deep Knowledge Graph for the Plant Insect Pest and Disease (DKG-PIPD)

Yingying Liu[1,2]
[1]Data and Target Engineering College, Information Engineering University, ZhengZhou 450001, China
[2]Information Engineering College, Henan University of Animal Husbandry and Economy, ZhengZhou 450046, China

Corresponding author: Yingying Liu (e-mail: conslexanve1985@gmx.com).

ABSTRACT In this study, a method about building Deep Knowledge Graph for the Plant Insect Pest and Disease, namely DKG-PIPD, was proposed. Specifically, the semi-automatic extraction of semi-structured and unstructured knowledge was carried out on the basis of domain ontology, and the knowledge graph was stored in the third-party knowledge database according to the corpus characteristic of the plant insect pest and disease, to realize the visual display of entity interactive relationship and knowledge inference. Furthermore, DKG-PIPD performed joint extraction about the entity and the relationship in unstructured knowledge in a corpus tagging method that is suitable for domain data. In this way, the entity and the relationship were annotated synchronically, and the triplet can be obtained directly through label matching and label mapping, which not only effectively improved the annotation efficiency, but also solved the problem of one-versus-many overlapping relation extraction. In addition, DKG-PIPD used a novel end-to-end model to train and predict on the crawled dataset. The experimental contrast results with other classical benchmark methods demonstrated the effectiveness of the proposed method. Moreover, the related work in this paper first introduced the general architecture required for the building of knowledge graph, and then summarized its key points, that is, named entity recognition, entity relationship extraction and knowledge inference using deep learning are emphatically introduced. Finally, the improvement direction of this paper was also introduced in the discussion section.

INDEX TERMS DKG-PIPD, semi-structured knowledge, unstructured knowledge, end-to-end model

I. INTRODUCTION

In 2012, Google introduced the concept about knowledge graph, which provides a new way for knowledge management. Knowledge graph is essentially a structured semantic knowledge base that describes concepts, entities and their relationships in the objective world in a structured form, generally in the form of a triplet of (entity, relationship, entity) or (entity, attribute, attribute value). Knowledge graph can structure the heterogeneous knowledge of the field and it is good at describing the interaction between entities, which makes the field knowledge explicitly precipitated and associated, and well solves the problem of scattered, complex and siloed data in the field. In conclusion, knowledge graph is widely used in medical, biological and financial fields [1]. According to the different knowledge coverage, knowledge graph is divided into open-domain knowledge graph [2] and vertical-domain knowledge graph [6]-[8]. The former focus more on breadth while the latter focus on depth, but due to the lack of annotated training corpus and excessive reliance on experts, the scale is generally small and the construction cost is expensive.

The plant insect pests and diseases have always been an important factor affecting plant production. With the development of information technology, Internet has become the main source of the prevention and control knowledge about insect pests and diseases. However, the current open-source knowledge in the field of plant insect pests and diseases is mainly stored in the form of traditional databases, which has poor aggregation ability, low utilization rate and difficult knowledge sharing. In view of the good performance of knowledge graph for field knowledge management, there are some achievements about knowledge graph in agriculture, but the in-depth studies on knowledge graph for plant insect pests and diseases are still relatively few. Based on fragmented agricultural big data, literature [4] constructed a knowledge graph for smart agriculture and its application system. Literature [5] firstly generated an ontology layer based on the classification criteria of plant insect pest and disease

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

data, and then on its basis extended the entity layer to initially form the knowledge graph and visualize the knowledge graph. Literature [6] used the ontology and other techniques to construct the knowledge graph in agriculture, which covers plant varieties, plant insect pests and diseases, pesticide data and fertilizer data. Literature [7] constructed the knowledge graph of rice, and etc. However, these knowledge graphs still have much room for improvement in terms of scale, intelligence and systematization, and it is still very challenging to extract semi-structured or unstructured data, solve the extraction of overlapping relationships in text, and reduce the input of manual features effectively.

The construction of knowledge graph is a combination of knowledge representation, knowledge extraction, knowledge storage and other techniques. Knowledge representation is a computer-acceptable data structure for describing knowledge, but early knowledge representation was not very expressive and lacked flexibility, so now the ontology has become the most commonly used method for knowledge representation, knowledge sharing, and knowledge reuse. Knowledge extraction is the core part of knowledge graph construction, including name-entity recognition (NER) task and relation extraction (RE) task. According to the order of completing NER and RE, knowledge extraction can be divided into the pipeline approach and the joint learning approach. The pipeline approach [8] divides NER and RE into 2 independent subtasks to first identify the entities in the text and then classify the semantic relationships between entity pairs, which is more flexible and easier to model, but dividing the 2 tasks suffers from error propagation, information loss, entity redundancy and other problems. Therefore, the entity relationship joint learning methods have become the mainstream in recent years, and which are divided into 2 types of sub-methods, i.e., parameter sharing and sequence tagging, depending on the difference of modeling object. The parameter sharing approach is to model entities and relations separately and to realize the interaction between 2 subtasks by sharing a joint coding layer for joint learning [9], but there is still the problem that redundant entity information cannot be eliminated. Therefore, some scholars [10][11] have studied to transform the joint extraction of entity relations into a sequence tagging problem, which solves the entity redundancy as well as overlapping relations problem to some extent. Literature [12] carefully analyze the key techniques and methods of the construction of insect pest and disease knowledge graph in recent years based on the characteristics of insect pest and disease data, and therewith conclude that ontology learning, machine learning, and deep learning are the key techniques to achieve automatic knowledge extraction, and are also the current research hotspots of plant insect pest and disease based on knowledge graphs. There are mainly 2 types of storage methods for knowledge graph, i.e., resource description framework (RDF)-based storage and graph database-based storage. The important design principle of RDF is the easy release and sharing of data, while the graph database uses the attribute graph as the basic representation, which is easier to express the real business scenarios and achieve efficient graph query and search. Therefore, knowledge graph storage based on graph database has become the mainstream approach in recent years, and Neo4j therein, as an open-source graph database system, is the main way to store knowledge graph at present.

How to accurately extract useful knowledge such as the causal factor, the harm site, and the control agent from the huge amount of complex plant insect pest and disease data is the key problem about the construction of plant insect pest and disease knowledge graph. With the development of information technology, deep learning has gradually penetrated into all aspects of knowledge graph construction, and the application of deep learning in the key link of knowledge graph construction will be introduced in the next section [13]. In order to improve the efficiency and accuracy of knowledge extraction and reduce the cost of knowledge graph construction, this study proposes a method about building Deep Knowledge Graph for the Plant Insect Pest and Disease, namely DKG-PIPD.

Specifically, the contributions of this paper are as follows:

1. We implements a novel corpus tagging model based on the field ontology to achieve joint extraction of the entity and the relation, simultaneous tagging of the entity and the relation, direct modeling of the triple, which can be obtained by label matching and label mapping.

2. Bidirectional encoder representations from transformers (BERT)- bi-directional long-short term memory (BiLSTM) & conditional random field (CRF) such an end-to-end model is used for training and prediction.

3. The extracted triadic data are stored in the Neo4j graph database to realize the visual display and knowledge inference of the knowledge graph.

4. The related work in this paper first introduces the general architecture required for the building of knowledge graph, and then summarizes its key points, that is, named entity recognition, entity relationship extraction and knowledge inference using deep learning are emphatically introduced.

5. The proposed knowledge graph can provide a high-quality knowledge base for downstream applications such as intelligent question and answer system, recommendation system, and intelligent search for plant insect pests and diseases, which can further be effectively used in agricultural production such as plant variety selection, insect pest & disease control, and fertilization & irrigation.

II. RELATED WORKS

Knowledge graph was first proposed by Google, who developed a project based on the knowledge graph and applied knowledge graph to semantic search. Moreover, Google could accurately search the required information through the constructed knowledge graph. The definition given by Google is that knowledge graph is an auxiliary knowledge base which Google uses to enhance its search

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

engine function. In general, knowledge graph is a knowledge base which is associated with structured information by the way of the graph structure, and the construction of knowledge graph based on deep learning is to build an "entity-relations-entity" triplet model with the data information of a certain field by means of deep learning algorithm and store it in the graph database.

A. THE CONSTRUCTION OF KNOWLEDGE GRAPH

The structure of knowledge graph refers to the technical system of realizing the construction of knowledge graph, which is mainly divided into two parts: data acquisition and data processing. Data acquisition refers to the selection of raw materials for the construction of knowledge graph. Knowledge graph based on deep learning requires a large amount of training data for model training. Thus, data acquisition is one of the important steps about the construction of knowledge graph. Data processing refers to proceeding the relevant algorithm operation with the collected data and further completing the corresponding tasks. As shown in Figure 1, the construction of knowledge graph is mainly divided into the following processes:

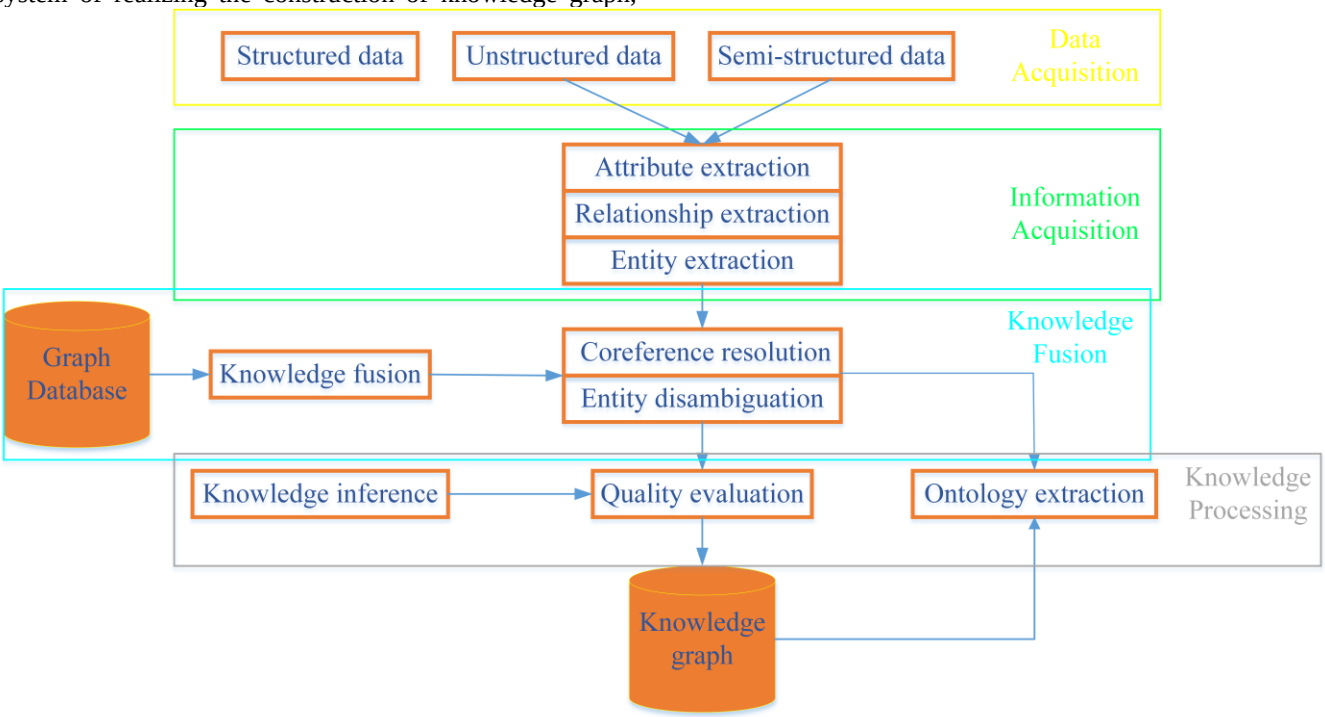

FIGURE 1. The general building of knowledge graph

DATA ACQUISITION. Data set can be acquired by web crawler, database, manually made data or downloaded from the corresponding official website. The data collected generally falls into three forms:

1) Structured data: For the existing information in the network database, the database can be read and written directly. This kind of data is screened or sorted into two-dimensional content in advance, and which tends to have a high degree of confidence due to manual screening, so this kind of data is the most important way to build the knowledge graph in the early stage. However, since structured data requires a lot of manual operation, the cost of manual production of structured data is too expensive when based on a large amount of data.

2) Unstructured data: Unstructured data is usually data that has no any structure, such as picture, audio, text and other information, which is usually stored or read and written as a whole. Most of the construction of the knowledge graph requires the mining of these unstructured data. Hence, the main data source of the construction of the knowledge graph is unstructured data. At the same time, the related research mainly takes the unstructured data as the raw material.

3) Semi-structured data: Semi-structured data refers to the content displayed in the form of web, such as Baidu Encyclopedia, Wikipedia, etc. Such data usually exists in the form of XML, JSON, and so on, which is between structured and unstructured. This kind of data needs a series of preprocessing methods to transform it into structured data.

INFORMATION ACQUISITION. After data collection, corresponding data operations needed and the key part of data operations in the knowledge graph is information extraction, which mainly includes three steps: Named Entity Recognition (NER), Entity Relationship Extraction/ Relationship Classification (RC) and Attribute Extraction (AE).

1) NER: Named entity recognition is the first step of information extraction from semi-structured data and unstructured data, and entities are often the main carriers of information. An entity can be a person, a place name or something, and it can be a concept. In the early stage, the required entities were extracted through string matching or

manual operation. Afterwards, people extracted the entities through Natural Language Processing (NLP) and machine learning. Furthermore, in the knowledge graph construction based on deep learning, NER is recognized through sequence labeling.

2) RC: Entity relationship extraction is also known as relationship classification. In order to determine the "entity-relation-entity" triplet, the relationship between entities needs to be classified. This process is also known as semantic information extraction. In the early stage, relational extraction was carried out manually by pattern matching according to the grammatical rules of the language. Although this method was highly accurate, it required professionals from various fields to operate as well as a large amount of labor costs.

In the knowledge graph architecture based on deep learning, the relationship label of sentences containing two related entities is carried out through feature engineering to realize supervised learning. Now there is also relationship extraction based on self-supervised learning. In addition, the joint learning of NER and RC proposed by [14] integrate the two steps together to form a joint learning method, which improves the accuracy of the model to a certain extent.

3) AE: After the construction of triples, it is necessary to extract the attributes of entities and relationships. Attribute extraction can often be obtained directly through the network, and properties can also be treated as entities or relationships through NER or RC.

Named entity recognition, entity relationship extraction and attribute extraction are the main parts of the construction of knowledge graph, which are also the preparation for the next step.

KNOWLEDGE FUSION. Since the triplets obtained by information extraction often have a certain degree of error and the accuracy of the model is often not 100% from the perspective of the model optimization through NER and RC, and accordingly, which could lead to the emergence of misidentified entities or misclassified relations. Therefore, in order to improve the confidence of the knowledge graph, it needs to be processed, which is mainly in the following ways:

1) Entity disambiguation: The same entity may have different names, and the same name may represent different types of entities. For example, "the basketball god" and "Air Jordan", both of which mean the same thing, namely "Michael Jeffrey Jordan". However, they are not merged in the process of information extraction, hence, the main purpose of entity disambiguation is to eliminate the ambiguity caused by entities with the same name. Four methods provided by [15]: Entity disambiguation can be realized by spatial vector model, semantic model, social network model and encyclopedia knowledge model.

2) Coreference resolution: In a sentence, there are often multiple references pointing to the same entity. This kind of problem can not only be dealt with through syntactic analysis, but can be transformed into a classification or clustering problem based on machine learning.

3) Knowledge fusion: Knowledge systems that often independently established are relatively isolated with limited information, and in order to make the self-built knowledge system echo the existing knowledge base of the network, it is necessary to merge the knowledge. The established knowledge system can be stored in the graph database in the form of graph structure and merged through entity disambiguation. In addition, the knowledge system can also be stored in the relational database in the form of relationship and then merged by means of database technology. Knowledge fusion is an important step to expand autonomous learning and build a knowledge base.

In reality, in the process of self-building knowledge graph, knowledge fusion is often neglected, but it is also extremely important.

KNOWLEDGE PROCESSING. Through information extraction, knowledge elements such as entity, relationships and attributes can be extracted from the original corpus. After integrating knowledge, the ambiguity between entity reference and entity object can be eliminated and a series of basic factual expressions can be obtained. However, fact itself is not equal to knowledge. In order to gain a structured and networked knowledge system, going through the phase of knowledge processing is requisite. To sum up, knowledge processing mainly includes three aspects: ontology construction, knowledge inference and quality evaluation.

1) Ontology construction: Ontology is a set of terms used to describe a certain field, with the goal to obtain, describe and represent knowledge of the related field, and provide a common understanding of the knowledge of the field. Moreover, ontology not merely determines the commonly accepted terms in the field, but also gives the clear definition of these terms and their relationships from different levels of formal models.

2) Knowledge inference: As its name implies, it is the relationship reasoning between knowledge, knowledge inference includes logical relationship inference and graph relationship inference. The former belongs to semantic analysis. For example, the proposition "People who play in the NBA are professional, but not all the professional play in the NBA", from which we can infer that Michael Jordan, who played in the NBA, was also a professional. Correspondingly, graph relationship inference extends the relationship according to the graph model. Another example is that the established triples include "The Bulls is in Chicago" and "Chicago is in Illinois", and " The Bulls is in Illinois" can be inferred.

3) Knowledge Update: Specially, a good knowledge graph needs to be constantly updated and iterated. There are two types of updates: full update and incremental update.

At present, the construction of knowledge graph is a huge course research system in the field of scientific research, which involves many technologies. Therefore, the focus of this paper is to introduce merely named entity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

recognition, entity relationship extraction and knowledge inference that are based on deep learning.

B. NAMED ENTITY RECOGNITION USING DEEP LEARNING

In recent years, deep learning technology derived from neural network model has become a hotspot in the field of machine learning. In particular, the method of using word vector to represent words, on the one hand, which solves the problem of data sparsity caused by high latitude vector space. On the other hand, the word vector itself contains more semantic information than manually selected features. Moreover, this method can obtain feature representation under the unified vector space from the heterogeneous text, which definitely brings a strong impetus for the development about the typical serialization annotation task of NER. Therefore, although NER is no longer a hot topic in many research fields related to named entities, many scholars still apply the latest deep learning technology to the NER problem in order to further improve the effect of NER. Thereinto, using word vectors as features is still the simplest and most effective method [16]. However, more studies are also trying to learn from and improve existing models and methods. For example, some studies refer to the good results obtained by LSTM in automatic word segmentation and put forward a model combining LSTM and CRF, which improves the F value by 5% compared with the previous methods [17]. In order to prove that data from the real world can be used to improve the effect of NER, Tomori et al. [18] trained a DNN+R model by using the commentary corpus and data from Japanese chess match. At length, they found that the effect of this model was much better than that of the simple DNN (Deep Neural Networks) model. Lample et al. [19] proposed two kinds of neural network models i.e., LSTM-based and transformation-based, and obtained features from annotated and unannotated corpus at the same time, which achieved the best NER effect in all four languages. Besides, Bharadwaj et al. [20] added a layer of phoneme feature into LSTM, and achieved a better NER effect in languages with complex morphological changes such as Turkish. In addition, deep learning methods such as convolution neural network (ConvNet) [21] and hybrid neural network (HNN) [22] have also been successfully used to solve NER problem and achieved good results.

C. ENTITY RELATIONSHIP EXTRACTION USING DEEP LEARNING

Entity relationship extraction based on deep learning is mainly divided into two categories: supervision and distant supervision. In the supervision model, the methods to solve entity relationship extraction can be divided into two kinds: pipeline learning and joint learning. The former method means to directly extract the relationship between entities based on the completion of entity recognition, while the latter method is mainly based on the end-to-end model of neural network, which simultaneously completes the identification of entities and the extraction of relationships between entities. Compared with supervised entity relationship extraction, distant supervision method lacks manual annotation dataset. Therefore, the distant supervision approach is one more step than the process of marking unlabeled data by distant alignment of the knowledge base, and the part of constructing relation extraction model is similar to the pipeline method of supervised domain.

SUPERVISION ENTITY RELATIONSHIP EXTRACTION METHOD BASED ON DEEP LEARNING. In recent years, relationship extraction based on supervised method in deep learning has become a research hotspot of relationship extraction, which can solve the two main problems of manual feature selection and error propagation of feature extraction in typical methods, and combine low-level features to form more abstract high-level features, which can be used to find distributed feature representation of data. Supervised entity relationship extraction based on deep learning can be divided into: pipeline method and joint learning-based method. These two methods are based on ConvNet, RNN, LSTM three frameworks for extension optimization.

First, the main process of the pipeline-based method for relation extraction can be described as follows: relation extraction is carried out for sentences with labeled target entity pairs, and the triples with entity relations are output as the predicted results. In the pipeline-based method, the extension based on RNN model includes adding dependency analysis tree information and word dependency matrix information on the basis of RNN. The extension based on ConvNet model includes adding category ranking information, dependency analysis tree and attention mechanism on the basis of ConvNet. The extension based on the LSTM model includes adding the shortest dependency path (SDP) on the basis of LSTM or combining LSTM with ConvNet. However, the pipeline-based method has some problems, such as error accumulation propagation, neglecting the relationship dependence between subtasks, and generating redundant entities. Specifically, the pipeline method has the following disadvantages.

1) Error propagation: the error of entity recognition module will affect the following relationship classification performance;

2) The relationship between the two subtasks is ignored, i.e., the loss of information affects the extraction effect;

3) Generating redundant information: Since identified entities are paired with each other and then classified into relationships, unrelated entity pairs can be loaded with redundant information and lead to increasing the error rate. Therefore, the joint learning model gradually began to receive attention.

Second, compared with the pipelining method, the joint learning method [23] can make use of the close interaction information between entities and relationships, further extract entities and classify the relationships of entity pairs at the same time, which covers the shortage of the pipeline-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

based method well. In addition, the joint learning method obtains the entity triple with relationship directly through the joint model of entity recognition and relation classification. Moreover, owing to the different modeling objects in the joint learning method, the joint learning method can be divided into the parameter sharing method and the sequence labeling method. The parameter sharing method models entities and relationships respectively, while sequence labeling method models entity-relationship triple directly. In addition, BI-LSTM is used in the coding layer of the parameter sharing method, while the decoding layer is optimized and extended based on the methods of BI-LSTM, dependency number and attention mechanism. Whereas, the sequence labeling method solves the problem of redundant entities in pipeline model by using an end-to-end model of a new annotation strategy. To sum up, the joint learning method includes the extraction method of entity relation based on parameter sharing and new sequence labeling. For one thing, the former can improve the problem of error accumulation propagation and the neglect of the relationship dependence between two sub-tasks in the pipeline method. For another, the latter solves these two problems as well as the problem of redundant entities in pipeline methods. Whereas, these two methods fail to provide relevant solutions to the overlapping entity relationship identification problem existing in the current supervision field.

DISTANT SUPERVISION ENTITY RELATIONSHIP EXTRACTION METHOD BASED ON DEEP LEARNING. In the face of a large number of unlabeled data, supervised relationship extraction consumes a lot of manpower, which seems to be labored. Therefore, that is where the distant supervision entity relationship extraction came from at that historic moment. In 2009, Mintz [24] first proposed that the application of distant supervision should be used to the task of relation extraction, which solved the problem of automatic labeling of large amounts of unlabeled data in open domain by automatically aligning the distant knowledge base with data. There are two main problems when labelling data in a distant supervision way, i.e., noise problem and feature extraction error propagation problem. The noise problem is due to the strong hypothesis of distant supervision, which leads to the mislabeled relationship of a large number of data, resulting in a large amount of noise in the training data. The problem of error propagation in feature extraction is that the traditional feature extraction mainly uses NLP tools to extract the features of data sets. Consequently, a large number of propagation errors will be introduced. As for the problem of wrong labeling, the multi-instance multi-label learning (MIML) method proposed by Surdeanu [25] and the attention mechanism proposed by Lin [26] both effectively weakened the influence of distant supervised wrong labeling on extraction performance. What's more, owing to the rise of deep learning and its good effect about relationship extraction in the supervised field, it is a very natural idea to replace feature engineering with the idea of feature extraction by deep learning, i.e., representing the entities and other words in the sentence with the word vector or the position vector; modeling sentences and constructing sentence vectors by means of the deep learning model; Finally, carrying out a classification about relationship. The deep learning model and its characteristics include: ConvNet's extended models, i,e., PCNN+MIL [27] and PCNN+ATT [28] (attention mechanism as a generalization of multi-instance mechanism) for weakening the problem of mislabeling; LSTM [29] obtained the directional information of entity; COTYPE [30] extracted entity and relationship information jointly; Deep residual network [31] prevented the accumulation of mislabeled noise layer by layer.

A COMPARISON OF DISTANT SUPERVISION AND SUPERVISION RELATIONSHIP EXTRACTION METHOD BASED ON DEEP LEARNING. Supervision entity relationship extraction relies on manual annotation method to obtain data set, which has high accuracy and purity, and the trained relationship extraction model has good effect and experimental value. However, the method of manually annotating data, on the one hand, which consumes a lot of manpower cost, on the other hand, which is limited in quantity, poor in scalability and confined in field. As a result, the constructed supervision entity relational extraction model is excessively dependent on the manually annotated data, which is not beneficial to the cross-domain generalization ability of the model and likewise poor in domain migration. In the face of a large amount of unlabeled data, distant supervision has obvious advantages over supervised entity relationship extraction. Since it is impractical for humans to label a large amount of unlabeled data, the distant supervision automatically labels the data by adopting the method of aligning the distant knowledge base, which not only greatly reduces the error of human and but also has strong mobility in the field. However, the accuracy of data obtained from distant supervision and automatic labeling is relatively low. Therefore, when training the model, the error of wrong labeling will spread layer by layer and eventually affect the whole model. Therefore, the effect of the current distant supervised entity relationship extraction model is generally worse than that of the supervision model.

D. KNOWLEDGE INFERENCE USING DEEP LEARNING

The main idea of knowledge inference based on deep learning is to use the learning ability and generalization ability of the neural network to model the fact tuple of the knowledge graph. Among them, modeling and predicting the elements of the triplet are generally single-step inference, while multi-step inference is to model the continuous path constituted by the tuples and predict the information such as the entities at the beginning and the end of the path and the implied relationship between them. More specifically, knowledge inference based on deep learning can be further divided into three parts: Semantic-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

based inference, structure-based inference, auxiliary storage-based inference and other inference according to the inference basis:

SEMANTIC-BASED INFERENCE. Semantic-based inference is based on the mining and exploitation of semantic information, such as the name, description, and context information of entities and relationships. As the potential semantic information contained in the text is very rich and there is a deep semantic correlation between the information, so the text and semantic information naturally become the main inference basis in the field of knowledge inference. There are several typical semantic-based inference models as follows. The NTN model proposes a Neural Tensor Network and designs a new representation of long-tailed entities [32]. The DKRL (Description-embodied Knowledge Representation Learning) model adds entity description to entity representation through the CBOW (Continuous Bag-Of-Words) method and ConvNet [33]. ProjE (Embedding Projection) model designs a combined operator to combine the inputs into new vectors and then project them onto the candidate set [34]. Besides, the MT-KGNN (Multi-Task Neural Network) model is the first one to use a neural network to model the attribute information in the knowledge graph [35]. Moreover, the ConMask model defines "open world knowledge graph completion", which can link new entities outside the knowledge base to the knowledge graph [36]. In addition, the HNM (Holistic Neural Matching) model is applied in the field of intelligent question-answering, which is different from the traditional question-answering system based on the pipe [37]. In fact, it combines the semantic information of characters and words to avoid the problem of error propagation in the pipeline method.

STRUCTURE-BASED INFERENCE. Structure-based inference refers to using the structural connections within or between the triplet in the knowledge base for reasoning, which is often used in multi-step inference problems. There are some typical inference methods based on structural information, which can be defined from the perspective of structure as inference based on adjacent entities, inference based on multi-hop relations, and inference based on combined paths. Among them, the inference based on adjacent entities makes use of the relationship and entity information adjacent to the target entity, and the RGCN (Relational Graph Convolutional Networks) model is improved by the graph-structured GCN network combined with the directed relationship of the knowledge graph. In addition, encoders and decoders are used to model the relationship connected with the entity, which can realize entity classification and relationship prediction. The inference based on multi-hop relationship makes use of the information of multi-hop relationship on the path of continuous tuple group, and combines the multi-hop relationship into one, and deduces the "merge relationship" of the entities connecting the beginning and ending of entity path; The inference based on combined path is improved on the basis of multi-hop relationship inference, which takes into account the relationship and the information of intermediate entities in multiple paths, and makes the prediction result more accurate.

AUXILIARY STORAGE-BASED INFERENCE. Inference based on auxiliary storage is analogous to the storage, reading and writing of knowledge by human brain. It uses shared memory components or external storage matrix to store the intermediate results or necessary information required for inference, and simulates the inference process of human inference and thinking through auxiliary storage to facilitate the inference process to obtain implicit information and improve the efficiency of inference. There are some typical approaches as follows. The IRN (Implicit ReasoNets) model shares a memory component that is used to store knowledge base information and can be read by the model at any time [38]. DNC (Differentiable Neural Computer) model simulates the process of adding, deleting and changing knowledge memory of human brain by reading and writing shared external matrix [39].

THE OTHER INFERENCE. Knowledge inference oriented to knowledge graph has always belonged to the field of Natural Language Processing (NLP). Some researchers have innovatively extended knowledge inference to the field of computer vision, hoping to combine the inference with visual information and solve the problem of relational reasoning behind image pattern recognition. For example, Wang et al. [40] combined knowledge inference with image recognition and discussed an interesting social relation inference problem. The method trains a Graph Reasonable Model (GRM), which combines with Gated Graph Neural Network (GGNN) [41] and can infer over the social relations of the characters in the pictures. The main method is to generate a task relationship knowledge graph on the basis of the social relationship data set, i.e., PISC, and use GRM to initialize the relationship node according to the characteristics of the task region in the image. Next, the pre-trained Fast-RCNN detector [42] is used to search the semantic objects in the image, extract their features, and initialize the corresponding object nodes. After that, GGNN is responsible for calculating node features, propagating node messages through graphs to fully explore the interaction between people and context objects. What's more, adaptively selecting nodes with the largest amount of information by using the graph attention mechanism to facilitate recognition by measuring the importance of each object node. Different objects in the picture may correspond to different social relationships. For example, when the oven is identified, the probability of family relationship is relatively large, while when the keyboard is identified, it is easier to identify as a professional relationship.

III. PROPOSED METHOD

There are 2 methods about knowledge graph construction, i.e., bottom-up and top-down. Bottom-up refers to the data-driven approach, which is more applicable to the open field

knowledge graph, while the vertical field mostly adopt the top-down construction mode [43] due to their industry-specific specialties, complex and changing business needs and requirements for high-quality data, i.e., the ontology and the data scheme are defined first, and then the entity and their interrelationship are populated into the knowledge graph. This study adopts the top-down knowledge graph construction approach, and the specific construction process is shown in Figure 2, which mainly includes data acquisition, ontology construction, knowledge extraction and knowledge storage.

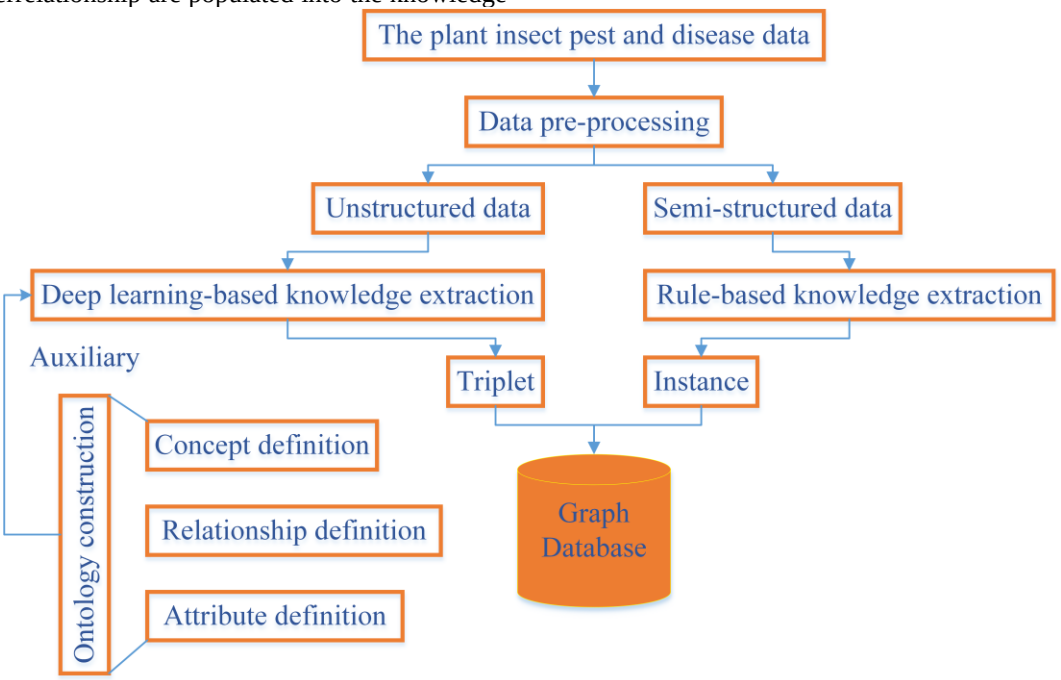

FIGURE 2. Proposed building flowchart

A. DATA ACQUISITION AND PRE-PROCESSING

The main data source for this study is Plant Germplasm Information Network-Plant Insect Pest and Diseases Knowledge Website [44]. The data are crawled by using the Scrapy framework of Python programming language, and which further were preprocessed by combining the corresponding rule and the manual review to obtain a noise-free plain text corpus. Since the path of XPath of the website is irregular, it is not possible to use the unified XPath page parsing method to crawl the webpage content directly, so a plant insect pest and disease data is taken as a basic unit, and a total of 1,745 data are crawled by multi-level page crawling, including rice, wheat, beans, corns, potatoes, cottons, oilseeds, sugar, tobacco, tea and mulberry. In general, the insect pest and disease data of 11 types of plants are crawled. Since the crawled data also contains irrelevant contents such as web page navigation, advertisements, and other redundant and missing data, the redundant and missing data are cleaned up and complemented by using the regular expression combined with the manual auditing. However, the pre-processed text still retains the semi-structured data form inherent in the original web page, mainly containing attributes such as the name of the plant insect pest and disease and its symptoms, the pathogens, the transmission pathways, the pathogenic conditions, and the control methods.

B. THE BUILDING OF PLANT INSECT PEST AND DISEASE ONTOLOGY

The ontology is the clear specification about the conceptual model [45]. By analogy, the plant insect pest and disease ontology are the description and organization about plant insect pest and disease knowledge in the form of a language that can be understood by computer, and which can be used to organize and manage the data layer effectively through the construction of the upper-level ontology. This study uses Protégé [46], an open-source ontology construction tool, for defining top-level logical concepts, relationships between entities, entity attributes and setting corresponding constraints on the definition domains and value domains of relationships and attributes without the need for the complex and difficult ontology construction language. The plant insect pest and disease ontology are controlled into 4 layers as shown in Figure 3, and which includes 5 types of parent concepts, namely insect pest and disease, plant, pathogen, taxonomy, and pesticide. In order to describe the relationship between insect pest and disease entities and other entity types more accurately, combined with the practical business requirements and the guidance of domain experts, the relationship set between entities and the attribute set of entities are predefined based on the data representation characteristics. The relationship set includes harm plant, harm site, and so on, while the attribute set includes symptom, harm characteristic and control method and so on.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

At the same time, the corresponding definition domains and value domains are set for the relationships and attributes to clarify the boundary of knowledge extraction. The meaning of the definition domains and value domains is to set a certain range of constraints on the values of the relations and attributes. For example, for the relation of *harm plant*, the subject can only be the insect pest and disease entity and the object can only be the plant entity.

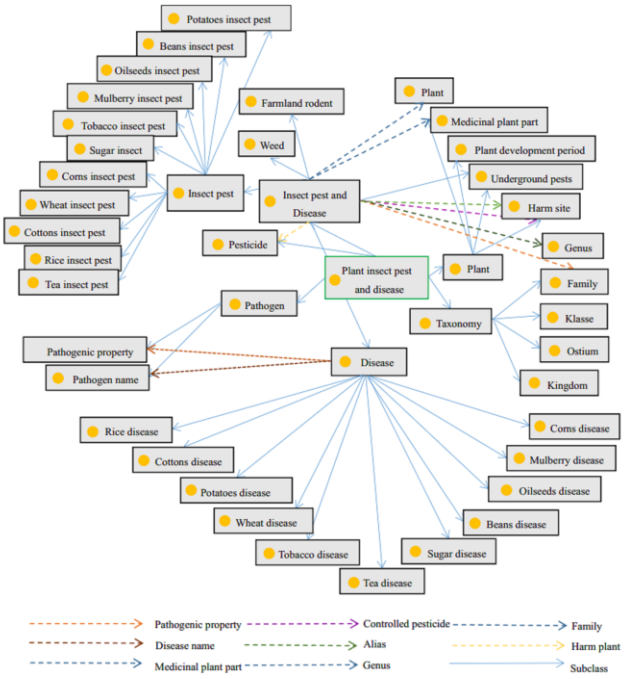

FIGURE 3. The knowledge graph ontology of plant insect pest and disease

C. THE SEMI-STRUCTURED KNOWLEDGE EXTRACTION

When crawling down the data from Plant Germplasm Information Network-Plant Insect Pest and Diseases Knowledge website, this study also obtained its semi-structured information, such as title, paragraph level and subheading, etc. Through practice, we found that we can use these semi-structured features to construct corresponding rules for extracting instances, i.e., (name: plant insect pest and disease; attribute 1: attribute value 1; attribute 2: attribute value 2; .....; attribute n: attribute value n). The text is first parsed into a structured .json format, where each plant insect pest and disease entity is an object, and each attribute of the insect pest & disease and attribute value form a key-value pair, and then 1,745 plant insect pest and disease instances are stored in the Neo4j graph database based on the py2neo module of the Python programming language by directly passing in Cypher statements, where each instance is a node, and each node contains information on entity attributes and attribute values such as plant insect pest and disease entity name, symptom, pathogen, and control method, e.g. {name: rice bacterial leaf streak ; symptom: rice bacterial leaf streak is a disease of rice caused by.....; pathogen: xanthomonas oryzae pv. oryzicola (Fang et al.) Swing et al. .....; control methods: (1) the rice seeds were soaked in 85% trichloroisocyanurate powder 500 times for 24 hours, and .....}.

D. THE UNSTRUCTURED KNOWLEDGE EXTRACTION

In the process of semi-structured knowledge extraction, where a whole text is used as an attribute value. However, the text of the attribute value also contains a lot of latent information that has not been mined. For example, in the attribute value of the symptoms about rice bacterial leaf streak, there is also hidden information of entity relationships such as alias, harm site, and the extraction of these relationships belongs to the knowledge extraction based on unstructured data. Extracting triplets from the unstructured text is a challenging task, and compared with the general corpus, the plant insect pest and disease corpus in this study has the following three special features: 1) A piece of data unfolds around only one plant insect pest and disease entity, so the head entity is fixed in the triplet extraction of the same piece of data, and only the tail entity and the relationship between the two needs to be extracted. 2) The entity distribution is dense, i.e., plant insect pest and disease entities generating relationship pairs with multiple entities in the text, and long distances between head and tail entities. However, the high-density entity distribution in the sentence seems to promote the fitting of named entity recognition model, but the same entity is involved in the composition of different types of relationship pairs several times, which will easily lead to the underfitting of interleaved relationships with limited annotation information support once the model lacks the ability to characterize the semantic information at the sentence level. In addition, the relationship between 2 entities with long distance is difficult to be extracted [47]. 3) The relationship between entities is complex. Controlled pesticide and banned pesticide entities often appear in the text at the same time, therefore the names of the entities are very similar, but the types of relationships to which they belong are completely different or even mutually exclusive, which makes the work of relationship extraction more difficult to a certain extent.

In order to transform the joint extraction of entity relation into the sequential tagging task, and based on the above-mentioned features of the corpus in this field, this study uses a corpus tagging pattern, i.e., main entity & relation & begin-inside-end-single-other (ME&R&BIESO), to achieve joint extraction of entities and relations to synchronize annotation of entities and relations, so as to directly model triplets instead of modeling entities and relations separately. The method of obtaining triplet data directly through label matching and label mapping effectively improves the tagging efficiency and solves the extraction problem of overlapping relation. To further characterize more

comprehensive sentence-level semantic features and alleviate the problems of interleaved entity relations and long distances between entities, this study introduces the BERT pre-trained language model, i.e., uses the BERT-BiLSTM&CRF end-to-end model for training and predicting, which not only extracts word-level features but also enables deeper mining and learning of sentence-level semantic features.

THE ME&R&BIESO TAGGING MODEL. In order to extract entities and relations in a corpus text where the data is described only around one main entity (ME), essentially only the entities $\{X_1, X_2, ..., X_i, ... X_n\}$ that are related to the ME and the relations between 2 entities $\{R_1, R_2, ..., R_i, ... R_n\}$ should be extracted, where X_i denotes the i-th entity that has a relationship with ME, and R_i denotes the type of relationship between X_i and ME. In addition, in order to reduce entity redundancy, only the relations within the set of predefined relationships in the ontology are extracted.

The ME&R&BIESO tagging mode aims at simultaneous annotation about the main entity and the relationship between the main entity and each entity. Firstly, the main entity is annotated with the ME tag, and when there is a relationship R_i between an entity X_i and ME in the text, the tag of X_i is directly set to R_i and the location information of the words in ME and entity X_i is indicated by the Begin-inside-end-single-other (BIESO) flag (as shown in TABLE I).

TABLE I THE LABEL SOLUTION OF ME&R&BIESO MODE

Tags	Connotations
ME	Main entity
R_i	The relation type between X_i and ME
B-ME	The first character of ME
I-ME	The internal character of ME
E-ME	The tail character of ME
S-ME	ME is a single character
B- R_i	The first character of entity X_i
I- R_i	The internal character of entity X_i
E- R_i	The tail character of entity X_i
S- R_i	Entity X_i is a single character
O	Other characters

Whenever a complete BIE, BE or S set with label ME and the same relation R_i in the data is matched, the entity ME and X_i corresponding to the label set are taken out, and the (ME, R_i, X_i) triplet is formed by label mapping and data parsing. Taking the data describing the entity of *rice bacterial leaf streak* as an example (as shown in Figure 4).

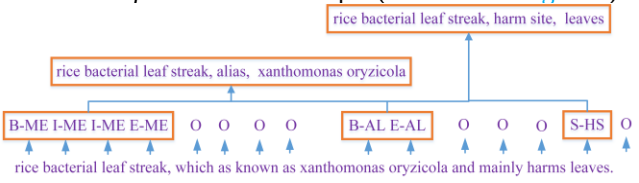

FIGURE 4. The annotation method instance

Firstly, *rice bacterial leaf streak* is labeled as ME. Secondly there is an alias relationship between *xanthomonas oryzicola* and *rice bacterial leaf streak*, so *xanthomonas oryzicola* is labeled as ALias (AL). Thirdly, the relationship between *leaves* and *rice bacterial leaf streak* is the harm site, then the leaf is labeled as Harm Site (HS). When the set of BE tags of ME and relationship AL is matched, and the triplet (rice bacterial leaf streak, alias, xanthomonas oryzicola) is generated; when the set of BIE of ME and HS is matched, the triad (rice bacterial leaf streak, harm site, leaves) is generated. When the next ME tag is matched, this means that all triplets corresponding to the previous main entity have been extracted.

The ME&R&BIESO tagging method only focuses on the relationship type R_i between main entity and each entity without focusing on the type of the entity itself, and only labels and extracts on the set of predefined relationships to reduce the redundancy and error propagation of irrelevant entity pairs. Meanwhile, for the problem of overlapping relationships between ME and multiple X_i, multiple corresponding triples can be obtained by label matching and label mapping. In addition, the traditional tagging-based and pipeline-based entity & relationship extraction methods need to annotate and identify the entity first, and then annotate and classify the relationships between pairs of entities that exist, while the ME&R&BIESO tagging method can annotate the entity and the relationship simultaneously, saving at least half of the annotation cost. However, this tagging method also has some limitations, i.e., it only considers the case of one-versus-many overlapping relationships, and the overlapping relationships for many-versus-many will be the future exploration direction.

THE BERT-BiLSTM&CRF MODEL. Based on the ME&R&BIESO tagging model, the tags are trained and predicted using the BiLSTM&CRF end-to-end model based on BERT word embedding. The overall framework of the model is shown in Figure 5, in which $\{E_1, E_2, \ldots, E_n\}$ is the embedding of BERT, and each word in the sequence is obtained by adding the three parts included the word vector, the segment vector and the position vector; $\{T_1, T_2, \ldots, T_n\}$ is the target of BERT, which is a sequence vector with rich semantic features obtained after feature extraction by the bi-directional converter. Specifically, the architecture consists of three parts: the annotated corpus first generates the word vector based on the context information through the BERT pre-trained language model; Then the word vector is fed to the BiLSTM module for bi-directional encoding, and the predicted score of each tag is the output; Finally, the output of the BiLSTM module is decoded using the CRF module, and the label transfer probability and the constraint condition are obtained by training and learning to further obtain the final predicted tagged sequences.

In the natural language processing (NLP) task, a language model is used to convert the word into the vector form for computer understanding. Traditional language models such as Word2Vec [48], Glove [49] and other single-layer neural networks cannot characterize the polysemy of the word well. Therefore, Devlin et al [50] proposed the BERT pre-trained language model, which is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

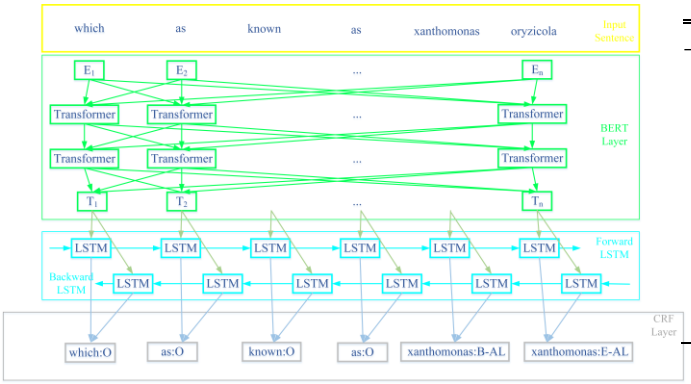

FIGURE 5. BERT-BILSTM&CRF model

responsible for converting the original input into vector form and then inputting the vector to the BiLSTM layer to learn contextual features. BERT is the first unsupervised and deep bi-directional model for pre-training and NLP technique, innovatively using 2 tasks of masked language model (MLM) and next sentence prediction for pre-training, enabling the word vectors obtained by BERT not only contain contextual word-level features, but also capture sentence-level features effectively [51].

BiLSTM [52] uses the word vector generated by BERT as input to obtain more comprehensive semantic information by capturing contextual features. LSTM [53] is a variant of Recurrent Neural Network (RNN) [54], which introduces memory units and gating mechanism on the basis of RNN to forget, update and pass selectively on contextual history information so as to learn the long-range semantic dependency, and at the same time can reduce the network deepness and alleviate the gradient disappearance and gradient explosion problem effectively. BiLSTM is a combination of a forward LSTM and a backward LSTM, which transforms the original sequential input sequence into two inputs, i.e., one positive and one negative, enabling the whole network to obtain both forward and backward information, which can better capture the long-range bi-directional semantic dependency and has better performance in the sequence annotation.

Although BiLSTM adequately captures contextual information, it sometimes does not consider dependency information between the labeled tags. For example, the B-AL tag can be followed by the I-AL tag or E-AL tag but if the tags such as B-HS, I-HS, E-HS, O, etc. are followed, then it is an illegal tag sequence. CRF [55] can be trained to learn to obtain the label transfer probability and add some constraints to the predicted labels to prevent the appearance of illegal labels. Therefore, using CRF as the output layer of BiLSTM can obtain the best triad labeling results.

The training procedure of BiLSTM&CRF as shown in Algorithm 1.

Algorithm 1 The training procedure of BiLSTM & CRF

for each epoch:
 for each batch:
 1) BiLSTM&CRF model forward pass:
 forward pass for forward state LSTM
 forward pass for backward state LSTM
 2) CRF layer forward and backward pass
 3) BiLSTM&CRF model backward pass:
 backward pass for forward state LSTM
 backward pass for backward state LSTM
 4) update parameters
 end for
end for

IV. EXPERIMENT

A. EXPERIMENTAL SETUP

In order to evaluate the performance of the proposed model accurately, this study uses three basic evaluation metrics in the field of entity relationship extraction, i.e., precision, recall and F1-score, to evaluate the model performance. The calculation of each evaluation metric is shown in (1) to (3):

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \tag{1}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - score = \frac{2\,\mathrm{Pr}ecision \times \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \tag{3}$$

where TP is the correctly predicted positive sample, FP is the incorrectly predicted positive sample, and FN is the incorrectly predicted negative sample.

During the training phase, this study sets the batch size according to the memory capacity, sets the maximum length of the sequence according to the average length of statements, judges the convergence of the loss function according to the training log, fine-tunes the dropout rate and the learning rate until the trained loss converges stably, and sets the number of LSTM units for expanding the system output capability. After several tunings and experiments, the optimal combination of core parameters is chosen as follows: batch size of 64, maximum length of sequence of 256, dropout rate of 0.4, learning rate of 0.01, and number of LSTM units of 200.

The computer equipment configuration and environment for this study are: Intel(R) Xeon(R) Bronze 3106 CPU @1.70GHz; GPU: NVIDIA GeForce RTX 2080 Ti (11G); memory 32GB; Python3.7; and Tensorflow2.2.0.

B. EXPERIMENTAL RESULTS AND ANALYSIS

In this study, a total of 1,745 plant insect pest and disease trial data (TABLE II) were divided into training and test sets in the ratio of 7:3 based on the resampling strategy of cross-validation.

In order to verify the effectiveness of ME&R&BIESO tagging method and BERT-BILSTM&CRF model, the pipeline method and other classical models of joint learning method were selected as benchmark models for contrast

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

TABLE II THE APPORTION OF THE COLLECTED DATASET

Data types	The documents amount	Number within different sets	
		Train set	Test set
Diseases	874	612	262
Insect and pests	729	510	219
Weeds	84	59	25
Farmland rodent harm	58	40	18
Total	1745	1221	524

TABLE III THE CONTRAST OF DIFFERENT METHODS (%)

Methods	Models	Precis -ion	Recall	F1-score
Pipeline	BERT&BERT	94.87	30.57	45.85
Sequence Annotation	PA-LSTM&CRF	77.64	83.43	80.57
	ETL-SPAN	90.16	91.39	88.23
Graph Structure	Novel Graph Scheme	96.04	69.95	82.09
	GraphRel	95.34	88.05	90.14
Parameter Sharing	BiLSTM&CRF	88.43	80.75	84.13
	CNN-BiLSTM&CRF	87.65	79.98	83.46
	BERT-BiLSTM&CRF	95.55	90.44	92.87

experiments, and the test results of each model are shown in TABLE III.

In order to verify the superiority of ME&R&BIESO tagging method and BERT-BiLSTM&CRF model in entity and relationship extraction tasks, BERT&BERT model in the traditional pipeline method and BiLSTM&CRF and ConvNet-BiLSTM&CRF models of joint learning based on parameter sharing method, PA-LSTM&CRF[54] and ETL-SPAN[55] of joint learning based on sequence annotation method, Novel Graph Scheme[56] and GraphRel [57] of joint learning based on Graph structure method are selected for comparison experiments in this study.

The pipeline-based method adopts the traditional entity and relationship labeling method, in which BIO method is used to annotate entities, and then the entity pairs with relationships are classified and labeled. A classification model of the relationships is first built using BERT, followed by an entity extraction model using BERT with the predicted relationship and plant insect pest and disease text. As thus, the entity extraction model is to predict the labeling of each token, and finally the entity pairs can be extracted based on the labeling.

PA-LSTM&CRF [54] proposes a novel joint extraction model, which generates N tag sequences for sentences of N words, marks entities and relational labels according to query word position P, and introduces positional attention mechanism to generate different sentence representations for each query position. The model can simultaneously extract entities as well as entity types and all overlapping relationships. ETL-SPAN [55] regard the entity relationship extraction task as the marking task of the head entity and corresponding tail entity, and mark corresponding head and tail entities for each entity relationship. The joint learning method based on sequence annotation models entity and entity relation simultaneously and obtains entity relation triplet in the same model. To annotate both entity and entity pair, code together in a model, and transform the joint extraction of entity and entity relationship into the problem of sequence annotation.

Novel Graph Scheme [56] designs a transformation framework to transform entity relation extraction into directed Graph, so as to facilitate the capture of the relations between entities and relations and the relations between entities, and cross entity extraction and relation extraction tasks. GraphRel [57] proposes a graph-based convolutional network (GCNs) joint extraction model i.e., GraphRel to automatically learn features through stacked BiLSTM encoders and GCN dependent tree encoders, extracts text sequence features and regional features using linear and dependent structure graphs, and extracts implicit features between all words in text using word graphs. This model can solve the problem of entity overlap and relation overlap effectively by establishing a full connection graph and taking into account the relationship between all pairs of words and the interaction between entities and relations. The method based on graph structure makes use of graph to model entity and entity relation, and the graph structure composed of entity and relation can fully consider the relationship between all entity pairs, which can improve the problem of entity overlap and relation overlap to some extent.

With the proposed ME&R&BIESO tagging method, the parameter sharing-based joint learning entity and relationship extraction method is experimented by using BiLSTM&CRF, ConvNet-BiLSTM&CRF and BERT-BiLSTM&CRF end-to-end models, respectively. It is worth mentioning that the joint learning method based on parameter sharing models the entity and entity relationship respectively, shares some parameters in the model, and adds the loss of entity recognition and the loss of relationship extraction as the overall loss of the joint model.

From the experimental results, it can be seen that although the pipeline method has a high precision rate of 94.87%, but the overall effect is biased. Specifically, the severely low recall rate of 30.57% results in an F1-score of only 45.85%. Through the analysis of the generated final prediction data, it is found that the relationships between pairs of entities in the text that are close together can generally be predicted accurately, but the pairs of entities that are far away are basically unpredictable, which indicates that the traditional pipeline method has great limitations when used for long-distance relationship prediction.

In the comparison test of the joint extraction model, the BERT-BiLSTM&CRF model significantly outperforms the BiLSTM&CRF and ConvNet-BiLSTM&CRF. Compared with BiLSTM&CRF and ConvNet-BiLSTM&CRF, the precision of BERT-BiLSTM&CRF increases by 7.12-7.9, the recall increases by 9.69-10.46, the F1 score increases by 8.74-9.41, respectively, and its F1 score reaches 92.87%. The ConvNet-BiLSTM&CRF model adds a ConvNet layer to BiLSTM&CRF, but the effect is not optimized, and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

F1 score is reduced by 0.67% instead. However, after adding the BERT pre-trained language model based on the BiLSTM&CRF layer, the F1 score improves by 8.74, indicating that BERT can assist in improving the model's semantic representation of the text and capture the interrelated entity relationships in the plant insect pest and disease text to a greater extent, thus enhancing the effect of the entity relationship extraction task.

The prediction results of the BERT-BiLSTM&CRF model for the relationship between main entity and each entity are shown in TABLE IV, and the overall effect is relatively balanced with an F1 score of about 91.47%. However, the prediction result of the relationship between main entity and harm site are significantly lower than the average, especially the recall rate is only 58.53%, which is an important factor that lowers the overall effect of the model. Through the analysis of the corpus text and the final prediction result of the harm site, it is found that the descriptions of the same plant site are not uniform, such as blade, leaf surface, blade back, leaf sheath, tender leaf, young leaf, mesophyll, and etc. are all used to describe the site of leaf. As a result, such a situation leads to many negative samples with wrong predictions in the prediction process, making the recall rate severely low and thus affecting the overall prediction level of the model.

The comparison of the running time of each epoch of different methods can be seen in TABLE V. The comparison of the running time of different methods can be seen in TABLE V. It can be seen from TABLE V and TABLE III that the proposed method achieves optimal results in terms of efficiency or accuracy. And the reason for that, the entity relationship extraction in this study about the construction of knowledge graph of plant insect pests and diseases is based on the set of relationships predefined by the ontology, which defines the boundary for unstructured knowledge extraction and reduces the invalid extraction of redundant information, while combining ME&R&BIESO tagging method and BERT-BiLSTM&CRF model for experiment, which largely improves the efficiency and accuracy of entity relationship extraction and ensures the quality of the knowledge graph.

TABLE IV THE PERFORMANCE DATA FOR MAIN ENTITY AND THE RELATIONSHIP TYPE BETWEEN MAIN ENTITY AND OTHER ENTITIES USING THE PROPOSED MODEL (%)

The relationship type between main entity and other entities	Precision	Recall	F1-score
Main entity	99.98	96.86	98.4
Alias	96.86	89.92	93.26
Scientific name	95.76	94.67	95.21
Harm site	80.68	58.53	67.84
Genus	99.03	99.03	99.03
Family	97.4	96.83	97.12
Harm plant	95.32	91.11	93.17
Pathogen name	91.08	82.61	86.64
Pathogenic property	94.62	84.62	89.34
Medicinal plant part	94.57	95.67	95.12
Medicinal plant efficacy	98.34	98.34	98.34
Controlled pesticide	91.73	89.34	90.52

TABLE V A COMPARISON OF THE RUNTIME OF DIFFERENT MODELS FOR EACH EPOCH (SECONDS)

Models	Runtime
BERT&BERT	212.44
BiLSTM&CRF	175.09
CNN-BiLSTM&CRF	168.63
PA-LSTM&CRF	114.85
BERT-BiLSTM&CRF	89.72

V. THE STORAGE OF KNOWLEDGE GRAPH

The current knowledge graph storage methods are divided into RDF (Resource Description Framework) triples-based and graph database based. The RDF triples are generally stored in relational databases, which are more flexible and efficient in querying, but at the same time, they store a lot of redundant information and need to be maintained regularly. The graph database stores the entities and concepts of the knowledge graph as graph vertices and stores entity attributes and relationships as edges in the form of graphs, which reflects the internal structure of the knowledge graph more intuitively, facilitates graph query and knowledge inference, and it is more scalable. Neo4j is an open-source graph database system, which uses graph data structure for storage at the bottom to substantially improve the performance of data retrieval, and it is the main way currently used for knowledge graph storage. Therefore, in this study, the plant insect pest and disease knowledge graph are stored in the Neo4j graph database.

Since the amount of data in this study is not particularly large, we use the LOAD CSV method in Cypher language that comes with the Neo4j database. We first save the entity nodes and relationships obtained through parsing as .csv files and place them in the import folder of Neo4j, and then import the nodes and relationships through the LOAD CSV statement. The Cypher statement is used to store the relationships between entities and entities in the Neo4j graph database to form the plant insect pest and disease knowledge graph, which includes 1,745 insect pest and disease instances and 29,496 triples. A partial visualization is shown in Figure 6, where the pink nodes are plant insect pest and disease entities, the blue nodes are entities with relationships to plant insect pest and disease entities, and the edges are the types of relationships between them.

The interactively associated nodes in the knowledge graph provide a good knowledge base for reasoning about latent relationships. For example, the edge between the node *monographella* and *leaf blight* is denoted as alias, and the edge between *monographella* and the node *50% thiophanate-methyl wettable powder* is denoted as pesticide, then it can be inferred that there is also a relationship of pesticide between *leaf blight* and *50% thiophanate-methyl wettable powder*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116467, IEEE Access

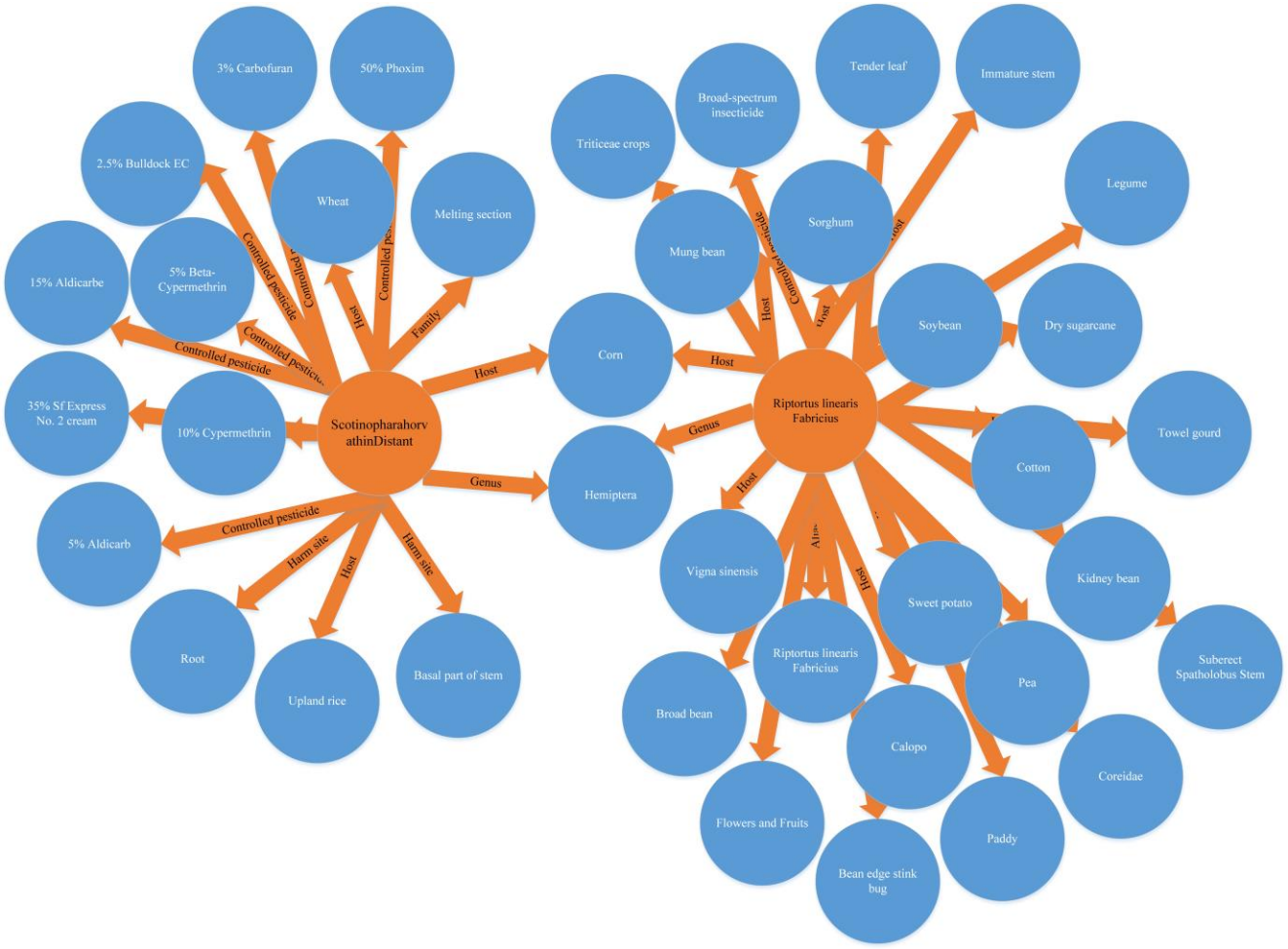

FIGURE 6. The visual display about knowledge graph in this study

VI. DISCUSSION

Although the knowledge map of plant insect pests and diseases realized in this study has begun to take shape, there is still room for improvement. In the future, the exploration will be carried out in the building method, many-to-many overlapping, relationship extraction, automatic update and other aspects. A top-down & bottom-up approach can be adopted to construct the knowledge graph, which combines the customized ontology model with the data-driven approach. In this way, a clear logical concept hierarchy can be set, and automatic knowledge extraction can be carried out from the public data set. Meanwhile, the quality and scale of the knowledge graph can be guaranteed. In addition, the research on more extensible and portable entity and relationship annotation method and training model can solve the problem of many-versus-many overlapping relationship extraction in corpus. Finally, with the rapid update of network data, it is necessary to update and supplement the knowledge graph data in time, and realize the automatic updating and upgrading of knowledge graph through knowledge fusion and knowledge inference.

VII. CONCLUSION

In view of the problems in the field of plant insect pests and diseases, such as the cross-correlation of entity relations, poor aggregation ability of multi-source heterogeneous data, and difficulty in knowledge sharing, this study uses the advantage of knowledge graph to describe the complex relationship between entities in a structured form, and proposes a method about building Deep Knowledge Graph for the Plant Insect Pest and Disease, namely DKG-PIPD. Based on the domain ontology, this method implements the joint extraction of entities and relationships with a new annotation pattern suitable to the domain corpus. The task of entity and relation extraction was transformed into a sequence annotation problem, and the entity and relation were annotated simultaneously, which effectively improved the efficiency of annotation. In order to solve the problem of overlapping relationship extraction, the triplet data can be obtained by label matching and mapping instead of modeling the entity and relationship respectively. In addition, an end-to-end model is used, and the contrast results show that the experimental data are better than the pipeline method based on the general labeling method and the classical models in the joint learning method. Finally, the extracted knowledge was stored in the third-party graph database to intuitively reflect the internal structure of the knowledge graph and

realize knowledge visualization and knowledge inference. The knowledge map constructed in this study can provide a high-quality knowledge base for downstream applications such as intelligent question answering system, recommendation system and intelligent search for plant insect pests and diseases. Moreover, the related work in this paper first introduced the general architecture required for the build of knowledge graph, and then summarized its key points, that is, named entity recognition, entity relationship extraction and knowledge inference using deep learning are emphatically introduced. In addition, the improvement direction of this paper was also introduced in the discussion section.

REFERENCES

[1] DieterFensel, Umutcanimek, KevinAngele, et al. Introduction: What Is a Knowledge Graph? [M]. 2020.
[2] Dodds K. Popular geopolitics and audience dispositions: James Bond and the Internet Movie Database (IMDb)[J]. Transactions of the Institute of British Geographers, 2006, 31(2):116-130.
[3] D Vrandečić. Wikidata: a new platform for collaborative data collection. ACM, 2012.
[4] https://github.com/qq547276542/Agriculture_KnowledgeGraph
[5] Kishimoto K, Hayashi K, Akai G, et al. Binarized Knowledge Graph Embeddings[J]. European Conference on Information Retrieval, 2019.
[6] Fensel D, Imek U, Angele K, et al. How to Use a Knowledge Graph[J]. 2020.
[7] Fensel D, Imek U, Angele K, et al. How to Build a Knowledge Graph[M]. 2020.
[8] Marrero M, Urbano J, S Sánchez-Cuadrado, et al. Named Entity Recognition: Fallacies, challenges and opportunities[J]. Computer Standards & Interfaces, 2013, 35(5):482-489.
[9] Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.
[10] Sun J, Zhao D, Wang L, et al. Remote supervision relation extraction method of power safety regulations knowledge graph based on ResPCNN-ATT[C]// 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA). IEEE, 2021.
[11] D Dai, X Xiao, Lyu Y, et al. Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:6300-6308.
[12] Liu, Xiaoxue, Bai, et al. Review and Trend Analysis of Knowledge Graphs for Crop Pest and Diseases[J]. IEEE Access, 2019, 7:62251-62264.
[13] Hasan S, Rivera D, Wu X C, et al. Knowledge Graph-Enabled Cancer Data Analytics[J]. IEEE Journal of Biomedical and Health Informatics, 2020, PP(99):1-1.
[14] Parret H. Kripke, S.A. Naming and Necessity. 1983.
[15] Sundheim B M. Named entity task definition, version 2.1[J]. Proc. Sixth Message Understanding Conf. (MUC-6), Nov. 1995, 1995.
[16] Guo H. The Unreasonable Effectiveness of Word Representations for Twitter Named Entity Recognition[C]// Conference of the North American Chapter of the Association for Computational Linguistics. 2015.
[17] Peng N, Dredze M. Improving Named Entity Recognition for Chinese Social Media with Word Segmentation Representation Learning[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016.
[18] Tomori S , Ninomiya T , Mori S . Domain Specific Named Entity Recognition Referring to the Real World by Deep Neural Networks[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016.
[19] Chaptal V , Kwon S , Sawaya M R , et al. Crystal structure of lactose permease in complex with an affinity inactivator yields unique insight into sugar recognition[J]. Proceedings of the National Academy of Sciences of the United States of America, 2011, 108(23):9361-9366.
[20] Bharadwaj A , D Mortensen, Dyer C , et al. Phonologically Aware Neural Model for Named Entity Recognition in Low Resource Transfer Settings[C]// Conference on Empirical Methods in Natural Language Processing. 2016.
[21] Dong X , Qian L , Yi G , et al. A multiclass classification method based on deep learning for named entity recognition in electronic medical records[C]// Scientific Data Summit. IEEE, 2016.
[22] Shao Y , Hardmeier C , Nivre J . Multilingual Named Entity Recognition using Hybrid Neural Networks. 2016.
[23] Y Chen, Kuang J , Cheng D , et al. AgriKG: An Agricultural Knowledge Graph and Its Applications[M]. 2019.
[24] Mintz M , Bills S , Snow R , et al. Distant supervision for relation extraction without labeled data[C]// ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore. Association for Computational Linguistics, 2009.
[25] Surdeanu M , Tibshirani J , Nallapati R , et al. Multi-instance Multi-label Learning for Relation Extraction[C]// Joint Conference on Empirical Methods in Natural Language Processing & Computational Natural Language Learning. 2012.
[26] Lin Y , Shen S , Liu Z , et al. Neural Relation Extraction with Selective Attention over Instances[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.
[27] D Zeng, Kang L , Chen Y , et al. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks[C]// Conference on Empirical Methods in Natural Language Processing. 2015.
[28] Zhou F Y, Jin L P, Dong J. Review of convolutional neural network[J]. Chinese Journal of Computers,2017,40(6):1229-1251.
[29] He D , Zhang H , Hao W , et al. A Customized Attention-Based Long Short-Term Memory Network for Distant Supervised Relation Extraction[J]. Neural Computation, 2017, 29(7):1-22.
[30] Ren X , Wu Z , He W , et al. CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases[J]. the 26th International Conference, 2016.
[31] Huang Y Y , Wang W Y . Deep Residual Learning for Weakly-Supervised Relation Extraction[J]. 2017.
[32] Socher R , Chen D , Manning C D , et al. Reasoning with neural tensor networks for knowledge base completion. Curran Associates Inc. 2013.
[33] Xie R , Liu Z , Jia J , et al. Representation learning of knowledge graphs with entity descriptions. 2016.
[34] Shi B , Weninger T . ProjE: Embedding Projection for Knowledge Graph Completion[J]. 2016.
[35] Yi T , Tuan L A , Phan M C , et al. Multi-Task Neural Network for Non-discrete Attribute Prediction in Knowledge Graphs[C]// CIKM'17. 2017.
[36] Shi B , Weninger T . Open-World Knowledge Graph Completion[J]. 2017.
[37] Lukovnikov D , Fischer A , Lehmann J , et al. Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level[C]// International World Wide Web Conference 2017. International World Wide Web Conferences Steering Committee, 2017.
[38] Shen Y , Huang P S , Chang M W , et al. Implicit ReasoNet: Modeling Large-Scale Structured Relationships with Shared Memory[J]. 2016.
[39] Graves A , Wayne G , Reynolds M , et al. Hybrid computing using a neural network with dynamic external memory[J]. Nature, 2016.
[40] Yang Z , Tang J , Cohen W . Multi-Modal Bayesian Embeddings for Learning Social Knowledge Graphs[J]. 2015.
[41] Scarselli F , Gori M , AC Tsoi, et al. The Graph Neural Network Model[J]. IEEE Transactions on Neural Networks, 2009, 20(1):61.
[42] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. 2017.

[43] Xie K , Jia Q , Jing M , et al. Data Analysis Based on Knowledge Graph[M]. 2020.
[44] https://www.aphis.usda.gov/aphis/resources/pests-diseases
[45] Gruber T . Ontolingua: A Translation Approach to Providing Portable Ontology Specifications[J]. Knowledge Acquisition, 2012, 5(2):199-220.
[46] Noy N F , Crubezy M , Fergerson R W , et al. Protégé-2000: an open-source ontology-development and knowledge acquisition environment.[C]// Proc of Amia Open Source Expo Amia Symposium. 2003.
[47] Liu X , Huang H , Gao F . Dynamic Analysis of Low-carbon Agriculture Research Based on Knowledge Graph Visualization Method[J]. Science and Technology Management Research, 2019.
[48] Xu R , Chen T , Xia Y , et al. Word Embedding Composition for Data Imbalances in Sentiment and Emotion Classification[J]. Cognitive Computation, 2015, 7(2):226-240.
[49] Gharibi M , Zachariah A , Rao P . FoodKG: A Tool to Enrich Knowledge Graphs Using Machine Learning Techniques[J]. Frontiers in Big Data, 2020, 3:12.
[50] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.
[51] Garijo D , Osorio M , Khider D , et al. OKG-Soft: An Open Knowledge Graph with Machine Readable Scientific Software Metadata[C]// 2019 15th International Conference on eScience (eScience). IEEE, 2020.
[52] Graves A , S Fernández, Schmidhuber J . Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition[C]// Artificial Neural Networks: Formal Models & Their Applications-icann, International Conference, Warsaw, Poland, September. DBLP, 2005.
[53] Sundermeyer M , Ney H , Schluter R . From Feedforward to Recurrent LSTM Neural Networks for Language Modeling[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(3):517-529.
[54] D Dai, X Xiao, Lyu Y , et al. Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:6300-6308.
[55] Yu B , Zhang Z , Shu X , et al. Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy[J]. 2019.
[56] Wang S , Yue Z , Che W , et al. Joint Extraction of Entities and Relations Based on a Novel Graph Scheme[C]// Twenty-Seventh International Joint Conference on Artificial Intelligence {IJCAI-18. 2018.
[57] Fu T J , Ma W Y . GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction[C]// ACL. 2019.

Liu Yingying received her BE degree from Network Engineering of Information Engineering University in 2006, received master degree from Information Management of Zhengzhou University. Later, she worked as a lecturer in College of Information Engineering, Henan University of Animal Husbandry and Economy (2010–2020). And now, she is a doctoral student in the Data and Target Engineering college at Information Engineering University. She has published a paper entitled “A Robust Malware Detection System Using Deep Learning on API Calls” in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) and has many articles accepted by key journals and conferences. Her research interests include big data and artificial intelligence.