

Chinese Relation Extraction on Forestry Knowledge Graph Construction

Qi Yue, Xiang Li and Dan Li*

College of Information and Computer Engineering, Northeast Forestry University, Harbin, 150040, China

*Corresponding Author: Dan Li. Email: ld725725@126.com

Received: 17 September 2020; Accepted: 20 October 2020

Abstract: Forestry work has long been weak in data integration; its initial state will inevitably affect the forestry project development and decision-quality. Knowledge Graph (KG) can provide better abilities to organize, manage, and understand forestry knowledge. Relation Extraction (RE) is a crucial task of KG construction and information retrieval. Previous researches on relation extraction have proved the performance of using the attention mechanism. However, these methods focused on the representation of the entire sentence and ignored the loss of information. The lack of analysis of words and syntactic features contributes to sentences, especially in Chinese relation extraction, resulting in poor performance. Based on the above observations, we proposed an end-to-end relation extraction method that used Bi-directional Gated Recurrent Unit (BiGRU) neural network and dual attention mechanism in forestry KG construction. The dual attention includes sentence-level and word-level, capturing relational semantic words and direction words. To enhance the performance, we used the pre-training model FastText to provide word vectors, and dynamically adjusted the word vectors according to the context. We used forestry entities and relationships to build forestry KG and used Neo4j to store forestry KG. Our method can achieve better results than previous public models in the SemEval-2010 Task 8 dataset. By training the model on forestry dataset, results showed that the accuracy and precision of FastText-BiGRU-Dual Attention exceeded 0.8, which outperformed the comparison methods, thus the experiment confirmed the validity and accuracy of our model. In the future, we plan to apply forestry KG to question and answer system and achieve a recommendations system for forestry knowledge.

Keywords: Forestry; KG; RE; BiGRU; dual attention; FastText

1 Introduction

In the big data and intelligent era, internet information resources have exploded exponentially. The World Wide Web has become a huge database that stored data generated by billions of netizens. Traditional web content organizations are loose and difficult, the procession of huge data posed great challenges to the knowledge organization. Knowledge Graph (KG) can help to build relational frameworks from massive data. With its powerful semantic processing and open interconnection capabilities, KG becomes a driving force in the development of artificial intelligence [1].



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

All industries are flooded with redundant data. For a long time, China's forestry informatization construction fund is insufficient and the degree of information integration is too low [2], which has caused the forestry development to lag. Most forestry business application systems are built by data providers themselves. These phenomena have caused information isolated islands and many problems, such as, the inability to share information resources and the serious waste of resources. The forestry initial state and forestry information accuracy will inevitably affect the development and decision-quality of forestry projects [3]. The forestry industry is also faced with massive forestry science data, it is an important requirement of forestry researchers and workers to manage these data flexibly and efficiently. The current search methods of China's current forestry information platforms are using full-text keyword searches [4]. There are four weaknesses in Chinese retrieval methods. Firstly, the retrieval methods matching by the literal meaning, resulting in low precision. Secondly, the retrieval methods pursue recall rate and return too many results. Then, the retrieval methods are not querying in user's NLP, inaccurate understanding of query content. Finally, a separate query for forestry science data is one-sided.

KG will provide efficient resources that are sharing with forestry practitioners. KG helps integrate the forestry information data onto highly dispersed and heterogeneous fragmentation knowledge. Google coined the term of KG dating back to 2012, referring to semantic knowledge usage in Web Search, usually described as "Things, not strings" [5]. KG also refers to Semantic Web Knowledge Bases at first, like Freebase [6], YAGO [7], and Probase [8]. There are existing many famous Chinese KGs, like CN-DBpedia [9] is a large-scale general domain encyclopedia KG, CN-Probase is a large-scale concept KG [10], Zhishi.me [11] is a general KG, XLORE [12] is an encyclopedia KG constructed by Tsinghua University. KG technology for forestry management has been applied for forest species diversity [13], agroforestry knowledge management [14], forest carbon sinks [15], forest health [16], forestry economy [17], and so on. Researchers are devoted to applying KGs the field of agroforestry. For example, Zschocke et al. [18] showed that enriching Darwin Information Typing Architecture (DITA) topics made its content more comprehensible and accessible, and improved sharing with Linked Data techniques by exploiting the mapping and linking to DITA-topics.

There are many challenges in constructing a Chinese forestry KG. First, the differences between Chinese and English languages are very distinct. From a computer perspective, because the Chinese emphasize parataxis and not form, the sentence structure is relatively loose. There are not so many function words in English as the semantic glue between content words but rely on the lexical sequence relationship to implicitly express the sentence structure. The Chinese forestry datasets need to be collected from websites and books. So, they also bring challenges to the entity recognition and relation extraction in the construction of forestry KG.

Based on the above observations, forestry KG will contribute to high-quality and efficient forestry management. We massively collected forestry data, trained relation extraction model on forestry dataset and built the forestry KG. This paper proposed an end-to-end relation extraction method that used Bi-directional Gated Recurrent Unit (BiGRU) neural network and dual attention mechanism. The dual attention includes sentence-level and word-level, capturing relational semantic words and direction words. To enhance the performance, we used the pre-training model FastText to provide word vectors, and dynamically adjusted the word vectors through FastText according to the context. By training the model on forestry dataset and SemEval-2010 Task 8 dataset, the results showed that our model outperformed the comparison methods. The main contributions are summarized as follows:

1. To build the forestry KG, we formed a forestry dataset. The dataset was crawled from Baidu Encyclopedia and Wikipedia; it was also collected from forestry book and forestry industry-standard PDF resources by hand.

2. The FastText-BiGRU-Dual Attention model is composed of BiGRU and Dual Attention. The model focuses on words that have a decisive effect on sentence relation extraction. And captures relational semantic words and direction words. To enhance the performance, it uses the pre-training model FastText to generate word vectors, and dynamically adjusts the word vectors according to the context.
3. Based on the above framework, our model obtained a competitive experimental result on the SemEval-2010 dataset. By training it on forestry dataset, the accuracy and precision exceeded 0.8, the results showed that our model outperformed BiLSTM-Attention, thus confirmed the validity and accuracy of constructing forestry KG. We used graph database Neo4j to store forestry KG and used Django to build forestry KG that contains domain prior knowledge.

2 Related Work

Along with theoretical foundation development, powerful storage, and computing capabilities, the automatic acquisition of knowledge from internet becomes a trend. KnowItAll [19], TextRunner [20], and Never-Ending Language Learner [21] are the same type of knowledge base project, and researchers have applied them in different KG. This paper explores forestry relation extraction technology, KG construction method, and storage. The related work provides a systematic study on the current methods of constructing KG.

2.1 Forestry Knowledge Graph

With the rapid development of forestry science research, forestry science data is constantly increasing. Each part of the forestry data has different degrees of connection with each other, the existing information retrieval technology cannot reflect the data's connection, and cannot identify complex retrieval requirements. Retrieving information accurately and comprehensively efficiently has become one of the urgent problems in forestry information sharing.

Chirici et al. [22] proposed that the K-Nearest Neighbors (K-NN) technique was important for producing spatially contiguous forestry predictions, by using the combination of field and remotely sensed data. Mithal et al. [23] have discussed forestry monitoring and other problems, they used global remote sensing datasets and time-series algorithms to experiment, finally illustrated results of land covered changing detection. Li et al. [24] performed a cross-sectional dataset analysis using the method of data envelopment analysis to investigate the forestry resources efficiency. Czimber et al. [25] introduced a novel decision support system (DSS), based on geospatial data analyses that they developed for the Hungarian forestry and agricultural sectors by using machine learning techniques (Maximum likelihood and Fuzzy logic). Han et al. [26] used the thesaurus in the address field and the forestry fields as the experimental object, and used the Bootstrapping algorithm to design an automatic seed generation algorithm based on the thesaurus.

There are few research applications of KGs in the field of forestry verticals. In terms of theoretical research, Miao [27] used the forestry patent data as the data source and used the KG analysis method to display the KG of forestry patent data from the year, the applicant, and keyword aspects. Chen et al. [28] used the visualization KG software CiteSpace to sort out the research literature on natural forest protection engineering in China. Most of the forestry studies based on KGs in the past used scientific KG visualization tools to analyze scientific data, objectively showed the current status and future development of forestry research, and cannot make full use of unstructured and structured texts data of forestry, nor can existing forestry data be effectively managed.

2.2 Knowledge Graph Construction

KG is supported by Knowledge Base (KB) [29] such as Freebase and DBpedia. KG can semantically label and correlate data resources, establish a relational network by deep semantic analyzing and mining.

Relying on powerful semantic processing capabilities and open interconnection capabilities to visualize interfaces for users. KG is essentially a semantic network and great achievements of Semantic Web [30] technology. The semantic KG system is formed by extracting interrelated entities from the World Wide Web [31] through a certain process.

There are two ways to construct KG: top-down and bottom-up [32]. In the top-down process, first, it learns entity from structured data and hierarchical relation extraction, and related rules, then carries out the process of entity learning, and incorporates entities into the previous concept system. The bottom-up construction process is the opposite of top-down, it starts with the inductive entity, further abstracts, and gradually forms a layered conceptual system [33]. In the early days of KG techniques, almost all participating companies and institutions built basic KGs in a top-down manner.

For example, the Freebase project and DBpedia project used Wikipedia as a core data source. With the automatic knowledge extraction and process techniques being continuously matured, most researchers used the bottom-up ways to build KGs currently. The most typical example, such as the public Google Knowledge Vault [34]. Usually, the data source can be the massive internet web pages, and then use the existing KB to build; researchers enriched and improved the KGs by automatically extracting knowledge at last. In the process of building the KG, deep learning framework has accelerated the emergence of novel NER and RE methods. Researchers studied entity relation extraction based on deep learning framework, provided support for the application of hot deep learning to KG, and studied segment-level Chinese NER method based on neural network. Since the neural network model automatically learns sentence features without complex feature engineering, many works used relevant neural network models to construct KG. Garcez et al. [35] used the Deep Belief Network to extract knowledge and build KG. Tab. 1 has summarized methods applied for KG construction, including word vector training, named entity recognition and relation extraction.

Table 1: The methods applied for knowledge graph construction

Methods	Example methods
Word vector training	CBOW [36], Skip-gram Model [37], Huffman Tree [38]
Named entity recognition	RNN, BiRNN, BiLSTM-CRF [39], BiLSTM-CNN-CRF [40]
Relation extraction	(1) Supervised learning method (SVM [41], Bayesian-network learning algorithms [42]) (2) Weakly supervised learning method (APCNN [43], CNN-RL [44], Bootstrapping [45]) (3) Deep learning method (CNN-ATT [46], SDP-BLSTM [47], GCNs [48], LSTM-RNN model [49])

2.3 Relation Extraction

Deep learning methods have been widely applied to NLP. Deep neural networks have also been successfully applied to NER and entity RE methods, and achieved good results. Compared with traditional statistical models, deep learning-based methods take the vector of words in the text as input, and implement end-to-end feature extraction through neural networks, which no longer depend on artificially defined features.

The relation can obtain from a discrete text *corpus* data source but it lacks the semantic information. The early research methods of relation extraction mainly consist of artificially grammar and semantic regulations. The pattern matching [50] method was used to identify the relationship between entities. Then academic communities and institutes have started to adopt statistical ML methods [51–53] to suit the rules. Kambhatla et al. used the lexical, syntactic and semantic features in natural language to model the entity relation [54], and successfully implemented the entity relation extraction without the hard coding by the maximum entropy method. Subsequently, many supervised learning methods appeared to make sure algorithm effective and researchers usually manually label big *corpus* as a training set.

In 2007, Turing Center proposed an open information extraction (OIE) framework [55] for open domains to solve the key problem of this constraint extraction technology to practical application. Later, researchers have gradually turned to semi-supervised and unsupervised learning. Carlson et al. mentioned an automatically extracting entity relationships method, which combines semi-supervised learning with the Bootstrap algorithm [56]. However, in practical applications, it is very difficult to define a perfect system of entity relation classification. The open domain-oriented relation extraction technology directly used the relational vocabulary in the *corpus* to model the entity relation [57], Wu et al. used this technology based on OIE. Zeng et al. exploited a convolutional Deep Neural Network (DNN) to extract lexical and sentence level features [58], and found it performed better than traditional statistical ML.

The most important features of the deep learning framework neural network model are the feature representation of words and the automatic learning of features. The neural network model can retain all the features of the text. The most critical use of the neural network model to achieve relation extraction is to select the appropriate word vector according to the task, and then use the different network models, such as PCNN, CNN, RNN, and LSTM, to automatically extract features and abandon complex artificial feature engineering [59]. Wang et al. did an experiment on the SemEval-2010 dataset, BiLSTM-Attention also reached 0.84. Yogatama D et al. focused on self-attention models and recurrent neural network models [60] to extract factual knowledge from a *corpus*, such as the BERT model. Our work improved on the BiGRU model, combined FastText and dual Attention to extract relationship from forestry dataset, and finally formed a forestry KG.

2.4 Knowledge Graph Storage

The scale of KG grows faster, data management becomes more important. To better store ternary arrays, a graph database for managing attribute graphs has been developed in the database field. At present, although no database is currently recognized as the dominant knowledge graph database; it is foreseeable that with ternary the integration and development of groups and graph databases [61], the storage and data management methods of knowledge graphs are becoming richer and more powerful.

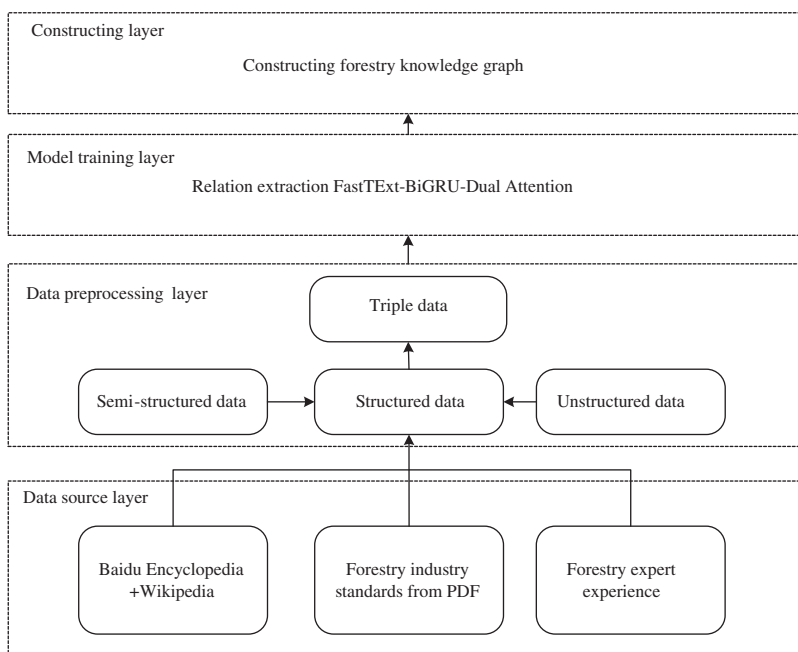
From the data model perspective, KG is a graph constructed by representing each item [62]. To visualize KG in different fields, it must follow the corresponding data model, as mentioned in this paper, the triples of the forestry KG are expressed as (h, r, t) . The mainstream KG database now includes relational database-based storage solutions, RDF-oriented ternary databases, and native graph databases. There are many popular graph databases, like ArangoDB, TitanDB [63], Neo4j [64] and OrientDB [65]. The most important features of databases for a complete and effective application, such as flexible schema, query language, and scalability. Tab. 2 has summarized the comparison of Graph database features. We presented the legend for comparison according to our experiences and literature review, 1–4 represents bad, average, good and great. The advantages of the Neo4j graph database, the query speed of long-range relationships of Neo4j is fast in terms of performance; Neo4j is good at finding a hidden relationship. Neo4j belongs to the native graph database. The storage backend used by Neo4j is customized and optimized for the storage and management of graph structure data. Neo4j is the best choice for storing forestry data.

Table 2: Graph database features comparison

Features\DB	TitanDB	OrientDB	ArangoDB	Neo4j
Flexible Schema	3	3	3	4
Backups	3	3	2	4
Multi-model	3	4	4	2
Multi-architecture	4	3	4	4
Scalability	3	3	4	4
Total	16	16	17	18

3 Methods

Compared with traditional models, deep learning-based methods directly take word vectors in the text as input and implement end-to-end relation extraction through neural networks. We proposed a deep learning-based method FastText-BiGRU-Dual Attention model to extract forestry relation. The model framework is described in Fig. 1.

**Figure 1:** A model framework of constructing forestry KG

This paper constructed forestry KG by using forestry data to improve the efficiency and convenience of forestry knowledge usage. To realize forestry KG construction, we first preprocessed the raw forestry data to structured triples and represented the forestry knowledge. Then we fed entity, relation, and the related sentence into the neural network to train forestry relation extraction.

3.1 Entity Relation Extraction

Entity relation extraction can be roughly divided into the following aspects, as shown in Fig. 2. In this section, we used the FastText-BiGRU-Dual Attention model to extract the forestry relationship.

1. Input layer: input structured forestry information data to the BiGRU-Dual Attention training model.
2. Embedding layer: use the FastText model framework to create word-level low-dimensional forestry entity vectors, and both use word embedding and position embedding results to feed the BiGRU layer.
3. Encoder Layer: utilize BiGRU to get the high-level feature, through the multilayer neural network; the probability of entity relations in each sentence is generated in the training set.
4. Selector layer: that is the dual attention layer, produce a weight vector, and merge word-level features from each time step into a sentence-level feature vector, by multiplying the weight vector.
5. Classified layer: the sentence-level feature vector is finally used for relation classification. Iteratively loop step 2, step 3, and step 4. Finally, train forestry dataset through the model to obtain the prediction accuracy and probability results.

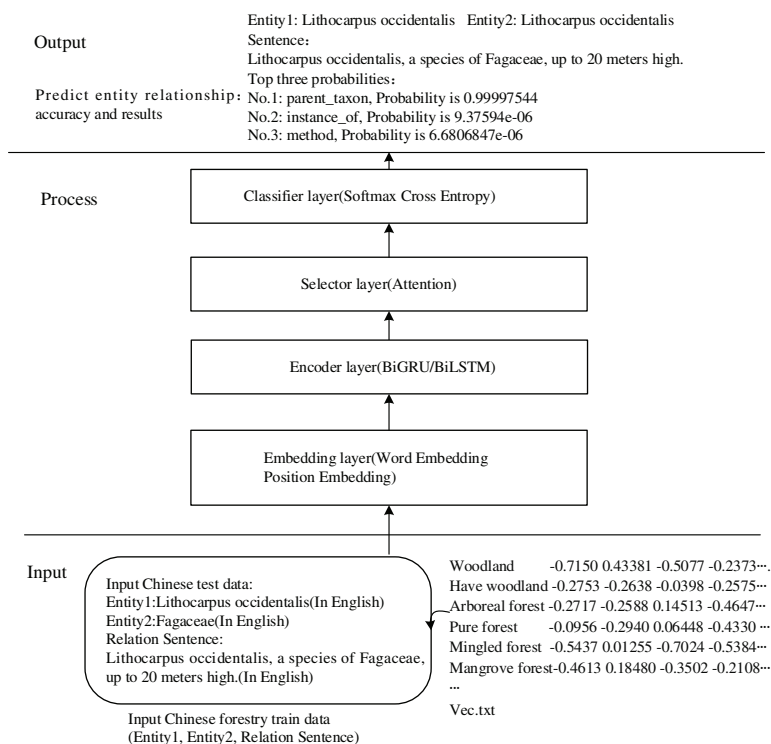


Figure 2: The pipeline of forestry entity relation extraction

3.1.1 FastText

FastText [66] is a text classification and vectorization tool, it calculates the similarity in vectorizing each forestry entity in the sentence for the entity relation extraction. In the FastText model, the input data $(x_1, x_2, \dots, x_{N-1}, x_N)$ represents n-gram vectors in a text, and each feature is the average of the word vectors. FastText architecture is similar to word2vec's CBOW model. CBOW uses context to predict the central word, and FastText uses all n-grams to predict the specified category. In the FastText model, it uses *softmax* as an activation function at the neural network layer. FastText uses all n-grams to predict the probability of a specified category as Eqs. (1) and (2). Among them, w is a word and its context is *context* (w), the

context refers to the words in the left and right windows, subwords, and word n-grams.

$$p(w|context(w)) = \prod_{j=2}^j p(d_j^w | X_w, \theta_{j-1}^w) \quad (1)$$

$$p(d_j^w | X_w, \theta_{j-1}^w) = \begin{cases} \sigma(X_w^T \theta_{j-1}^w), & d_j^w = 1 \\ 1 - \sigma(X_w^T \theta_{j-1}^w), & d_j^w = 0 \end{cases} = [\sigma(X_w^T \theta_{j-1}^w)]^{d_j^w} [1 - \sigma(X_w^T \theta_{j-1}^w)]^{1-d_j^w} \quad (2)$$

And its loss function is Eq. (3), C is the *corpus*.

$$L = \sum_{w \in C} \log p(w|context(w)) \quad (3)$$

FastText has been used in short text classification tasks, entity recognition disambiguation tasks, and feature extraction tasks of text vectorization in recommendation systems, and so on. Practical experience shows that FastText is more suitable for tasks with large samples and many class labels and it can get good results generally. In most cases, it is stronger than the traditional CBOW-LR or SVM classifier [67]. We used the FastText trained by Wikipedia with one billion bytes of information for forestry word vector learning and then used *softmax* as an activation function at the output layer of the neural network to finally generate the forestry expected word vectors.

3.1.2 Word and Position Embedding

Word embedding is a typical representation of words. Words with similar meanings have similar representations. It is a general term with a method of mapping words in a sentence S to a real number vector, $S = \{x_1, x_2, \dots, x_T\}$, every word x_j is converted into a vector e_j . Word embedding converts a word into a fixed-length vector representation for mathematical processing. The word embedding matrix assigns a fixed-length vector representation to each word, and the angle between two-word vectors can be used as a measure of their relationship. Through a simple cosine function, the correlation between two words is calculated efficiently.

The word embedding model cannot learn the order of sequence. If the sequence information is not learned, then the performance of the model will be reduced. Position embedding can solve the problem of learning non-sequential information. By combining the position vector and the word vector, certain position information is introduced into each word, so that Attention can distinguish the words in different positions, and then learn the position information. Google directly gave Eq. (4) for constructing the position vector [68], the position p is mapped to a position vector of d_{pos} dimension, and the value of this i_{th} element is $PE_i(p)$.

$$\begin{cases} PE_{2i}(p) = \sin(p/10000^{2i/d_{pos}}) \\ PE_{2i+1}(p) = \cos(p/10000^{2i/d_{pos}}) \end{cases} \quad (4)$$

3.1.3 BiGRU

GRU (Gated Recurrent Unit) [69] is a variant Long Short-Term Memory (LSTM) [70] model, it's also an improved model based on Recurrent Neural Network (RNN). In recent years, Bi-directional Long Short-Term Memory (BiLSTM) [71] and Bi-directional Gated Recurrent Unit (BiGRU) have made great steps in Natural Language Processing (NLP), especially in text classification and relation extraction. But RNN has many limitations, such as a serious vanishing gradient problem, too many parameters, and so on. GRU model maintains the performance of LSTM and has the advantages of simple structure, few parameters, and good convergence. Different from LSTM [72], the GRU model has only two gates, update gate and reset gate, namely z and r in Fig. 3. The update gate controls the degree to which the

state information at the previous moment is passed into the current state. The larger the value of the updated gate, the more the status information of the previous moment is passed in. The reset gate controls the degree of forgetting the status information of the previous moment. Fig. 3 shows the illustration of LSTM and GRU.

GRU can only obtain the forward context information, ignore the following backward information, and cannot use multi-directionally GRU. A Bi-directional GRU neural network is used in the forestry relation

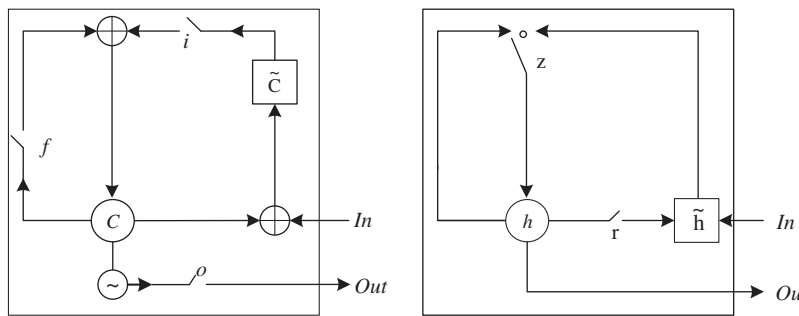


Figure 3: This is a comparison figure of GRU cell and LSTM cell

extraction experiment. That is, the context information is obtained simultaneously from the front to the back; In this way, it can improve the accuracy of extraction. And the BiGRU has the advantages of small dependence on word vectors, low complexity, and fast response time. Its structure likes Fig. 4.

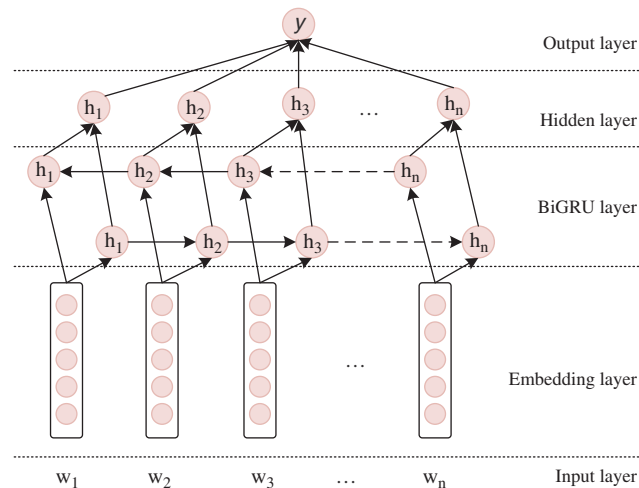


Figure 4: The architecture of Bidirectional Gated Recurrent Unit

Forestry vectors are fed into the BiGRU layer. The BiGRU processes the input sequences x_n in both directions and two sublayers. Then it calculates separately forward sequences \vec{h}_n and backward hidden sequences \overleftarrow{h}_n at the BiGRU layer, and then combine them to calculate the current hidden state h_n and the output y_i of BiGRU, as shown in the following Eq. (5).

$$\vec{h}_n = GRU(x_n, \vec{h}_{n-1})$$

$$\begin{aligned}
\overleftarrow{h}_n &= GRU(x_n, \overleftarrow{h}_{n-1}) \\
h_n &= W^T \overleftarrow{h}_n + W^V \overleftarrow{h}_n \\
y_i &= \text{sigmoid}(W^i h_n + b_i)
\end{aligned} \tag{5}$$

In the BiGRU model, W^T and W^V are weight coefficients that point to the forward hidden state \overleftarrow{h}_n and the reverse hidden state \overleftarrow{h}_n . Finally, the outputs of the BiGRU neural network are fed to the classifier for attribute prediction.

3.1.4 Dual Attention Mechanism

Attention mechanism [71] is an important concept in deep learning. The goal of the attention mechanism is to ignore unimportant information from massive information, selectively filtering out a small amount of important information and focusing. Its purpose is to reduce the loss of key information on text sequence. When RNN is processing, the attention mechanism first calculates the weight of each feature in the sentence and then applies the weight to the feature. The attention mechanism weights matrix w is calculated by adding the product of different initialization probability weights assigned by Attention mechanism and output vector at each time step of the BiGRU layer, and finally calculating the value through softmax function. In Eq. (6), H represents a matrix consisting of vectors $[h_1, h_2, \dots, h_n]$, where n is the sentence length. The sentence S is formed by a weighted sum of these output vectors h_i , where $H \in R^{d^w \times n}$, d_w is word vector dimension:

$$\begin{aligned}
M &= \tanh(H) \\
\alpha &= \text{softmax}(w^T M) \\
S &= H\alpha^T
\end{aligned} \tag{6}$$

After the BiGRU network processing, a word-level attention mechanism, and a sentence-level attention mechanism are added, so that the model can encode the input sentence through the BiGRU network to obtain bidirectional contextual semantic information of the training instance, and focus on the word-level attention mechanism. Then uses the sentence-level attention mechanism to describe the same entity pair. It calculates the correlation between each instance and the corresponding relationship, and dynamically adjust the weight of each instance to reduce the noise data impact and make full use of the semantic information contained in each instance. Fig. 5 shows the architecture of the BiGRU-Dual Attention model.

For describing forestry entity pair $\langle e_i, e_j \rangle$ in instance set $E = (V_1, V_2, V_3, \dots, V_N)$, where V_i is the instance vector. Set E is used for relation classification. The calculation of the instance set vector E will depend on each instance vector V_i in the set and weighted by the instance-level attention weight α_{V_i} . Where α_{V_i} measures the degree of correlation between the input entity V_i and its corresponding relationship r , and calculation formulas are Eqs. (7) and (8).

$$\alpha_{V_i} = \text{softmax}(V_i A_r) \tag{7}$$

$$E = \sum_i \alpha_{V_i} V_i \tag{8}$$

Then, we calculated the conditional probability of the predicted relation through the *softmax* function layer and its calculation formulas are Eqs. (9) and (10). R is the relation representation matrix, and b is the offset vector. Finally, use the *argmax* function to get the final relation prediction \hat{r} .

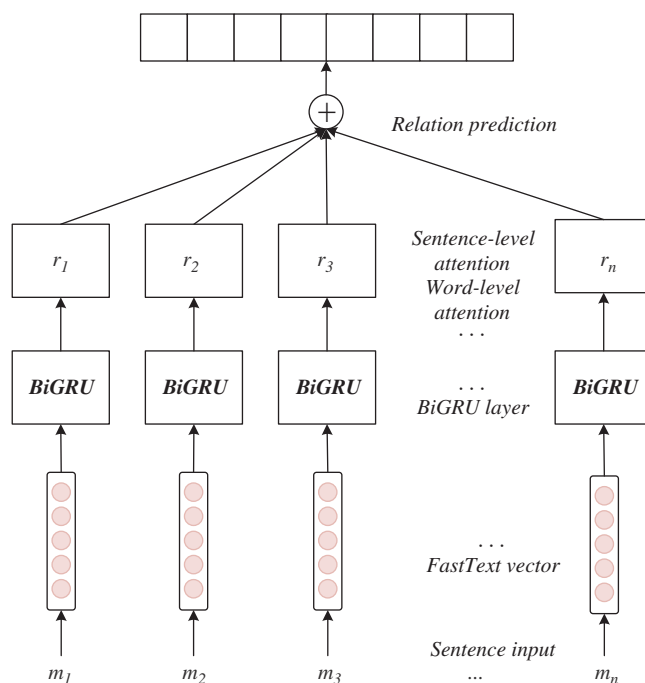


Figure 5: BiGRU-Dual Attention relation extraction model

$$P(r|E) = \text{softmax}(RE + b) \quad (9)$$

$$\hat{r} = \text{argmax} P(r|E) \quad (10)$$

Due to the effectiveness of the attention mechanism, we used FastText to vectorize forestry entity words and combined BiGRU with the dual attention mechanism to implement relation extraction experiments on forestry dataset to achieve better performance.

3.2 Knowledge Graph Construction

This paper adopts a bottom-up approach to construct forestry KG. First, inductively organizes entities to form the underlying concept, and then gradually abstracts online to form the upper-level concept. This method transforms data patterns based on existing industry standards. The construction of the forestry KG mainly depends on the Baidu Encyclopedia and Wikipedia, which have rich entities and concept information. In the forestry text, use relation extraction algorithms to extract certain types of relationships and uncertain types of relationships. The storage design of forestry KG needs to take into account the map data of entities, encyclopedia texts and entity relationships. It is recommended using the graph database Neo4j to store forestry KG.

4 Dataset and Experiments

This section is designed to test the effectiveness of the FastText-BiGRU-Dual Attention model and the bottom-up forestry KG construction approach. Relation extraction is implemented by deep learning method as the FastText-BiGRU-Dual Attention model in above section. The experiment trained relation extraction model on forestry dataset. Finally, constructed forestry KG and visualized it.

4.1 Dataset

We chose a common representative dataset SemEval-2010 Task 8 and a self-collected forestry dataset. Since SemEval-2010 Task 8 is a widely used dataset for relation extraction in previous researches, we used it to evaluate our model performance and analyze the effectiveness of our model. To illustrate the effectiveness of the method in forestry relation extraction, we conduct a comparative experiment on the forestry dataset. The forestry entities may refer to trees, the animals in forestry, diseases, symptoms, and pests. To get the forestry dataset, we crawled related items from Baidu Encyclopedia (baike.baidu.com) and Forestry Net (www.forestry.gov.cn). We also transferred Forestry Industry Standard PDF resources and many professional definitions in forestry books. The processing of unstructured forestry data is complex and important. The following are the processing steps:

1. First, cleaned dirty data and used regular expressions to process JavaScript, CSS styles, etc. To make the training data standardized and reduce the error of special characters, SQL statements are used to remove the irregular special characters and spaces in the text.
2. Second, defined structured information and summarized the types of entities to be extracted based on the forestry field. For example, the forestry field has entity tags such as trees and forests; summarized the relationship between entity tags to facilitate the extraction of the next relationship.
3. The NLP model cannot directly process documents in image formats such as PDF, and usually requires PDF Reader to convert documents in image format into text documents;
4. The document may be very long, usually the document is divided into paragraphs, and forestry-related paragraphs are intercepted;

Finally, we extracted 23838 concepts, 17948 entities, and 12346 relationships as the forestry dataset. After obtaining the raw textual data, we applied the relation extraction model to train the raw data. Then judged the category of relationship, represented the relations as a form of triples (h, r, t) , h represents the head entity, t represents the tail entity, and r represents the relationship between two entities. We extracted the most common relationship in forestry dataset between these entities like “rank”, “category”, “instance of”, “subclass of”, “description”, “parent taxon” and “method”. After processing, formed `realtion2id.txt`, `vextor.txt`, `test.txt`, and `train.txt`.

4.2 Model Training

Most of the works on relation extraction used the SemEval-2010 Task 8 dataset. To show the efficiency of the model, firstly the experiment applied our model to the SemEval-2010 Task 8 dataset. Secondly, to verify the efficiency of our method in Chinese forestry knowledge, this paper conducted a comparative experiment on the self-collected forestry dataset. Follow the previous work [73,74] and we adopted Precision, Recall, F1 score, and Accuracy rate for evaluating our model performance.

The Dual attention model focuses on sentences to calculate the relationship between entity pairs. In [Tab. 3](#), we showed examples of visualizing dual attention weights on the SemEval-2010 Task 8.

In our experiment, The FastText was used to generate word vectors. The position embedding was randomly initialized from Gaussian distribution. The detail of the hyper-parameters setting in experiment is shown in [Tab. 4](#).

4.3 Results on SemEval-2010 Task 8

Since our method combined Dual attention on BiGRU and FastText, this paper chose some representative models to compare. We compared our method with baselines which are recently published for SemEval-2010 Task 8. All of these methods are relying on the original sentences which will lose some important information. However, FastText-BiGRU-Dual Attention can make full use of the

characteristics of original sentences. The results of different models are shown in Tab. 5. The WE remark refers to Word embedding. The PE, PI remark refers to Word around nominals, Position embedding, Position indicator. The GR, POS refers to grammar relation, part of speech. It listed the model name, features set and F1-score. The FastText-BiGRU-Dual Attention achieves an F1-score of 84.76% on SemEval-2010 Task 8. Our model was compared with the RNN-based models, our model outperformed.

Table 3: The visualization of dual attention weights on Semeval-2010 task 8

	Sentence	Relationship
Original	<e1>burst</e1>, <e2>pressure</e2>, The burst has been caused by water hammer pressure, Cause-Effect	<e1>burst</e1> <e2>pressure</e2>
$\vec{\alpha}$	The burst has been caused by water hammer pressure.	Cause-Effect (e1,e2)
$\bar{\alpha}$	The burst has been caused by water hammer pressure.	
Original	<e1>stable</e1>, <e2>hounds</e2>, She soon had a stable of her own rescued hounds, Member-Collection	<e1>stable</e1> <e2>hounds</e2>
$\vec{\alpha}$	She soon had a stable of her own rescued hounds.	Member-Collection (e1,e2)
$\bar{\alpha}$	She soon had a stable of her own rescued hounds .	

Table 4: Hyper-parameters setting

Embedding dimension	100
Position size	82
Position dimension	25
Hidden dimension	200
Batch	128
Epochs	200

Table 5: Comparison with other RNN-based methods on Semeval-2010 Task 8

Paper	Model Name	Features Set	F1-score
Zhang et al. [75]	BiLSTM	WE, PE	84.30%
Zhou et al. [71]	Attention-BiLSTM	WE, PI	84.00%
Xu et al. [76]	DRNNs	WE, GR, POS,	84.16%
Lee et al. [77]	BiLSTM-Entity-Aware Attention	WE, PE, Attention	84.70%
Our Paper	FT-BiGRU-Dual Attention	WE, PE, Dual Attention	84.76%

4.4 Results on Forestry Dataset

Another experiment was applied the Attention mechanism to BiLSTM and BiGRU for comparative experiments of forestry relation extraction. And the experiment used FastText to generate word vector, BiLSTM/BiGRU, and Dual Attention for comparison experiments on the self-collected forestry dataset. The forestry dataset was randomly divided into a train set and a test set, we used the train set to train the

model, and used the test set to evaluate the model's accuracy and recall rate, etc. The performance of the model will be represented by the average value of the evaluation metric for each experiment.

From Fig. 6, the FastText model with higher accuracy. the vectorization of FastText in forestry relation extraction has better performance. Due to the FastText vectors are 1200-dimensional, and FastText can learn the representation of a word in isolation. The curve of BiGRU-Dual Attention fluctuates a bit, and it fluctuates up and down relatively gently. The BiGRU model maintains the performance of LSTM and has the advantages of simple structure, few parameters, and good convergence.

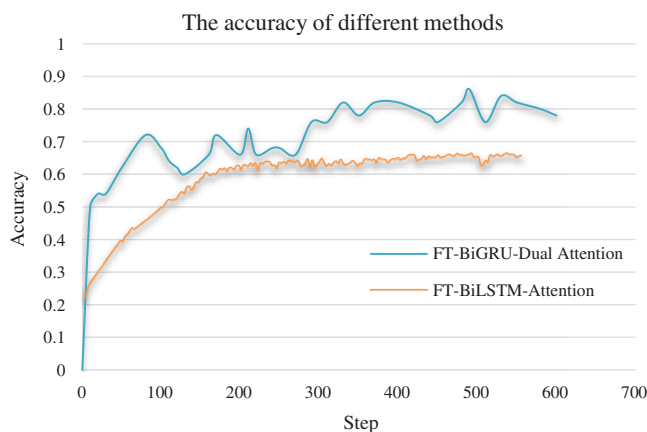


Figure 6: The Accuracy of BiGRU-Dual Attention and BiLSTM-Attention

Obviously, the performance of BiGRU-Dual Attention is better than BiLSTM-Attention. From the measurement precision and recall, the precision and recall of BiGRU can reach 0.77 and 0.82 respectively, as shown in Tab. 6.

Table 6: Different model performance of relation extraction (%)

Model\Metrics	Precision	Recall	Accuracy
BiGRU-Dual Attention	76.51	82.03	80.23
BiLSTM-Attention	63.64	64.49	64.06

We designed a set of comparative experiments between FastText-BiLSTM-Attention and FastText-BiGRU-Dual Attention. By using the forestry dataset to test, our model has exceeded BiLSTM-Attention above 0.15 in Recall and Precision rate. It means that most forestry relationship can be efficiently extracted from forestry structured or unstructured data.

4.5 Construction of Forestry Knowledge Graph

In this paper, we extracted forestry relations from forestry sentences to construct KG. To complete forestry KG construction, we adopted the best model FastText-BiGRU-Dual Attention to apply to forestry relation extraction. Forestry KG is composed of triples, the nodes which represent forestry entities, and the edges which represent the relationships between two entities. Fig. 7 shows the forestry KG example.

trained the model that supports GPU acceleration to realize relation extraction. Finally, we used the distributed graph database neo4j to store data.

4.6 Discussion

This paper constructed a forestry KG from data collection, data process, relation extraction, and data storage. The process is applicable to construct all KGs are similar to the forestry industry, and the method has universality. Moreover, forestry KG can improve the efficiency of forestry information searching and provide a certain reference value for the accuracy of the construction of a forestry KG. The proposed FastText-BiGRU-Dual Attention model for extracting forestry relationship in the natural field is tested and evaluated. And it makes the accuracy, precision, and recall rate are exceeding other methods. It obviously improves the efficiency of relation extraction.

5 Conclusions

Continuously increasing demand for forestry conservancy information on social production leads to challenges to data integration. The complexity of data involved in forestry conservancy information is increasing, which leads to low data utilization. Constructing forestry KG can realize a layered coverage and node interconnection network based on forestry information network currently. The KG can build links between concepts or entities by extracting knowledge from forestry web pages. Organizing data about forestry-related accumulations has the lowest cost and makes massive data become usable.

To meet the needs of users for the integration of forestry data information, we proposed a bottom-up method of constructing a KG and storing the KG into Neo4j. This paper constructed forestry KG by combining entities and entity relationships. Previous researches on relation extraction methods focused on the representation of the entire sentence and ignored the loss of information. Such as the contributions to words and syntactic features, especially in Chinese relation extraction, resulting in poor performance. A novel deep learning-based method FastText-BiGRU-Dual Attention is adopted to extract forestry relationship. In our experiment, we find that the values of precision and recall are better than other comparative methods, the accuracy and recall rate exceed 0.8. There are some limitations should be addressed in future work. One lies in the partially missing data when constructing the forestry KG and another is that manually reviewed and corrected forestry data. In the future, we would like to mine data more deeply and improve the accuracy of extracting relationships. We plan to apply forestry KG to knowledge question and answer and achieve recommendations for forestry knowledge.

Funding Statement: This work is supported by Heilongjiang Provincial Natural Science Foundation of China (Grant No. LH2019G001), Q. Yue would like to thank the foundation for financial support.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] L. Deng and Y. Liu, *Deep Learning in Natural Language Processing*, Springer, 2018. [Online]. Available: <https://link.springer.com/book/10.1007%2F978-981-10-5209-5>.
- [2] P. Ji, Y. D. Xiao, R. X. Hou and N. J. Zhang, "Application of data integration technology to forestry in China and its progress," *World Forestry Research*, vol. 31, no. 6, pp. 49–54, 2018.
- [3] K. Duvemo and T. Lamas, "The influence of forest data quality on planning processes in forestry, (in English)," *Scandinavian Journal of Forest Research*, vol. 21, no. 4, pp. 327–339, 2006.
- [4] N. J. Zhang, "Study on semantic retrieval based on forestry science data," Ph.D. dissertation. Chinese Academy of Forestry, 2013.

- [5] A. Singhal, "Introducing the knowledge graph: Things, not strings," *Official Google Blog*, vol. 16, 2012. [Online]. Available: <http://googleblog.blogspot.ie/2012/05/introducing-knowledgegraph-things-not.html>.
- [6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ICMD*, Vancouver, Canada, pp. 1247–1250, 2008.
- [7] T. Rebele, F. M. Suchanek, J. Hoffart, J. Biega, E. Kuzey *et al.*, "Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames," in *Proc. ISWC*, Kobe, Japan, pp. 177–185, 2016.
- [8] W. Wu, H. Li, H. Wang and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in *Proc. ICMD*, Scottsdale Arizona, USA, pp. 481–492, 2012.
- [9] B. Xu, Y. Xu, J. Liang, C. Xie, B. Liang *et al.*, "Cn-dbpedia: A never-ending chinese knowledge extraction system," in *Proc. IEA/AIE*, Arras, France, pp. 428–438, 2017.
- [10] J. Chen, A. Wang, J. Chen, Y. Xiao, Z. Chu *et al.*, "Cn-probase: A data-driven approach for large-scale chinese taxonomy construction," in *Proc. ICDE*, Macau SAR, China, pp. 1706–1709, 2019.
- [11] X. Niu, X. Sun, H. Wang, S. Rong, G. Qi *et al.*, "Zhishi.Me: Weaving Chinese linking open data," in *Proc. ISWC*, San Francisco, CA, USA, pp. 205–220, 2011.
- [12] Z. Wang, J. Li, Z. Wang, S. Li, M. Li *et al.*, "Xlore: A large-scale English-Chinese bilingual knowledge graph," in *Proc. ISWC*, Sydney, Australia, pp. 121–124, 2013.
- [13] F. Skov and J. Svenning, "Predicting plant species richness in a managed forest," *Forest Ecology and Management*, vol. 180, no. 1, pp. 583–593, 2003.
- [14] M. E. Isaac, E. Dawoe and K. Sieciechowicz, "Assessing local knowledge use in Agroforestry management with cognitive maps," *Environmental Management*, vol. 43, no. 6, pp. 1321–1329, 2009.
- [15] Z. Zhang, Y. Q. Shen, F. Long, Z. Zhu and X. R. He, "Knowledge mapping of research on forest carbon sinks," *Journal of Zhejiang AF University*, vol. 30, no. 4, pp. 567–577, 2013.
- [16] M. H. Shi, C. W. Zhao, Z. H. Guo and S. R. Liu, "Forest health assessment based on self-organizing map neural network: A case study in Baihe forestry bureau, Jilin province," *Chinese Journal of Ecology*, vol. 30, no. 6, pp. 1295–1303, 2011.
- [17] Q. F. Duan, "Knowledge mapping analysis of forestry economy in China," *Forestry Economics*, vol. 1, no. 4, pp. 115–119, 2013.
- [18] T. Zschocke and S. Closs, "Preliminary exploration of using RDF for annotating DITA topics on agroforestry," in *Proc. MSR*, Zurich, Switzerland, pp. 276–288, 2012.
- [19] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. M. Popescu *et al.*, "Web-scale information extraction in knowitall: (preliminary results)," in *Proc. WWW*, New York, NY, USA, pp. 100–110, 2004.
- [20] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead *et al.*, "Textrunner: Open information extraction on the web," in *Proc. NAACL-HLT*, Rochester, NY, USA, pp. 25–26, 2007.
- [21] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka *et al.*, "Toward an architecture for never-ending language learning," in *Proc. AAAI 2010*, Atlanta, Georgia, USA, pp. 1306–1313, 2010.
- [22] G. Chirici, M. Mura, D. McInerney, N. Py, E. O. Tomppo *et al.*, "A meta-analysis and review of the literature on the k-nearest neighbors technique for forestry applications that use remotely sensed data, (in English)," *Remote Sensing of Environment*, vol. 176, pp. 282–294, 2016.
- [23] V. Mithal, A. Garg, S. Boriah, M. Steinbach, V. Kumar *et al.*, "Monitoring global forest cover using data mining, (in English)," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 4, pp. 36, 2011.
- [24] L. Li, T. T. Hao and T. Chi, "Evaluation on China's forestry resources efficiency based on big data, (in English)," *Journal of Cleaner Production*, vol. 142, pp. 513–523, 2017.
- [25] K. Czimer and B. Galos, "A new decision support system to analyse the impacts of climate change on the Hungarian forestry and agricultural sectors, (in English)," *Scandinavian Journal of Forest Research*, vol. 31, no. 7, pp. 664–673, 2016.
- [26] Q. C. Han, Y. W. Zhao, Z. Yao and L. J. Fu, "Initial seed set automatic generation algorithm for domain knowledge graph based on thesaurus," *Journal of Chinese Information Processing*, vol. 32, no. 08, pp. 1–8, 2018.

- [27] R. Q. Miao, "The development of project management system and knowledge mapping analysis in forestry," M.S. dissertation. Beijing Forestry University, 2016.
- [28] L.R. Chen, Y. K. Cao, Z. F. Zhu and L. Su, "Knowledge mapping analysis of research on Chinese natural forest protection projects," *Journal of AgroForestry Economics and Management*, vol. 14, no. 06, pp. 622–629, 2015.
- [29] J. H. Yan, C. Y. Wang, W. L. Cheng, M. Gao and A. Y. Zhou, "A retrospective of knowledge graphs, (in English)," *Frontiers of Computer Science*, vol. 12, no. 1, pp. 55–74, 2018.
- [30] T. Berners-Lee and J. Hendler, "Publishing on the semantic web," *Nature*, vol. 410, no. 6832, pp. 1023–1024, 2001.
- [31] T. Berners-Lee and M. Fischetti, *Weaving the web: The original design and ultimate destiny of the world wide web by its inventor*. DIANE Publishing Company, Collingdale, PA, USA, 2001.
- [32] L. Tao, W. Cichen and L. Huakang, "Development and construction of knowledge graph," *Journal of Nanjing University of Science and Technology*, vol. 41, pp. 22–34, 2017.
- [33] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar and K. Achan, "Product knowledge graph embedding for e-commerce," in *Proc. WSDM '20*, Houston, TX, USA, pp. 672–680, 2020.
- [34] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao *et al.*, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proc. KDDM*, New York, NY, USA, pp. 601–610, 2014.
- [35] S. N. Tran and A. S. d'Avila Garcez, "Deep logic networks: Inserting and extracting knowledge from deep belief networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 246–258, 2018.
- [36] T. Kenter, A. Borisov and M. De Rijke, "Siamese cbow: Optimizing word embeddings for sentence representations," in *Proc. ACL*, Berlin, Germany, pp. 941–951, 2016.
- [37] A. Lazaridou and M. Baroni, "Combining language and vision with a multimodal skip-gram model," *Arxiv: Computation Language*, 2015. [Online]. Available: <https://arxiv.org/pdf/1501.02598.pdf>.
- [38] L. Perros-Meilhac and C. Lamy, "Huffman tree based metric derivation for a low-complexity sequential soft vlc decoding," in *Proc. ICC*, New York, NY, USA, pp. 783–787, 2002.
- [39] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural architectures for named entity recognition," in *Proc. NAACL*, San Diego, California, USA, pp. 260–270, 2016.
- [40] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf." *Arxiv: Learning*, 2016. [Online]. Available: <https://arxiv.org/pdf/1603.01354.pdf>.
- [41] B. Rink, S. M. Harabagiu and K. Roberts, "Automatic extraction of relations between medical concepts in clinical texts," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 594–600, 2011.
- [42] J. A. D. M. Brito, L. S. Ochi, F. Montenegro and N. Maculan, "An iterative local search approach applied to the optimal stratification problem," *International Transactions in Operational Research*, vol. 17, no. 6, pp. 753–764, 2010.
- [43] M. Mintz, S. Bills, R. Snow and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. IJCNLP*, Suntec, Singapore, pp. 1003–1011, 2009.
- [44] J. Feng, M. Huang, L. Zhao, Y. Yang and X. Zhu, "Reinforcement learning for relation classification from noisy data," in *Proc. IAAI-18*, New Orleans, Louisiana, USA, pp. 5779–5786, 2018.
- [45] C. Zhang, W. Xu, Z. Ma, S. Gao, Q. Li *et al.*, "Construction of semantic bootstrapping models for relation extraction," *Knowledge Based Systems*, vol. 83, no. 83, pp. 128–137, 2015.
- [46] L. Wang, Z. Cao, G. De Melo and Z. Liu, "Relation classification via multi-level attention CNNs," in *Proc. ACL*, Berlin, Germany, pp. 1298–1307, 2016.
- [47] P. Wang, Z. Xie and J. Hu, "Relation classification via CNN, segmented max-pooling, and SDP-BLSTM," in *Proc. ICONIP*, Guangzhou, China, pp. 144–154, 2017.
- [48] Y. Zhang, P. Qi and D. M. Christopher, "Graph convolution over pruned dependency trees improves relation extraction," in *Proc. EMNLP*, Brussels, Belgium, pp. 2205–2215, 2018.
- [49] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," in *Proc. ACL*, Berlin, Germany, pp. 1105–1116, 2016.
- [50] N. Tongtep and T. Theeramunkong, "Pattern-based extraction of named entities in thai news documents," *Thammasat International Journal of Science*, vol. 15, no. 1, pp. 70–81, 2010.

- [51] A. Culotta and J. S. Sorensen, "Dependency tree kernels for relation extraction," in *Proc. ACL*, Barcelona, Spain, pp. 423–429, 2004.
- [52] Y. Peng, A. Rios, R. Kavuluru and Z. Lu, "Chemical-protein relation extraction with ensembles of SVM, CNN, and RNN models." *Arxiv: Computation Language*, 2018. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1802/1802.01255.pdf>.
- [53] J. Q. Guo, S. B. Liu and X. Liu, "Construction of visual cognitive computation model for sports psychology based on knowledge atlas, (in English)," *Cognitive Systems Research*, vol. 52, pp. 521–530, 2018.
- [54] N. Kambhatla, "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations," in *Proc. ACL-IPDS*, Barcelona, Spain, pp. 22, 2004.
- [55] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead and O. Etzioni, "Open information extraction from the web," in *Proc. IJCAI*, Hyderabad, India, pp. 2670–2676, 2007.
- [56] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr and T. M. Mitchell, "Coupled semi-supervised learning for information extraction," in *Proc. WSDM*, New York City, USA, pp. 101–110, 2010.
- [57] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proc. ACL*, Uppsala, Sweden, pp. 118–127, 2010.
- [58] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. COLING*, Dublin, Ireland, pp. 2335–2344, 2014.
- [59] A. G. Luo, S. Gao and Y. J. Xu, "Deep semantic match model for entity linking using knowledge graph and text, (in English)," *Proceedings of the International Conference on Identification, Information and Knowledge in the Internet of Things*, vol. 129, pp. 110–114, 2018.
- [60] D. Yogatama, C. D. M. Dautume, J. Connor, T. Kociský, M. Chrzanowski *et al.*, "Learning and evaluating general linguistic intelligence." *Arxiv: Learning*, 2019. [Online]. Available: <https://arxiv.org/pdf/1901.11373.pdf>.
- [61] I. Robinson, J. Webber and E. Eifrem, *Graph databases*. O'Reilly Media, Inc., Boston, MA, USA, 2013.
- [62] Y. Duan, L. Shao, G. Hu, Z. Zhou, Q. Zou *et al.*, "Specifying architecture of knowledge graph with data graph, information graph, knowledge graph and wisdom graph," in *Proc. SERA*, London, United Kingdom, pp. 327–332, 2017.
- [63] V. Mishra, "Titan graph databases with Cassandra," in *Beginning Apache Cassandra Development*. Boston, MA, USA: Apress, pp. 123–151, 2014.
- [64] F. Holzschuher and R. Peinl, "Performance of graph query languages: Comparison of cypher, gremlin and native access in neo4j," in *Proc. EDBT/ICDT*, Genoa, Italy, pp. 195–204, 2013.
- [65] C. Tesoriero, *Getting Started with Orientdb*, Packt Publishing Ltd, 2013. [Online]. Available: <https://proquest.safaribooksonline.com/9781782169956>.
- [66] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jegou *et al.*, "Fasttext.Zip: Compressing text classification models." *arXiv: Computation and Language*, 2017. [Online]. Available: <https://arxiv.org/pdf/1612.03651.pdf>.
- [67] A. Kołcz and C. H. Teo, "Feature weighting for improved classifier robustness," in *Proc. CEAS*, Redmond, Washington, USA, 2009.
- [68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.*, "Attention is all you need," in *Proc. ANIPS*, California, USA, pp. 5998–6008, 2017.
- [69] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling." *Arxiv: Neural Evolutionary Computing*, 2014. [Online]. Available: <https://arxiv.org/pdf/1412.3555.pdf?ref=hackernoon.com>.
- [70] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [71] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. ACL*, Berlin, Germany, pp. 207–212, 2016.
- [72] Z. Geng, G. Chen, Y. Han, G. Lu and F. Li, "Semantic relation extraction using sequential and tree-structured LSTM with attention," *Information Sciences*, vol. 509, pp. 183–192, 2020.
- [73] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss *et al.*, "Cotype: Joint extraction of typed entities and relations with knowledge bases," in *Proc. WWW*, Perth, Australia, pp. 1015–1024, 2017.

- [74] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou *et al.*, “Joint extraction of entities and relations based on a novel tagging scheme,” in *Proc. ACL*, Vancouver, Canada, pp. 1227–1236, 2017.
- [75] S. Zhang, D. Zheng, X. Hu and M. Yang, “Bidirectional long short-term memory networks for relation classification,” in *Proc. PACLIC*, Shanghai, China, pp. 73–78, 2015.
- [76] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen *et al.*, “Improved relation classification by deep recurrent neural networks with data augmentation.” *Arxiv: Computation and Language*, 2016. [Online]. Available: <https://arxiv.org/pdf/1601.03651.pdf>.
- [77] J. Lee, S. Seo and Y. S. Choi, “Semantic relation classification via bidirectional LSTM networks with entity-aware attention using latent entity typing,” *Symmetry*, vol. 11, no. 6, pp. 785, 2019.