2020 International Conference on Information Science and Education (ICISE-IE) | 978-1-6654-2261-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICISE51755.2020.00094

PaintKG: the painting knowledge graph using bilstm-crf

[1st] Han Wu
Computer Science and Engineering
Beijing Information Science Technology University
Beijing, China
unchained_hank@foxmail.com

[2nd] Shuo Yan Liu
Computer Science and Engineering
Beijing Information Science Technology University
Beijing, China
panilsy@icloud.com

[3rd] Wenkai Zheng
Computer Science and Engineering
Beijing Information Science Technology University
Beijing, China
3531104786@qq.com

[4th] Yifu Yang
Computer Science and Engineering
Beijing Information Science Technology University
Beijing, China
y2000613@yeah.net

[5th] Han Gao
Computer Science and Engineering
Beijing Information Science Technology University
Beijing, China
840119580@qq.com

***Abstract*—One of the most important elements of culture is painting, which reserves ethnic and national immaterial relics. However, Chinese Web is currently short of a specialized knowledge graph on museums and it still crave accesses to effectually assimilate extensive types of discrete knowledge for applications. To facilitate museum knowledge sharing, we propose a painting knowledge graph, namely PaintKG, utilizing Bi-LSTM with CRF layer which obviously rises the F-1 measure to recognize and extract knowledge and relations of different types of paintings and their painters from existing encyclopedia and unstructured Web text data and physically restore them in neo4j as graph data eventually. What's more, we depict and demonstrate typical scenarios of PaintKG and implement them by real-world applications, such as painting recommendation, painter entity association.**

Keywords—Knowledge graph, Painting, Bi-LSTM, CRF

I. INTRODUCTION

Painting is ethnic and national immaterial relic which reflects the ancient people's inner cognition of nature, society, philosophy, politics, religion, morality , literature and art. It is desired to reserve national culture and improve the aesthetic of people. A painting knowledge repository of graph will function as a solution to attain these targets.

As the Semantic Web grows and expands, structured knowledge in distinct domains has been produced and broadcasted as linked data online. Linking Open Data however, is still short of a specialized and typical knowledge graph on painting and it still crave accesses to effectually assimilate extensive discrete plain text information for applications. To facilitate painting knowledge sharing, we aim to build the first Chinese painting knowledge graph named PaintKG. Knowledge on painting does exist in some large-scale encyclopedia, like Wikipedia. To amass and assimilate extensive knowledge effectually, we develop a regular expression crawler program driven by python and its scrapy library to collect meta-data trying to recognize entities from within and extract relations and properties of distinct entities via Bi-LSTM and CRF.

We illustrate a painting knowledge graph named PaintKG in this paper, containing the knowledge of painters and paintings including their name, figure, era etc. Besides the dataset PaintKG itself, there are two goals of this demo system which can be summarized as follows:

- Painting entity retrieval: PaintKG offers the entity attribution and relation retrieval in multiple methods. It's available for professional or common users to search the painting entities or reveal relation by offering specific options and key words.

- Knowledge graph performing application: We used elastic search to build a search engine that accelerates search query time from database. And a distributed web application was constructed to provide data access and more user-friendly interface via spring data, spring cloud, vue.js, and roc-graph.

II. SYSTEM OVERVIEW AND DEVELOPMENT METHOD

A. System overview

As illustrated in figure 1, PaintKG consists of four key components: (i)crawler engine collects half structured data from Web; (ii) entity recognition identifies the painting entities from organized text; (iii) relation extraction proposes finding the connection of the attributions of entities and extracting relations;

978-1-6654-2261-1/20/$31.00 ©2020 IEEE
DOI 10.1109/ICISE51755.2020.00094

(iv)the application of PaintKG including entity retrieval, access API and web application.

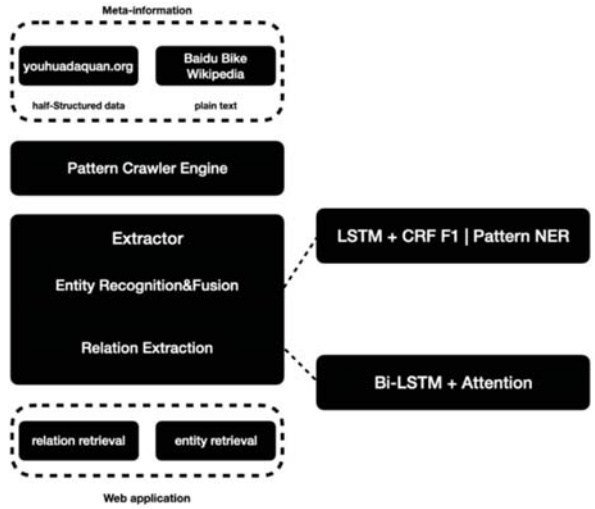

Fig. 1. The framework of PaintKG

B. *crawler engine*

The first development module is crawler engine, which is utilized to crawl taxonomy entities representing painting and their corresponding knowledge images or other attributes from youhuadaquan.org and dpm.org which are half structured and acquire large scale of raw text from Website like Baidu Baike and Wikipedia. Then we integrate collected information via pattern match program into exact raw in SQL to form data rows of entities and its properties.

C. *knowledge collection*

We observation reveals that there are some identifying patterns for features of the entities representing paintings or painters. For example, the labels representing painters contain characters like "." in the middle which inspires us to clarify meta-data via pattern match like regular expression. We manually summarize 20 naming rules for painting and 7 rules for painters. If the feature of an entity fulfill or meet one of these patterns, this entity knowledge can be identified and assembled.

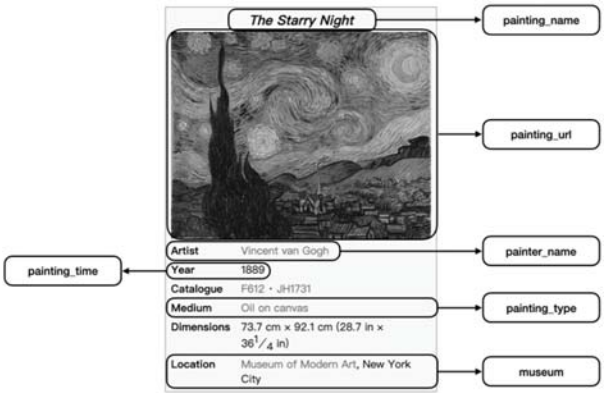

Fig. 2. An Example of Extracted Source Page of Painting

```
Birth Date
((\d+year)－(\d+)year)
(\d+year)birth2[，。]
((\d+)--(\d+))
 (era(\d+year)(~|-)era(\d+)year)
(\d+year(\d+month){0,1}(\d+day){0,1})(.*)(birth|was born at|born from)(.*)
Death Date
[～—~](\d+)year)
--(\d+))
(\d+年(\d+月3){0,1}(\d+日4){0,1})(.*)passed away
Works
.*representative work was([\S]+)。.*
.*has representative work ([\S]+?)。.*
Era
(East Han|West Han|Han|Wei|Shu|Wu|East Jin|West Jin|Jin|Tang|Sui|South Song|North
Song|Song|Yuan|Ming|Qing|Republic of China|Ancient|Modern)([\u4e00-\u9fa5]{2,10})Dynasty
Nationality
(India/Japan/China/Russia/Italy/Germany/France/Belgium/UK/Republic of China)
painting Style
has (.*?) style
a painting of (.*?) style
painting Date
was (produced|created|painted|built|drew)at.*(。|，)
was created by (.*) as  .*
created (at .*.*?)
[，|。](.*era)creation[，|。]
(was painted at \\d*)[，|。]
```

Fig. 3. Naming rules' pattern of clarified data collection.

D. *Knowledge fusion and resolution*

Since all the collected data rows of entity are from different structured type of data, we need to fuse the entities and resolve its attribute. For each assembled knowledge which can be a potential relation or entity, our automatic crawler program will attempt to generate its synonym collection by retrieving its namespace. If the two sets overlap or overlap each other when the second one was created from different data source, these two entities are considered equivalent. Besides entity resolution, in the period of collecting procedures entity's property are fused as well. We designed5 fixed info box property names manually to describe painting entities, which are "painter_name"(the painting's name), "time"(when the painting is created), "museum"(where the painting is collected), "type"(painting style), "URL"(the universal resource location to access the painting's image), And 6 info box properties for painter which are "name" (painter's name), "country"(painter's country), "birth year"(birth year), "death year"(death year), "avatar"(painter's avatar universal resource location), "description"(painter's description text). Data rows containing same subject will be fused under the rules.

All entities in PaintKG is grouped into four predefined categories, including painter, painting and attribute. We use pattern to match and classify given entity in text system which procedure is to enumerate all spans and their labels, which are considered as the candidates of the entities. If an entry satisfied the naming rules, it is considered as an entity and then classified into exact one of the four categories.

Finally, we simply remove Redundant data, multivalued properties and duplicated trips.

III. KNOWLEDGE GRAPH CONSTRUCTION

Knowledge graph construction procedures mainly includes two tasks which are entity recognition and relation extraction. Since meta-data contains huge scale of long vectors, we choose typical LSTM (Long Short-Term Memory) neural network and CRF (Conditional Random Field) to implement entity finding so as to improve efficiency and accuracy. And We preset seven kinds of labels to label words from different data sources to

construct relationship hash-map(dictionary) and then input labeled entity plain text information as dataset to train Bi-LSTM (Bi-directional Long Short-Term Memory) neural network to implement relation classification and to increase evaluation index.

A. Entity recognition

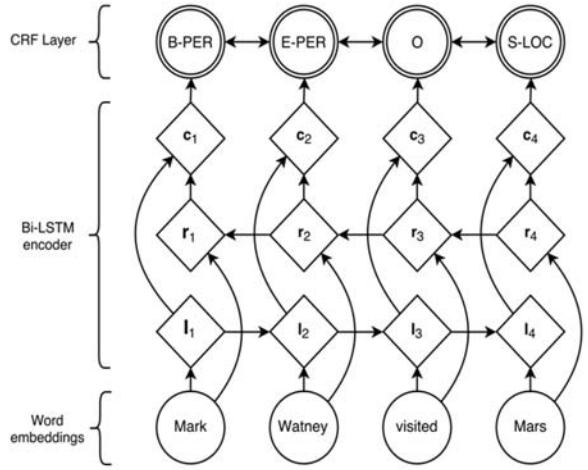

Fig. 4. LSTM neural network and CRF layer

Typical LSTM and CRF are discriminative models which infer posterior probabilities $p(C_k|x)$ .LSTM uses past input features and CRF uses sentence-level annotation information, which can effectively utilize past and future annotations to predict current labels.

As LSTM neural network based on RNN architecture has feedback connections which are input gate, forget gates, output gates, and memory cells that it can record extra information. And we also utilize CRF's key advantage which is great elastic arbitrary variety containing capability, and non-independent features to form sequential data and linear chain which takes form:

$$\mathrm{p(Y|X)} = \frac{1}{\mathrm{Z(x)}}\exp\{\sum_{\mathrm{m=1}}^{\mathrm{M}} \lambda_{\mathrm{m}} \mathrm{f_m}(\mathrm{y_n}, \mathrm{y_{n-1}}, \mathrm{X_n})\} \quad (1)$$

where

$$\mathrm{Z(x)} = \sum_{\mathrm{y}} \exp\{\sum_{\mathrm{m=1}}^{\mathrm{M}} \lambda_{\mathrm{m}} \mathrm{f_m}(\mathrm{y_n}, \mathrm{y_{n-1}}, \mathrm{X_n})\} \quad (2)$$

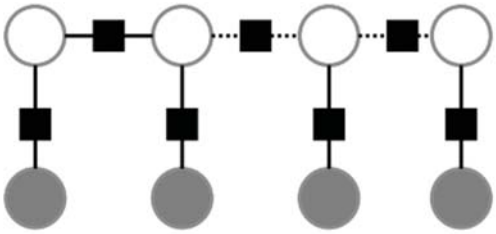

Fig. 5. CRF layer linear form

And therefore, we can take LSTM's sequence labeling feature to complete word2vec level classification task aiming to recognize entity from current data source like unstructured web knowledge or plain text. We build data source by labeling B-SUB, I-SUB, B-SUB, I-SUB, B-SUB, I-SUB to identify nature of each word in a sentence. For instance, "Van Gogh came to the small town of Al, France, and created *The Suspension Bridge in Al*", would be labeled as "B-SUB,I-SUB O O O O O O O O B-PREDICATE I-PREDICATE O B-SUB I-SUB I-SUB I-SUB I-SUB".

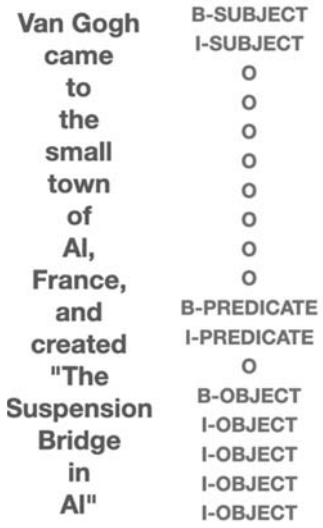

Fig. 6. Data mapping labels rules example

Then naturally generated dictionary by labels would be converted to id which is described as number sequence. After that we imported it into LSTM neural network which was built by Gluon RNN LSTM class and set loss function to cross entropy loss

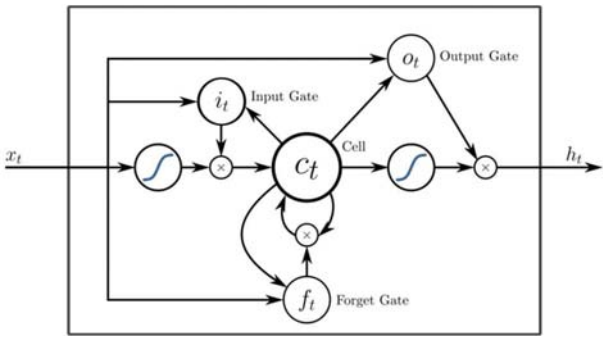

Fig. 7. LSTM neural network

h: hidden units, input: X, previous step hidden status: H, time step t input: I, deviation parameter:

$$I_t = \sigma(x_t W_{xi} + H_{t-1} W_{hi} + b_i) \quad (3)$$

$$F_t = \sigma(x_t W_{xf} + H_{t-1} W_{hf} + b_f) \quad (4)$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \quad (5)$$

The trained neural network will scan dataset by rows and we import the output of it via Neo4j database drive program.

All entities are restored in Neo4j, and the remaining data is restored in MySql.

B. Relation Extraction

To solve long vector relation extraction, we introduce attention layer which gives weight index to the output of bidirectional LSTM that generates weighted results.

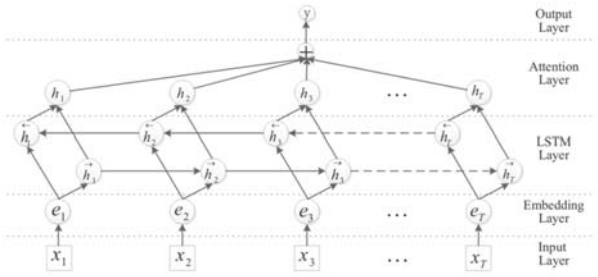

Fig. 8. Bidirectional LSTM model with Attention

We implement a data preprocessor that predefines 2 kinds of relations which are SAME_ ERA&MADE_ BY and put them into dictionary with map (relation, value). And preprocessor read train-text by rows recursively to count Objects' and Subjects' location relatively in sentences. And thus, we get a three-dimension vector like <words, The distant background between each word and B-SUB, The distant background between each word and I-SUB> which is finally converted to bidirectional queue like:

<vector>: ['Van' 'Gogh', 'created', 'The' 'suspension' 'bridge', 'in' 'Al'], [0, 1,2, 3, 4, 5, 6, 7, 8], [-8, -7, -6, -5, -4, -3, -2, -1, 0]

<bidirectional queue>: ['The', 'Suspension', 'Bridge', 'Van', 'Gogh', 'created', 'in', 'Al'].

And then convert queue to a two-dimension data structure, data frame like:

({'words':datas,'tags':labels,'positionE1':positionE1,'position E2':positionE2}, index=range(len(datas))).

We thus get the basic vector set and we define a set of stop-words to remove redundant words and signature and form index by labels. Since Bi-LSTM with can't resolve Chinese language that we need to count and sequence the statistics of index to compute discrete id value of each Chinese character which and conclude id2word array like:

'oil', 'chalk', 'paint', 'inspire'… =1, 2, 3, 4…

Before input such queues into model, we rank them by frequency to convert character vector to frequency vector and fill the queue's element with frequency less than 50 which like.

['inspired', 'Van' 'Gogh', 'to created', 'suspension' 'bridge'… to [4, 20, 35, 338, 6, 111…

We write this word2id array, vector set, id2word array, relation2id array to training PKL text. Currently we completed pre-train procedures to get vectors of model input.

To train the LSTM model with attention, we prepare data loader and configure dictionary variables beforehand. We take Pytorch's tensor to create Dataset object so as to create a dataloader object. Pytorch recursively parses data loader object to train data and label in model.

With every 10 epoch we test and evaluate model output that we can use back-propagation algorithm to fix arguments of model. In order to satisfy retrieve requirement which is breadth-first and raise recall index we intend to take F-1 measure to evaluate our model. Here is definition of F-1:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R} \tag{6}$$

β: argument, P: precision index, R: recall index.

Let β=1 then we get F-1 measure

$$F_1 = \frac{2 \times PR}{P+R} \tag{7}$$

Apparently, we expect to get as much results as possible in such retrieve system like PaintKG that recall index is thus more important. Thus we configure β to 3.5 to increase influence of recall index.

F-1 measure grows slower as epoch loop continuous and it reaches the maximum index of 0.335 approximately.

C. Knowledge database and model evaluation

We input dataset to this model and save output result in Neo4j which is a graph database that can store relation data.

Compared to other models, Our model (LSTM+CRF and Bi-LSTM with attention layer) has following advantages:

- Loosen conditional assumption which is independent of observed linear data with given feature.
- Capability of containing unpredictable labels.
- Exclude restrictions of generated Markov models and avoid biased successor states.
- Unified probability for whole sequential features of single exponential model with observation features given.

From perspective of quantitative analysis, our model satisfied retrieval requirements and reach mark of 0.335 in F-1 measure which increases recall ratio by 30% compared to naïve bayes and logistic regression.

```
this is 8 times
precision:  0.29647680625393114
recall ratio:  0.3174687010954616
f1:  0.30661387846202454
epoch: 8
this is 9 times
precision:  0.32590673571822565
recall ratio:  0.3437284820031299
f1:  0.3345804538901519
epoch: 9
this is 10 times
precision:  0.3193007855587718
recall ratio:  0.3451134585289515
f1:  0.33170570738290694
```

Fig. 9. F-1 measure evaluation

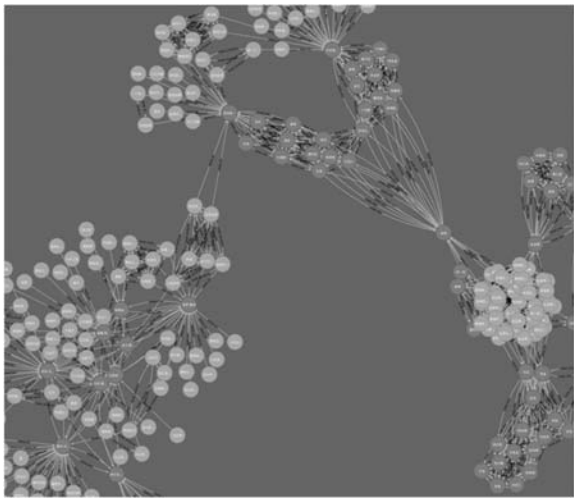

Fig. 10. PaintKg database example

D. Painting Entity retrival

To support smart painting applications, such as painting recommendation, painter discovery. Users can retrieval painting entities or relations by offering specified options and keyword. And PaintKG search engine based on elastic search returns analyzed match results.

E. Online API

To make PaintKG more publicly available, we use web structure technology like Vue.js, Spring Boot, Spring Data to provide an online API which can be queried via a interface that returns JSON. Only GET method request is available and will contain of entity and its corresponding knowledge like its image or maker.

F. Knowledge graph performing application

We used elastic search(a distributed full-text search engine based on lucence) to build a search engine that accelerates search query time from averagely 2 seconds to 5milliseconds. A distributed web application was built based on this elastic search that provides a more user-friendly interface and user experience via spring framework and vue.js technology. Moreover, roc-map module is implemented to realize knowledge graph data visualization.

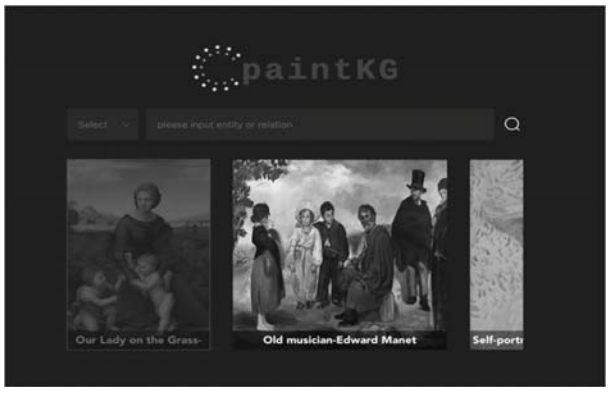

Fig. 11. Web application search engine(translated)

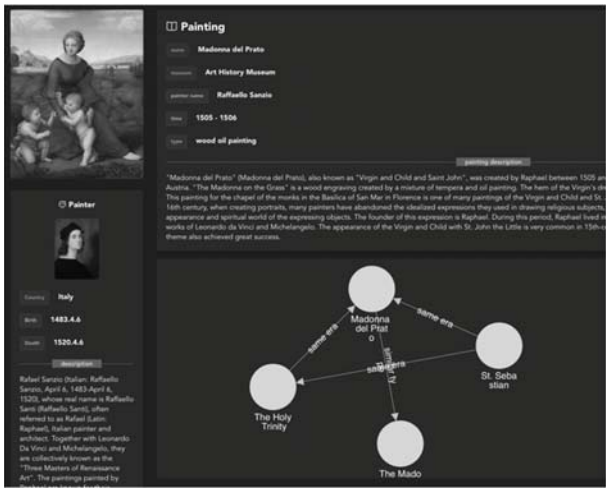

Fig. 12. Web application search result(translated)

IV. Conclusion and Future Work

We demonstrate the very first trial to construct Chinese knowledge graph on paintings and painters, called PaintKG, in this paper, currently covers mainstream online Chinese knowledge linking data like Baidu Baike, youhuadaquan.org and Wikipedia. We extracted semantic data representing the knowledge of painting from these Web-based cyclopedia and assimilate them via pattern match crawler and LSTM+CRF model. Then we take adopted strategy and Bi-LSTM with attentat layer to discover relation links between equivalent resources.

Although our job has improved and raise F-1measure ratio of relation extraction, the arguments of vector's length, batch size and dictionary mapping rules still expect researching and proving to find the best fit parameters and rules.

We provided distributed web search engine and application to our knowledge graph databases and apply services like entity retrieval, relation retrieval for both professional and common users.

Acknowledgements

Funded by Beijing Information Science Technology University 2020 undergraduate innovation training program (5102010805)

References

[1] Wu T, Gao C, Qi G, et al. KG-Buddhism: The Chinese Knowledge Graph on Buddhism[C]//Joint International Semantic Technology Conference. Springer, Cham, 2017: 259-267.

[2] Chen Y, Kuang J, Cheng D, et al. AgriKG: an agricultural knowledge graph and its applications[C]//International Conference on Database Systems for Advanced Applications. Springer, Cham, 2019: 533-537.

[3] Niu X, Sun X, Wang H, et al. Zhishi. me-weaving chinese linking open data[C]//International Semantic Web Conference. Springer, Berlin, Heidelberg, 2011: 205-220.

[4] Li Z, Sun M. Punctuation as implicit annotations for Chinese word segmentation[J]. Computational Linguistics, 2009, 35(4): 505-512.

[5] Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data[C]//Proceedings of the Joint Conference of the 47th

Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 2009: 1003-1011.

[6] Mahdisoltani F, Biega J, Suchanek F M. A knowledge base from multilingual Wikipedias–yago3[R]. Technical report, Telecom ParisTech. http://suchanek. name/work/publications/yago3tr. pdf, 2014.

[7] Lehmann J, Isele R, Jakob M, et al. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia[J]. Semantic web, 2015, 6(2): 167-195.

[8] Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016: 207-212.

[9] Maass W, Natschläger T, Markram H. Real-time computing without stable states: A new framework for neural computation based on perturbations[J]. Neural computation, 2002, 14(11): 2531-2560.

[10] Graves A. Generating sequences with recurrent neural networks[J]. arXiv preprint arXiv:1308.0850, 2013.

[11] Zhang S, Zheng D, Hu X, et al. Bidirectional long short-term memory networks for relation classification[C]//Proceedings of the 29th Pacific Asia conference on language, information and computation. 2015: 73-78.