Learning Multi-granular Features for Harvesting Knowledge from Free Text

Zheng Zhou, Huaming Wang, Zhixing Li, Feng Hu(✉), and Guoyin Wang(✉)

Chongqing Key Laboratory of Computational Intelligence,
Chongqing University of Posts and Telecommunications,
Chongqing 400065, People's Republic of China
{hufeng,wanggy}@cqupt.edu.cn

Abstract. Extracting entities and their relations expressed in free text is essential to correct and populate knowledge graphs. Traditional methods assume that only the information of entities benefits the extraction of relations. They view this task as a two-step task, named entity recognition (NER) and relation classification (RC). However, the inadequate use of information and the error propagation problem constrain methods following this pipeline fashion. Joint extraction methods are proposed to incorporate useful interaction information between the two tasks for improvement, which solve NER and RC simultaneously. Although they have been proved to be superior to pipeline models, their performance is still far from satisfaction. In this paper, we try to combine the idea of data-driven granular cognitive computing and deep learning in joint extraction task. Accordingly, a neural-based joint extraction model named Joint extraction with Multi-granularity Context (JMC) is proposed. It explores the multi-granularity context of natural language sentences and uses neural networks to learn representations of these context automatically. Experiments results on NYT, a large data set produced by the distant supervision technique, show that JMC achieves comparative results to state-of-the-art methods.

Keywords: Knowledge extraction · Joint extraction · Data-driven granular cognitive computing · Deep learning

1 Introduction

There is massive free text containing considerable fragmented knowledge on the Web, which computers can only process with many constraints. With effective extracting methods, knowledge expressed in free text can be organized into structural knowledge bases, such as Knowledge Vault [6], Freebase [2] and Wikidata [28]. Then, the knowledge can be used to build question answering, semantic search and recommendation systems. However, existing knowledge graphs are mostly incomplete and noisy [7], as may lead to wrong decisions in knowledge-based systems. Coping with these problems still counts on knowledge expressed in free text, which is helpful to correct and populate the facts in knowledge graphs.

T. Mihálydeák et al. (Eds.): IJCRS 2019, LNAI 11499, pp. 210–224, 2019.
https://doi.org/10.1007/978-3-030-22815-6_17

An effort of handling knowledge in free text is open information extraction (OpenIE). However, its relation words are picked from the raw text, but it is common that relations are not expressed explicitly in natural language. Relation extraction, aiming to predict semantic relations between named entity pairs, has no such constrains. As a result, semantic relations conveyed implicitly in natural language can be uncovered effectively. Traditional relation extraction methods are often conducted on a pipeline fashion of two separated tasks: named entity recognition (NER) and relation classification (RC) [4,13,14,23]. The main drawback is that the error of entity recognition task may be propagated to relation classification task, limiting the final performance. Moreover, only the result of NER is applied to help RC task in a pipeline fashion.

Actually, entity recognition and relation classification are highly interrelated. Not only the results of NER can help determine the relations among entities, but the results of RC can also help improve the performance of NER. For example, the sentence *"Mrs. Tsuruyama is from Kumamoto Prefecture in Japan."* denotes that the person named *Mrs. Tsuruyama* lives in *Kumamoto Prefecture*. With such prior information that *Mrs. Tsuruyama* is a person and *Kumamoto Prefecture* is a location, the possibility of there is *Live_In* relation between these two entities is high. Besides, given that relation *Live_In* exists in *Mrs. Tsuruyama* and *Kumamoto Prefecture*, one can easily determine that *Mrs. Tsuruyama* is a person and *Kumamoto Prefecture* is a location. Under similar observation, joint extraction methods were designed to make NER and RC benefit from each other by incorporating the interaction information between them. Although joint extraction methods have been proved to be superior to pipeline methods, most of them still rely on millions of lexicalized features and higher-order term features like other natural language processing tasks [10,15,22]. These features are incomplete, sparse and costly in computing [3].

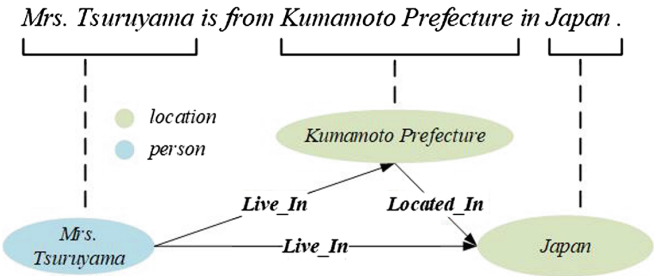

Fig. 1. Illustration of the joint extraction task.

Motivated by data-driven granular cognitive computing model [29], this paper explores multi-granular features for joint extraction task, including word-level features, local context features, segment context features and sentential context features. Moreover, we introduce these multi-granular prior knowledges to neural network architecture and propose a neural-based joint extraction method named

Joint Extraction with Multi-granular Context (JMC). Unlike traditional methods, JMC counts on neural network to learn representations of multi-granular context automatically instead of using hand-crafted features.

The main contributions of this paper are three-fold: (1) A neural model named JMC which extracts entities and relations jointly from unstructured text is proposed. (2) The idea of granular computing is introduced to joint extraction task to find multi-granular context features and design the corresponding neural network. (3) Experiments are conducted to evaluate the effectiveness of the proposed methods. Results imply that multi-granular context features can bring improvement to joint extraction task.

The rest of this paper is organized as follows. Section 2 briefly introduces related works of knowledge extraction. Section 3 states the joint extraction task and gives the multi-task objective. Section 4 depicts the proposed model. Section 5 gives the experiment results on a distant supervision corpus. Conclusions are shown in Sect. 6.

2 Related Works

2.1 Pipeline

Most existing works view relation extraction as a two-step task, where named entity recognition [13] is first conducted to determine the type of entities. Then, the information of entities are taken as input to identify the relations for entity pair [14,23]. Collobert et al. [5] propose a convolutional neural network based model for part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. However, it eliminates the interactions among the predications. Lample et al. [13] modify it by replacing CNNs with bi-directional LSTMs to extract features. A conditional random layer is also adopted to solve the structural predication problem. Chiu and Nichols [4] add richer features for words as the input of neural based NER model, including word embeddings, capitalization information and character embeddings extracted by CNNs.

For relation classification, neural based models have achieved state-of-the-art performances. Given a sentence and an entity pair it contains, Nguyen and Grishman [23] adopt convolutional neural networks to extract representation automatically and determine semantic relations between entities that a sentence expresses. Distant supervised technique has been used widely to generate massive training data automatically for the relation classification task. For an entity pair, there is more than one sentence in distant supervised data set. Only part of them express the considered relation in extract operation, other sentences are noisy samples. To cope with the noise in distant supervised data sets, Lin et al. [17] take a batch of sentences as input and weight them using attention [31] to reduce the influence of noisy sentences. Considering information consistency and complementarity among texts in different languages, Lin et al. [16] generalizes the model to multi-lingual scenario.

2.2 Joint Extraction

Recent studies focus on designing more integrated models to capture the interdependencies between named entity recognition and relation classification tasks. Roth and Yih [27] adopt linear programming formulation to infer entities and relations simultaneously. Kate and Mooney [10] introduce a card-pyramid structure which encodes the entities and relations in a sentence. It adopts dynamic programming to solve the joint extraction task by labeling nodes in a card-pyramid structure jointly. Li and Ji [15] use a segment-based decoder based on the idea of semi-Markov chain to simultaneously extract entity mentions and relations with beam search. Miwa and Sasaki [22] propose the table representation that encodes entities and relations in a sentence. Besides, a history-based structured learning approach is proposed. Miwa and Bansal [21] present a joint model stacking bidirectional tree-structured LSTMs on bidirectional LSTMs to capture word sequence and dependency tree substructure.

Gupta et al. [9] view the entity recognition and relation classification as a table filling problem and design neural models based on multi-task recurrent neural networks to solve it. Zheng et al. [32] transform the joint extraction to a single tagging problem by fusing the relation types with the tags of NER. Ren et al. [26] first embed entity mentions, relation mentions, text features and type labels into two low-dimensional spaces where objects whose types are close also have similar representations. Then, the types of test mentions are estimated based on the learned embeddings. Katiyar and Cardie [11] propose an attention-based recurrent neural network for joint extraction of entity mentions and relations without using dependency trees. Adel and Schütze [1] utilize convolutional neural networks and linear-chain conditional random fields for joint extraction.

In this paper, we design an architecture for the joint extraction task. Different from existing joint extraction methods, it benefits from multi-granular context feature extracted automatically. Experiments results show that the proposed model achieves comparative or better results to state-of-the-art methods.

3 Problem Statement

This paper focuses on extracting facts from single sentence, leaving the integrating of information in multiple sentences for future study. Given a sentence $S = (w_1, w_2, ..., w_n)$, where w_i is the i-th word in the sentence and n is the sentence length. Let R be the set of the predefined semantic relations or the relations in knowledge graph. Set T contains the abstracted types of entities such as `PERSON` and `LOCATION`. Joint extraction is aimed at finding the mentions as well as types of entities and the relations between entities in S. The types of entities and relations are picked from T and R respectively. Challenges are three-fold. First, the extraction of entities and relations are highly related. Second, the assignment for entities are not independent. Third, the results could turn to be a multi-relational graph with the entities and relations in the sentence increasing, as Fig. 1 shows.

Actually, this task can be well represented as a table filling task [22]. As Table 1 shows, the table representation encodes the whole entity and relation structure in a sentence. The diagonal cells are tagged according to the relative position to its corresponding entity and the type of the entity. Other cells are filled with relation types and directions between words (→ denotes the direction of relations and ⊥ denotes the non-relation pair). Its relations are defined on word pairs, instead of entities, as enables it extracting relations from raw sentences directly. Besides, that the table structure captures multiple relations in a single sentence comes for free.

Table 1. The table representation of a sentence in joint extraction task.

	Mrs.	Tsuruyama	is	from	Kumamoto	Prefecture	in	Japan	.
Mrs.	B-PER,⊥								
Tsuruyama	⊥	L-PER,⊥							
is	⊥	⊥	O,⊥						
from	⊥	⊥	⊥	O,⊥					
Kumamoto	Live_in→	⊥	⊥	⊥	B-LOC,⊥				
Prefecture	⊥	⊥	⊥	⊥	⊥	L-LOC,⊥			
in	⊥	⊥	⊥	⊥	⊥	⊥	O,⊥		
Japan	Live_in→	⊥	⊥	⊥	⊥	Located_in→	⊥	U-LOC,⊥	
.	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	O,⊥

4 Model

We consider the joint extraction task from granular computing perspective and propose to introduce multi-granular context features. Section 4.1 gives the details of multi-granular context. Section 4.2 introduce the details of the proposed model.

4.1 Multi-granular Features

For table filling tasks, relations are assigned on words. Only taking word itself as features would be very deficient. As a result, capturing rich contextual information is essential for determining the non-diagonal cells. This paper explores information from multi-granular context for the table filling task. For the convenience of statement, word on position i is marked as w_i, its tag, which corresponds with the diagonal cell in the table representation, is marked as t_i. The representation of i-th word is h_i.

Word Feature. Word feature is the representation of tokens. For filling the diagonal cells, only the very basic feature h_i is used. The word feature can be formulated as

$$feat_i^w = h_i \tag{1}$$

When determining other cells, feature h_i as well as its tag t_i is used. The word feature turns to be

$$feat_i^w = [h_i, t_i] \tag{2}$$

where $[\cdot]$ is the concatenation operation.

Local Context Feature. In natural language processing tasks, the surrounding words contribute to the understanding of current word. The local context feature is constituted by the information of surrounding words within the predefined window size. Taking the window size as c, the local context feature is

$$feat_i^{lc} = g(h_{i-c/2}, ..., h_{i+c/2}) \tag{3}$$

where $g(\cdot)$ is the feature extraction function. i is the index of the corresponding word.

Segment Context Feature. Previous works have shown the effectiveness of segment features in dependency parsing task. Table filling and dependency parsing share the characteristic that relations are defined on word pairs. Inspired by the graph-based dependency parsing model [30], we also divide a sentence into three parts (prefix, infix and suffix). The segment context of the dependency word pair is composed of these segments (parts). In this paper, the segment feature is used to produce the relation on word pair. For cell c_{ij} in the table representation, three types of segment feature are considered

$$\begin{aligned} feat_{ij}^{ps} &= k(h_0, ..., h_i) \\ feat_{ij}^{is} &= k(h_{i+1}, ..., h_j) \\ feat_{ij}^{ss} &= k(h_{j+1}, ..., h_n) \end{aligned} \tag{4}$$

where $k(\cdot)$ is the feature extraction function. $feat_{ij}^{ps}$, $feat_{ij}^{is}$ and $feat_{ij}^{ss}$ represent the segments which split by the indexes i and j. The final segment feature is the concatenation of the representations of three segments, formulated as

$$feat_{ij}^{seg} = [feat_{ij}^{ps}, feat_{ij}^{is}, feat_{ij}^{ss}] \tag{5}$$

Sentential Context Feature. The global information can also help the determination of relations. For example, given the prior knowledge that only the `Live_In` relation exists in the given sentence, one could avoid illegal assignments

to cells. Sentential context feature captures the global information over the entire sentence, which can be formulated as

$$feat^s = o(h_1, h_2, ..., h_n) \tag{6}$$

where $o(\cdot)$ is the feature extraction function and n is the sentence length.

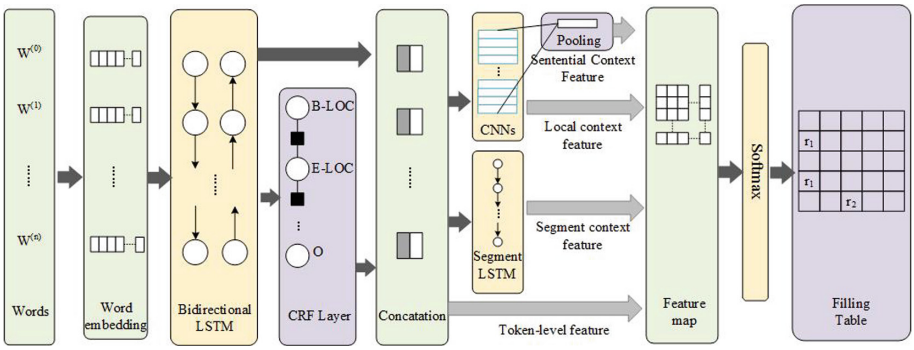

Fig. 2. The JMC architecture. Bidirectional LSTM layer, CNNs&Pooling layer and segment LSTM layer produce multi-granular features, including word feature, local context feature, segment context feature and sentential context feature.

4.2 The Proposed Joint Extraction Model

Different from traditional methods, we propose to learn these features automatically with neural model instead of designing extraction functions by hand. Word feature is generated by feeding the embedding of words into a bi-directional long-short term memory network. Local and sentential context feature are given by convolutions and polling. For segment context feature, a forward LSTM layer is adopted following [30].

Figure 2 depicts the architecture of the proposed joint extraction model. JMC takes only word unigram as input and then leaves the feature combinations learned by the model automatically. First, it embeds words into dense vectors using pre-trained word2vec. Second, following the structure of BiLSTM-CRF (bidirectional long-short term memory network and conditional random field) for NER, dense vectors of words are feed into bi-directional LSTM layer, dense hidden layer and CRF layer sequentially. Then, the NER tags are produced by CRF layer. Third, the outputs of BiLSTM are concatenated with the one-hot vectors of NER tags as word features. They are feed into a forward LSTM and CNNs to generate segment context feature, local context feature and global context feature. The concatenation of these features is taken as the basic representation of cells in feature map.

Word Embedding. Words are discrete and sparse in nature. We adopt a word embedding layer to represent the word. It maps a word to a dense vector of pre-defined dimensionality. The word embedding layer is initialized with the pre-trained 300 dimensional GloVe[1] word vectors trained on Wikipedia corpus.

BiLSTM Layer. Bi-directional LSTM (BiLSTM), presenting each sequence forwards and backwards to two separate hidden states to capture past and future information, has been proved to be effective in sequence labeling tasks. The representation of a word produced by Bi-LSTM is obtained by concatenating its left and right context.

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{7}$$

where h_t is the output of the Bi-LSTM layer. $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the output vector of forward and backward LSTM respectively.

Suppose x_t and $\overrightarrow{h_t}$ are the word embedding and the hidden state at time t. The states of forward LSTM unit at time t can be formulated as

$$\begin{aligned}
i_t &= \sigma(W_i\overrightarrow{h_{t-1}} + U_i x_t + b_i) \\
f_t &= \sigma(W_f\overrightarrow{h_{t-1}} + U_f x_t + b_f) \\
\tilde{c}_t &= \tanh(W_c\overrightarrow{h_{t-1}} + U_c x_t + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
o_t &= \sigma(W_o\overrightarrow{h_{t-1}} + U_o x_t + b_o) \\
\overrightarrow{h_t} &= o_t \odot \tanh c_t
\end{aligned} \tag{8}$$

where $\sigma(\cdot)$ is element-wise sigmoid function and $\odot$ is the element-wise product. U_i, U_f, U_c, U_o and W_i, W_f, W_c, W_o denote the weight matrices of different gates. b_i, b_f, b_c and b_o are the weight matrices and bias vectors. The formulation of the backward LSTM is similar to Eq. 8.

CRF Layer. Conditional Random Field(CRF) layer has been successively used in tagging models. We also use it to model the interdependencies among NER tags. Given an input sentence $X = (x_1, x_2, ..., x_n)$, $P = (p_1, p_2, ..., p_n)$ is considered as the score vectors delivered by the BiLSTM. p_i is a score vector of word x_i whose size is $1 \times k$, where k is the number of distinct tags for NER task.

Given the prediction tags $Y = (y_1, y_2, ..., y_n)$, where y_i is chosen from the tag set $T = \{t_1, t_2, ..., t_k\}$. The score is defined as

$$s(X, Y) = \Sigma_{i=0}^{n} A_{y_i, y_{i+1}} + \Sigma_{i=1}^{n} p_i^{y_i} \tag{9}$$

[1] https://nlp.stanford.edu/projects/glove/.

where A is the transition matrix and $A_{i,j}$ denotes the transition score from tag i to tag j. We predict the output sequence by maximizing the score

$$Y^* = argmax_{y \epsilon \bar{Y}} s(X, y) \tag{10}$$

where $\bar{Y}$ contains all possible output sequences of the input sentence X.

Segment LSTM Layer. We consider three segments described and adopt a forward LSTM layer to learn their representations in Eq. 4. The representation of infix segment is considered as the hidden state of the head word. The representation of inner segment is obtained by subtraction between the hidden vector of the tail word and the head word. For the suffix segment, its representation is the subtraction of the last hidden vector and the hidden state of the tail word. When there has no prefix or suffix, the corresponding embedding is set to zero vector.

Softmax Layer. A Softmax classifier is adopted to determine the relation that the word pair hold. The relation between word i and word j is produced by

$$r_{ij} = softmax(WT'_{ij} + b) \tag{11}$$

where the W and b are weight matrix and bias vector. Besides, instead of feeding the feature table generated by the table convolution layer into the Softmax classifier directly, we add a hidden layer ahead of it, which transforms the representation of each cell into a new feature space with much lower dimensionality.

Objective Function. This paper follows the multi-task framework to avoid the error propagation problem in the pipeline framework. Basic features learned automatically are shared by these two tasks and their objectives are optimized jointly. Let the given sentence be $S = (w_1, w_2, ..., w_n)$. For named entity recognition, the objective function is

$$L^{ner} = \sum_{i=1}^{|D|} \sum_{t=1}^{n_i} (log(p_t^{(i)} = y_t^{(i)} | x^{(i)}, \Theta)) \tag{12}$$

where $|D|$ is the size of training set, n_i is the length of sentence $x^{(i)}$. $y_t^{(i)}$ is the correct tag[2] of word t in sentence $x^{(i)}$ and $p_t^{(i)}$ is the normalized probabilities of tags produced by the model. Besides, Θ is the parameter of the joint model. For relation classification, the objective function is

$$L^{rc} = \sum_{i=1}^{|D|} \sum_{m,n=1}^{n_i^2} (log(c_{mn}^{(i)} = y_{mn}^{(i)} | x^{(i)}, \Theta)) \tag{13}$$

[2] Entity type encoded in BILOU (Begin, Inside, Last, Outside, Unit) scheme.

where $c_{mn}^{(i)}$ is the ground truth relation between m-th word and n-th word in the sentence $x^{(i)}$. The multi-task objective function is

$$L = \alpha L^{ner} + (1 - \alpha) L^{rc} \tag{14}$$

where α is the trade-off weight between named entity recognition and relation classification tasks.

5 Experiments

5.1 Implement Details

We use Tensorflow[3] framework to implement our joint extraction model. All hyper-parameters are tuned on the development set. The weights of word embedding are pre-trained by [24] and the dimensionality of embedding vectors is 300. The numbers of hidden units of forward and backward LSTM are both 64. The weights and biases are updated using gradient based optimizer Adam [12] by minimizing crossentropy of the output of CRF layer and softmax layer. The learning rate is initialed to 0.01 and reduced half when there has no decrements of loss. To avoid overfitting, we add dropout operations after the BiLSTM with the dropout rate of 0.2. Early stop technique is also adopted. More detailed setting of parameters can be found in the source code[4].

5.2 Data Set

Distant supervision methods can produce a large amount of training data automatically. With manually labeled test set, its quality can be ensured despite containing noise. Distant supervision has been used in many natural language processing tasks [19,26]. To evaluate the effectiveness of our methods detailedly, we test the proposed method on the public dataset NYT [26], produced by distant supervision technique. There are 353k triplets in the training data and 3,880 triplets in the test set. Besides, the number of valid relations is 24 and `None` is viewed as the undefined relation `UND`.

5.3 Compared Methods

We choose joint extraction methods producing state-of-the-art results on NYT as comparatives. **DS+Logistic** [20] trains a multi-class logistic classifier to predict relations. **DeepWalk** [25] embeds mention-feature co-occurrences and mention-type associations as a homogeneous network. **FCM** [8] adopts neural language model to perform compositional embedding. **Cotype** [26] first runs text segmentation algorithm to extract entity mentions. Then, entity mentions, relation mentions, text features and type labels are embedded into two low-dimensional

[3] www.tensorflow.org.
[4] https://github.com/MingYates/JMC.

spaces. In each space, mentions with close types also have similar representations. **LSTM-LSTM** [32] converts the joint extraction task to a tagging problem and solves it using LSTMs. **REHESSION** [19] benefits from heterogeneous information source, for example, knowledge base and domain heuristics. Besides, state-of-the-art tagging model **BiLSTM+CRF** is also selected as a comparative on the named entity recognition task.

5.4 Results of Named Entity Recognition

We take Strict-F1, Macro-F1 and Micro-F1 proposed in [18] as evaluations for NER. Results are shown in Table 2. BiLSTM+CRF and JMC outperform other methods with more than 0.30 on Strict-F1. The reason might be that **DeepWalk** and **Cotype** have a preprocess step of entity mention detection and the error of entity mention detection will propagate to entity typing. Moreover, the results denote that tagging based NER can also achieve comparative results on distant supervision data set.

Table 2. Performance of named entity recognition on NYT

Methods	Strict-F1	Macro-F1	Micro-F1
DS+Logistic [20]	-	-	-
DeepWalk [25]	0.49	0.54	0.53
FCM [8]	-	-	-
LSTM-LSTM [32]	-	-	-
Cotype [26]	0.60	0.65	0.66
REHESSION [19]	-	-	-
BiLSTM+CRF	0.89	0.91	0.90
JMC (proposed)	**0.94**	**0.93**	**0.91**

5.5 Results of Relation Classification

For a sentence, it is considered correct if the predicted relations are correct without considering the results of entities. Besides, we ignore `BLANK` and `UND` relations and only report the accuracy for valid relations as [26] does. As Fig. 3 shows, JMC produces the best results on relation classification task. It is worth to mention that the proposed method only takes words as input, while **Cotype** and **REHESSION** introduce external knowledge bases.

5.6 Results of Joint Extraction

Performances on the setting of end-to-end relation extraction are also reported in Table 3. A sentence is considered correct if the entities and relations are correct. The results of comparative methods are reported in their original papers

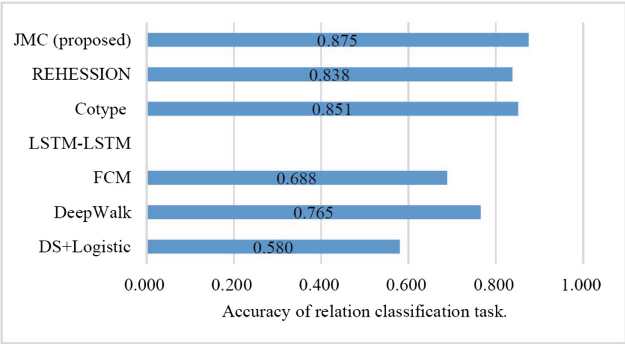

Fig. 3. Accuracy of relation classification on NYT

adopting the same criteria [19,26]. As Table 3 says, JMC produces the highest recall and F1 score compared to other methods. Besides, it gives comparative results evaluated by precision.

Table 3. Performance of joint extraction on NYT

Methods	Precision	Recall	F1
DS+Logistic [20]	0.258	0.393	0.311
DeepWalk [25]	0.176	0.224	0.197
FCM [8]	0.553	0.154	0.240
LSTM-LSTM [32]	**0.615**	0.414	0.495
Cotype [26]	0.423	0.511	0.463
REHESSION [19]	0.412	0.573	0.479
JMC (proposed)	0.524	**0.657**	**0.583**

6 Conclusions

This paper studies joint extraction of entities and relations from free text. Considering that the ground truth of part-of-speech tags and dependency trees are not available in real applications, we design a neural model extracting entities and relations jointly which only takes words as input. Different from existing joint extraction methods, the proposed model needs no hand-designed features and learns representations of multi-granular context among outputs on feature automatically. Results on distant supervision data set show that the proposed method produces comparative performance compared to state-of-the-art methods in the setting of named entity recognition, relation classification and end-to-end joint extraction. For the future works, incorporating heterogeneous source

such as knowledge bases, rules and prior knowledge may bring improvement for extraction entities and relations from free text.

Acknowledgment. The author would like to thank the anonymous reviewers for their help. This work was supported by the National Key Research and Development Program of China (Grant no. 2016YFB1000905), the National Natural Science Foundation of China (Grant nos. 61572091, 61772096).

References

1. Adel, H., Schütze, H.: Global normalization of convolutional neural networks for joint entity and relation classification. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017, pp. 1723–1729 (2017)
2. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, 10–12 June 2008, pp. 1247–1250 (2008)
3. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A Meeting of SIGDAT, A Special Interest Group of the ACL, 25–29 October 2014, Doha, Qatar, pp. 740–750 (2014)
4. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. TACL **4**, 357–370 (2016)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
6. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, New York, NY, USA, 24–27 August 2014, pp. 601–610 (2014)
7. Färber, M., Bartscherer, F., Menne, C., Rettinger, A.: Linked data quality of DBpedia, freebase, OpenCyc, Wikidata, and YAGO. Semant. Web **9**(1), 77–129 (2018)
8. Gormley, M.R., Yu, M., Dredze, M.: Improved relation extraction with feature-rich compositional embedding models. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 1774–1784 (2015)
9. Gupta, P., Schütze, H., Andrassy, B.: Table filling multi-task recurrent neural network for joint entity and relation extraction. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016, pp. 2537–2547 (2016)
10. Kate, R.J., Mooney, R.J.: Joint entity and relation extraction using card-pyramid parsing. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL 2010, Uppsala, Sweden, 5–16 July 2010, pp. 203–212 (2010)
11. Katiyar, A., Cardie, C.: Going out on a limb: joint extraction of entity mentions and relations without dependency trees. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, vol. 1: Long Papers, pp. 917–928 (2017)

12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014)
13. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, USA, 12–17 June 2016, pp. 260–270 (2016)
14. Lee, J.Y., Dernoncourt, F., Szolovits, P.: MIT at semeval-2017 task 10: relation extraction with convolutional neural networks. In: Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, 3–4 August 2017, pp. 978–984 (2017)
15. Li, Q., Ji, H.: Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Baltimore, MD, USA, 22–27 June 2014, vol. 1: Long Papers, pp. 402–412 (2014)
16. Lin, Y., Liu, Z., Sun, M.: Neural relation extraction with multi-lingual attention. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, vol. 1: Long Papers, pp. 34–43 (2017)
17. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, 7–12 August 2016, Berlin, Germany, vol. 1: Long Papers (2016)
18. Ling, X., Weld, D.S.: Fine-grained entity recognition. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, Ontario, Canada, 22–26 July 2012 (2012)
19. Liu, L., et al.: Heterogeneous supervision for relation extraction: a representation learning approach. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017. pp. 46–56 (2017)
20. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: 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, Singapore, 2–7 August 2009, pp. 1003–1011 (2009)
21. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016, vol. 1: Long Papers (2016)
22. Miwa, M., Sasaki, Y.: Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A Meeting of SIGDAT, A Special Interest Group of the ACL, Doha, Qatar, 25–29 October 2014, pp. 1858–1869 (2014)
23. Nguyen, T.H., Grishman, R.: Relation extraction: perspective from convolutional neural networks. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, VS@NAACL-HLT 2015, Denver, Colorado, USA, 5 June 2015, pp. 39–48 (2015)
24. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)

25. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, New York, NY, USA, 24–27 August 2014, pp. 701–710 (2014)
26. Ren, X., et al.: CoType: joint extraction of typed entities and relations with knowledge bases. In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, 3–7 April 2017, pp. 1015–1024 (2017)
27. Roth, D., Yih, W.T.: Global inference for entity and relation identification via a linear programming formulation. In: Introduction to Statistical Relational Learning, pp. 553–580 (2007)
28. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
29. Wang, G.: Data-driven granular cognitive computing. In: Polkowski, L., et al. (eds.) IJCRS 2017. LNCS (LNAI), vol. 10313, pp. 13–24. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60837-2_2
30. Wang, W., Chang, B.: Graph-based dependency parsing with bidirectional LSTM. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, 7–12 August 2016, Berlin, Germany, vol. 1: Long Papers (2016)
31. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. CoRR abs/1502.03044 (2015)
32. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, vol. 1: Long Papers, pp. 1227–1236 (2017)