

Relationship Classification based on Dependency Parsing and Pre-training Model

Baosheng Yin (✉ 19971981@sau.edu.cn)

Shenyang Aerospace University

Yifei Sun

Shenyang Aerospace University

Research Article

Keywords: Relationship Extraction, Dependency Parsing, Pre-training Model

Posted Date: December 9th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1105737/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

As an important part of information extraction, relationship extraction aims to extract the relationships between given entities from natural language text. On the basis of the pre-training model R-BERT, this paper proposes an entity relationship extraction method that integrates entity dependency path and pre-training model, which generates a dependency parse tree by dependency parsing, obtains the dependency path of entity pair via a given entity, and uses entity dependency path to exclude such information as modifier chunks and useless entities in sentences. This model has achieved good F1 value performance on the SemEval2010 Task 8 dataset. Experiments on dataset show that dependency parsing can provide context information for models and improve performance.

1 Introduction

Information extraction (IE) aims to extract structured information from large-scale semi-structured or unstructured natural language text ^[1]. Information extraction tasks are applied, for example, to knowledge graph construction ^[2], information retrieval ^[3], question-answering systems and text summarization. Entity relationship extraction is an important part of information extraction tasks, and its results will affect the performance of follow-up tasks.

Entity relationship extraction based on deep learning falls into one of the following two major categories: supervised entity relationship extraction and distantly supervised entity relationship extraction. In supervised entity relationship extraction, the entity relationship extraction can be achieved by either pipeline learning or joint learning ^[4]. The pipeline learning method is to extract the relationships between entities directly on the basis of entity identification, and the joint learning method is to identify entities while extracting the relationships between entities mainly based on an end-to-end model of neural network. Compared with supervised entity relationship extraction, the distantly supervised entity relationship extraction, due to lack of human-annotated dataset, takes one more step to distantly align knowledge base to label unlabeled data. As for the construction of relationship extraction model, there is little difference between distantly supervised entity relationship extraction and the pipeline learning method of the supervised entity relationship extraction. The main difference between the supervised entity relationship extraction and the distantly supervised entity relationship extraction is the difference in the annotation level of dataset. For the supervised method, entity and relationship type have been given in the dataset. At this time, the relationship extraction task can be done by the method of classification task.

Due to the rise of the pre-training models, the tasks of entity relationship extraction have gradually moved closer to the direction of pre-training models. Researchers have achieved very good results only by fine-tuning the pre-training model BERT and then performing entity relationship extraction experiments. In 2019, the Ali team ^[5] took the lead in applying BERT to the relationship extraction tasks and achieved the best results at that time. The result made more researchers focus on pre-training model. After these experiments, most entity relationship extraction models are based on pre-training models, usually by

training their own pre-training models after changing the initialization parameters of BERT structure, or integrating external knowledge.

The existing models pay too much attention to the impact of the whole sentence on the relationship classification, without considering the noise caused by such content as modifier chunks in sentences. Moreover, external knowledge is cited to assist the model in sentence classification, but the syntactic knowledge of the sentence itself is ignored.

This paper proposes a method of using dependency parsing, which establishes a dependency tree for each data and obtains the shortest dependency path between entities via the dependency trees. This paper mainly focuses on the word information in the dependency path between entities, rather than using the type of dependency relationship between words in the past. Parsing is used to enhance the context information of model learning, so as to avoid noise caused by information such as modifier chunks and unannotated entities in sentences.

2 Related Work

Parsing ^[6] is one of the key technologies in natural language processing, and its basic task is to determine the syntactic structure of a sentence or to clarify the dependency relationships between words in a sentence. Dependency parsing analyzes a sentence into a dependency syntax tree to describe the dependency relationships between words. The dependency relationship is represented by a directed arc, which is called the dependency arc. The shortest dependency path refers to the shortest path of two words in dependency syntactic structure. Entity dependency path is the shortest path between two entity nodes in the dependency syntactic structure. The shortest dependency path can express the syntactic relationships between two nodes. According to the characteristics of the shortest dependency path, the entity dependency path can concisely express the syntactic relationships between entities, remove modifier chunks, and retain the backbone mode that can clearly express the entity relationships. Therefore, dependency parsing is widely used in relationship extraction.

Entity relationship extraction, as one of the most critical tasks in natural language processing, is widely used in such fields as information extraction, natural language understanding, and information retrieval. Early relationship extraction methods include feature-based methods and kernel-based methods. As early as in the feature-based methods, syntactic knowledge has been used for relationship extraction. Today's relationship extraction methods can be divided into two categories: statistical relationship extraction and neural relationship extraction ^[7]. The statistical relationship extraction is to annotate the relationships of the target entity pair in the text based on traditional machine learning methods. Among them, classical entity relationship extraction methods can be divided into four categories: supervised, semi-supervised, weakly-supervised and unsupervised methods, which are distinguished by the degree of annotation of dataset. Neural relationship extraction applies deep learning to relationship extraction tasks, and the entity relationship extraction tasks of deep learning can be divided into supervised tasks and distantly supervised tasks.

Among the classical statistical relationship extraction methods, Zhou^[8] and Guo Xiyue et al.^[9] used SVM as a classifier to study the effects of lexical, syntactic and semantic features on entity relationship extraction; Craven et al.^[10] first proposed the idea of weakly-supervised machine learning in the process of extracting structured data from text to establish a biological knowledge base; Brin^[11] used the Bootstrapping method to extract the relationships between named entities; and Hasegawa et al.^[12] first proposed an unsupervised method of extracting relationships between named entities at the ACL meeting.

Traditional methods have the problem of error propagation of feature extraction, so the entity relationship extraction method based on deep learning, which can effectively solve this problem, has been paid attention to and achieved good results. Zeng et al.^[13] first proposed using CNN to extract the meaning of a word and applying softmax for classification in 2014. Zhang et al.^[14] proposed using Bi-LSTM for relationship classification in 2015. Xu et al.^[15] reintroduced the traditional method and proposed CNN that is based on the shortest path. In addition, in recent years, many attention-based models have been applied to relationship extraction tasks. Katiyar et al.^[16] first used Attention, an attention mechanism, together with Bi-LSTM to jointly extract entity and classification relationships in 2017.

Scholars have also proposed a variety of improvements based on the basic methods, such as the fusion method of PCNN and multi-instance learning^[17], and the fusion method of PCNN and attention mechanism^[18]. Ji et al.^[19] proposed adding entity description information on the basis of PCNN and Attention to assist in learning entity representation. The COTYPE model proposed by Ren et al.^[20] and the residual network proposed by Huang^[21] both enhanced the effect of relationship extraction.

After the pre-training model was proposed, Wu et al. first applied the pre-training model to the relationship extraction tasks in 2019, and explored the mode of combination of entities and entity locations in the pre-training model by adding identifiers before and after the entities to indicate the entity locations, rather than using the traditional location vector. The best results were achieved at that time, which made more researchers focus on the pre-training model. Later, Livio Baldini Soares et al.^[22] from the Google team proposed a pre-training model of BERTEM+MTB in 2019. In that paper, the effects of input and output on the results of relationship classification under different conditions were discussed, and the Matching the blank pre-training task was proposed according to the results to eliminate the error caused by over utilization of entities. Peng et al.^[23] conducted experiments on the basis of BERT and MTB in 2020, explored through experiments the information types used by the existing models in the entity relationship extraction tasks, designed experiments, and finally concluded that the existing models did not make full use of context information. In this paper, the BERT model is used for experiments. Besides the entity information, entity dependency path is used as syntactic representation, and sentence information, entity information and syntactic information are used as sentence representation for relationship classification.

3 Model Introduction

In supervised entity relationship extraction tasks, since the dataset have fully annotated entities and the corresponding relationship types are given, the existing models all use these tasks as classification tasks. The model outputs a vector as a sentence representation and predicts the relationship type. This paper proposes a model framework that uses context information for relationship extraction, whose architecture is shown in Figure 1.

In this paper, the pre-training model BERT is used as the basic model for relationship extraction, and its structure including three parts. Given a sentence, the shortest dependency path between entities is obtained first after dependency parsing, which, together with the sentence, will be used as the input to the model. Token of the sentence obtained through word segmentation is input to an encoder for coding to obtain the vector representation of each Token, and the sentence vector, entity vector and dependency vector are spliced to obtain the final representation of the sentence, which is also the final vector for classification. This vector is input to the Softmax classifier for prediction.

3.1 Dependency syntactic parsing

Parsing is one of the key technologies in natural language processing, and its basic task is to determine the syntactic structure of a sentence or the dependency relationships between words in a sentence. The dependency syntax was first proposed by French linguist L. Tesniere (1959) in his works, which analyzed a sentence into a dependency syntax tree to describe the dependency relationships between words. In the structure of dependency grammars, there is a direct dependency relationship between words to form a dependency pair, one of which is the core word, also known as the governing word, and the other is called the modifier, also known as the dependent word. The dependency relationship is represented by a directed arc, called the dependency arc. Take a sentence "The <e1>show</e1> centered around a <e2>beach theme</e2>." as an example for dependency parsing. The analyzed dependency relationship of the sentence is shown in Figure 2:

In order to obtain the entity dependency path, the dependency tree should be obtained from the dependency structure of the sentence first. According to the dependency tree and the annotated entities, the path between entities e_1 and e_2 on the dependency tree can be found, which is the entity dependency path. The entity dependency path is shown in Figure 3, where the red nodes represent entity nodes, and the dotted line is the entity dependency path.

3.2 Input

The pre-training model BERT is a multi-layer two-way transformer encoder. The input to BERT can be a sentence or a pair of sentences. A special tag [CLS] is the first tag of each sequence.

Given a sentence S , the dependency parsing tree is obtained through dependency parsing, and the shortest dependency path between entities is found according to the target entities (e_1, e_2). In order to prevent the path length from being 0 because one entity is dependent on the other entity, the entity

dependency path is a path that contains the entities. Special identifiers are inserted before and after the two target entities to emphasize the entities and assist the model in capturing locations of the entities.

The processed sentences and the entity dependency path are entered into the model. The location of the node words can be obtained by the entity dependency path, and the input of the one-hot vector is entered in the path. A [CLS] tag is added to the beginning of the sentence, and the data is input into a tokenizer to obtain its Token sequence. The vector representation of each Token is generated by the encoder.

3.3 Sentence representation

The hidden state sequence H output by the BERT module corresponds to each Token. H_0 is the hidden state vector corresponding to the [CLS] token, for which an activation operation is performed, the result is input to a fully connected layer, and the resulting vector H_0' is used as the representation of the sentence vector.

$$H = ENC(T)$$

1

$$H_0' = W_0 \left(\tanh(H_0) \right) + b_0$$

2

The hidden state vectors of the tags of the two pairs of entities are represented by H_i, H_j, H_m and H_n . For the vectors of the target entities, the vector between H_i and H_j represents the vector of entity e_1 , and the vector between H_m and H_n represents the vector of entity e_2 . These vectors are summed and averaged to obtain a single representation, for which a tanh activation is performed, and the result is input to a fully connected layer to obtain the required entity vector representation.

$$H^{e_1} = W_1 \left[\tanh \left(\frac{1}{n - m + 1} \sum_{t=m}^n H_t \right) \right] + b_1$$

3

$$H^{e_2} = W_2 \left[\tanh \left(\frac{1}{j - i + 1} \sum_{t=i}^j H_t \right) \right] + b_2$$

4

For the vector representation of dependency syntax and the representation of entity vector, the same method is adopted. POS represents the position of the word in the sentence. According to the shortest dependent path, we can obtain the position of the word on the path in the sentence. Then a tanh

activation is performed, and the result is input to a fully connected layer to obtain a single syntax vector representation.

$$p = \text{pos}(\text{word}_{\text{SDP}})$$

5

$$H^{\text{sdp}} = W_3 \left[\tanh \left(\text{Average} \left(\sum_{t=s} H_s \right) \right) \right] + b_3 \quad (s \in p)$$

6

After the single representations of sentence vector, entity vector and dependency syntactic vector are obtained, the four vectors are spliced to obtain a vector z. The vector z is input to a fully connected layer, and the resulting vector is the final sentence representation r used for classification.

$$r = W_4 [\text{concat}(H_0, H^{e1}, H^{e2}, H^{\text{sdp}})] + b_4$$

7

3.4 Classification

Given a sentence x, containing the entire sentence and the analyzed shortest dependency path sequence, a vector representation r can be obtained by inputting x to a relational encoder. After the relationship representation is obtained, a fully connected softmax layer is used to predict the relationships of the sentence. Then a probability distribution P covering all predefined relationship types is obtained.

$$p(y|x, \theta) = \text{softmax}(W^T r + b^T)$$

8

where $y \in \mathcal{Y}$ is the target relationship type, and θ refers to all learnable parameters, including W^T and b^T .

4. Experiment

4.1 Dataset

In this experiment, the SemEval-2010 Task 8 dataset was used. The dataset was collected from major data sources according to nine pre-set incompatible relationships, which contains 10,717 pieces of data, including 8,000 use cases for training and 2,717 use cases for testing. All examples in the dataset were annotated with nine relationships and an Other relationship. The distribution of quantity of the nine relationship types is shown in Table 1:

Table 1
Relationship Distribution of SemEval-2010 Task 8
Dataset

Relationship Type	Training Set	Testing Set
Cause-Effect	1003	328
Component-Whole	941	312
Entity-Destination	845	292
Product-Producer	717	261
Entity-Origin	716	258
Member-Collection	690	233
Message-Topic	634	231
Content-Container	540	192
Instrument-Agency	504	156
Other	1410	454

In addition to the annotated relationship types, each data also contains two annotated entities e_1 and e_2 . The relationship types other than Other type are directional. For example, Cause-Effect (e_1, e_2) and Cause-Effect (e_2, e_1) are different. Therefore, in the experiments, 19 relationship types are usually set to make predictions.

In this paper, the macro average F1 value in the official scoring script provided by the SemEval-2010 Task 8 dataset was used for scoring. According to this scheme, the macro average F1 value scores of 9 actual relationships (excluding relationship of Other type) were calculated, and the directionality of the relationships was taken into account. The calculation of F1 values requires precision and recall. The calculation formula is shown in equations (9) to (11):

$$\text{precision} = \frac{TP}{TP + FP}$$

9

$$\text{recall} = \frac{TP}{TP + FN}$$

10

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where true positive (TP) represents the number of correct predictions in positive prediction cases, false positive (FP) represents the number of wrong predictions in positive prediction cases, and false negative (FN) represents the number of wrong predictions in the negative prediction cases.

4.2 Hyper-parameter settings

The settings of hyper-parameters are as follows:

Table 2
Hyper-parameter Settings

Parameter	Value
Batch_size	8
Max_sequence_length	384
Learning_rate	2e-5
Train_epoch	5
Adam_epsilon	1e-8
dropout_rate	0.1
seed	42

4.3 Comparison of experimental results

Table 3 compares the performance of the model in this paper with various neural network models on the SemEval-2010 Task 8 dataset, which proves that the method proposed in this paper has achieved good results. The highest value in each column of indicators is shown in bold.

Table 3
Experimental Results

Model	F1/%
RNN	77.6
Bi-LSTM	82.7
CNN+softmax	82.7
CR-CNN	84.1
Attention Bi-LSTM	85.2
Attention CNN	85.9
BERT-base	87.1
R-BERT	89.25
R-BERT+SDP	89.97

It can be seen from the results in the table that the effect of the pre-training model is much better than those of such neural network models as CNN and LSTM. In this paper, the pre-training model was also used for experiments, and the R-BERT model was selected as the Baseline model. The R-BERT model was based on the pre-training model and highlighted the entity information with special identifiers to indicate the entity location, which achieved the best results at that time, and the official F1 evaluation value reached 89.25%. On this basis, the shortest dependency path was obtained through dependency parsing and integrated into the R-BERT model in this paper, so that the model could learn the context information of sentences. The results show that the F1 value performance of the model reaches 89.97% after parsing is introduced, which fully proves that the context information provided by the dependency parsing is effective.

4.4 Ablation Experiments

The method proposed in this paper has been proved by the above experimental results. We wanted to further understand what factors besides BERT contributed to the experimental results in the method based on the pre-training model, and therefore, three ablation experiments were designed. Since the entity tags "<e1>" and "<e2>" were added to emphasize the entity and add boundary information to the entity, which significantly improved the classification prediction, these entity tags were reserved and used in each ablation experiment.

In the first experiment, a [CLS] token was added before the sentence input, the hidden layer vector of this token was used as a vector representation of sentence classification, and only this vector was used for classification. In the second experiment, [CLS] and the hidden vector of entity dependency path were spliced to obtain a vector as the sentence representation, in which the entity dependency path did not contain entity information. In the third experiment, [CLS] and the hidden vector of the entity were spliced

as sentence representation, and in this case, the entity information contained the tags of the entity and integrated the boundary information of the entity. The SDP represents the shortest dependency path.

Table 4
Comparison of Different Components
of BERT-based Method

Relationship Representation	F1/%
[CLS]	87.99
[CLS]+SDP	89.15
[CLS]+ENT	89.23
[CLS]+ENT+SDP	89.97

It can be seen from the results in Table 4 that the experimental results are improved after the addition of entity identifiers, which provide the model with the boundary information of the entity and emphasize the entity. There is little difference between the result of using the hidden vector of entity dependency path information as sentence representation and that of using the hidden vector of entity as sentence representation, but the result of using entity information is better. Experimental results show that the model can make use of context information, but the model still needs entity information for supplementation. After combining the entity information with the context information provided by the dependency parsing, the model can predict the classification better.

4.5 Case study

This section analyzes the results of the R-BERT model and the model proposed in this paper in detail, and compares the results of various relationship types, as shown in Table 5.

Table 5
Comparison of F1 Values of Various Relationship Types

Relationship	R-BERT	BERT+ENT+SDP
Cause-Effect	93.11	92.47
Component-Whole	87.34	87.72
Content-Container	90.03	92.93
Entity-Destination	94.18	93.68
Entity-Origin	89.14	89.15
Instrument-Agency	82.87	84.04
Member-Collection	87.82	88.48
Message-Topic	90.77	91.31
Product-Producer	87.82	90.00
Other	64.22	67.14
Official Score	89.23	89.97

The results in the table show that the classification effect for most relationship types is improved compared with Baseline after the introduction of the entity dependency path, and the effect is more obvious for such relationship types as Content-Container, Product-Producer and Instrument-Agency, indicating that this experiment has successfully integrated the entity dependency path into the pre-training model, and is beneficial to improving the effect of relationship classification.

However, the classification effect for Cause-Effect and Entity-Destination has not improved, but reduced significantly. Therefore, we reviewed the classification results obtained by the two models in detail, and extracted the examples of wrong classification results of the two models respectively. Table 6 provides detailed examples of classifications errors in these two types.

Table 6
Comparative Examples of Results Generated by Models

A few days before the service, Tom Burris had thrown into Karen's <e1> casket </e1> his wedding <e2> ring </e2>.			
prediction	Official	R-BERT	BERT+ENT+SDP
	Entity-Destination(e2, e1)	Other	Entity-Destination(e1, e2)

Each time a <e1> neuron </e1> unleashes its tiny <e2> jolt </e2>, it needs to replenish its stores of energy for the next spark.			
prediction	Official	R-BERT	BERT+ENT+SDP
	Cause- Effect (e1, e2)	Other	Cause-Effect(e2, e1)

These wind <e1> turbines </e1> generate <e2> electricity </e2> from naturally occurring wind.			
prediction	Official	R-BERT	BERT+ENT+SDP
	Cause-Effect (e1, e2)	Content-Container(e1, e2)	Cause-Effect(e2, e1)

From the results of data classification in the table, we can see that in the prediction results of these two types, the model proposed in this paper correctly predicted the relationship types, but mispredicted the relationship directions, and the relationship types predicted by the baseline model were different from the standard results. Therefore, taking Cause-Effect as an example, when the accuracy of this type is calculated on the premise that the recall rates of the two models are not much different, due to the wrong relationship directions in the prediction results on some data, the model in this paper predicted more data to be of Cause-Effect type than that of the baseline model, so the accuracy rate obtained is lower. As a result, the F1 value evaluation of the Cause-Effect classification results is lower than that of the baseline model.

It can be seen from the above results that the method proposed by the model in this paper not only allows the model to learn the context information provided by the dependency syntax, but also improves the prediction of the model. However, the model underutilizes the context information of the data in some relationship types, resulting in correct classification of relationship types and wrong classification of relationship directions. In this case, it shows that there is still room for improvement in the use of context information, which is also the focus of our following work.

5. Experiment

Based on the pre-training model, this paper proposes a pre-training model integrating dependency parsing for supervised entity relationship extraction. The shortest dependency path between entities is obtained by dependency parsing, which concisely expresses the syntactic relationships between entities, retains the main part of the expression relationship type, and removes useless modifier chunks and redundant entity information. The context information between entities can be obtained through dependency parsing, and a syntactic representation can be obtained by adopting the same processing method as the entity representation in the R-BERT model, which is spliced to the sentence vector and entity vector to

obtain the vector representation for classification. The F1 value is increased to 89.97% on the SemEval-2010 Task 8 dataset, which is an increase of 0.72% compared with R-BERT. Through the analysis of the results and the comparison of the results of the models, the model in this paper has achieved good results, successfully learned the context information of sentences, basically solved the problems raised and achieved the expected results.

In the detailed analysis of the results, it is found that the length of the dependency path between entities in some sentences is 0. In order to avoid this situation, this paper also counts entities as nodes on the path during data processing. However, the model cannot obtain enough context information from these data, and the entity information is only reused, which lead to the situation that in some relationship types, entity relationship is accurately predicted, while the direction of the relationship is incorrectly predicted, affecting the final relationship prediction results to some extent. Therefore, in the next work, we will try to design strategies to extract context information for these sentences to improve the overall effect of relationship extraction.

Declarations

Authors' contributions Authors have equal contributions.

Funding This work was supported by the National Defense Science and Technology Industrial Technology Research Project (JSQB2017206C002).

Availability of data and material Data for this work were obtained from the web (Access from www.semeval2.fbk.eu/ semeval2.php)

Conflict of interest The authors declare no conflict of interest.

Ethical approval The authors declare that they do not have any conflict of interest. This research does not involve any human or animal participation. All authors have checked and agreed the submission.

Consent for publication Author have taken all the consent for publication.

References

1. Golshan PN, Dashti HR, Azizi S. A study of recent contributions on information extraction. arXiv preprint arXiv:1803.05667, 2018.
2. Liu Q, Li Y, Duan H, Liu Y, Qin ZG. A Survey of Knowledge Mapping Construction Techniques. Journal of Computer Research and Development, 2016,53:582–600 (in Chinese with English abstract).
3. Wu F, Weld D S. Open information extraction using Wikipedia[C]// ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010:118-127.
4. Li DM, Zhang Y, Li DY, et al. Review of Entity Relation Extraction Methods[J]. Journal of Computer Research and Development, 2020(7):1424-1448.

5. Wu SC, He YF. Enriching Pre-trained Language Model with Entity Information for Relation Classification[C]// the 28th ACM International Conference. ACM, 2019:2361-2364.
6. Tesnière L. Elements de syntaxe structurale[M]. Paris:clincksieck, 1959.
7. E HH, Zhang WJ, Xiao SQ, Cheng R, Hu YX, Zhou XS, Niu PQ. Survey of entity relationship extraction based on deep learning. Ruan Jian Xue Bao/Journal of Software, 2019,30(6):1793-1818 (in Chinese). <http://www.jos.org.cn/1000-9825/5817.htm>
8. Zhou GD, Su J, Zhang J, et al. Exploring various knowledge in relation extraction[C]. In: Proc. of the Conf. on Meeting of the Association for Computational Linguistics (ACL 2005). University of Michigan, 2002:419-444.
9. Guo XY, He TT, Hu XH, Chen QJ. Chinese Entity Relationship Extraction based on Syntactic and Semantic Features. Journal of Chinese Information Processing, 2014,183-189 (in Chinese with English abstract).
10. Craven M, Kumlien J. Constructing Biological Knowledge Bases by Extracting Information from Text Sources[J]. Proceedings / International Conference on Intelligent Systems for Molecular Biology; ISMB. International Conference on Intelligent Systems for Molecular Biology, 1999:77-86.
11. Brin S. Extracting Patterns and Relations from the World Wide Web[J]. Springer Berlin Heidelberg, 1998:172-183.
12. Hasegawa T, Sekine S, Grishman R. Discovering Relations among Named Entities from Large Corpora[C]. Proceedings of the Meeting on Association for Computational Linguistics, 2004:415-es.
13. Zeng D, Liu K, Lai S, Zhou G, Zhao J. Relation classification via convolutional deep neural network[C]. Proceedings of the 25th Int'l Conf. on Computational Linguistics: Technical Papers (COLING 2014). 2014. 2335-2344.
14. Zhang D, Wang D. Relation classification via recurrent neural network. arXiv preprint arXiv:1508.01006, 2015.
15. Xu K, Feng Y, Huang S, Zhao D. Semantic relation classification via convolutional neural networks with simple negative sampling[J]. Computer Science. 2015;71: 941-9.
16. Katiyar A, Cardie C. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees[C]. Proceedings of the Conference on Association for Computational Linguistics. 2017. 917-928.
17. Zeng D, Liu K, Chen Y, et al. Distant supervision for relation extraction via piecewise convolutional neural networks[C]. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2015. 1753-1762.
18. Lin Y, Shen S, Liu Z, et al. Neural relation extraction with selective attention over instances[C]. In: Proc. of the Meeting of the Association for Computational Linguistics. 2016. 2124-2133.
19. Ji GL, Liu K, He SZ, et al. Distant supervision for relation extraction with sentence-level attention and entity descriptions[C]. In: Proc. of the AAAI. 2017. 3060-3066.

20. Ren X, Wu Z, He W, et al. CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases[C]// the 26th International Conference. International World Wide Web Conferences Steering Committee, 2016:1015-1024.
21. Huang YY, Wang WY. Deep Residual Learning for Weakly-Supervised Relation Extraction[C]. Proceedings of the Conference on EMNLP 2017: 1803-1807.
22. Soares L B, Fitzgerald N, Ling J, et al. Matching the Blanks: Distributional Similarity for Relation Learning[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.
23. Peng H, Gao T, Han X, et al. Learning from Context or Names? An Empirical Study on Neural Relation Extraction[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, November 16–20, 2020:3661–3672.
24. Tao Q, X Luo, H Wang. Enhancing Relation Extraction Using Syntactic Indicators and Sentential Contexts[J]. arXiv, 2019.

Figures

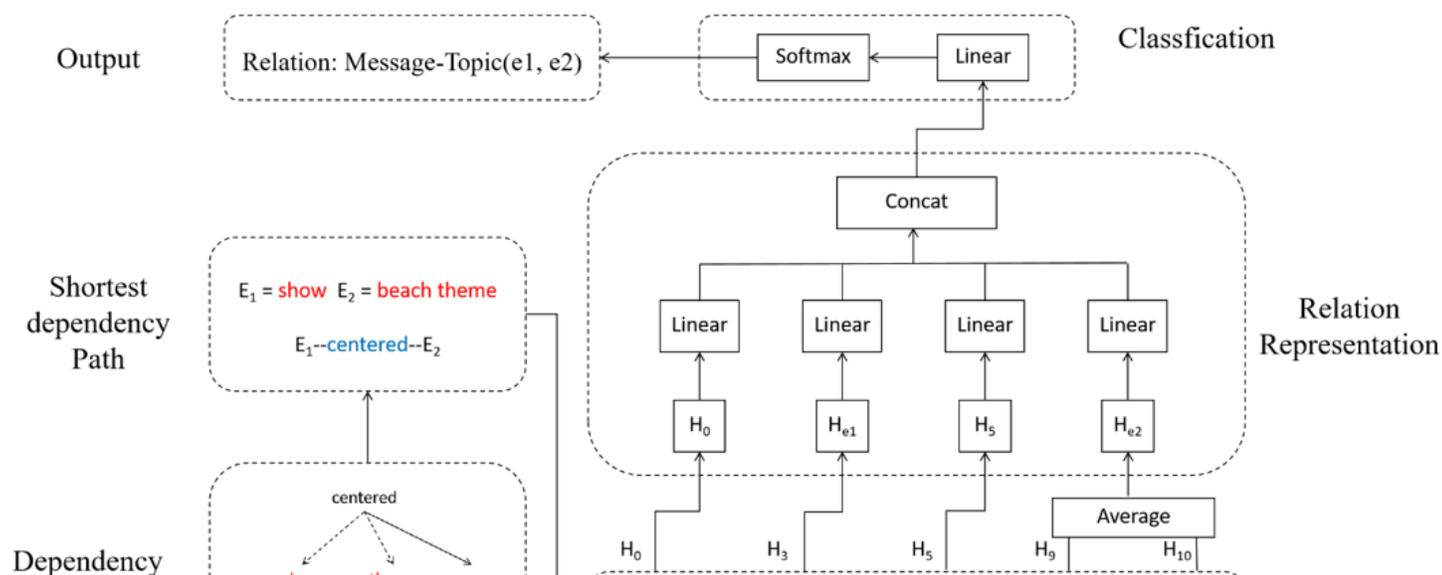


Figure 1

Model Structure Diagram.

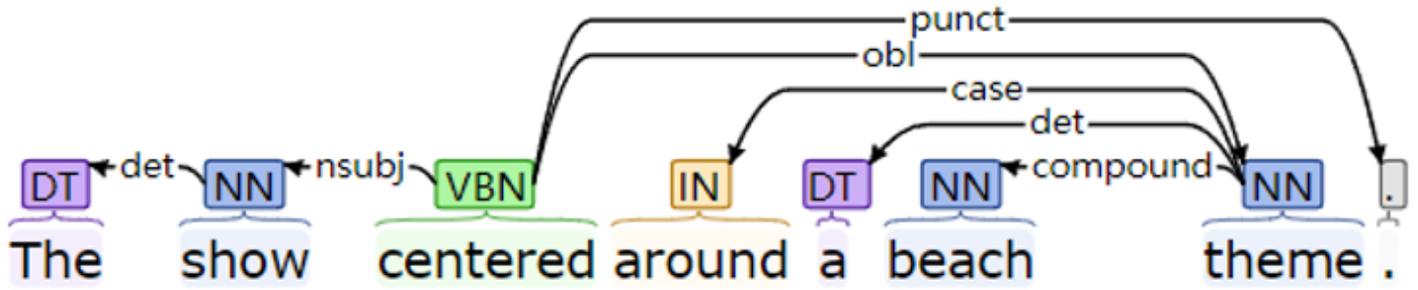


Figure 2

Dependency Relationship Graph.

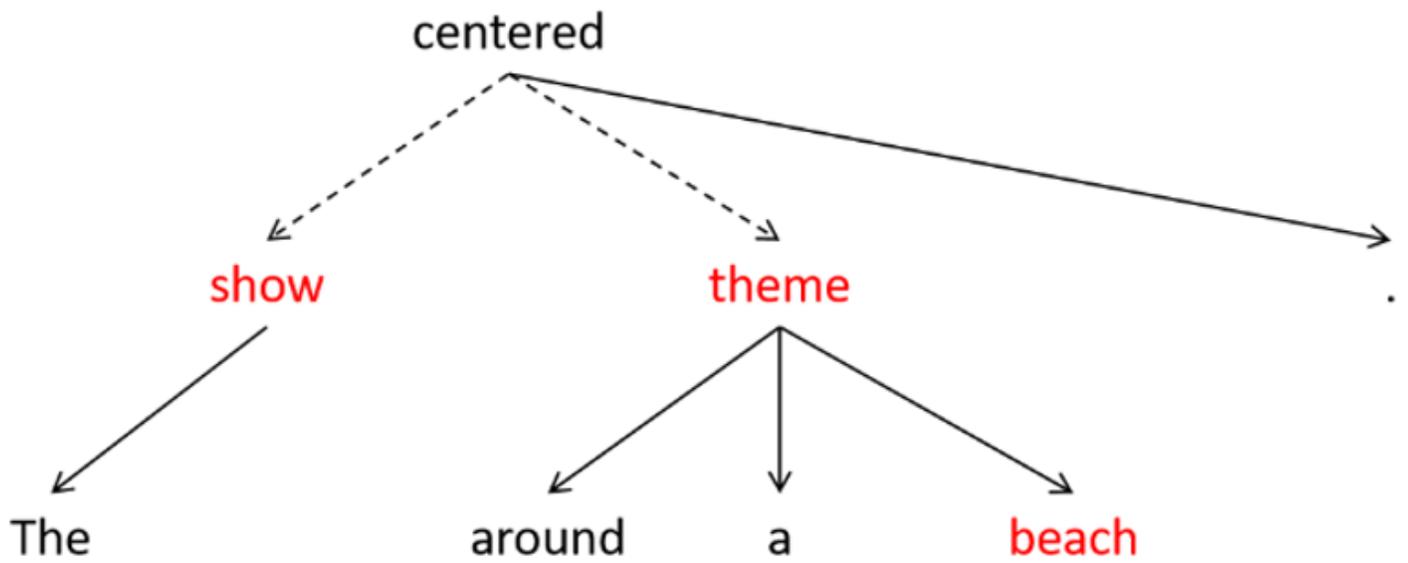


Figure 3

Entity Dependency Path.