

Improving Semantic Relation Extraction System with Compositional Dependency Unit on Enriched Shortest Dependency Path

Duy-Cat Can, Hoang-Quynh Le, and Quang-Thuy Ha

Faculty of Information Technology, University of Engineering and Technology
Vietnam National University Hanoi, Vietnam
{catcd, lhquynh, thuyhq}@vnu.edu.vn

Abstract. Experimental performance on the task of relation extraction/classification has generally improved using deep neural network architectures. In which, data representation has been proven to be one of the most influential factors to the model’s performance but still has many limitations. In this work, we take advantage of compressed information in the shortest dependency path (SDP) between two corresponding entities to classify the relation between them. We propose (i) a compositional embedding that combines several dominant linguistic as well as architectural features and (ii) dependency tree normalization techniques for generating rich representations for both words and dependency relations in the SDP. We also present a Convolutional Neural Network (CNN) model to process the proposed SDP enriched representation. Experimental results for both general and biomedical data demonstrate the effectiveness of compositional embedding, dependency tree normalization technique as well as the suitability of the CNN model.

Keywords: Relation extraction · Dependency unit · Shortest dependency path · Convolutional neural network.

1 Introduction

Relation extraction (RE) is an important task of natural language processing (NLP). It plays an essential role in knowledge extraction tasks from information extraction [17], question answering [13], medical and biomedical informatics [4] to improving the access to scientific literature [5], etc. The relation extraction task can be defined as the task of identifying the semantic relations between two entities e_1 and e_2 in a given sentence S to a pre-defined relation type [5].

Many deep neural network (DNN) architectures are introduced to learn a robust feature set from unstructured data [15], which have been proved effective, but, often suffer from irrelevant information, especially when the distance between two entities is too long. Previous researches have illustrated the effectiveness of the shortest dependency path between entities for relation extraction [4]. We, therefore, propose a model that using convolution neural network (CNN) [9] to learn more robust relation representation through the SDP.

The on-trending researches demonstrated that machine learn a language better by using a deep understanding of words. The better representation of data may help machine learning models understanding data better. Word representation has been studied for a long time, several approaches to embed a word into an informative vector has been proposed [11,1], especially with the development of deep learning. Up to now, enriching word representation is still attracting the interest of the research community; in most cases, sophisticated design is required [7]. Meanwhile, the problem of representing the dependency between words is still an open problem. In our knowledge, most previous researches often used a simple way to represent them, or even ignore them in the SDP [18].

Considering these problems as motivation to improve, in this paper, we present a compositional embedding that takes advantage of several dominant linguistic and architectural features. These compositional embedding then are processed within a dependency unit manner to represent the SDPs.

The main contributions of our work can be concluded as:

1. We introduce a enriched representation of SDP that utilizes a major part of linguistic and architectural features by using compositional embedding.
2. We investigate the effectiveness of dependency tree normalizing before generating the SDP.
3. We propose a deep neural architecture which processes the above enriched SDP effectively; we also further investigate the contributions of model components and features to the final performance that provide a useful insight into some aspects of our approach for future research.

2 Related Work

Relation extraction has been widely studied in the NLP community for many years. There has been a variety of computational models applied to this problem, and supervised methods have shown to be the most effective approach. Generally, these methods can be divided into two categories: feature engineering-based methods and deep learning-based methods.

With feature-based methods, researchers concentrate on extracting a rich feature set. The typical studies are of Le et al. [8] and Rink et al. [14], in which variety of handcrafted features that capture the, semantic and syntactic information are fed to an SVM classifier to extract the relations of the nominals. However, these methods suffer from the problem of selecting a suitable feature set for each particular data that requires tremendous human labor.

In the last decade, deep learning methods have made significant improvement and produced the state-of-the-art result in relation extraction. These methods usually utilize the word embeddings with various DNN architectures to learn the features without prior knowledge. Socher et al. [15] proposed a Recursive Neural Network (mvRNN) on tree structure to determine the relations between nominals. Study of Zhou et al. [21] presents an ensemble model using DNN with syntactic and semantic information. Some other studies use all words in sentence with position feature [20] to extract the relations within it.

In recent years, many studies attempt other possibilities by using dependency tree-based methods. Panyam et al. [12] exploit graph kernels using constituency parse tree and dependency parse tree of a sentence. The SDP also receives more and more attention on relation extraction researches. CNN models (Xu et al. [18]) are among the earliest approaches applied on SDP. Xu et al. [19] rebuilt an Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) unit on the dependency path between two marked entities to utilize sequential information of sentences. Various of improvements have been suggested to boost the performance of RE models, such as negative sampling [18], exploring subtrees go along with SDP’s node [10], voting schema and combining several deep neural networks [7].

3 Enriched Shortest Dependency Path

3.1 Dependency Tree and Shortest Dependency Path

The dependency tree of a sentence is a tree-structural representation, in which each token is represented as a node and each token-token dependency is represented as a directed edge. The original dependency tree provides the full grammatical information of a sentence, but some of this information may be not useful for the relation extraction problem, even bring noises.

The Shortest Dependency Path (SDP) is the shortest sequence go from a starting token to the ending token in the dependency tree. Because the SDP represents the concise information between two entities [3], we suppose that the SDP contain necessary information to shows their relationship.

3.2 Dependency Tree Normalization

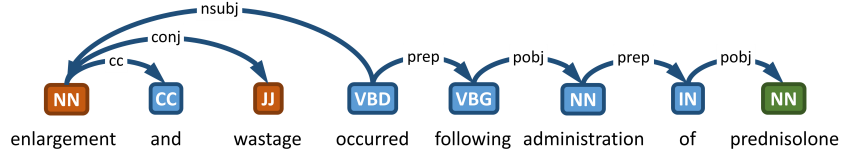
In this work, we applied two techniques to normalize the dependency tree, in order to reduce noise as well as enrich information in the SDP extracted from the dependency tree (see Figure 1 for example).

Preposition normalization: We collapse the “*pobj*” dependency (object of preposition) with the predecessor dependency (*e.g.*, “*prep*”, “*acl*”, etc) into a single dependency, and cut the preposition off from the SDP.

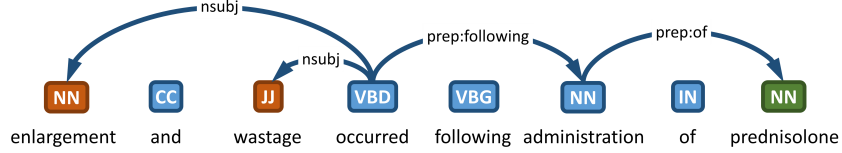
Conjunction normalization: Base on the assumption that two tokens that linked by a conjunction dependency “*conj*” should have the same semantical and grammatical roles; we then add a skip-edges to ensure that these conjuncted tokens have same dependencies with other tokens.

3.3 The Dependency Unit on the SDP

According to the study of [7], a pair of a token and its ancestor has the difference in meaning when they are linked by a different dependency relation. We make use of this structure and represent the SDP as a sequence of substructures like “ $t_a \xleftarrow{r_{ab}} t_b$ ”, in which t_a and t_b are token and its ancestor respectively; r_{ab} is the dependency relation between them. This substructure refers to the Dependency Unit (DU) as described in Figure 2.



(a) Subtree from original dependency tree.



(b) Subtree from normalized dependency tree.

Fig. 1: Example of normalized dependency tree.

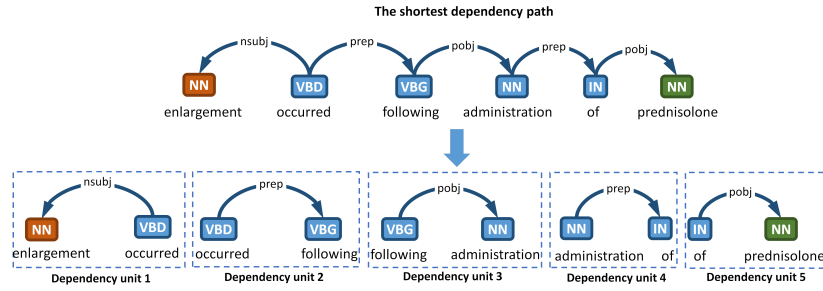


Fig. 2: Dependency units on the SDP.

4 Proposed Model

We design our cduCNN model to learn the features on the sequence of DUs that consist of both token and dependency information. Figure 3 depicts the overall architecture of our proposed model. The model mainly consists of three components: compositional embeddings layer, convolution phase, and a softmax classifier.

Given the dependency tree of a sentence as input, we extract the shortest path between two entities from the tree, pass it through an embedding generation layer for token embeddings and dependency embeddings. These two embeddings matrix are then composed into dependency units. A convolution layer is applied to capture local features from each unit and its neighbors. A max pooling layer thereafter gathers information from these features combines these features into a global feature vector, and a softmax layer is followed to perform a $(K + 1)$ -class classification. This final $(K + 1)$ -class distribution indicates the probability of each relation respectively. The details of each layer are described below.

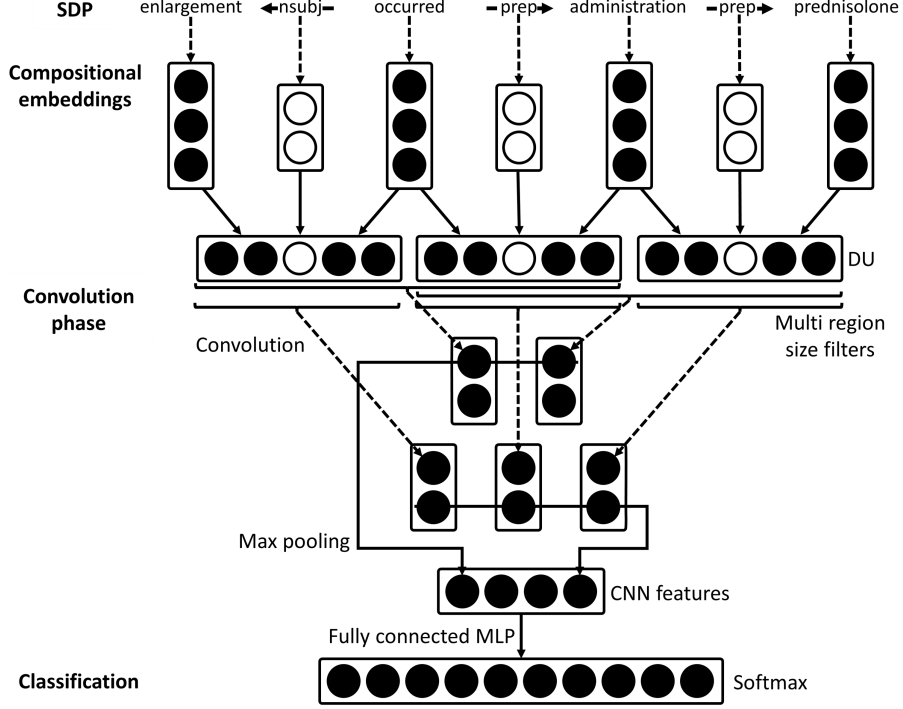


Fig. 3: An overview of proposed model.

4.1 Compositional embeddings

In the embeddings layer, each component of the SDP (*i.e.*, token or dependency) is transformed into a vector $w_e \in \mathbb{R}^d$, where d is the desired embedding dimension. In order to capture more features along the SDP, we compositionally represent the token and dependency on SDP with various type of information.

Dependency embeddings: The dependency directions are proved effective for the relation extraction task [18]. However, treated the dependency relations with opposite directions as two separated relations can induce that two vectors of the same relation are disparate. We represent dependency relation dep_i as a vector that is the concatenation of dependency type and dependency direction. The concatenated vector is then transform into a final representation d_i of dependency relation as follow:

$$d_i = \tanh(W_d [d_i^{typ} \oplus d_i^{dir}] + b_d) \quad (1)$$

where $d^{typ} \in \mathbb{R}^{d^{typ}}$ represents the dependency relation type among 62 labels; and $d^{dir} \in \mathbb{R}^{d^{dir}}$ is the direction of the dependency relation, *i.e.* from left-to-right or vice versa on the SDP.

Token embeddings: For token representation, we take advantage of five types of information, including:

- ***Pre-trained fastText embeddings*** [1]: which learned the word representation based on its external context, therefore allows words that often appear in similar context to have similar representations. Each token in the input SDP is transformed into a vector t_i^w by looking up the embedding matrix $W_w^e \in \mathbb{R}^{d^w \times |V^w|}$, where V^w is a vocabulary of all words we consider.
- ***Character-based embeddings***: CNN is an effective approach to learn the character-level representations that offer the information about word morphology and shape (like the prefix or suffix of word). Given a token composed of n characters c_1, c_2, \dots, c_n , we first represent each character c_i by an embedding r_i using a look-up table $W_c^e \in \mathbb{R}^{d^{char} \times |V^c|}$, where V^c is the alphabet. A deep CNN with various window sizes is applied on the sequence $\{r_1, r_2, \dots, r_n\}$ to capture the character features. A pooling layer is followed to produce the final character embedding t_i^c .
- ***Position embeddings***: To extract the semantic relation, the structure features (*e.g.*, the SDP between nominals) do not have sufficient information. The SDP is lack of in-sentence location information that the informative words are usually close to the target entities. We make use of position embeddings to keep track of how close each SDP token is to the target entities on the original sentence. We first create a 2-dimensional vector $[d_i^{e1}, d_i^{e2}]$ for each token that is combination of relative distances from current token to two entities. Then, we obtain the position embedding t_i^p as follow:

$$t_i^p = \tanh(W_p[d_i^{e1}, d_i^{e2}] + b_p) \quad (2)$$

- ***POS tag embeddings***: A token may have more than one meaning representing by its grammatical tag such as noun, verb, adjective, adverb, etc. To address this problem, we use the part-of-speech (POS) tag information in the token representation. We randomly initialize the embeddings matrix $W_t^e \in \mathbb{R}^{d^t \times 56}$ for 56 OntoNotes v5.0 of the Penn Treebank POS tags. Each POS tag is then represented as a corresponding vector t_i^t .
- ***WordNet embeddings***: WordNet is a large lexical database containing the set of the cognitive synonyms (synsets). Each synset represents a distinct concept of a group and has a coarse-grained POS tag (*i.e.*, nouns, verbs, adjectives or adverbs). Synsets are interlinked by their conceptual-semantic and lexical meanings. For this paper, we heuristically select 45 F1-children of the WordNet root which can represent the super-senses of all synsets. The WordNet embedding t_i^n of a token is in form of a sparse vector that figure out which sets the token belongs to.

Finally, we concatenate the word embedding, character-based embedding, position embedding, POS tag embedding, and WordNet embedding of each token into a vector, and transform it into the final token embedding as follow:

$$t_i = \tanh(W_t[t_i^w \oplus t_i^c \oplus t_i^p \oplus t_i^t \oplus t_i^n] + b_t) \quad (3)$$

4.2 CNN with Dependency Unit

Our CNN receives the sequence of DUs $[u_1, u_2, \dots, u_n]$ as the input, in which two token embeddings t_i , t_{i+1} and dependency relation d_i are concatenate into a d -dimensional vector u_i . Formally, we have:

$$u_i = t_i \oplus d_i \oplus t_{i+1} \quad (4)$$

In general, let the vector $u_{i:i+j}$ refer to the concatenation of $[u_i, u_{i+1}, \dots, u_{i+j}]$. A convolution operation with region size r applies a filter $w_c \in \mathbb{R}^{rd}$ on a window of r successive units to capture a local feature. We apply this filter to all possible window on the SDP $[u_{1:r}, u_{2:r+1}, \dots, u_{n-r+1:n}]$ to produce convolved feature map. For example, a feature map $c^r \in \mathbb{R}^{n-r+1}$ is generated from a SDP of n DUs by:

$$c^r = \left\{ \tanh(w_c u_{i:i+r-1} + b_c) \right\}_{i=1}^{n-r+1} \quad (5)$$

We then gather the most important features from the feature map, which have the highest values by applying a max pooling [2] layer. This idea of pooling can naturally deal with variable sentence lengths since we take only the maximum value $\hat{c} = \max(c^r)$ as the feature to this particular filter.

Our model manipulates multiple filters with varying region sizes (1 – 3) to obtain a feature vector f which take advantage from wide ranges of n-gram features that can boost relation extraction performance.

4.3 Classification

The features from the penultimate layer are then fed into a fully connected multi-layer perceptron network (MLP). The output h_n of the last hidden layer is the higher abstraction-level features, which is then fed to a softmax classifier to predict a $(K + 1)$ -class distribution over labels \hat{y} :

$$\hat{y} = \text{softmax}(W_y h_n + b_y) \quad (6)$$

4.4 Objective Function and Learning Method

The proposed cduCNN relation classification model can be stated as a parameter tuple θ . The $(K + 1)$ -class distribution \hat{y} predicted by the softmax classifier denotes the probability that SDP is of relation R . We compute the the penalized cross-entropy, and further define the training objective for a data sample as:

$$L(\theta) = - \sum_{i=0}^K y_i \log \hat{y}_i + \lambda \|\theta\|^2 \quad (7)$$

where $y \in \{0, 1\}^{(K+1)}$ indicating the one-hot vector represented the target label, and λ is a regularization coefficient. To compute the model parameters θ , we minimize $L(\theta)$ by applying mini-batch gradient descent (GD) with Adam optimizer [6] in our experiments. θ is randomly initialized and is updated via back-propagation through neural network structures.

Table 1: System’s performance on SemEval-2010 Task 8 dataset

Model	Feature set	F1
SVM (Rink et al., 2010)	Lexical features, dependency parse, hypernym, NGrams, PropBank, FanmeNet, NomLex-Plus, TextRunner	82.2
CNN (Zeng et al., 2014)	Word embeddings + Lexical features, WordNet, position feature	69.7 82.7
mvRNN (Socher et al., 2012)	Word embeddings + WordNet, NER, POS tag	79.1 82.4
SDP-LSTM (Xu et al., 2015b)	Word embeddings + WordNet, GR, POS tag	82.4 83.7
depLCNN (Xu et al., 2015a)	Word embeddings + WordNet, word around nominals + Negative sampling	81.9 83.7 85.6
Baseline	Word embeddings + DU	83.4 83.7
cduCNN (our model)	Compositional Embedding, DU + Normalize conjunction + Ensemble + Normalize object of a preposition	84.7 85.1 86.1 80.6

5 Experimental evaluation

5.1 Dataset

Our model was evaluated on two different datasets: SemEval-2010 Task 8 for general domain relation extraction and BioCreative V CDR for chemical-induced disease relation extraction in biomedical scientific abstracts.

The SemEval-2010 Task 8 [5] contains 10,717 annotated relation classification examples and is separated into two subsets: 8,000 instances for training and 2,717 for testing. We randomly split 10 percents of the training data for validation. There are 9 directed relations and one undirected **Other** class.

The BioCreative V CDR task corpus [16] (BC5 corpus) consists of three datasets, called training, development and testing set. Each dataset has 500 PubMed abstracts, in which each abstract contains human annotated chemicals, diseases entities, and their abstract-level chemical-induced disease relations.

In the experiments, we fine-tune our model on training (and development) set(s) and report the results on the testing set, which is kept secret with the model. We conduct the training and testing process 20 times and calculate the averaged results. For evaluation, the predicted labels were compared to the golden annotated data using standard precision (P), recall (R), and F1 score metrics.

5.2 Experimental results and discussion

System’s performance: Table 1 summarizes the performances of our model and comparative models. For a fair comparison with other researches, we implemented a baseline model, in which we interleave the word embeddings and

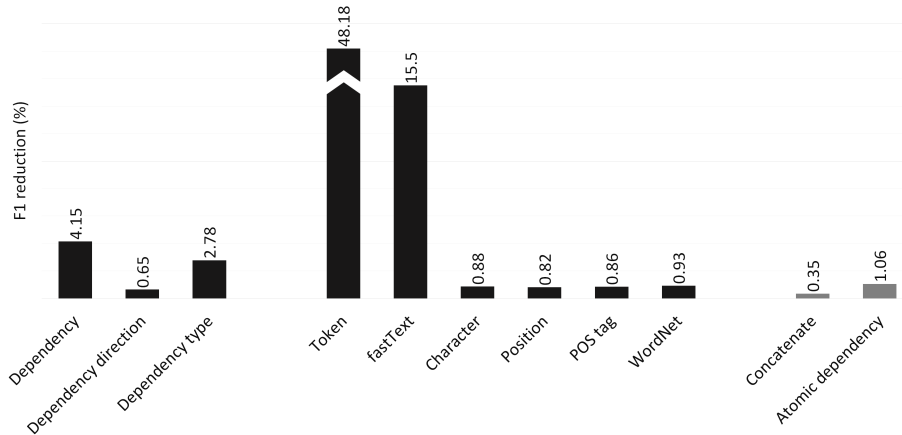


Fig. 4: Contribution of each component. The black columns indicate the kick-out of components. The grey columns indicate the alternative methods of embedding.

dependency type embeddings for the input of CNN. It yields higher F1 than competitors which are feature-based or DNN-based with information from pre-trained Word embeddings only. With the improvement of 0.3% when applying DU on the baseline model, our model achieves the better result than the remaining comparative DNN approaches which utilized full sentence and position feature without the advanced information selection methods (*e.g.*, attention mechanism). This result is also equivalent to other SDP-based methods.

The results also demonstrate the effectiveness of using compositional embedding that brings an improvement of 1.0% in F1. Our cduCNN model yields an F1-score of 84.7%, outperforms other comparative models, except depLCNN model with data augmented strategy, by a large margin. However, the ensemble strategy by majority voting on the results of 20 runs drives our model to achieve a better result than the augmented depLCNN model.

It is worth to note that we have also conducted two techniques to normalize the dependency tree. Unfortunately, the results did not meet our expectations, with only 0.4% improvement of conjunction normalization. Normalizing the object of preposition even degrades the performance of the model with 4.1% of F1 reduction. A possible reason is that the preposition itself represent the relation on SDP, such as “*scars from stitches*” shows **Cause-Effect** relation while “*clip about crime*” shows **Message-Topic** relation. With the cut-off of prepositions, the SDP is lack of information to predict the relation.

Contribution of components on enriched SDP: Figure 4 shows the changes in F1 when ablating each component and information source from the cduCNN model. The F1 reductions illustrate the contributions of all proposals to the final result. However, the important levels are varied among different components and information sources. Both dependency and token embeddings have a great

Table 2: System’s performance on BioCreative V CDR dataset

Model	Feature set	P	R	F1
BioCreative benchmarks	Average result*	47.09	42.61	43.37
	Rank no. 1 result*	55.67	58.44	57.03
UET-CAM (Le et al., 2016)	SVM, rich feature set	53.41	49.91	51.60
	+ silverCID corpus	57.63	60.23	58.90
hybridDNN (Zhou et al., 2016)	Syntactic feature, word embeddings	62.15	47.28	53.70
	+ Context	62.39	47.47	53.92
	+ Position	62.86	47.47	54.09
ASM (Panyam et al., 2018)	Dependency graph	49.00	67.40	56.80
Baseline	Word embeddings	60.25	49.37	54.27
	+ DU	60.33	50.36	54.90
cduCNN (our model)	Compositional Embedding, DU	57.24	55.27	56.24
	+ Normalize conjunction	56.95	56.14	56.54
	+ Ensemble	58.74	56.10	57.39
	+ Post processing	52.09	70.09	59.75
	+ Normalize object of a preposition	56.66	55.94	56.30

* results are provided by the BioCreative V.

influence on the model performance. Token embedding plays the leading role, eliminating it will reduce the F1 by 48.18%. However, dependency embedding is also an essential component to have the good results. Removing fastText embedding, dependency embedding and dependency type make significant changes of 15.5% 4.15% and 2.78% respectively. The use of other components brings a quite small improvement.

An interesting observation comes from the interior of dependency and token embeddings. The impact of kicking the whole component out is much higher than the total impact of kicking each minor component out. This proves that the combination of constituent parts is thoroughly utilized by our compositional embedding structure.

Another experiment on using alternative methods of embedding also proves the minor improvement of compositional embedding. The result lightly reduces when we concatenate the embedding elements directly without transforming into a final vector or treat two divergent directional relations as to atomic relations.

Model’s adaptation to other domain: Table 2 shows our results on biomedical BioCreative V CDR corpus compared to some related researches. Our model outperforms the traditional SVM model using rich feature set without additional data and the hybrid DNN model with position feature. The average result is lower than ASM model using dependency graph. However, the conjunction normalization and ensemble technique can boost our F1-score 1.15%.

We further apply the post processing rules on the predictions of the model to improve the recall and achieve the best result among competing models with 59.75%. The results also highlight out the limitation of our model about cross-sentence relation. We leave this issue for our future works.

6 Conclusion

In this paper, we have presented a neural relation extraction architecture with the compositional representation of the SDP. The proposed model is capable of utilizing the dominant linguistic and architectural features, such as word embeddings, character embeddings, position feature, WordNet and part-of-speech tag.

The experiments on SemEval-2010 Task 8 and BioCreative V CDR datasets showed that our model achieves promising result when compared with other comparative models. We also investigated and verified the rationality and contributions of each model’s constituent parts, features, and additional techniques. The result also demonstrated the adaptability of our model on classifying many types of relation in different domains.

Our limitation of cross-sentence relations extraction is highlighted since it resulted in low performance on the BioCreative V CDR corpus compared to state-of-the-art results which handled this problem significantly. Moreover, the SDP between two nominals may be lack of supported information, raising the motivation to take advantages of more informative feature for augmenting the SDP. We aim to address these problems in our future works.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**, 135–146 (2017)
2. Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. pp. 111–118 (2010)
3. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: *Proceedings of the conference on human language technology and empirical methods in natural language processing*. pp. 724–731. Association for Computational Linguistics (2005)
4. Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., Kalinin, A.A., Do, B.T., Way, G.P., Ferrero, E., Agapow, P.M., Zietz, M., Hoffman, M.M., et al.: Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface* **15**(141), 20170387 (2018)
5. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: *Proceedings of the Workshop on Semantic Evaluations*. pp. 94–99 (2009)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
7. Le, H.Q., Can, D.C., Vu, S.T., Dang, T.H., Pilehvar, M.T., Collier, N.: Large-scale exploration of neural relation classification architectures. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pp. 2266–2277 (2018)

8. Le, H.Q., Tran, M.V., Dang, T.H., Ha, Q.T., Collier, N.: Sieve-based coreference resolution enhances semi-supervised learning model for chemical-induced disease relation extraction. *Database* **2016** (07 2016). <https://doi.org/10.1093/database/baw102>
9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
10. Liu, Y., Wei, F., Li, S., Ji, H., Zhou, M., Houfeng, W.: A dependency-based neural network for relation classification. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. vol. 2, pp. 285–290 (2015)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. pp. 3111–3119 (2013)
12. Panyam, N.C., Verspoor, K., Cohn, T., Ramamohanarao, K.: Exploiting graph kernels for high performance biomedical relation extraction. *Journal of biomedical semantics* **9**(1), 7 (2018)
13. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pp. 2383–2392 (2016)
14. Rink, B., Harabagiu, S.: Utd: Classifying semantic relations by combining lexical and semantic resources. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. pp. 256–259. Association for Computational Linguistics (2010)
15. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. pp. 1201–1211. ACL (2012)
16. Wei, C.H., Peng, Y., Leaman, R., Davis, A.P., Mattingly, C.J., Li, J., Wieggers, T.C., Lu, Z.: Overview of the biocreative v chemical disease relation (cdr) task. In: *Proceedings of the fifth BioCreative challenge evaluation workshop*. pp. 154–166 (2015)
17. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. pp. 118–127. Association for Computational Linguistics (2010)
18. Xu, K., Feng, Y., Huang, S., Zhao, D.: Semantic relation classification via convolutional neural networks with simple negative sampling. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. pp. 536–540 (2015)
19. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. pp. 1785–1794 (2015)
20. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*. pp. 2335–2344 (2014)
21. Zhou, H., Deng, H., Chen, L., Yang, Y., Jia, C., Huang, D.: Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database* **2016** (04 2016). <https://doi.org/10.1093/database/baw048>