Improving Pre-Trained Language Model for Relation Extraction Using Syntactic Information in Persian

Mohammad Mahdi Jafari*, Somayyeh Behmanesh*,
Alireza Talebpour†
Faculty of Computer Science and Engineering,
Shahid Beheshti University, Tehran, Iran
mohama.jafari@mail.sbu.ac.ir,
s_behmanesh@sbu.ac.ir, talebpour@sbu.ac.ir

Ali Nadian Ghomsheh
Cyberspace Research Institute,
Shahid Beheshti University, Tehran
a_nadian@sbu.ac.ir

Abstract

Relation classification is an essential task in NLP to identify relationships between entities. The state-of-the-art methods for relation classification are primarily based on deep learning and pre-trained BERT methods. This paper presents U-Bert and T-BERT methods and is submitted to the Second Workshop on NLP Solutions for Under Resourced Languages (NSURL2021) (Taghizadeh et al., 2021). In this paper, we focus on the optimal use of the syntactic features in pre-trained language models. First, we extract the syntactic properties and then feed them by a new embedding layer. This work achieved third place in NSURL-2021 task 1: Semantic Relation Extraction in Persian. Our results in this competition are 59.44 and 57.6 macro-average F1-score, respectively, in U-BERT and T-BERT evaluation.

1 Introduction

One of the main tasks in NLP is the relation classification which predicts semantic relation between two tagged entities in a sentence (Hendrickx et al., 2019). Various NLP applications such as information extraction, document summary, knowledge base population, and question answering use the relation classification.

According to the syntactic structures of sentences, using the information of Shortest Dependency Path (SDP) is a popular way in most solutions for relation classification in sentences (K. Xu et al., 2015; Y. Xu et al., 2015). However, the use of SDP increases the parsing time of the sentence exponentially as the sentence length increases (Lee et al., 2019). Using pre-trained language models such as BERT causes good results that have been reported for the relation classification without considering syntactic features directly (Wu and He, 2019; Wang and Yang, 2020). But syntactic information still plays an influential role in NLP applications (Kiperwasser and Ballesteros, 2018). Therefore, researchers have proposed solutions to effectively add the syntax tree to pre-trained transformers (Bai et al., 2021; Sundararaman et al., 2019).

This paper applies the pre-trained BERT model for relation classification and uses syntactic information in Embeddings Level. In the first method, called the U-BERT, two solutions have been considered to improve the algorithm's accuracy. The first solution is based on the inequality of the number of samples during training in different classes. By oversampling the samples into smaller classes, we covered the inequality. In the second solution, we used the Pairwise ranking loss function to reduce the effect of the "Other" class.

In the second method, called the T-BERT, we use sentence syntax features. The relation classification problem depends on the SDP in the dependency tree. Therefore we use a new embedding layer at the input of the BERT network, called Dependency Tree Embedding. Dependency Tree Embedding is obtained from Part-of-Speech (POS) Tag and Dependency Tree Tag in Persian. We use HAZM tools[1] in the Persian language to extract POS and Dependency Tree tags. Moreover, we apply the average Entity Words for

* Equal contribuation
† Corresponding author

[1] https://github.com/sobhe/hazm

classification. Our contributions in this paper are as follows: (1) we put forward an innovative approach to exploit syntax-level information for relation classification in the Persian dataset. (2) We apply syntactic information without degrading the model's pre-trained knowledge.

The remainder of this paper is organized as follows. Section 2 provides a summary of the related literature. In Section 3, we introduce the applied methodology, dataset, pre-processing, and model architecture. We Presented Experimental results and discussed them in Section 4. Finally, in section 5, we conclude our work and propose future careers.

2 Related Work

In recent years, a variety of methods proposed by researchers for relation classification. We could divide the Relation classification methods into non-neural-based models (Rink and Harabagiu, 2010) and neural-based models (Tai et al., 2015; Socher et al., 2012). Regarding the broad application of deep learning, many works use deep neural networks to perform the relation classification task. Applied neural and deep learning models include supervised (Socher et al., 2012; Zeng et al., 2014) and distant supervised (Min et al., 2013) based on the labeling of the dataset. Deep neural network categorized into two groups for the relation classification task, including the End-to-End model (Socher et al., 2012; Zeng et al., 2014) and SDP-based model (X et al., 2015; Socher et al., 2012; Liu et al., 2015; Y. Xu et al., 2015).

Among End-to-End- methods, R-BERT (Wu and He, 2019) and BERTEM-MTB (Soares et al., 2019) methods marked entities with special tokens. The tokens before and after each entity are different in the R-BERT and BERTEM-MTB methods. Furthermore, Wang and Yang (Wang and Yang, 2020) utilized BERT and attention-based Bi-LSTM (Att-Bi-LSTM).

Syntactic characteristics play a critical role in the relation identification in a sentence. The grammatical relations and structure of a sentence show a dependency tree (Culotta and Sorensen, 2004). When subjects and objects are long-distance, some neural network models suffer from irrelevant information. Xu et al. (K. Xu et al., 2015) proposed learning more robust relation representations based on the SDP through a convolution neural network. Some studies have attempted to incorporate syntactic information structures into their network architectures, such as Tree-LSTM (Tai et al., 2015) and Linguistically-informed self-attention (LISA) (Strubell et al., 2018).

The use of language models such as BERT (Devlin et al., 2018), RoBERTa (Joshi et al., 2020), and T5 (Raffel et al., 2019) has shown remarkable results in various language processing tasks. Tao et al. (Tao et al., 2019) showed that synthetic indicators, specific phrases, and words like propositions contained information to find semantic relationships. They use the BERT network to take advantage of both semantic and syntactic methods. Since the entity provides only a small amount of information for categorization, they used 'syntactic indicators.' Sundararaman et al. introduced Syntax-Infused Transformer and BERT models for Machine Translation and Natural Language Understanding (Sundararaman et al., 2019). As novel contributions, they fed in syntax information to modify pretrained $\text{BERT}_{\text{BASE}}$ embeddings, and the performance of $\text{BERT}_{\text{BASE + POS}}$ outperforms $\text{BERT}_{\text{BASE}}$ on many GLUE benchmark tasks was calculated.

Bai et al. (Bat et al., 2021) proposed a novel framework named Syntax-BERT for relation identification. Reported experiments based on Syntax-BERT verify the effectiveness of syntax trees and show better performance over multiple pre-trained models, including BERT, RoBERTa, and T5. Some studies (Hewitt and Manning, 2019; Jawahar et al., 2019) have shown that pre-trained transformers can implicitly learn certain syntactic information from sufficient examples. However, Bai et al. (Bai et al., 2021) showed that there was still a big gap between the syntactic structures which are implicitly learned and the syntactic trees created by human experts as a target point.

For Extracting the relation from the text in Persian, the non-neural network method has been utilized (Saheb-Nassagh et al., 2020; Rahat and Talebpour, 2018; Fadaei and Shamsfard, 2010). These works have used syntactic features. Fadaei and Shamsfard (Fadaei and Shamsfard, 2010) proposed a relation extraction system for the Persian language. They used raw texts and Wikipedia articles to learn conceptual relations. Saheb-Nassagh and et al. introduced RePersian as

a relation extraction method (Saheb-Nassagh et al., 2020). RePersian depends on POS tags of a sentence and particular relation patterns extracted from the analysis of sentence structures. Rahat and Talebpour (Rahat and Talebpour, 2018) proposed a novel OIE extractor named Parsa that encompasses tree-structured patterns. It applies an efficient matching technique for pattern trees and a function for extraction confidence measurement. Moreover, Asgari-Bidhendi et al. (Asgari-Bidhendi et al., 2021) address Persian relation extraction utilizing language-agnostic algorithms. It used six neural and non-neural models for relation extraction on the bilingual dataset. The non-neural model was set as the baseline, while one CNN-based model, two RNN-based models, and two deep learning models were fed by multilingual-BERT contextual word representations.

3 Methodology

Theoretically, models based on transformer architecture can derive semantic and syntactic features of the language. But, these models must be trained with sufficiently diverse and large datasets. Some works (X et al., 2015; Socher et al., 2012; Liu et al., 2015; Y. Xu et al., 2015) provide a superficial understanding of the syntactic features in natural language to solve explicit training on syntactic features. In the learning task for the relation classification, knowing the position and type of the verb, prepositions, and other terms in the sentence can help distinguish different classes. The hypothesis uses the sentence dependency tree, which paves the way for recognizing the relationship between sentence entities. It has been substantiated in several kinds of research, including (Bai et al., 2021).

To learn the syntactic properties of the language, first, we extracted the syntactic properties of each word in the sentence using the dependency tree. Then the words were broken into the sub-words by BERT-tokenizer, and we designed an additional layer to embed the syntactic information. This additional layer was trained with different learning rates to eliminate the model's shortcomings in learning syntactic information.

3.1 Dataset and Preprocessing

We used a Persian edition of the famous semeval 2010-task8 database, translated into Persian (Asgari-Bidhendi et al., 2021). In the first step of pre-processing the dataset, all records whose structure contradicted the valid structure (legal and non-empty tags) were discarded. Entities tags in each record were then removed to match the sentence structure with the standard language. The sentence was then converted to a dependency tree using the HAZM dependency parser. The label corresponding to the syntactic features of each word consists of POS tags and a grammatical role in the dependency tree. In addition, indicator signs are exploited for entities to localize them for the model.

The imbalance in the classes in the database made us use weighted sampling to help supply more samples in the smaller classes. First, the frequency of each class was added, then the probability of a sample in each class is the inverse ratio of class frequency/total frequency. Sample counts before and after filtering for each class are presented in Table 1.

Category	Before filtering (e1-e2)/(e2-e1)	After filtering (e1-e2)/(e2-e1)
Other	1410	1374
Component-Whole	470/ 471	454/ 449
Instrument-Agency	97/ 407	95/ 397
Member-Collection	78/ 612	75/ 601
Cause-Effect	344/ 659	333/ 637
Entity-Destination	844/ 1	827/ 1
Content-Container	374/ 166	364/ 161
Message-Topic	490/ 144	481/ 140
Product-Producer	323/ 394	314/ 384
Entity-Origin	844/ 148	553/ 138

Table 1: Distribution of samples in different classes before and after filtering samples in the wrong format

3.2 Model Architecture

In the U-Bert method, we use the BERT model for task relation classification. We considered two solutions to improve the accuracy of the algorithm. The first solution is based on the inequality of samples during training in different classes, and we applied oversampling the samples in smaller classes to cover the inequality. Our analysis showed that the “other” class is the noisiest. In the

second solution, we used the Pairwise ranking loss function to reduce the effect of the "other" class.

The main characteristic of the proposed T-BERT method is the use of sentence syntax features. Since the relation classification problem depends on the shortest dependency path problem in the dependency tree, this feature inspires the use of a new embedding layer at the input of the BERT network. In this step, the vector for each word is reinforced with Pos Tag and Dependency Tree Tag. We use available tools in the Persian language to extract Pos and Dependency Tree tags. In addition to the Bert network output, we apply the average Entity Words for classification.

To use the syntactic properties extracted in the previous section, we add a new layer to the embedding part of the BERT architecture. This layer is precisely like the other embedding layers in terms of quantification and initialization strategy ($E \sim N(0, 0.02)$), called dependency tree embedding (E^{DT}). Then we add this layer's output to other embeddings, including token embeddings (E^T), positional embeddings (E^P), and segmentation embeddings (E^S).

$$E = E^T + E^P + E^S + E^{DT} \quad (1)$$

The only difference between this layer and other embedding layers was the learning rate during the training phase. According to Figure 1, there are four different embeddings for each sub-word, the first three were trained in the pre-training phase, but the last was filled with random initialization. Complementary information on the number of tokens and the initialization probability distribution function is presented in Table 1. After passing the embedding of input tokens through the BERT network, their semantic display in the $x \in \mathrm{R}^{768}$ space would appear. They are marked as $X_0, X_1, X_2 \dots X_{ml}$ in Figure 1. The vector for each entity (E_1 and E_2) is converted to a $768\ d$ vector using the mean operation.

$$E_1 = mean([X_i\ for\ i \in\ entity1]) \quad (2)$$

$$E_2 = mean([X_i\ for\ i \in\ entity2]) \quad (3)$$

After longitudinal concatenation, these two vectors are projected to $19d$ space through a dense layer of neurons with bias. This layer is equipped with a dropout, and the probability is presented in Table 1: Distribution of samples in DIFFERENT classes BEFORE and after filtering samples in the wrong format

$$logits = (W[E_1; E_2] + b) \quad (4)$$

4 Experimental Results and Discussion

Two apparent challenges in classifying relationships are the high noise in the "Other" class and the imbalance between classes, making it difficult to distinguish between classes. Table 1 clearly shows the considerable difference between the number of samples in different classes. This study tries to improve class imbalance and noisy samples in the "Other" class by choosing the Loss function under the problem structure. Using Pairwise Ranking Loss would eliminate the error surface sensitivity to "Other" class noisy samples. We utilized dropout to prevent the network from overfitting.

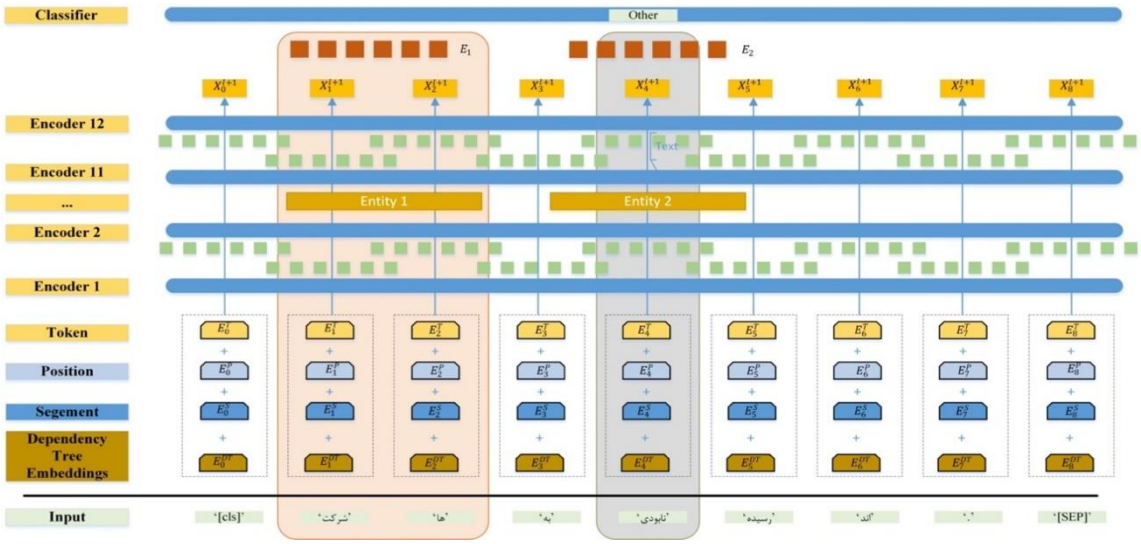

Figure 1: Model architecture determines the location of entities in the sequence by using the markers.

In addition, weighted sampling increases the chances of supplying samples related to smaller classes. Table 2 depicts the hyper-parameters related to the loss function and the weighting sampling method during the training phase. After filtering in the pre-processing phase, the number of training samples was equal to 7778, and the number of test samples was 2653. The maximum length in the training samples is 83, including special tokens. The batch size used in the training process was equal to 16. The learning rate was related to all network parameters except the embedding layer related to syntactic features equals $5e-5$. The cosine scheduler is used along with the learning rate decay mechanism with a coefficient of 1.1. Table 3 shows the results of the evaluation reported by competition for our models. Based on the obtained results, macro-average F1-score are 59.44 and 57.6 in U-BERT and T-BERT evaluation, respectively[2].

Parameters	Value
Dependency Tree Embedder Learning Rate	0.0001
Positive Margin	1.75
Loss Gomma	2
Negative Margin	0.25
Drop_out Ratio	0.45

Table 2: Hyperparameter details

	U-BERT	T-BERT
Cause-Effect	58.33	56.74
Content-Container	50.91	49.14
Entity-Destination	69.48	71.43
Entity-Origin	59.06	56.93
Instrument-Agency	66.92	59.93
Member-Collection	47.23	43.87
Message-Topic	65.93	60.95
Other	28.97	27.34
MACRO-averaged-F1	**59.44**	**57.6**

Table 3: THE MACRO-averaged-F1 of U-BERT and T-BERT methods on the test dataset.

To analyze the effect of adding syntactic information to U-BERT in Transformers models for the Persian language, we applied the combination of T-BERT and U-BERT. Table 4 shows the number of direction errors, precision, recall, and F1-score based on two methods for each class: the combination of T-BERT and U-BERT (top-row) and U-BERT (bottom row). The precision and recall are scorer script v1.2 of the semeval-task 8. Precision is calculated by tp/(tp+fp+direction error) and recall is obtained by tp/(tp+fn). Based on the obtained results, F1-score is 71.32 for the combination of T-BERT and U-BERT methods and 70.65 for the U-BERT method. It shows that by adding syntactic information to U-BERT, we achieve better results. The results show fewer direction errors for the combination of T-BERT and U-BERT methods. Therefore, this combination predicts a better relation direction in most classes than the U-BERT model. Furthermore, in two classes, Instrument-Agency and Product-Producer, the combination of T-BERT and U-BERT methods have the greatest improvement in relation detection.

Class name	# Direction errors	Precision	Recall	F1-Score
Component-Whole	45	56.76%	56.95%	56.85%
	45	56.55%	60.00%	58.22%
Instrument-Agency	2	73.45%	53.55%	61.94%
	4	64.84%	53.55%	58.66%
Member-Collection	7	72.96%	62.45%	67.29%
	6	71.43%	65.50%	68.34%
Cause-Effect	13	83.54%	83.02%	83.28%
	14	79.53%	83.95%	81.68%
Entity-Destination	1	84.05%	87.24%	85.62%
	1	83.56%	85.86%	84.69%
Content-Container	1	78.07%	78.07%	78.07%
	2	78.46%	81.82%	80.10%
Message-Topic	4	69.88%	71.83%	70.84%
	12	68.07%	76.98%	72.25%
Product-Producer	13	68.78%	62.39%	65.43%
	21	63.41%	57.52%	60.32%
Entity-Origin	3	76.52%	69.02%	72.58%
	3	74.79%	68.63%	71.57%
MACRO-averaged result (excluding "Other"):		73.78%	69.39%	71.32%
		71.18%	70.42%	70.65%

Table 4: Number of Direction errors, precision, recall, and F1-Score for two methods for each class: the combination of T-BERT and U-BERT (top-row) and U-BERT (bottom row).

5 Conclusion

This paper presented U-Bert and T-BERT's methods, submitted to the Second Workshop on

[2] Code is available at https://github.com/DeepKBQA/Pre-Trained-Language-Model-for-Relation-Extraction-Using-Syntactic-Information

NLP Solutions for Under Resourced Languages (NSURL2021). We emphasized the syntactic features in pre-trained language models. Based on the obtained results, macro-average F1-score are 59.44 and 57.6 in U-BERT and T-BERT evaluation, respectively. Furthermore, we proposed a new method by combining T-BERT and U-BERT to show the effect of adding syntactic information to U-BERT in Transformers models for the Persian language. The results depict better performance in F1-score in most analyzed classes.

References

Majid Asgari-Bidhendi, Mehrdad Nasser, Behrooz Janfada, and Behrouz Minaei-Bidgoli. 2021. "PERLEX: A Bilingual Persian-English Gold Dataset for Relation Extraction." Scientific Programming 2021.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees. arXiv preprint arXiv:2103.04350.

Aron Culotta, and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction." In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pp. 423-429.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Hakimeh Fadaei, and Mehrnoush Shamsfard. 2010. Extracting conceptual relations from Persian resources." In 2010 Seventh International Conference on Information Technology: New Generations, pp. 244-248. IEEE.

Iris Hendrickx, Kim S. Nam, Zornitsa Kozareva, Preslav Nakov, Diarmuid O. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. arXiv preprint arXiv:1911.10422.

John Hewitt, and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4129-4138.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In ACL 2019-57th Annual Meeting of the Association for Computational Linguistics.

Naman GJDM Joshi, Danqi Chen Omer Levy Mike Lewis, Luke Zettlemoyer Veselin Stoyanov Yinhan Liu, and Myle Ott. 2020. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In Submitted to International Conference on Learning Representations. https://openreview.net/forum.

Eliyahu Kiperwasser, and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. Transactions of the Association for Computational Linguistics 6: 225-240.

Joohong Lee, Seo Sangwoo, and Yong Suk Choi. 2019. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. Symmetry 11, no. 6: 785.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification." arXiv preprint arXiv:1507.04646.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 777-782. 2013.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR, abs/1910.10683.

Mahmoud Rahat, and Alireza Talebpour. 2018. Parsa: An open information extraction system for Persian. Digital Scholarship in the Humanities 33, no. 4: 874-893.

Bryan Rink, and Sanda Harabagiu. 2010. "Utd: Classifying semantic relations by combining lexical and semantic resources." In Proceedings of the 5th international workshop on semantic evaluation, pp. 256-259.

Raana Saheb-Nassagh, Majid Asgari, and Behrouz Minaei-Bidgoli. 2020. RePersian: An Efficient Open Information Extraction Tool in Persian. In 2020 6th International Conference on Web Research (ICWR), pp. 93-99. IEEE.

Livio B. Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. arXiv preprint arXiv:1906.03158.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. 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.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. 2019. Syntax-infused transformer and bert models for machine translation and natural language understanding. arXiv preprint arXiv:1911.06156.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. arXiv preprint arXiv:1804.08199.

Nasrin Taghizadeh, Ebrahimi Ali, and Faili Heshaam. 2021. NSURL-2021 task 1: Semantic Relation Extraction in Persian. In Proceedings of the Second International Workshop on NLP Solutions for Under Resourced Languages, NSURL ’21, Trento, Italy.

Kai S. Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.

Qiongxing Tao, Xiangfeng Luo, Hao Wang, and Richard Xu. 2019. Enhancing relation extraction using syntactic indicators and sentential contexts. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1574-1580. IEEE.

Shanchan Wu, Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In Proceedings of the 28th ACM international conference on information and knowledge management, pp. 2361-2364.

Zihan Wang, and Bo Yang. 2020. Attention-based Bidirectional Long Short-Term Memory Networks for Relation Classification Using Knowledge Distillation from BERT. In 2020 IEEE (DASC/PiCom/CBDCom/CyberSciTech), pp. 562-568..

Kun Xu, Feng Yansong, Songfang Huang, and Dongyan Zhao. 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. arXiv preprint arXiv:1506.07650.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. 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.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network." In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335-2344.