Structural block driven enhanced convolutional neural representation for relation extraction

Dongsheng Wang [a,1], Prayag Tiwari [b,*,1], Sahil Garg [c], Hongyin Zhu [d], Peter Bruza [e]

[a] *Department of Computer Science, University of Copenhagen, Copenhagen, Denmark*
[b] *Department of Information Engineering, University of Padova, Padova, Italy*
[c] *École de technologie supérieure, Montréal, QC H3C 1K3, Canada*
[d] *Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[e] *School of Information Systems, Queensland University of Technology, 2 George St, Brisbane City, QLD 4000, Australia*

ARTICLE INFO

ABSTRACT

In this paper, we propose a novel lightweight relation extraction approach of structural block driven convolutional neural learning. Specifically, we detect the essential sequential tokens associated with entities through dependency analysis, named as a structural block, and only encode the block on a block-wise and an inter-block-wise representation, utilizing multi-scale Convolutional Neural Networks (CNNs). This is to (1) eliminate the noisy from irrelevant part of a sentence; meanwhile (2) enhance the relevant block representation with both block-wise and inter-block-wise semantically enriched representation. Our method has the advantage of being independent of long sentence context since we only encode the sequential tokens within a block boundary. Experiments on two datasets i.e., SemEval2010 and KBP37, demonstrate the significant advantages of our method. In particular, we achieve the new state-of-the-art performance on the KBP37 dataset; and comparable performance with the state-of-the-art on the SemEval2010 dataset.

1. Introduction

Relation extraction (RE) is an essential task in natural language processing (NLP) to extract the relation between two entities for some given context. In the past years, numerous massive scale knowledge bases (KBs), including DBpedia [1],YAGO [2], and Freebase [3], have been constructed and being broadly utilized in many NLP tasks and applications, such as question answering and web search. These knowledge bases mainly consist of relational facts with some format, e.g., (*Google*, founder, *Larry Page*).

Though existing KBs contain a large number of facts, they are not close to encompassing the vast number of facts embedded in plain text. RE is a method of automatically extracting hidden relational facts from plain text to supplement the KBs.

The relation extraction (RE) tasks are considered into two steps, relation detection, and relation classification. Specifically, the first step is to detect candidate relation mentions in sentences involving pairs of entities, and the second is to classify these relation mentions into predefined categories [4–6]. In this paper, we focus on the latter task.

Non-relation instances can be managed as a normal relation class. On the other hand, RE comes with a vastly unbalanced dataset where the quantity of non-relation instances far surpasses relation instances, making this RE task more challenging but more pragmatic than relation classification.

In past decades, most of the work in RE has been dominated by two approaches which can be differentiated by the essence of the relation description: kernel-based approaches [7–14], and feature-based approaches [13,15–17]. There is a common goal in these approaches which is to leverage a substantial body of knowledge resources and linguistic analysis in order to map relation mentions into some rich representation. The purpose of the rich representation is that it can be utilized by some statistical classifiers, for example, maximum entropy [18,19] or support vector machines (SVM) [20,21]. The pipeline of linguistic analysis consists of several manually designed steps for example, tokenization, chunking, part of speech tagging, parsing and name tagging, which are often executed by the existing NLP module. Due to the knowledge founded by the NLP research community, these approaches enable the RE module to inherit knowledge from the pre-processing tasks. For example, the indicated tasks in the pipeline above are well known to be subject to significant levels of error when applied to out-of-domain dataset [22–24],

* Corresponding author.
E-mail addresses: wang@di.ku.dk (D. Wang), prayag.tiwari@dei.unipd.it (P. Tiwari), sahil.garg@ieee.org (S. Garg), zhuhongyin2014@ia.ac.cn (H. Zhu), p.bruza@qut.edu.au (P. Bruza).

[1] Dongsheng Wang and Prayag Tiwari contribute equally and share the co-first authorship.

triggering the RE module to collapse. Thus, our main aim is to propose a novel RE model which reduces the complexity of the feature engineering task, reduce the error propagation rate and improve the performance in the RE task.

In this paper, we propose a novel structural block-driven convolutional neural representation for RE. To be specific, we detect the essential spans associated with entities through dependency relation analysis, by obtaining the parent, siblings, and children nodes of entities. These are ranked into selective sequential tokens in the same order as they appear in the text. We enhance the selective sequential tokens by enriching them with semantic tags (semantic role and part-of-speech tags), all of which is encoded with multi-scale CNNs. Furthermore, we add two more inter-block representations with one subtract layer of the block representation subtracted by the two entity representations, and one multiply layer of the two entity representations. Then, we concatenate the block-wise and inter-block wise representations to infer the relation. As a result, the encoding of a selective part of a sentence and the enhanced encoding of the block leads to an improvement in both performance and efficiency. We achieved a new state-of-the-art performance in the KBP37 dataset with an F1 of 60.9, and a comparable result in the SemEval dataset with an F1 of 81.1.

2. Literature survey

Over recent years, many approaches have been proposed for relation extraction and classification. Most related work are based on applying NLP system or pattern matching to derive lexical attributes. In general, pattern matching is the base for traditional relation classification task [25,26] which can be categorized into kernel-based approaches [9,10] and feature-based approaches [16,27]. The preceding category depends on manually designed patterns and so its time consuming as well as requiring the need the input from experts. Consequently, data sparsity is a challenge facing the latter approaches. Furthermore, extra tools are required for these methods to derive linguistic features.

Distant supervision [28–32] came into much recognition since 2009 to address the challenge of pattern design, and also because of the scarcity of manually annotated data. This kind of approach integrates knowledge graphs and textual datasets, where the knowledge graph is utilized to automatically identify patterns from the textual dataset.

Our approach is inspired by neural models that learn features automatically, e.g., Collobert et al. [33]. Currently, deep learning is [34,35] very widely applied to learn the underlying feature automatically, so the remainder of the literature survey will cover such approaches. For example, Lin [36] proposed a sentence-level attention mechanism to alleviate the wrong labeling problem, expecting to reduce the weight of the instances of noise. They employ a CNNs encoder on multiple sentences with selective attention on expected correctly labeled sentences, by adding a learn-able weight to each sentence. Zhang et al. [37] proposed a model that in the first stage amalgamates the Long Short-Term Memory (LSTM) sequential approach with the type of entity position-aware attention and shows improvement in relation extraction. In the later stage, TACRED (106,264 instances), a large supervised relation extraction dataset was attained from crowdsourcing and focused towards TAC KBP relations. This combination of an effective model with high quality supervised data yields superior relation extraction performance. The proposed model outperforms all the neural-based baselines. Culotta et al. [38] proposed a probabilistic extraction model that yields mutual advantage to both "bottom-up" and "top-down" relation extraction. This work demonstrates that amalgamating the relation extraction with pattern discovery improves the performance of each task.

Zeng et al. [39–41] used a deep neural network to extract sentence and lexical level features. The proposed architecture takes the input (all the word tokens) without complex pre-processing. Primarily, all the word tokens are converted to vectors by using word embedding. Furthermore, lexical level based features are extracted according to the stated nouns. Meanwhile, the convolutional model is used to learn the sentence level features. The final extracted feature vector is formed by combining these two-level features. In the end, the obtained features are given to the softmax classifier to predict the association between two marked nouns. The obtained results show that the proposed model outperforms the baselines.

One interesting work by Nguyen et al. [42] used CNNs for relation extraction that learns features from the sentence automatically and hence reduces the dependencies on external resources and toolkits. The proposed architecture takes benefit of various window sizes for pre-trained word embedding and filter on a non-static architecture as an initializer in order to enhance performance. The relation extraction issues because of unbalanced data have been highlighted in this work. Results shown improvement, not only over baselines for relation extraction but also relation classification models. Liu et al. [43] explored how dependency information can be used. Firstly, the newly termed augmented dependency path (ADP) model is proposed, which comprises the shortest dependency path among the two subtrees and entities joined to the shortest path. In order to explore the semantic relation behind the ADP architecture, they proposed dependency-based neural networks: CNNs to apprehend the most essential features on the shortest path, and a recursive neural network (RNN) is constructed to model the subtrees.

Huang et al. [44] proposed an attention-based CNNs for the relation classification task. The proposed architecture makes full use of word embedding, position embedding and part-of-speech tag embedding information. which part of the sentence is essential and influential w.r.t the two entities of interest, which is determined by a word level attention approach. This model allows learning of some essential features from labeled data, thus removing the dependency on exterior knowledge, for example, the plain dependency structure. The model was tested on the SemEval-2010 Task 8 benchmark dataset and outperforms all the state-of-art neural network models. Furthermore, the model can obtain this performance with minimum feature engineering.

Zhang et al. [45] tried to address the lack of ability to learn temporal features in CNNs, and focussing mainly on long-distance dependencies among nominal pairs. They proposed a general architecture based on recurrent neural networks (RNN's) and contrasted it with CNNs-based approach. A new dataset was introduced which is the refined version of MIMLRE [28]. Experimental results on two datasets demonstrated that RNN's-based approach can enhance the performance of relation classification, and, in particular, is proficient at learning long distance relationships.

3. Technical background

We introduce the essential technical background of our model, including word embedding, CNNs, and semantic parsing. In particular, word embedding initializes the vectors of words; CNNs are the representation models that we employ to encode the text; and semantic parsing is the NLP approach we adopted.

3.1. Multi-scale CNNs

CNNs was originally proposed for computer vision but has subsequently been used for text classification [46] where it has

shown high performance [47,48] which is superior to traditional NLP based methods.

CNNs is first utilized in a sentence-level classification by Kim et al. [49] where they demonstrated improvement on NLP classification tasks.

Multi-scale CNNs have been demonstrated successful [50] where they employ multi-scale CNNs with different kernel sizes to overcome the drawback of the simple convolutional kernel with fixed window size over encoded semantics of documents, as shown in Fig. 1. The reason underlying the design is that determining a fixed window size using a simple convolutional kernel is demanding since small window normally requires deeper networks to gain critical information while large window sizes lead to loss of local information. Hence, multi-scale CNNs with multiple window sizes are used to represent the comprehensive contextual information of the text.

When we employ this structure, the last Dense and predict layer will be popped out, resulting in a CNN representation, which can be adopted to encode texts of various types, followed by some new neural encoding scheme.

3.2. Word embedding

Word embedding is a very prominent representation of document vocabulary. This allows capturing the context of some given word in a document, associations with other words, syntactic and semantic similarity, etc. Technically speaking, there is a mapping of words into vectors containing real number by utilizing the dimension reduction, probabilistic model, or neural networks on some word co-occurrence matrix. This is a kind of feature learning method and language model as well. Word embedding is a kind of process to execute the mapping utilizing neural networks.

Word embedding. Currently, there are several widely-used word embedding approaches for example, Glove (Stanford),[2] word2vec (Google),[3] and fastest (Facebook).[4] In this work, Glove word embedding is used. Glove[5] is type of unsupervised learning model for getting vector representation of words.

Tag Embedding. Besides, we employ one-hot encoding for POS and dependency tokens in our model, since they are not typical words or terms. Particularly, we encode a size of 24 dimensions embedding for POS labels (24 POS tags in total); and 41 dimensions embedding for dependency labels (41 dependency tags in total) in our model.

3.3. Semantic processing in NLP

3.3.1. Dependency parsing
Dependency parsing[6] is the way to investigate the framework of the sentence, forming an association between headwords and also those words which change the heads. The following Fig. 2 explains the dependency style investigation utilizing the traditional graphical approach which is accepted in the community of dependency parsing. Here it is important to note that the lack of nodes analogous to lexical categories or phrasal constituents in some dependency parse; the inner framework of the dependency parse includes simply the directed associations among lexical components in the sentences. Such associations directly encode hidden complex phrase structure parses to essential information. For example, arguments of a given verb *prefer* are directly connected in the dependency structure, although the relation to the main verb is far away in the phrase tree framework.

---

[2] https://nlp.stanford.edu/pubs/glove.pdf
[3] https://www.tensorflow.org/tutorials/representation/word2vec
[4] https://fasttext.cc/
[5] https://nlp.stanford.edu/projects/glove/
[6] https://nlp.stanford.edu/software/nndep.html
[3] https://web.stanford.edu/~jurafsky/slp3/13.pdf

3.3.2. POS Tags
POS[7] tags are very essential for constructing parse tree which are utilized in constructing named entities recognitions (NERs) and extract semantic connections among words. POS is also useful for constructing lemmatizers which minimize words to their root form. POS tagging is the way to highlight a word into the documents dataset to an associated POS tag that is based on some context in which then the word is utilized.

4. Proposed method

We propose a novel approach where the entity oriented structural block is detected, then the semantic representation for the block is encoded utilizing multi-scale CNNs, which is further enriched with inter-block representation. We first introduce the structural block detection in Section 4.1, followed by the enriched semantic encoding using CNNs in Section 4.2.

4.1. Structural block detection

We detect the block of coherent tokens for entities in a text. This design is to find directly relevant sequential tokens, whilst retaining their local integrity. First, we obtain the candidate entities, e.g., in the cases of SemEval and KBP37 dataset, they explicitly give two entities. Next, we detect their structurally related tokens to restore their coherent structural semantics. We achieve this by building up a dependency tree for each sentence with dependency relations between tokens and find the parent, siblings and children nodes as a single block for each entity e_i, which is defined as Eq. (1).

$$single\_block(e_i) = \sum_{t^j \in e_i} (t^j + head(t^j) + siblings(t^j) + children(t^j))$$

(1)

where t^j is a token from the entity with its rank position j in text; $head(t^j)$ basically refers to the relation name (mostly verbs) while $siblings(t^j)$ refers to those tokens that share the same relation name; and the $children(t^j)$ indicates those tokens that depend on t^j. A single block literally covers all the structurally related tokens in terms of a single entity. Nevertheless, we provide an alternative version without including children, i.e., $single\_block(e_i) = \sum_{t^j \in e_i} (t^j + head(t^j) + siblings(t^j))$. This version empirically leads to comparable or better performance than the first version, the comparison of which is detailed in Section 5.4.

The selected tokens for a set of entity E is defined as Eq. (2) where all single blocks are aggregated into one block defined as $aggreg\_block(E)$. All the token indexes that are selected are denoted as $J = set(\sum_{t^j \in aggreg\_block(E)} j)$, indicating that duplicate j will be removed.

$$aggreg\_block(E) = \sum_{e_i \in E} single\_block(e_i)$$

(2)

Then, the sequential tokens for the block are defined in Eq. (3).

$$seq\_tokens(E) = \bigoplus_{j \in ranked(J)}^{|J|} t^j$$

(3)

As the block is locally integrated instead of being integrated through the whole sentence, we additionally concatenate its semantic role $role(t)$ and the part-of-speech $pos(t)$ to assist the generalized learning on the selective sequential tokens. The resulting

---

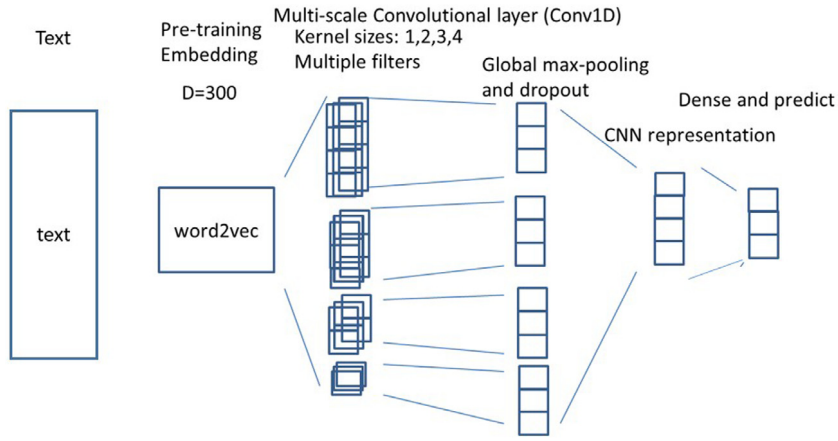[7] https://nlp.stanford.edu/software/tagger.shtml
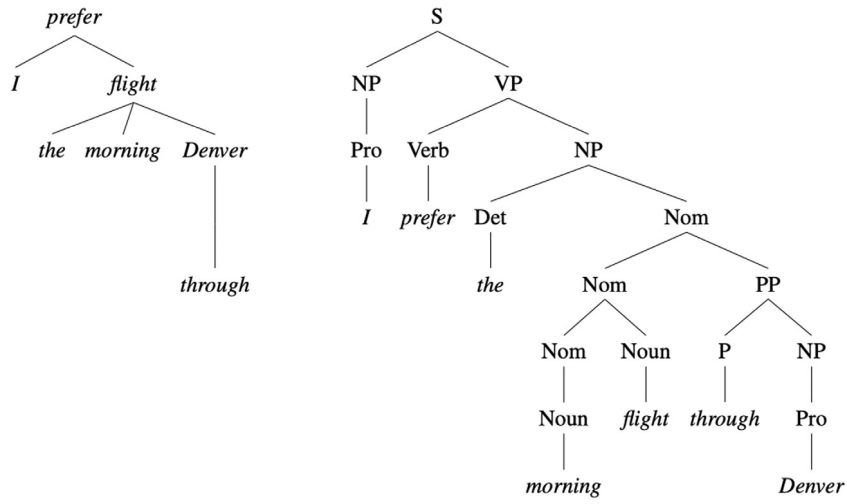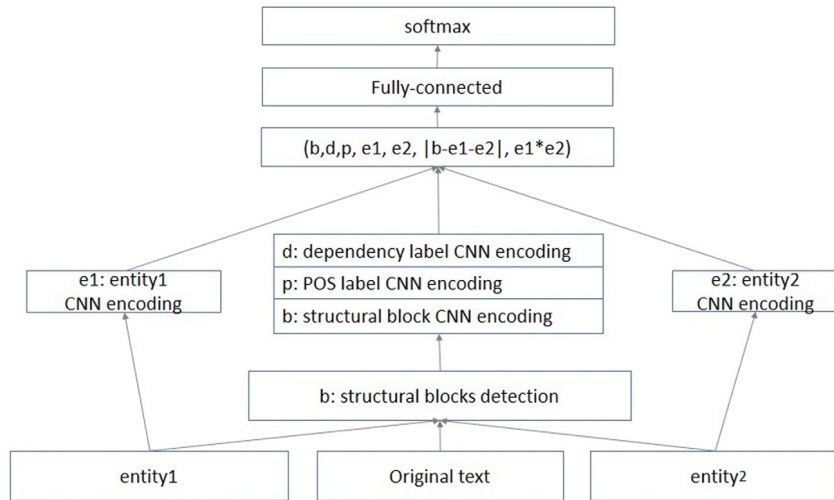
Fig. 1. Multi-scale CNNs [50].

Fig. 2. A dependency-style parse alongside the associated constituent-based investigation for *I prefer the morning flight through Denver* [7].

Fig. 3. Our prediction scheme.

structural block for a sentence is defined as Eq. (4),

$$structural\_block(E) = \sum_{t \in seq\_tokens(E)} (t \oplus role(t) \oplus pos(t)) \qquad (4)$$

where $structural\_block(E)$ is the enhanced representation for the selective structural block.

As a result, the relation predicting is defined with softmax function of the probability distribution as below,

$$p(r|\theta, s) = softmax(M(h \odot s) + b) \qquad (5)$$

Table 1

Dataset statistics.

Dataset	Train (instances)	Test (instances)
SemEval2010	8000	2717
KBP37	15,917	3405

where $softmax(x) = \frac{Exp(x)}{\sum_{k=1}^{K} Exp(x_k)}$ (K is the number of the relation); M is the matrix representation of the relation, s is the enhanced structural block representation; h is the hidden layer; and b is a bias vector in terms of the output.

4.2. Modeling

Inspired by [51], we propose an adapted model scheme as shown in Fig. 3. The design is to take full advantage of the selective structural blocks with both block and inter-block representation.

For a given sentence and two entities, we first detect the structural block with dependency analysis, resulting in a subset of sequential tokens. We then enrich the semantics by concatenating their semantic role tags and POS tags as demonstrated in Eq. (4). As a result, every single token of the block is represented with three tokens, working together to represent the whole structural block. The semantic role tokens are supposed to give rise to the generalized learning on the selective sequential tokens.

Then, we encode these sequential tokens with multi-scale CNNs, as shown in Fig. 1. Specifically, we pop out the softmax layer of the model, resulting in the previous CNN representation layer, to encode the sequential tokens of the entities, dependency labels, POS labels, and structural block.

Besides, we explicitly encode the inter-block representation to gain the connection among the block and two entities. The inter-block representation includes a subtract layer between the block and the two entities, expressed as $b - e1 - e2$; and a multiply layer between two entities expressed as $e1 * e2$, assisting the similarity inference between the two entities (as $sim(e1, e2) = \frac{e1 * e2}{|e1|.|e2|}$).

As a result, we fully connect the structural block representation and the inter-block representation and softmax the final relation classification.

Table 3

Hardware settings.

Property	modes	CPU	Memory	System	Threads per core
Value	64 bits	40	125G	Ubuntu 14.04	2

5. Experimental results

5.1. Data

Semantic Evaluation[8] dataset has 10 relations, as listed below. However, because they have order difference, therefore, there is a total of 19 relations, as the "Other" does not apply this rule.

The second KBP37 dataset[9] is the revised version of MIML-RE annotation dataset, which was provided by Gabor Angeli et al. [28]. They utilized both the 2013 and 2010 KBP document dataset, as well as a July 2013 dump of Wikipedia as the text dataset for further annotation. There are 33811 annotated sentences. More details about this dataset are presented in Tables 1 and 2.

5.2. Hardware setting

We list the hardware settings used to conduct the experiments. It is important to note that we employ ordinary CPU server settings instead of GPUs. Table 3 indicates that our model scheme overall has low setting requirements comparable to most existing servers. We will demonstrate that even on such basic servers the training speed is both fast and efficient.

5.3. Evaluation measures

There are several evaluation measures in machine learning for example, accuracy, precision, recall, f-score, etc. In line with most related work, we employ the F-score to evaluate the performance of our model and use it to compare the performance with other state-of-art neural models.

$$Precision(P) = \frac{Number\ of\ correctly\ extracted\ entity\ relations}{Total\ number\ of\ extracted\ entity\ relations} \quad (6)$$

$$Recall(R) = \frac{Number\ of\ correctly\ extracted\ entity\ relations}{Actual\ number\ of\ extracted\ entity\ relations} \quad (7)$$

8 https://www.cs.york.ac.uk/semeval2010_WSI/datasets.html

9 https://github.com/davidsbatista/Annotated-Semantic-Relationships-Datasets/issues/3

Table 2

Details about SemEval2010 and KBP37 datasets.

|			
SemEval2010	No. of relation types:	19	
	No. of relation classes:	10	
	Classes:	Cause–Effect	Instrument-Agency
		Product–Producer	Content–Container
		Entity-Origin	Entity-Destination
		Component–Whole	Member-Collection
		Communication-Topic	Other
KBP37	No. of relation types:	37	
	No. of relation classes:	19	
	Classes:	per:alternate_names	org:alternate_names
		per:origin	org:subsidiaries
		per:spouse	org:top_members/employees
		per:title	org:founded
		per:employee|_of	org:founded_by
		per:countries_of_residence	org:countries_of_headquarters
		per:stateorprovince_of_residence	org:stateorprovince_of_headquarters
		per:country_of_birth	org:member
		no_relation	
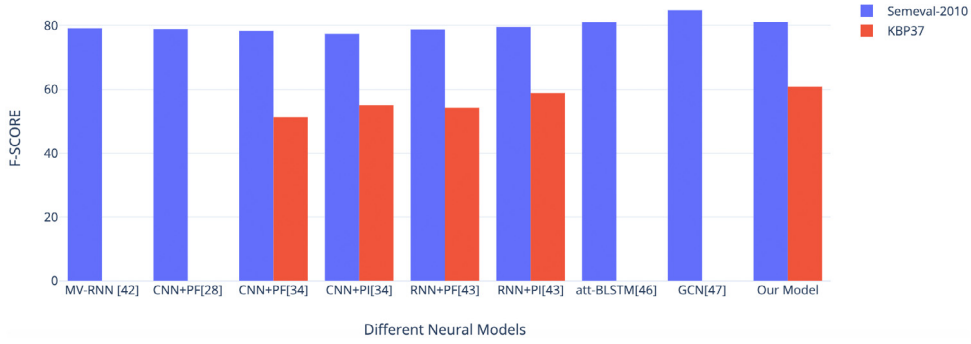
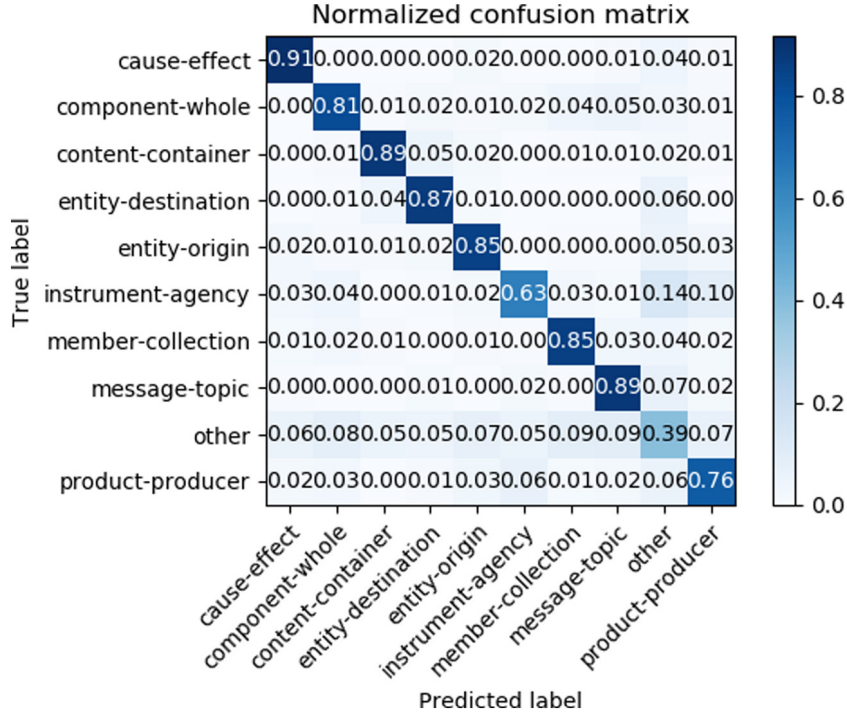**Fig. 4.** F1-score on Semeval-2010 and KBP37.

Fig. 5. SemEval2010 confusion matrix.

$$F1 = \frac{2P * R}{P + R} \qquad (8)$$

5.4. Results and discussion

As shown in Table 5 and Fig. 4, our model has superior performance on the KBP37 dataset and comparable performance on the SemEval2010 dataset. The main advantage of our method is that we do not rely on manually determined features while all the other methods adopt a feature set to some degree. As we achieve the new state-of-art performance in KBP37 (with an improvement of 2.1%), we found that text context in KBP37 is much longer than in SemEval2010. Consequently, it seems that our method has a superior advantage in long contexts, since the selective structural block strongly reduces the higher levels of noise associated with long contexts. The confusion matrix is shown in Fig. 6, where the mappings of label is listed at footnote.[10]

In SemEval2010, we found that the Instrument-Agency has the lowest F1 (below 70%), which is the bottleneck for the overall results. The reason behind this category can be that it has broader coverage of instances for the two types, namely, Instrument and Agency; while as we can observe that Cause-Effect, Member-Collection, etc. have more specific patterns, or smaller coverage of instances. There are some state-of-art models [52,53] whose F-score is slightly higher than our model, but our model take less time to compute than others. Further, our model rely on manual determined features while all the other methods adopt some level of manually selected features. The confusion matrix is shown in Fig. 5.

Our method exhibits satisfactory training speed as well. The average epoch training time is 4.6 s on SemEval2010 and 6.3 s on KBP37 datasets, due to CNNs and the structural block detection. Though we are not provided with the training time from

---

[10] no_relation = 0; org_alternate_names = 1; org_city_of_headquarters = 2; org_country_of_headquarters = 3; org_founded = 4; org_founded_by =

5; org_members = 6; org_stateorprovince_of_headquarters = 7; org_subsidiaries = 8; org_top_members = 9; per_alternate_names = 10; per_cities_of_residence = 11; per_countries_of_residence = 12; per_country_of_birth = 13; per_employee_of = 14; per_origin = 15; per_spouse = 16; per_stateorprovinces_of_residence = 17; per_title = 18
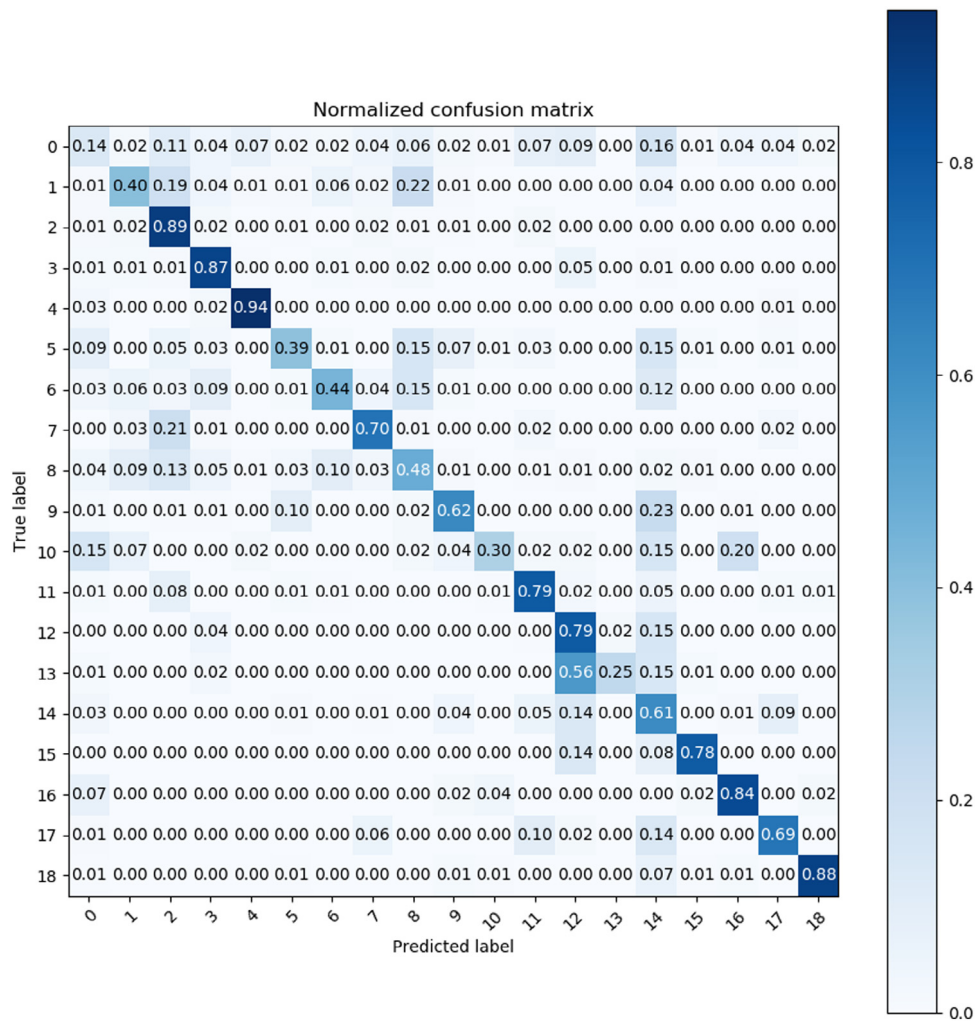
Fig. 6. KBP37 confusion matrix, label mappings is shown at the bottom of the result Section 5.4.

Table 4
Comparison of two versions of obtaining blocks.

Our model version	SemEval2010	KBP37
	Macro F1 (%)	
Block with-children	80.7	60.9
Block without-children	81.1	60.7

Table 5
Results of different neural models compared to our proposed models.

Model	Semeval-2010	KBP37
	Macro F1 (%)	
MV-RNN [54]	79.1	–
CNN+PF [39]	78.9	–
CNN+PF [45]	78.3	51.3
CNN+PI [45]	77.4	55.1
RNN+PF [55]	78.8	54.3
RNN+PI [55]	79.6	58.8
att-BLSTM [52]	84.0	–
GCN [53]	84.8	–
Our Model	**81.1**	**60.9**

other published models for comparison, we claim the satisfactory efficiency of our method.

As discussed in Eq. (1), we empirically found that the adoption of children nodes when detecting block does not necessarily influence the final performance. As shown in Table 4, the performance of with-children version is slightly lower than that of without-children; while it is the opposite for KBP37 dataset. Therefore, we maintain that the single block detection can include or exclude the children nodes.

A single model shows an F1 between 78% to 79% in SemEval2010 dataset, but improves to 81.1% when we ensemble the results from 2–4 models that are trained from the same scheme. Given the proposed scheme is independent of the fundamental encoding model, we therefore assume that different representation models, including LSTM, BiLSTM, or a mixture modeling can replace our multi-CNNs; and an ensemble of models with different fundamental representations can potentially improve the ultimate performance.

6. Conclusion and future work

Relation extraction plays an important role in the population of KBs and other NLP tasks. In this paper, we presented a novel relation extraction approach with few features called the structural block-driven convolutional neural model. The design of the model is to (1) eliminate the noise due to irrelevant parts of a sentence and (2) enhance the relevant block representation, by adopting semantic role embedding and concatenating the inter-block representation. The block is detected by entity oriented dependency analysis, and the enhanced encoding of the block is conducted not only on block-wise representation but with two

more layers, one subtract and one multiply layer, based on strong multi-scale convolutional neural models.

We validated our model on two datasets, i.e., SemEval2010 and KBP37, where we achieve the new state-of-the-art performance on the KBP37 dataset (60.9 F1) and comparable performance with the state-of-the-art on the SemEval2010 dataset (81.1 F1).

Our method has a superior advantage in long sentence contexts since our model only encodes sequential tokens within the block boundary. For example, the performance in KBP37 demonstrated the model's robustness in long contexts. Moreover, compared to most of the other relation extraction approaches, we do not rely on a manually constructed feature set. Finally, the model has a satisfactory training speed, e.g., 4.6 s epoch training time in SemEval2010 and 6.3 s in KBP37.

As the scheme we proposed is extensible, one avenue for future work is to apply the scheme based on other basic encoders, instead of CNNs. For instance, BERT [56] has proved to be highly efficient in multiple NLP tasks. We can replace the CNNs with BERT as an enhanced BERT representation to observe whether it can generate further improvement.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105913.

Acknowledgment

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: The Semantic Web, Springer, 2007, pp. 722–735.

[2] G. Kasneci, F. Suchanek, G. Weikum, Yago-a Core of Semantic Knowledge, Max-Planck-Institut für Informatik, 2006.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, AcM, 2008, pp. 1247–1250.

[4] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.

[5] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10, Association for Computational Linguistics, 2002, pp. 79–86.

[6] S.B. Kotsiantis, I. Zaharakis, P. Pintelas, Supervised machine learning: A review of classification techniques, Emerg. Artif. Intell. Appl. Comput. Eng. 160 (2007) 3–24.

[7] L. Qian, G. Zhou, F. Kong, Q. Zhu, P. Qian, Exploiting constituent dependencies for tree kernel-based semantic relation extraction, in: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 2008, pp. 697–704.

[8] D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, J. Mach. Learn. Res. 3 (Feb) (2003) 1083–1106.

[9] R.J. Mooney, R.C. Bunescu, Subsequence kernels for relation extraction, in: Advances in Neural Information Processing Systems, 2006, pp. 171–178.

[10] R.C. Bunescu, R.J. Mooney, A shortest path dependency kernel for relation extraction, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2005, pp. 724–731.

[11] M. Zhang, J. Zhang, J. Su, G. Zhou, A composite kernel to extract relations between entities with both flat and structured features, in: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2006, pp. 825–832.

[12] A. Culotta, J. Sorensen, Dependency tree kernels for relation extraction, in: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2004, p. 423.

[13] T.H. Nguyen, R. Grishman, Employing word representations and regularization for domain adaptation of relation extraction, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 2, 2014, pp. 68–74, Short Papers.

[14] C. Liu, W. Sun, W. Chao, W. Che, Convolution neural network for relation extraction, in: International Conference on Advanced Data Mining and Applications, Springer, 2013, pp. 231–242.

[15] J. Jiang, C. Zhai, A systematic exploration of the feature space for relation extraction, in: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, 2007, pp. 113–120.

[16] N. Kambhatla, Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction, in: Proceedings of the ACL Interactive Poster and Demonstration Sessions, 2004, pp. 178–181.

[17] Y.S. Chan, D. Roth, Exploiting background knowledge for relation extraction, in: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics, 2010, pp. 152–160.

[18] K. Nigam, J. Lafferty, A. McCallum, Using maximum entropy for text classification, in: IJCAI-99 Workshop on Machine Learning for Information Filtering, vol. 1, (1) 1999, pp. 61–67.

[19] M. Osborne, Using maximum entropy for sentence extraction, in: Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4, Association for Computational Linguistics, 2002, pp. 1–8.

[20] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.

[21] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond, MIT press, 2001.

[22] D. McClosky, E. Charniak, M. Johnson, Automatic domain adaptation for parsing, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 28–36.

[23] D. Jurafsky, E. Gaussier, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, in: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 2006.

[24] B. Sun, J. Feng, K. Saenko, Return of frustratingly easy domain adaptation, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[25] P. Tiwari, M. Melucci, Towards a quantum-inspired binary classifier, IEEE Access 7 (2019) 42354–42372.

[26] P. Tiwari, M. Melucci, Towards a quantum-inspired framework for binary classification, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, 2018, pp. 1815–1818.

[27] F.M. Suchanek, G. Ifrim, G. Weikum, Combining linguistic and statistical analysis to extract relations from web documents, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 712–717.

[28] G. Angeli, J. Tibshirani, J. Wu, C.D. Manning, Combining distant and partial supervision for relation extraction, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1556–1567.

[29] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, Association for Computational Linguistics, 2009, pp. 1003–1011.

[30] M. Surdeanu, J. Tibshirani, R. Nallapati, C.D. Manning, Multi-instance multi-label learning for relation extraction, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, 2012, pp. 455–465.

[31] J. Li, R. Hu, X. Liu, P. Tiwari, H.M. Pandey, W. Chen, B. Wang, Y. Jin, K. Yang, A distant supervision method based on paradigmatic relations for learning word embeddings, Neural Comput. Appl., 1–10.

[32] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2010, pp. 148–163.

[33] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (Aug) (2011) 2493–2537.

[34] S. Garg, K. Kaur, N. Kumar, J.J. Rodrigues, Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective, IEEE Trans. Multimed. 21 (3) (2019) 566–578.

[35] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A.Y. Zomaya, R. Ranjan, A hybrid deep learning-based model for anomaly detection in cloud datacenter networks, IEEE Trans. Netw. Serv. Manag. 16 (3) (2019) 924–935, http://dx.doi.org/10.1109/TNSM.2019.2927886.

[36] Y. Lin, S. Shen, Z. Liu, H. Luan, M. Sun, Neural relation extraction with selective attention over instances, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2016, pp. 2124–2133.

[37] Y. Zhang, V. Zhong, D. Chen, G. Angeli, C.D. Manning, Position-aware attention and supervised data improve slot filling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 35–45.

[38] A. Culotta, A. McCallum, J. Betz, Integrating probabilistic extraction models and data mining to discover relations and patterns in text, in: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Association for Computational Linguistics, 2006, pp. 296–303.

[39] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, et al., Relation Classification Via Convolutional Deep Neural Network, 2014.

[40] D. Zeng, K. Liu, Y. Chen, J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1753–1762.

[41] G. Ji, K. Liu, S. He, J. Zhao, Distant supervision for relation extraction with sentence-level attention and entity descriptions, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[42] T.H. Nguyen, R. Grishman, Relation extraction: Perspective from convolutional neural networks, in: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, 2015, pp. 39–48.

[43] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, H. Wang, A dependency-based neural network for relation classification, 2015, arXiv preprint arXiv:1507.04646.

[44] X. Huang, et al., Attention-based convolutional neural network for semantic relation extraction, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2526–2536.

[45] D. Zhang, D. Wang, Relation classification via recurrent neural network, 2015, arXiv preprint arXiv:1508.01006.

[46] A. Jacovi, O.S. Shalom, Y. Goldberg, Understanding convolutional neural networks for text classification, 2018, arXiv preprint arXiv:1809.08037.

[47] S. Bai, J. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, arXiv preprint arXiv: 1803.01271.

[48] R. Johnson, T. Zhang, Effective use of word order for text categorization with convolutional neural networks, 2014, arXiv preprint arXiv:1412.1058.

[49] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, a Meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1746–1751, http://aclweb.org/anthology/D/D14/D14-1181.pdf.

[50] D. Wang, J.G. Simonsen, B. Larsen, C. Lioma, The Copenhagen Team Participation in the Factuality Task of the Competition of Automatic Identification and Verification of Claims in Political Debates of the CLEF-2018 Fact Checking Lab., in: CLEF (Working Notes), 2018.

[51] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, 2017, arXiv preprint arXiv:1705.02364.

[52] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 2, 2016, pp. 207–212, Short Papers.

[53] Y. Zhang, P. Qi, C.D. Manning, Graph convolution over pruned dependency trees improves relation extraction, 2018, arXiv preprint arXiv:1809.10185.

[54] R. Socher, B. Huval, C.D. Manning, A.Y. Ng, 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, Association for Computational Linguistics, 2012, pp. 1201–1211.

[55] D. Zhang, D. Wang, Relation classification: Cnn or rnn? in: Natural Language Understanding and Intelligent Applications, Springer, 2016, pp. 665–675.

[56] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.