An Enhanced Deep Neural Network-Based Architecture for Joint Extraction of Entity Mentions and Relations

Elham Parsaeimehr, Mehdi Fartash, and Javad Akbari Torkestani

Department of Computer Engineering, Islamic Azad University - Arak Branch, Arak, Iran

Ijfis

Abstract

Named entity recognition and relation extraction are two principal tasks in most natural language processing systems. The majority of methods used in the field implement these two issues independently, thus leading to possible problems such as error propagation from one component (entity detection) to another (relation extraction). To solve such problems, we propose a new architecture for joint identification of entity mentions and their relation by employing a deep neural network framework. The model not only overcomes the error propagation challenge but also improves the detection results of both tasks owing to the cooperation with each other. Experiments on publicly available sources demonstrate that our joint model surpasses competitors in terms of accuracy. The results highlight the improvement achieved by the proposed deep neural network framework for the entity mention and relation classification tasks. Furthermore, we tested the effect of increasing the sentence length and demonstrated its negative influence on the performance.

Keywords: Entity classification, Relation extraction, Joint model, LSTM-RNN, CNN, NLP

1. Introduction

With the growing amount of stored data on the internet, substantial efforts have been devoted to the field of *information extraction*, which relates to the extraction of knowledge from such data warehouses. In this field also, two processes, named entity recognition (NER) and relation extraction (RE), have attracted more attention in text mining researches. These two processes are mainly employed for natural language processing (NLP) applications such as knowledge base construction (KBC) and question answering (QA) systems in various domains and languages [1].

Traditionally, NER and RE work together in a pipelined model. However, this strategy has some major disadvantages, such as: (i) errors propagate from one process (NER) to the other (RE), (ii) the helpful information acquired from one process cannot be applied to another (e.g., recognition of *born in* relation for two entity mentions may be conducive for NER process to extract the two entities types: *person*, *countries of birth*, and vice versa). Alternatively, a few researchers have introduced models where these two processes jointly contribute by interacting and sharing their parameters. Such joint models can overcome the aforementioned problems and consequently achieve a high performance.

In this paper, we propose a novel architecture for entity and relation classification using a

Received: Feb. 7, 2020
 Revised : Feb. 29, 2020
 Accepted: Mar. 5, 2020

Correspondence to: Mehdi Fartash
 (m-fartash@iau-arak.ac.ir)
 ©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

deep neural network framework. Our architecture incorporates the joint modeling of the NER and RE components in a single model by utilizing two different structures (sequential and tree) of LSTM-RNN (Long short term memory-Recurrent Neural Network) and CNN (Convolution Neural Network). More precisely, the novelty of our architecture is the contribution of two different neural network models (CNN and LSTM) in the NER component. This model enables the simultaneous classification of the entity mentions and their relation type with higher accuracy. In this neural network-based framework, the parameters are renewed via a classification feedback obtained from both the processes. More accurately, the model employs bidirectional LSTM-RNN (Bi-LSTM-RNN) and CNN models to automatically learn the features of an input sentence and then detect the relation between an entity pair by using a bidirectional tree-structured LSTM. Experiments on the ACE05 and KBP37 datasets prove that our architecture surpasses competitor models in performance.

2. Related Works

Multiple studies have been performed in recent years on entity pair detection and relation extraction. Based on their representative strategies, the studies can be divided into two main groups: where the NER and RE processes perform in a pipeline, and where the processes collaborate jointly.

The following subsections discuss previous works for the both tasks of NER and RE as well as their joint models.

2.1 Named Entity Recognition

In recent decade, several efforts have been devoted to study various methods for entity identification in unstructured documents across different fields. Most traditional methods, such as the maximum margin Markov network [2], conditional random field (CRF) [3], and support vector machine [4] need extreme computations for engineering and extracting the features. Recently, some architectures with RNN and CNN approaches have been partnered with the CRF algorithm for entity detection tasks [5, 6]. Such methods that do not employ manual feature extraction techniques have reported a noticeably good performance.

2.2 Relation Extraction

Relation extraction is another significant process in NLP that has attracted significant attention. Two approaches have been

used to perform this task: manual feature-based approach and deep neural network approach. With regard to the first approach, machine learning supervised algorithms such as kernel methods have been employed for detection [7]. These type of algorithms use the syntactic and semantic relations of the parse tree, in addition to the lexical features, to capture the structure of the input sentence [8]. To alleviate the dependence on feature engineering, deep neural network-based models have been developed. Researchers have also been able to reach a better semantic understanding of the natural language with the automatic extraction of lexical and sentence-level features by employing CNN-based [9] and RNN models [10].

2.3 Associative Identification of Entity Type and Relation Class

Most recent researches on associative (joint) identification of entity mentions and the relation types have been feature based, such as the probabilistic graphical model [11], card-pyramid parsing model [12], and history-based structured model [13]. These systems depend on the extraction of handcrafted features by NLP methods (e.g., POS), although manual engineering of features is a time-consuming process and using NLP tools may lead to an increase in computation and error propagation.

In [14–16], a deep neural network framework has been introduced that employs RNNs and CNNs to manage the issue in a joint structure. Specifically, Miwa and Bansal [14] employed bidirectional tree-structured LSTM-RNNs to formulate the dependency structure between a word pair. The researchers in [16] utilized the idea given in [14] for extraction of biomedical entities (drug and disease) and relations, reporting noticeable accuracy. The authors in [17] proposed a table-filling multi-tasking RNN that can model several relation instances. The study in [18] presented an attention-based LSTM that identifies the semantic relations between entity pairs without using a dependency tree.

Similar to Miwa and Bansal [14], we use a bidirectional tree LSTM for describing the dependency structure between the target entity pairs.

3. Proposed Architecture

In this section we describe our end-to-end (joint) architecture (Figure 1), which enables the simultaneous detection of the entity type and the relation class. Our model comprises four components: embedding layer, Bi-LSTM layer, entity type detection module, and relation extraction module. In the following

Figure 2. The mixture CNN (Mix-CNN) module.

on $j * 2$ words surrounding the target word *Minghella* (i.e., j words both before and after *Minghella*). $WE1_j^i$ represents the i -th filter of $CNN_{\pm j}^+$ in the Mix-CNN for extracting the entity $e1$ and $WE2_j^i$ indicates the i -th filter of $CNN_{\pm j}^-$ for entity $e2$. The feature extracted by $WE1_j^i$ for entity $e1$ is represented as Ze_j^i . Thus, the j -th contextual information of entity $e1$ is demonstrated as $E1_j = [Ze_j^1, Ze_j^2, \dots, Ze_j^{ne}]$, in which where ne represents the number of filters in the Mix-CNN. Considering the differences in the dependencies of various entities on the textual words, a max-pooling process is applied to combine the features obtained with $CNN_{\pm j}^+$ ($1, 2, \dots, j$), as shown in Eq. (3), to improve them for better use in the subsequent computations.

$$E1_s = \begin{pmatrix} \max(Ze_1^1, Ze_2^1, \dots, Ze_j^1) \\ \dots & \dots & \dots \\ \max(Ze_1^n, Ze_2^n, \dots, Ze_j^n) \end{pmatrix}. \quad (3)$$

Output phase. After obtaining the sentential context information from the Bi-LSTM layer and the semantic features of the two entities from the Mix-CNN, the collected information is concatenated to obtain $f = [h_t, E1_s, E2_s]$. In this phase, a softmax classifier with dropout is utilized, as shown in Eq. (4).

$$y = W_R \cdot (f \odot r) + b, \quad (4a)$$

$$p_i = \frac{\exp(y_i)}{\sum_{i=1}^m \exp(y_i)}. \quad (4b)$$

Here, $W_R \in R^{m \times (2 \times l + 2 \times ne)}$, is the weight vector between the concatenated vector of f and the layer of labels; m is the total number of entity type classes and $r \in R^{(2 \times l + 2 \times ne)}$ is a binary mask vector drawn from Bernoulli with the probability ρ . Dropout prevents overfitting and leads to a more robust model. In Eq. (4b), p_i represents the detection probability of entity

class i .

3.4 Relation Classification Module

This module recognizes a relation between the detected entities from the dependency tree. The *shortest path* is employed in this NLP process because it contains the summary of the significant words that represent the relation between two entities. For example, as Figure 2 depicts, the *shortest path* for the two entities *Minghella* and *England* contains the key phrase *born in*. We utilize the bidirectional tree LSTM [14] to extract a relation type by modeling the dependency structure of the entity pair and the words surrounding them in the input sentence. As explained in [14], this bidirectional structure propagates the information to each node from the leaves upward and also from the root downward. This bidirectional tree-structured LSTM shares the weight vectors of the children with similar type and furthermore permits variable number of children.

Bi-LSTM. At time-step t , the vectors in the LSTM unit with $C(t)$ children are calculated according to Eq. (5):

$$i_t = \sigma \left(W^{(i)} x_t^d + \sum_{l \in C(t)} U_{m(l)}^{(i)} h_{tl} + b^{(i)} \right), \quad (5a)$$

$$f_{tk} = \sigma \left(W^{(f)} x_t^d + \sum_{l \in C(t)} U_{m(k)m(l)}^{(f)} h_{tl} + b^{(f)} \right), \quad (5b)$$

$$o_t = \sigma \left(W^{(o)} x_t^d + \sum_{l \in C(t)} U_{m(l)}^{(o)} h_{tl} + b^{(o)} \right), \quad (5c)$$

$$u_t = \tanh \left(W^{(u)} x_t^d + \sum_{l \in C(t)} U_{m(l)}^{(u)} h_{tl} + b^{(u)} \right), \quad (5d)$$

$$c_t = i_t \odot u_t + \sum_{l \in C(t)} f_{tl} \odot c_{tl}, \quad (5e)$$

$$h_t = o_t \odot \tanh(c_t). \quad (5f)$$

Here, $m(\cdot)$ is a type mapping function. The LSTM unit in the tree structure receives $x_t^d = [h_t; e_t; x_t]$ as input, that is, the concatenation of the hidden state vectors h_t in the Bi-LSTM layer, the predicted entity label (e_t), and the embedded form of the word (x_t) as input.

Relation classification. We use the last words of the detected type of entity phrase for relation classification; in other words, the words with L or U tags in the BIOES scheme are considered. For instance, as shown in Figure 1, we perform a relation classification using *Minghella* with L-PER (Person) tag and *England* with U-COUNTRY tag.

The relation candidate vector is formed as the concatenation $d_p = [\uparrow hp; \downarrow hp_1; \downarrow hp_2]$, where $\uparrow hp$ is the hidden state vector of the top LSTM unit in the bottom-up LSTM-RNN (representing the lowest common ancestor of the target two entities) and $\downarrow hp_1, \downarrow hp_2$ are the hidden state vectors of the two LSTM units representing the first and the second entities in the top-down LSTM-RNN. $\uparrow hp, \downarrow hp_1$, and $\downarrow hp_2$ appear as arrows in Figure 1. According to Eq. (6), the output phase is a softmax classifier with weight matrices W and bias vectors b .

$$y_p = \text{softmax} (w^{rh} d_p + b^{rh}). \quad (6)$$

Here, w^{rh} and b^{rh} are the weight matrices and bias vectors, respectively.

4. Experiments

In this section, the datasets we used in our experiments are introduced, and the values of the hyper parameters and evaluation metrics are presented.

4.1 Dataset

We conducted experiments on two public datasets: ACE05 and KBP37.

ACE05. There are seven entity types and seven relation classes in this dataset. The entity types include: Organization, Location, Person, Vehicle, Geo-Political Entities, Facility, and Weapon. The relation classes are: Org-Affiliation, Physical, Artifact, Part-Whole, Person-Social, Gen-Affiliation, and Metonymy.

KBP37. This corpus defines eight coarse-grained entity types and 37 relation classes. The entity types are Person, Organization, Country, City, State or Province, Affiliation, Nationality, and Number. We divided the data source into three groups: training (15,917), development (1,724), and test (3,405). The relation types of the dataset are shown in Table 1. As exhibited, there are 18 directional relations and one additional *other* relation, resulting in 37 relation classes.

4.2 Metrics

The metrics we used to evaluate the performance of our architecture include Precision, Recall, and F-Measure (F1). We considered a relation type for a word pair as correct when the type of both the entities and their relation type were classified correctly.

Table 1. Relation types in KBP37 dataset

Relation type	Relation type
Person, alternate names	Organization, alternate names
Person, origin	Organization, subsidiaries
Person, spouse	Organization, top members/employees
Person, title	Organization, founded
Person, employee of	Organization, founded by
Person, countries of residence	Organization, country of headquarters
Person, state, or provinces of residence	Organization, state, or province of headquarters
Person, cities of residence	Organization, city of headquarters
Person, country of birth	Organization, members
No relation	-

Table 2. Hyper parameter list used in the proposed architecture

Hyper parameter	Explanation	Value
d	Dimension of word embedding	300
n_{lt}	The number of hidden units of Bi-LSTM	100
ne	The number of CNN filters in NER module	100
j	The number of CNNs in Mix-CNN layer	3
ρ	The ratio of dropout	0.3

4.3 Hyper Parameters

We simulated our end-to-end architecture using powerful language (Python and Keras library). The hyper parameters used in the experiments are presented in Table 2.

4.4 Results

Evaluation. We compared the working of our model with that of other systems on the ACE05 and KBP37 datasets. The compared systems included: (i) *NER without CNN*, which is a proposed NER model without the CNN phase and concatenation operation given by Miwa & Bansal [14], and (ii) pipeline NER model and pipeline RE model, with no interaction.

The comparison results show that our model works better than that of Miwa and Bansal [14] on the ACE05 dataset (Table 3). It is evident that the novelty of our architecture (adding CNN in the NER module) is effective and this enhancement propagates to the next module. Thus, we can see an improve-

Table 3. Comparison of proposed architecture on ACE05 dataset

Model	Entity			Relation		
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
Our architecture	82.5	83.8	83.1	51.1	53.8	52.4
Miwa and Bansal [14]	81.5	82.1	81.8	50.6	52.9	51.8

Table 4. Comparison of proposed architecture on KBP37 dataset

Model	Entity			Relation		
	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
Our architecture	80.1	81.7	80.9	47.3	48.6	47.9
Our architecture - CNN	79.0	80.2	79.6	46.1	47.8	46.9
Pipeline	78.3	79.5	78.9	46.0	46.2	46.1

ment of 1.3% and 1.4% in the F1-score for entity classification and relation identification, respectively. Therefore, this proves that the concatenation of the feature vectors of the Bi-LSTM-RNN and CNN obtains more effective information for entity extraction.

Table 4 displays the evaluation results of two different variations of our system on the KBP37 dataset: (i) the system without CNN and (ii) the system with a pipeline approach. The results prove that adding CNN in the proposed architecture increases the efficiency of the two processes by approximately 1% on this corpus. They also demonstrate that although the false entity tag assignment in the NER module has a negative effect on the performance of RE module in an end-to-end model, the RE module in our architecture provides better results than in the pipeline model. The reason is that both the modules (entity detection and relation extraction) in a pipeline model are trained independently without considering any interaction between them, such as sharing of the underlying Bi-LSTM-RNN layer. Conversely, in our model, both the NER and RE modules learn the entities and their relations interactively and simultaneously.

We can observe that better experimental results were obtained on ACE05 than on KBP37. This may be because the average length of the sentences in ACE05 is less, thus causing less complications and enabling higher accuracy. Another reason may be the number of relation types in the two corpora; there are 37 relation classes in KBP37, which are significantly more than the ACE05 relation types. Therefore, the performance reduces by increasing the complexity.

Effect of sentence length on performance. Figure 3 depicts the performance of the models for relation extraction on

Figure 3. Results vs. sentence length.

sentences with different lengths on ACE05 and KBP37, respectively. The x-axis represents the sentence length and the y-axis indicates the F-measure values. The number of words in the sentences of both the datasets do not exceed 60. The F-measure metric is calculated on the average length value of the sentences in the range $[m - 9, m]$, where $m = \{10, 20, \dots, 60\}$.

Compared with the pipeline approach, our model displays higher accuracy and this improvement suggests that the interaction of the two modules is really beneficial (e.g., identifying a *born in* relation may help the NER module in detecting the type of two other entities, i.e., Person, Countries_of_birth, and vice versa).

In addition, Figure 3 demonstrates that increasing the sentence length has negative impact on F-measure. In other words, longer the sentence, lower the F-measure.

5. Conclusion

In this research, we propose a novel architecture for entity and relation classification using a deep neural network framework.

Our architecture introduces joint modeling of the NER and RE components in a single model with the contribution of two different structures (sequential and tree) of LSTM-RNN and CNN. This model classifies entity mentions and their relation type simultaneously via a cooperative deep neural network. In this neural network-based framework, the parameters are renewed via a classification feedback received by both the processes. In particular, the model employs bidirectional LSTM-RNN and CNN models to automatically learn the features of the input sentence and then detects the relation between the pair entities by using a bidirectional tree-structured LSTM. The results of experiments on ACE05 and KBP37 datasets prove that our architecture surpasses competitor models in performance. The experiment results also show that the performance of the method deteriorates with increasing sentence length.

In future, we aim to work on Persian language texts and implement the NER and RE tasks on them. Furthermore, we would like to investigate sentences with more than two entity mentions and more relation types.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

- [1] B. Y. Kang and D. W. Kim, "A multi-resolution approach to restaurant named entity recognition in Korean web," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 4, pp. 277-284, 2012. <https://doi.org/10.5391/IJFIS.2012.12.4.277>
- [2] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," *Advances in Neural Information Processing Systems*, vol. 16, pp. 25-32, 2003.
- [3] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning*, San Francisco, CA, 2001, pp. 282-289.
- [4] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the 21st International Conference on machine Learning*, Helsinki, Finland, 2004, pp. 104-112. <https://doi.org/10.1145/1015330.1015341>
- [5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493-2537, 2011.
- [6] X. Ma, and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 1064-1074. <https://doi.org/10.18653/v1/p16-1101>
- [7] A. Culotta, and J. Sorensen, "Dependency tree kernels for relation extraction," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Barcelona, Spain, 2004, pp. 423-429. <https://doi.org/10.3115/1218955.1219009>
- [8] B. Rink, and S. Harabagiu, "UTD: classifying semantic relations by combining lexical and semantic resources," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, Los Angeles, CA, 2010, pp. 256-259.
- [9] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, 2014, pp. 2335-2344.
- [10] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "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*, Lisbon, Portugal, 2015, pp. 1785-1794. <https://doi.org/10.18653/v1/d15-1206>
- [11] B. Yang and C. Cardie, "Joint inference for fine-grained opinion extraction," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013, pp. 1640-1649.
- [12] R. J. Kate and R. J. Mooney, "Joint entity and relation extraction using card-pyramid parsing," in *Proceedings of the 14th Conference on Computational Natural Language Learning*, Uppsala, Sweden, 2010, pp. 203-212.
- [13] S. Singh, S. Riedel, B. Martin, J. Zheng, and A. McCallum, "Joint inference of entities, relations, and coreference," in *Proceedings of the 2013 Workshop on Automated*

knowledge Base Construction, San Francisco, CA, 2013, pp. 1-6. <https://doi.org/10.1145/2509558.2509559>

- [14] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," 2016, Available <https://arxiv.org/abs/1601.00770>
- [15] S. Zheng, Y. Hao, D. Lu, H. Bao, J. Xu, H. Hao, and B. Xu, "Joint entity and relation extraction based on a hybrid neural network," *Neurocomputing*, vol. 257, pp. 59-66, 2017. <https://doi.org/10.1016/j.neucom.2016.12.075>
- [16] F. Li, M. Zhang, G. Fu, and D. Ji, "A neural joint model for entity and relation extraction from biomedical text," *BMC Bioinformatics*, vol. 18, article no. 198, 2017. <https://doi.org/10.1186/s12859-017-1609-9>
- [17] P. Gupta, H. Schutze, and B. Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," in *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, 2016, pp. 2537-2547.
- [18] A. Katiyar and C. Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 917-928. <https://doi.org/10.18653/v1/p17-1085>
- [19] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no 11, pp. 2673-2681, 1997. <https://doi.org/10.1109/78.650093>
- [20] S. Zheng, J. Xu, P. Zhou, H. Bao, Z. Qi, and B. Xu, "A neural network framework for relation extraction: learning entity semantic and relation pattern," *Knowledge-Based Systems*, Vol. 114, pp. 12-23, 2016. <https://doi.org/10.1016/j.knosys.2016.09.019>

Elham Parsaeimehr is a Ph.D. student in Department of Computer Engineering, Arak Branch, Islamic Azad University. She received a master's degree in Computer Engineering from Science and Research branch, Islamic Azad University, Iran. Her research interests are datamining, machine learning, and text mining.

Mehdi Fartash received his Ph.D. from Science and Research Branch, Islamic Azad University, Tehran, Iran. He is currently an Assistant Professor in Department of Computer Engineering, Arak Branch, Islamic Azad University. His research interests are datamining, bio-inspired speech and acoustic processing, computational intelligence and Distributed computing.

Javad Akbari Torkestani received the Ph.D. degree in Computer Engineering from Science and Research University, Iran, in 2009. Currently, he is an Associate Professor with Department of Computer Engineering, Arak Branch, Islamic Azad University, Arak, Iran. Prior to his current position, he was an Assistant Professor from 2010 to 2013. He joined the faculty of Computer Engineering Department at Arak Azad University as a lecturer in 2005. His research interests include wireless networks, Web engineering, fault tolerant systems, grid computing, learning systems, parallel algorithms, soft computing, and datamining.