Relation Extraction among Multiple Entities using a Dual Pointer Network with a Multi-Head Attention Mechanism

Seongsik Park **Harksoo Kim**
Kangwon National University, South Korea
{a163912, nlpdrkim}@kangwon.ac.kr

Abstract

Many previous studies on relation extraction have been focused on finding only one relation between two entities in a single sentence. However, we can easily find the fact that multiple entities exist in a single sentence and the entities form multiple relations. To resolve this problem, we propose a relation extraction model based on a dual pointer network with a multi-head attention mechanism. The proposed model finds n-to-1 subject-object relations by using a forward decoder called an object decoder. Then, it finds 1-to-n subject-object relations by using a backward decoder called a subject decoder. In the experiments with the ACE-05 dataset and the NYT dataset, the proposed model achieved the state-of-the-art performances (F1-score of 80.5% in the ACE-05 dataset, F1-score of 78.3% in the NYT dataset)

1 Introduction

Relation extraction is the task of recognizing semantic relations (*i.e.*, tuple structures; subject-relation-object triples) among entities in a sentence. Figure 1 shows three triples that can be extracted from the given sentence.

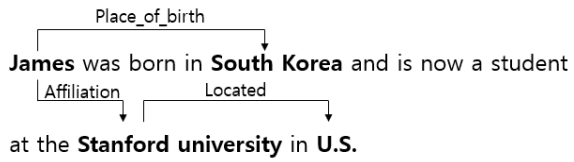

Figure 1: Subject-relation-object triples in a sentence

With significant success of neural networks in the field of natural language processing, various relation extraction models based on convolutional neural networks (CNNs) have been suggested (Kumar, 2017); the CNN model with max-pooling (Zeng et al., 2014), the CNN model with multi-sized window kernels (Nguyen and Grishman, 2015), the combined CNN model (Yu and Jiang, 2016), and the contextualized graph convolutional network (C-GCN) model (Zhang et al., 2018).

Relation extraction models based on recurrent neural network (RNNs) has been the other popular choices; the long-short term memory (LSTM) model with dependency tree (Miwa and Bansal, 2016), the LSTM model with position-aware attention mechanism (Zhang et al., 2017), and the walk-based model on entity graphs (Christopoulou et al., 2019). Most of these previous models have been focused on extracting only one relation between two entities from a single sentence. However, multiple entities exist in a single sentence, and these entities can form multiple relations. To address this issue, we propose a relation extraction model to find all possible relations among multiple entities in a sentence at once.

The proposed model is based on the pointer network (Vinyals et al., 2015). The pointer network is a sequence-to-sequence (Seq2Seq) model in which an attention mechanism (Bahdanau et al., 2015) is modified to learn the conditional probability of an output whose values correspond to positions in a given input sequence. We modify the pointer network to have dual decoders; an object decoder (a forward decoder) and a subject decoder (a backward decoder). The object decoder plays a role to extract n-to-1 relations as shown in the following example: *(James-BirthPlace-South Korea)* and *(Tom-BirthPlace-SouthKorea)* extracted from 'James and Tom was born in South Korea'. The subject decoder plays a role to extract 1-to-n relations as shown in the following example*: (James-Position-student)* and *(James-Affiliation-Stanford university)* extracted from 'James is a student at Stanford university'.

2 Dual Pointer Network Model for Relation Extraction

Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER), pages 47–51
Hong Kong, November 3, 2019.

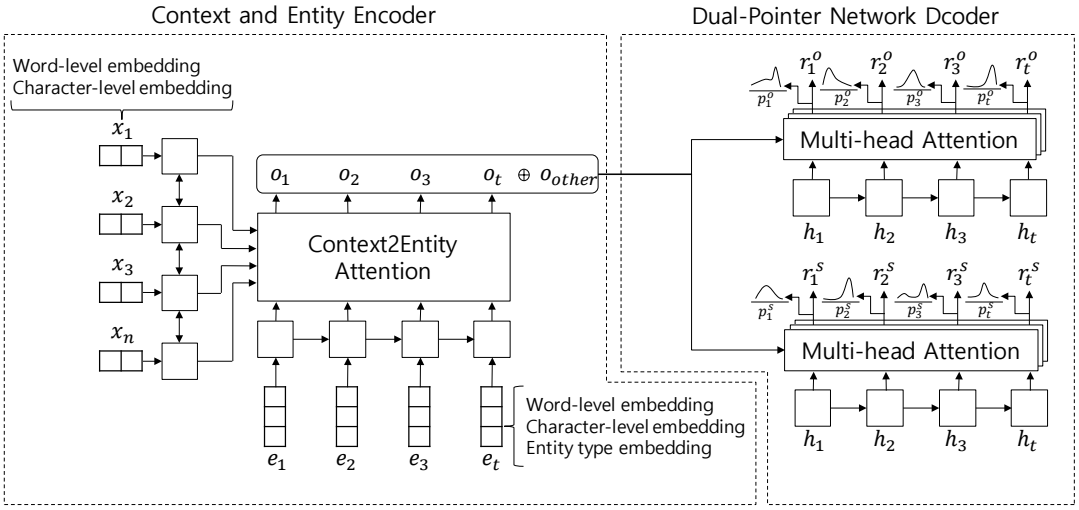

Figure 2: Overall architecture of the dual pointer networks for relation extraction

Figure 2 illustrates an overall architecture of the proposed model. As shown in Figure 2, the proposed model consists of two parts: One is a context and entity encoder, and the other is a dual pointer network decoder.

The context and entity encoder (the left part of Figure 2) computes degree of associations between words and entities in a given sentence. In the context and entity encoder, $\{ x_1, x_2, \dots, x_n \}$ and $\{e_1, e_2, \dots, e_t\}$ are word embedding vectors and entity embedding vectors, respectively. The word embedding vectors are concatenations of two types of embeddings; word-level GloVe embeddings for representing meanings of words (Pennington et al., 2014) and character-level CNN embeddings for alleviating out-of-vocabulary problems (Park et al., 2018). The entity embedding vectors are similar to the word embedding vectors except that entity type embeddings are additionally concatenated. The entity type embeddings are vector representations associated with each entity type[1] and are initialized as random values. The word embedding vectors are input to a bidirectional LSTM network in order to obtain contextual information. The entity embedding vectors are input to a forward LSTM network because entities are listed in the order appeared in a sentence. The output vectors of the bidirectional LSTM network and the forward LSTM network are input to the context-to-entity attention layer ('Context2Entity Attention' in Figure 2) in order to compute relative degrees of associations between words and entities according to the same manner with the Context2Query attention proposed in Seo et al. (2017).

In a pointer network, attentions show position distributions of an encoding layer. Since an attention is highlighted at only one position, the pointer network has a structural limitation when one entity forms relations with several entities (for instance, '*James*' in Figure 1). The proposed model adopts a dual pointer network decoder (the right part of Figure 2) to overcome this limitation. The first decoder called an object decoder learns the position distribution from subjects to objects. Conversely, the second decoder called a subject decoder learns the position distribution from objects to subjects. In Figure 1, 'James' should point to both 'south Korea' and 'Stanford university'. If we use a conventional forward decoder (the object decoder), this problem could not be solved because the forward decoder cannot point to multiple targets. However, the subject decoder (a backward decoder) can resolve this problem because 'south Korea' and 'Stanford university' can respectively point to 'James'.

Additionally, we adopt a multi-head attention mechanism in order to improve performances of the dual pointer network. The multi-head attention mechanism splits the input value into multiple heads and compute the attention of each head. The inputs $\{h_1, h_2, \dots, h_t\}$ of multi-head attention layer are the vectors that concatenate the entity embedding vectors $\{e_1, e_2, \dots, e_t\}$ and the output vectors $\{ o_1, o_2, \dots, o_t \}$ of the context-to-entity attention layer. The random initialized vector o_{other} is used

[1] We use seven entity types such as person, location, organization, facility, geo-political, vehicle and weapon in the ACE-2005 dataset. Then, we use three entity type such as person, location and organization in the NYT dataset.

for handling entities that do not have any relations with other entities. In other words, entities without any relations point to o_{other}. As shown in Figure 2, the dual pointer network decoder returns two kinds of value sequences. One is a sequence of relation labels $\{r_1, r_2, \dots, r_t\}$, and the other is a sequence of pointed positions $\{p_1, p_2, \dots, p_t\}$.

3 Evaluation

3.1 Datasets and Experimental Settings

We evaluated the proposed model by using the following benchmark datasets.

ACE-05 corpus: The Automatic Content Extraction dataset (ACE) includes seven major entity types and six major relation types. The ACE-05 corpus is not proper to evaluate models to extract multiple triples from a sentence. Therefore, if some triples in the ACE-05 corpus share a sentence (*i.e.*, some triples are occurred in the same sentence), we merged the triples. As a result, we obtained a data set annotated with multiple triples. Then, we divided the new data set into a training set (5,023 sentences), a development set (629 sentences), and a test set (627 sentences) by a ratio of 8:1:1.

New York Times (NYT) corpus (Riedel et al., 2010): the NYT corpus is a news corpus sampled from New York Times news articles. The NYT corpus is produced by distant supervision method. Zheng et al (2017) and Zeng et al (2018) used this dataset as supervised data. We excluded sentences without relation facts from Zheng's corpus. Finally, we obtained 66,202 sentences in total. We used 59,581 sentences for training and 6,621 for evaluate.

Optimization of the proposed model was done with the Adam optimizer (Kingma and Ba, 2014) with learning-rate = 0.001, encoder units = 128, decoder units = 256, dropout rate = 0.1.

3.2 Experimental Results

Table 1 shows performances of the proposed model and the comparison models when the ACE-05 corpus is used as an evaluation dataset. In Table 1, SPTree LSTM (Miwa and Bansal, 2016) is a model that applies the dependency information between the entities. FCM (Gormley et al., 2015) is a model in which handcrafted features are combined with word embeddings. CNN+RNN (Nguyen and Grishman, 2015) is a hybrid model of CNN and RNN. HRCNN (Kim and Choi, 2018) is hybrid model of CNN, RNN, and Fully-Connected Neural Network (FNN). WALK is a graph-based neural network model for relation extraction (Fenia et al., 2019). As shown in Table 1, the proposed model outperformed all comparison models.

Model	P	R	F1
SPTreeLSTM(Miwa+2016)	57.2	54.0	55.6
FCM(Gormley+2015)	71.5	49.3	58.2
CNN+RNN(Nguyen+2015)	69.3	66.3	67.7
HRCNN(Kim+2018)	-	-	74.1
WALK(Fenia+2019)	69.7	59.5	64.2
The Proposed Model	**79.1**	**81.7**	**80.5**

Table 1: Performance comparisons on ACE-05 (P: Precision, R: Recall rate, F1: F1-score in percentage)

Model	P	R	F1
NovelTag (Zheng+2017)	61.5	41.4	50.0
MultiDecoder(Zeng+2018)	61.0	56.6	58.7
The Proposed Model	**74.9**	**82.0**	**78.3**

Table 2: Performance comparisons on NYT (P: Precision, R: Recall rate, F1: F1-score in percentage)

Table 2 shows performances of the proposed model and the comparison models when the NYT corpus is used as an evaluation dataset. In Table 2, NovelTag (Zheng et al., 2017) MultiDecoder (Zeng et al., 2018) are models that jointly extract entities and relations. It is not reasonable to directly compare the proposed model with NovelTag and MultiDecoder because the proposed model needs gold-labeled entities while NovelTag and MultiDecoder automatically extracts entities from sentences. Although the direct comparisons are unfair, the proposed model showed much higher performances than expected.

# of entities	# of sentences	F1
2	316	91.5
3	108	80.1
4	68	74.5
More than 5	137	75.8

Table 3: Performance changes according to the number of entities per sentence (F1: F1-score in percentage)

Table 3 shows performance changes according to the number of entities per sentence in the ACE-05 corpus. As shown in Table 3, the more the number of entities per sentence was, the lower the performances of the proposed model were. We think that the decreasing of performances is due to the increasing of complexities. The performance when the number of entities is more than five was slightly improved as compared with the performance when the number of entities is four. The reason is that

many entities do not have any relations with the other entities.

4 Conclusion

We proposed a relation extraction model to find all possible relations among multiple entities in a sentence at once. The proposed model is based on a pointer network with a multi-head attention mechanism. To extract all possible relations from a sentence, we modified a single decoder of the pointer network to a dual decoder. In the dual decoder, the object decoder extracts n-to-1 subject-object relations, and the subject decoder extracts 1-to-n subject-object relations. In the experiments with the ACE-05 corpus and the NYT corpus, the proposed model showed good performances.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2013-0-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform.

References

Dzmitry Bahdanau, KyungHyun Cho and Yoshua Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473v7.*

Fenia Christopoulou, Makoto Miwa and Sophia Ananiadou1. 2019. A Walk-based Model on Entity Graphs for Relation Extraction. *arXiv preprint arXiv:1902.07023v1*.

Matthew R. Gormley, Mo Yu and Mark Dredze. 2015. Improved Relation Extraction with Feature-Rich Compositional Embedding Models. *arXiv preprint arXiv:1505.02419v3*.

SeonWo Kim and SungPil Choi. 2018. Relation Extraction using Hybrid Convolutional and Recurrent Networks. In *Proceedings of Korea Computer Congress 2018* (KCC 2018). pages 619-621

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Shantanu Kumar. 2017. A Survey of Deep Learning Methods for Relation Extraction. *arXiv preprint arXiv:1705.03645v1*.

Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. *arXiv preprint arXiv:1601.00770v3*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT 2015)*. pages 39-48.

Thien Huu Nguyen and Ralph Grishman. 2015. Combining Neural Networks and Log-linear Models to Improve Relation Extraction. *arXiv preprint arXiv:1511.05926v1*.

Jeffrey Pennington, Richard Socher and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. pages 1532–1543.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010)*. pages 148-163.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi and Hananneh Hajishirz. 2017. Bi-Directional Attention Flow for Machine Comprehension. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. 2017. Attention all you need. In *Advances in Neural Information Processing Systems (NIPS 2017)*. pages 5998-6008.

Oriol Vinyals, Meire Fortunato and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems (NIPS 2015)*. pages 2692-2700.

Jianfei Yu and Jing Jiang. 2016. Pairwise Relation Classification with Mirror Instances and a Combined Convolutional Neural Network. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*. pages 2366-2377.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2014)*. pages 2335-2344.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu and Jun Zhao. 2018. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. pages 506-514

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao,Peng Zhou and Bo Xu. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. *arXiv preprint arXiv:1706.05075v1*.

Yuhao Zhang, Peng Qi and Christopher D. Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*. pages 2205-2215.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. pages 35-45