applied sciences

MDPI

Article

Dual Pointer Network for Fast Extraction of Multiple Relations in a Sentence †

Seongsik Park [1] and Harksoo Kim [2,*]

[1] Computer and Communications Engineering, Kangwon National University, Chuncheon 24341, Korea; a163912@kangwon.ac.kr

[2] Computer Science and Engineering, Konkuk University, Seoul 05029, Korea

* Correspondence: nlpdrkim@konkuk.ac.kr; Tel.: +82-2-450-3499

† This paper is an extended version of paper published in The Second Workshop on Fact Extraction and VERification. (FEVER 2.0) at EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019.

Received: 25 April 2020; Accepted: 30 May 2020; Published: 1 June 2020

Featured Application: Ontology construction module for AI applications.

Abstract: Relation extraction is a type of information extraction task that recognizes semantic relationships between entities in a sentence. Many previous studies have focused on extracting only one semantic relation between two entities in a single sentence. However, multiple entities in a sentence are associated through various relations. To address this issue, we proposed 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 using a forward object decoder. Then, it finds *1*-to-*n* subject–object relations using a backward subject decoder. Our experiments confirmed that the proposed model outperformed previous models, with an F1-score of 80.8% for the ACE (automatic content extraction) 2005 corpus and an F1-score of 78.3% for the NYT (New York Times) corpus.

Keywords: relation extraction; dual pointer network; context-to-entity attention

1. Introduction

Relation extraction is a task that involves recognizing semantic relations (i.e., tuple structures; {subject, relation, object triples}) among entities in a sentence [1]. Zeng et al. [2] divided sentences into three types according to the triplet overlap degree, i.e., normal, entity pair overlap (EPO), and single entity overlap (SEO). In the normal type, the triples do not have overlapped entities; in the EPO type, some triples have an overlapped entity pair; and in the SEO type, some triplets have an overlapped entity, but these triplets do not have overlapped entity pairs. In this study, we focus on promptly extracting both the normal and SEO types because most relations are included in these types, as shown in Figure 1.

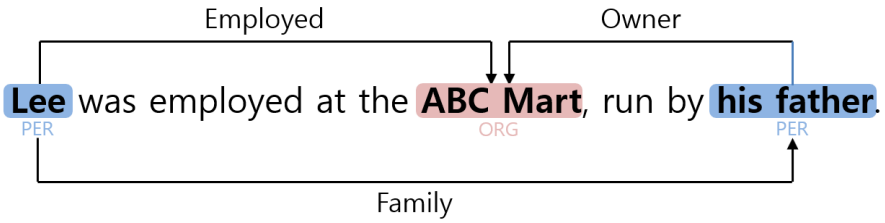


Figure 1. Subject-relation-object triples in a sentence. PER: person; ORG: organization.

In Figure 1, {Lee, employed, ABC Mart}, {Lee, Family, his Father} and {his Father, Owner, ABC Mart} are SEO types. To promptly extract these relations, we adopt the concept of dependency parsing

in which dependent words point to the head words by scanning each word in a sentence. We propose a dual pointer network model to efficiently extract multiple relations from a sentence through forward scanning (i.e., scanning from the first word to the last) and backward scanning (i.e., scanning from the last word to the first). The proposed model discovers an object of the current subject during forward scanning. Through forward scanning, all normal type relations can be found. However, SEO type relations are only partially found because a subject should point to only one object in the pointer network architecture. To address this limitation, the proposed model performs backward scanning to identify the subject of the current object.

The remainder of this paper is organized as follows. In Section 2, we review previous studies on relation extraction. Section 3 describes the proposed dual pointer network model. In Section 4, we elaborate on the experimental setup and results. Finally, we conclude the study in Section 5.

2. Previous Works

With the significant success of deep neural networks in the field of natural language processing, many researchers have proposed various relation extraction models based on convolutional neural networks (CNNs). These include the CNN model based on max-pooling [3], the CNN model based on multiple sizes of kernels [4], the combined CNN model [5], and the contextualized graph convolutional network (C-GCN) model [6]. Relation extraction models based on recurrent neural network (RNNs) have also been proposed, including the long-short term memory (LSTM) model based on the dependency tree [7] and the LSTM model using the position-aware attention technique [8]. These models have focused on normal type extraction (i.e., extracting only one relation between two entities from a single sentence). However, many entities in a single sentence can form multiple relations. Some studies have proposed multiple relation extraction to resolve this problem. For example, Luan et al. [9] treated triples in sentences as a graph and proposed a multiple relations extraction model that iteratively extracts spans between triples in the graph. In the present study, we propose a relation extraction model to simultaneously find all possible relations among multiple entities in a sentence. The proposed model is based on the pointer network [10]. The pointer network is a sequence-to-sequence (Seq2Seq) model in which an attention mechanism [11] is modified to learn the conditional probability of an output, where the values correspond to positions in a given input sequence. We modify the pointer network to include dual decoders, an object decoder (a forward decoder) and a subject decoder (a backward decoder). The object decoder extracts *n*-to-*1* relations as shown in the following example: {Lee, employed, ABC Mart} and {his Father, Owner, ABC Mart} are extracted from the sentence. The subject decoder extracts 1-to-n relations as shown in the following example: {Lee, employed, ABC Mart} and {Lee, Family, his Father} are extracted from the sentence.

3. Dual Pointer Network Model for Relation Extraction

Figure 2 illustrates the architecture of the proposed model. This consists of two parts, a context and entity encoder, and a dual pointer network decoder.

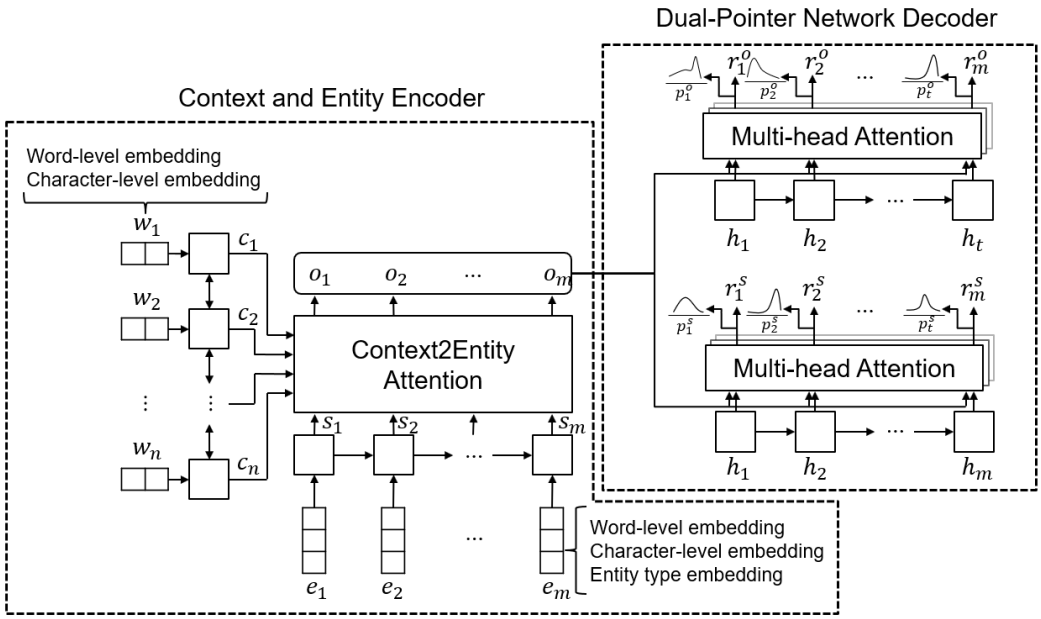
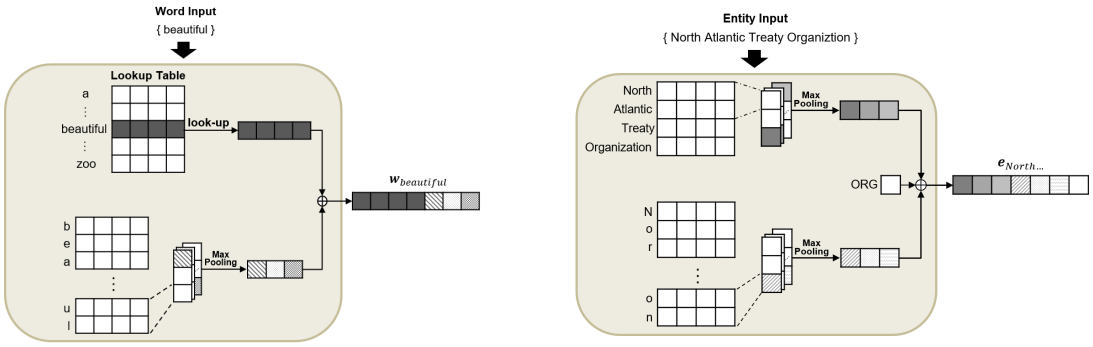


Figure 2. The overall architecture of dual pointer networks for relation extraction.

3.1. Context and Entity Encoder

The context and entity encoder computes the degree of association between words and entities in a given sentence. For example, $\{w_1, w_2, \ldots, w_i\}$ and $\{e_1, e_2, \ldots, e_m\}$ refer to word and entity embedding vectors, respectively. Figure 3 illustrates the process of word and entity embedding.



(a) (b)

Figure 3. (**a**) Word embedding process, (**b**) Entity embedding process.

As shown in Figure 3, the word embedding vectors are concatenations of two types of embeddings: word-level GloVe [12] embeddings for representing the meaning of words and character-level CNN embeddings [13] for addressing out-of-vocabulary problems. The entity embedding vectors are concatenations of three types of embeddings: word-level CNN embedding for representing the meaning of entities composed of multiple words, character-level CNN embedding for addressing out-of-vocabulary problems, and entity type embedding for representing the categorical information of input entities. Word-level GloVe embeddings represent each word in the word-level CNN embedding.

The word embedding vectors are used as input for a bidirectional LSTM network to obtain contextual information as follows:

$$\begin{aligned} \overrightarrow{c}_i &= \text{LSTM}(w_i, \overrightarrow{c}_{i-1}), \\ \overleftarrow{c}_i &= \text{LSTM}(w_i, \overleftarrow{c}_{i-1}), \\ c_i &= [\overrightarrow{c}_i; \overleftarrow{c}_i], \end{aligned} \tag{1}$$

where w_i is an embedding vector of the i-th word in a sentence, and $[\overrightarrow{c}_i; \overleftarrow{c}_i]$ is a concatenation of $\overrightarrow{c}_i$ and $\overleftarrow{c}_i$ that represents the output vectors of a forward LSTM and a backward LSTM, respectively. The entity embedding vectors are used as input for a forward LSTM network because the entities are listed in the order that they appear in a sentence, as shown below.

$$s_t = \text{LSTM}(e_t, s_{t-1}), \tag{2}$$

where e_t is an embedding vector of the t-th one among all entities occurring in a sentence, and s_t is an output vector encoded by a forward LSTM. The output vectors of the bidirectional LSTM network $\{c_1, c_2, \ldots, c_i\}$ and the forward LSTM network $\{s_1, s_2, \ldots, s_t\}$ are used as input for the context-to-entity attention layer (as shown in Figure 2), to compute the relative degrees of association between words and entities. This is similar to the well-known multi-head attention mechanism [14], as shown below.

$$\begin{aligned} q_j &= w^a * split(q, n)_j, \\ a_j &= softmax(\tfrac{q_j k_j}{\sqrt{d}}), \\ head_j &= a_j v_j, \\ o_t &= \text{relu}(w^o[head_0;\ head_1;\ head_2; \ldots; head_n] + b^o), \end{aligned} \tag{3}$$

where the query q is set to s_t, the key k and the value v are set to C's. The query q is split into n vectors, where n is the number of heads. The attention score a_j is calculated by a scaled-dot product, where d is a normalization factor. The context-to-entity layer output o_t is determined through a fully-connected neural network (FNN) using a concatenation of n heads as input.

3.2. Dual Pointer Network Decoder

In a pointer network, attentions show the position distributions of an encoding layer. Since attention is highlighted at only one position, the pointer network has a structural limitation when one entity forms relations with several entities (for instance, "Lee" in Figure 1). The proposed model adopts a dual pointer network decoder (see Figure 2) to overcome this limitation. The first decoder called an object decoder, learns the position distribution from subjects to objects as follows:

$$\begin{aligned} h_t &= [e_t; s_t], \\ g_t &= \text{LSTM}(h_t, g_{t-1}), \\ score_t^{obj} &= v^{obj}\tanh(w^{obj}[\text{O}; g_t]), \\ a_t^{obj} &= \text{softmax}(score_t^{obj}), \\ \hat{p}_t^{obj} &= \text{argmax}(a_t^{obj}), \\ \hat{r}_t^{obj} &= \text{argmax}(u^{obj}\tan\text{h}(z^{obj}[a_t^{obj} O; g_t])), \end{aligned} \tag{4}$$

where h_t is a concatenation of the entity embedding vector e_t and the LSTM-encoded entity embedding vector s_t, and the decoding vector g_t (i.e., the t-th entity to determine its objects) is calculated by the forward LSTM. Then, a_t^{obj} is the position distribution based on the attention scores $score_t^{obj}$ between g_t and the other entities $o_1, \ldots, o_{t-1}, o_{t+1}, \ldots, o_m$ in the context-to-entity attention layer. $\hat{p}_t^{obj}$ and $\hat{r}_t^{obj}$ represent a position and a relation name of g_t's object, respectively. The weighting parameters $v, w, u,$ and z are set during the training phase. Conversely, the second decoder, called a subject decoder,

learns the position distribution from objects to subjects in the same manner as the object decoder, as shown below.

$$\begin{aligned} score_t^{sub} &= v^{sub}\tanh(w^{sub}[\mathrm{O}; g_t]), \\ a_t^{sub} &= \mathrm{softmax}(score_t^{sub}), \\ \hat{p}_t^{sub} &= \mathrm{argmax}(a_t^{sub}), \\ \hat{r}_t^{sub} &= \mathrm{argmax}(u^{sub}\tan \mathrm{h}(z^{sub}[a_t^{sub}O; g_t])), \end{aligned} \tag{5}$$

where $\hat{p}_t^{sub}$ and $\hat{r}_t^{sub}$ represent a position and a relation name of g_t's subject, respectively. In Figure 1, "Lee" should point to both "ABC mart" and "his father." This problem cannot be solved using the conventional forward decoder because it cannot point to multiple targets. However, the subject decoder (a backward decoder) resolves this problem, because "ABC mart" and "his father" can point to "Lee." Additionally, we adopt a multi-head attention mechanism to improve the performance of the dual pointer network; this is shown in the following equation.

$$\begin{aligned} q_j &= w^l * split(q, n)_j, \\ a_j &= softmax(\tfrac{q_j k_j}{\sqrt{d}}), \\ head_j &= a_j v_j, \\ \hat{p}_t &= \mathrm{argmax}(\tfrac{1}{n}\textstyle\sum_{k=0}^{n} a_k), \\ \hat{r}_t &= \mathrm{argmax}(\mathrm{relu}(w^r[head_0;\ head_1;\ head_2; \ldots; head_n] + b^r)), \end{aligned} \tag{6}$$

where the query q is set to g_t, the key k and the value v are set to O's. The position distribution $\hat{p}_t$ is calculated by an average of n multi-head attention vectors, and the relation name $\hat{r}_t$ is determined through an FNN using a concatenation of n heads as the input.

3.3. Implementation detail

The context and entity encoder comprised 256 hidden units in each layer, and the dual pointer network decoder comprised 512 hidden units. We adopted a 0.1 drop-out probability for all the LSTM cells. We used 8 heads, with 32 units per head, for the multi-head attention. The vocabulary size and word-embedding size was set to 16,925 and 300, respectively. The filter size of the CNNs for character and word embeddings were 3, 4, and 5. The total number of filters was 100. 50-dimensional random initialized vectors were used for the character and entity embeddings. A cross-entropy function was used as a cost function to maximize the log-probability as follows:

$$\begin{aligned} CE(y, \widetilde{y}) &= -\textstyle\sum_i^C y_i \log(\widetilde{y}_i), \\ Loss &= \tfrac{\alpha}{2}\Big(CE(p^{sub}, \widetilde{p}^{sub}) + CE(p^{obj}, \widetilde{p}^{obj})\Big) + \tfrac{(1-\alpha)}{2}(CE(r^{sub}, \widetilde{r}^{sub}) + CE(r^{obj}, \widetilde{r}^{obj})), \end{aligned} \tag{7}$$

where y is the target answer, $\widetilde{y}$ is the score distribution of the model prediction, and C is the number of target classes. The loss is calculated by the cross-entropy combination of all targets and predictions. The weighting factor α was experimentally set to 0.6 as a scalar value.

4. Evaluation

4.1. Datasets and Experimental Setting

We evaluated the proposed model using the following benchmark datasets.

ACE-2005 corpus [15]: The automatic content extraction (ACE) dataset included seven major entity types and six major relation types. The ACE-2005 corpus does not properly evaluate models that extract multiple triples from a sentence. Therefore, if some triples in the ACE-2005 corpus share a sentence (i.e., some triples occur in the same sentence), the triples were merged, as shown in Figure 4.

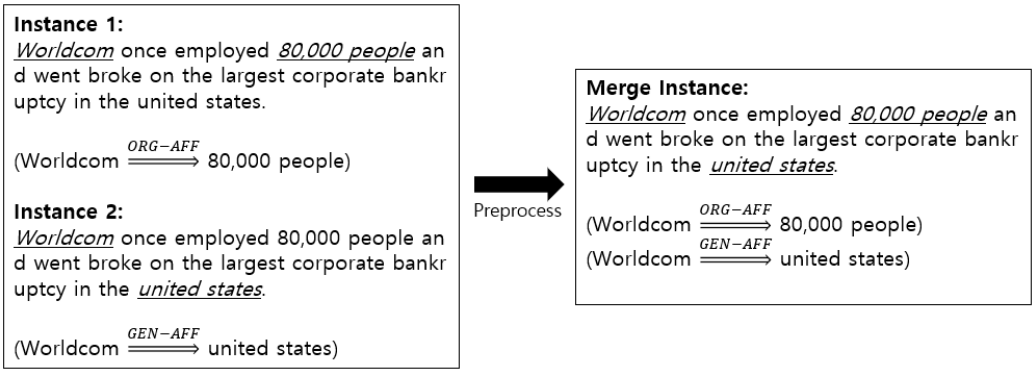


Figure 4. ACE-2005 data preprocess. ORG-AFF: organization-affiliation; GEN-AFF: general entity-affiliation.

As a result, we obtained a dataset annotated with multiple triples. We divided the new dataset into a training set (5023 sentences), a development set (629 sentences), and a test set (627 sentences) by a ratio of 8:1:1. Table 1 shows the composition of the preprocessed ACE-2005 corpus in detail.

Table 1. Composition of the preprocessed ACE-2005 corpus.

# of entities per sentence (avg/max)	3.5/22
# of triples per sentence (avg/max)	1.5/11
# of entity types	7
# of relation types	7

NYT corpus [16]: This is a news corpus sampled from news articles published in the New York Times (NYT). The training data is automatically labeled using distant supervision. The NYT corpus was manually converted to a relation extraction dataset by Zheng et al. [17]. 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 6621 for testing. Table 2 shows the composition of the NYT corpus in detail.

Table 2. Composition of the NYT corpus.

# of entities per sentence (avg/max)	3.2/20
# of triples per sentence (avg/max)	1.7/26
# of entity types	3
# of relation types	25

Table 3 shows sample sentences and their tripe relations in the ACE-2005 corpus and the NYT corpus.

Table 3. Sample of the ACE-2005 corpus and the NYT corpus. PER-SOC: person-social.

Dataset	Sentence	Triple
ACE-2005	Do you travel to meet up with family or friends during the holidays?	{you, PER-SOC, family}, {you, PER-SOC, friends}
NYT	Clarence Charles Newcomer was born on Jan. 18, 1923, in the Lancaster County town of Mount Joy, Pa.	{Lancaster County,/location/location/contains, Mount Joy}, {Clarence Charles Newcomer,/people/person/place_of_birth, Mount Joy}

To evaluate the experimental results, we adopted the standard micro precision, recall, and F1 score:

$$\begin{aligned} \text{Recall} &= \frac{\#\ of\ correct\ predict}{\#\ of\ all\ triple\ in\ the\ dataset} \\ \text{Precision} &= \frac{\#\ of\ correct\ predict}{\#\ of\ all\ triple\ in\ the\ model\ predict} \\ \text{F1} - \text{score} &= \frac{2{*}Precision{*}Recall}{Precision{+}Recall} \end{aligned} \tag{8}$$

4.2. Experimental Results

In the first experiment, we evaluated the effectiveness of the multi-head attention in the dual pointer network decoder; the results are summarized in Table 4. The evaluation was performed using the ACE-2005 corpus.

Table 4. Performance for different attention mechanisms in the dual pointer network decoder.

Model	Recall	Precision	F1-Score
Single-head	0.800	0.759	0.779
Multi-head	0.832	0.787	0.808

In Table 4, single-head refers to a conventional attention mechanism proposed by Bahdanau et al. [11]. As shown in Table 4, the multi-head attention mechanism used in the proposed model demonstrated better performance than the single-head one. Then, using the ACE-2005 corpus, we evaluated the effectiveness of multi-head attention in the context and entity encoder; the results are summarized in Table 5.

Table 5. Performance for different attention mechanisms in the context and entity encoder.

Model	Recall	Precision	F1-Score
BIDAF-C2Q	0.819	0.766	0.792
BIDAF-C2Q&Q2C	0.821	0.792	0.806
Multi-head	0.832	0.787	0.808

In Table 5, BIDAF [18] refers to a machine-reading and comprehension (MRC) model based on a co-attention mechanism between a query and a context. C2Q and Q2C are referring to mean context-to-query attention and query-to-context attention used in the BIDAF model, respectively. As shown in Table 5, the multi-head attention mechanism used in the proposed model showed the best F1-score. The *p*-values of F1-scores between multi-head and the comparison models were from 5.0E-4 to 0.0039. This implies that the performance differences are statistically significant at the 0.05 level.

In the second experiment, we compared the proposed model with previous state-of-the-art models. Table 6 compares the performance of the proposed model and with other models for the ACE-2005 corpus.

Table 6. Performance comparison on the ACE-2005 corpus.

Model	Recall	Precision	F1-Score
SPTree [6]	0.54	0.57	0.56
FCM [19]	0.49	0.72	0.58
DYGIE [9]	0.57	0.64	0.60
Span-Level [20]	0.58	0.68	0.63
HRCNN [21]	-	-	0.74
Walk-Based [22]	0.60	0.70	0.64
Our model	0.83	0.79	0.81

In Table 6, SPTree [6] is a model that applies the dependency information between the entities. In FCM [19], handcrafted features are combined with word embeddings. DYGIE [9] dynamically generates spans between entities and spans' representations. Span-Level [20] jointly performs entity mention detection and relation extraction. HRCNN [21] is a hybrid model of CNN, RNN, and FNN. Walk-Based [22] is a graph-based neural network model. As shown in Table 6, the proposed model outperformed all models across all metrics. The *p*-values of F1-scores between the proposed model and the comparison models were from 5.81E-8 to 1.37E-5. This implies that the performance differences are statistically significant at the 0.001 level. Table 7 compares the performance of the proposed model with existing models for the NYT corpus.

Table 7. Performance comparisons on the NYT corpus.

Model	Recall	Precision	F1-Score
NovelTag [17]	0.414	0.615	0.495
MultiDecoder [2]	0.566	0.610	0.587
GraphRE [23]	0.600	0.639	0.619
Our model	0.820	0.749	0.783

In Table 7, NovelTag [17] is an end-to-end model that extracts entities and their relations based on a novel tagging scheme designed for relation extraction. MultiDecoder [2] is a Seq2Seq-based model that combines the entity and relation extraction using a decoder with a copy mechanism. GraphRE [23] is a joint model that extracts entities and their relationships using graph convolutional networks (GCN) [24]. As shown in Table 7, the proposed model outperformed all models. It is not reasonable to directly compare the proposed model with these models because it requires gold-labeled entities, while the other models automatically extract entities from sentences. Although direct comparison is unfair, the proposed model exhibited considerably better performance. If we adopt a state-of-the-art named entity tagger based on BERT [25] with F1-scores of 0.9 or more, the proposed model is expected to show F1-scores of 0.662 or more based on simple multiplication. Figure 5 describes the performances according to relation types.

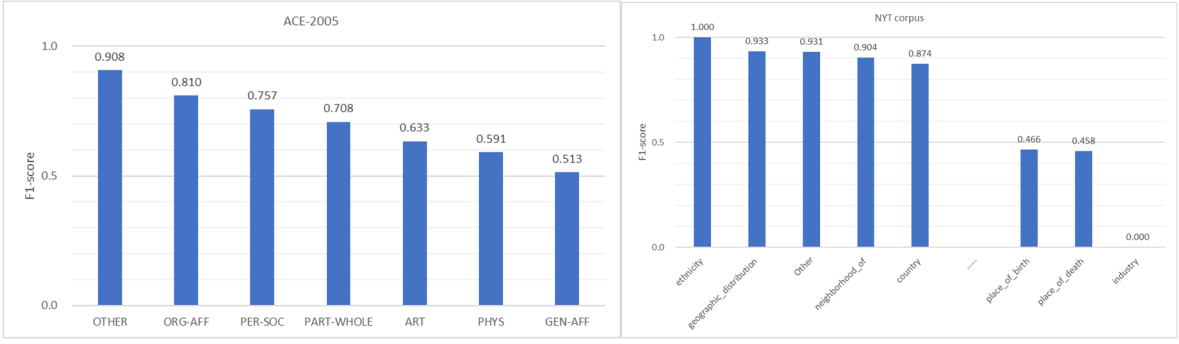


Figure 5. Performances per relation type.

As shown in the right graph of Figure 5, the proposed model obtained the F1-score of 1.0 for the relation type "ethnicity", but it obtained the F1-score of 0.0 for the relation type "industry". The imbalance of training data caused these performance differences. For example, the "industry" relation did not occur in the NYT training data at all.

The cases where the proposed model incorrectly extracted relations were also grouped in Table 8.

Table 8. Main reasons for errors in the ACE-2005 corpus (underline denotes incorrect results). ART: artifact; GEN-AFF: general entity-affiliation; PHYS: physical; PART-WHOLE: part of whole.

Input Sentence	Correct Relation	Predicted Relation
Iraqi forces responded with artillery fire	{Iraqi forces, ART, artillery} {Iraqi forces, GEN-AFF, Iraqi}	{Iraqi forces, PART-WHOLE, Iraqi} {Iraqi forces, GEN-AFF, Iraqi}
It is the first time they have had freedom of movement with cars and weapons since the start of the intifada	{they, ART, cars} {they, ART, weapons}	{they, ART, cars}
It was in northern Iraq today that an eight artillery round hit the site occupied by Kurdish fighters near Chamchamal	{Kurdish fighters, PHYS, the site} {the site, PHYS, Chamchamal} {Kurdish, GEN-AFF, Kurdish fighters} {the site, PART-WHOLE, northern Iraq}	{Kurdish fighters, PHYS, the site} {the site, PHYS, Chamchamal} {the site, PART-WHOLE, northern Iraq} {Kurdish fighters, ART, artillery}

Most incorrect predictions included cases where the decoders incorrectly pointed out subjects or objects, and these incorrect entities lead to incorrect relation names, as shown in the first and third sentences in Table 8. In some cases, the decoder did not point out subjects or objects. As a result, any triples in a sentence were not omitted, as shown in the second sentence.

5. Conclusions

We proposed a relation extraction model to find all possible relations among multiple entities in a sentence simultaneously. The proposed model is based on pointer networks with multi-head attention mechanisms. To extract all possible relations from a sentence, we modified a single decoder into 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. The results from the experiments with the ACE-2005 corpus and the NYT corpus confirmed that the proposed model shows an improvement in performance. Our future work will focus on an end-to-end model that directly extracts entities and their relations. In addition, we will focus on a method for improving performance using a large-scale language model like BERT [25].

Author Contributions: Conceptualization, H.K.; methodology, H.K.; software, S.P.; validation, S.P.; formal analysis, H.K.; investigation, H.K.; resources, S.P.; data curation, S.P.; writing—original draft preparation, S.P.; writing—review and editing, H.K.; visualization, H.K.; supervision, H.K.; project administration, H.K.; funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services). This work was also supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques).

Acknowledgments: We especially thank the members of the NLP laboratory at Kangwon National University and Konkuk University for their technical support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Choi, M.; Kim, H. Extraction of Instances with Social Relations for Automatic Construction of a Social Network. *J. KIISE Comput. Pract. Lett.* **2011**, *17*, 548–552. (In Korean)
2. Zeng, X.; Zeng, D.; He, S.; Liu, K.; Zhao, J. 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, Melbourne, Australia, 15–20 July 2018; pp. 506–514.
3. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation Classification via Convolutional Deep Neural Network. In Proceedings of the 24th International Conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
4. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Denver, CO, USA, 31 May–5 June 2015; pp. 39–48.
5. Yu, J.; Jiang, J. Pairwise Relation Classification with Mirror Instances and a Combined Convolutional Neural Network. In Proceedings of the 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 2366–2377.
6. Zhang, Y.; Qi, P.; Manning, C.D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2205–2215.
7. Miwa, M.; Bansal, N. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 1105–1116.
8. Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Positionaware Attention and Supervised Data Improve Slot Filling. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 35–45.
9. Luan, Y.; Wadden, D.; He, L.; Shah, A.; Ostendorf, M.; Hajishirzi, H. A General Framework for Information Extraction using Dynamic Span Graphs. In Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Minneapolis, MN, USA, 2–7 June 2019; pp. 3036–3046.
10. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2692–2700.
11. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the International Conference on Learning Representations 2015 (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
12. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
13. Park, S.; Jang, Y.; Park, K.; Kim, H. Named Entity Recognizer Using Gloval Vector and Convolutional Neural Network Embedding. *J. KITI Telecommun. Inf.* **2018**, *22*, 30–32. (In Korean)
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
15. ACE 2005 Multilingual Training Corpus. Available online: https://catalog.ldc.upenn.edu/LDC2006T06 (accessed on 31 May 2020).
16. Ren, X.; Wu, Z.; He, W.; Qu, M.; Voss, C.R.; Ji, H.; Abdelzaher, T.F.; Han, J. Cotype: Joint Extraction of Typed Entities and Relations with Knowledge Bases. In Proceedings of the International World Wide Web Conference, Perth, Australia, 3–7 April 2017; pp. 1015–1024.
17. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1227–1236.
18. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirz, H. Bi-Directional Attention Flow for Machine Comprehension. In Proceedings of the International Conference on Learning Representations 2017 (ICLR 2017), Toulon, France, 24–26 April 2017.

19. Gormley, M.R.; Yu, M.; Dredze, M. Improved Relation Extraction with Feature-Rich Compositional Embedding Models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1774–1784.
20. Dixit, K.; Onaizan, Y.A. Span-Level Model for Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5308–5314.
21. Kim, S.; Choi, S. Relation Extraction using Hybrid Convolutional and Recurrent Networks. In Proceedings of the Korea Computer Congress 2018 (KCC 2018), Jeju, Korea, 20–22 June 2018; pp. 619–621. (In Korean).
22. Christopoulou, F.; Miwa, M.; Ananiadou1, S. A Walk-based Model on Entity Graphs for Relation Extraction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 81–88.
23. Fu, T.J.; Li, P.H.; Ma, W.Y. GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1409–1418.
24. Kipf, T.; Welling, M. Semisupervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations 2017 (ICLR 2017), Toulon, France, 24–26 April 2017.
25. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.