

# Similarity-based Memory Enhanced Joint Entity and Relation Extraction

Witold Kościukiewicz<sup>1,2</sup>[0009-0001-0192-8850], Mateusz Wójcik<sup>1,2</sup>[0009-0008-0547-9467], Tomasz Kajdanowicz<sup>2</sup>[0000-0002-8417-1012], and Adam Gonczarek<sup>1</sup>

<sup>1</sup> Alphamoon Ltd., Wrocław, Poland

<sup>2</sup> Wrocław University of Science and Technology  
witold.kosciukiewicz@alphamoon.ai

**Abstract.** Document-level joint entity and relation extraction is a challenging information extraction problem that requires a unified approach where a single neural network performs four sub-tasks: mention detection, coreference resolution, entity classification, and relation extraction. Existing methods often utilize a sequential multi-task learning approach, in which the arbitral decomposition causes the current task to depend only on the previous one, missing the possible existence of the more complex relationships between them. In this paper, we present a multi-task learning framework with bidirectional memory-like dependency between tasks to address those drawbacks and perform the joint problem more accurately. Our empirical studies show that the proposed approach outperforms the existing methods and achieves state-of-the-art results on the BioCreative V CDR corpus.

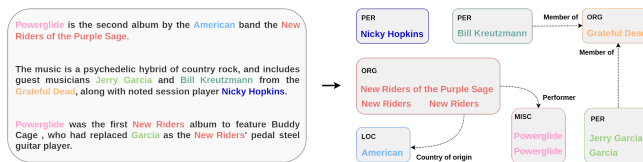
**Keywords:** Joint Entity and Relation Extraction · Document-level Relation Extraction · Multi-Task Learning

## 1 Introduction

In recent years text-based information extraction tasks such as named entity recognition have become more popular, which is closely related to the growing importance of transformer-based Large Language Models (LLMs). Such models are already used as a part of complex document information extraction pipelines. Even though important pieces of information are extracted, these pipelines still lack the ability to detect connections between them. The missing part, which is the relation classification task, was recognized as a significant challenge in recent years. The problem is even harder to solve if tackled with a multi-task method capable of solving both named entity recognition and relation classification task in a single neural network passage.

In this paper, we propose an approach to solve the multi-task problem of the joint document-level entity and relation extraction problem introduced with the DocRED dataset [16]. We follow the already existing line of research of learning

a single model to solve all four subtasks: mention detection, coreference resolution, entity classification, and relation extraction. The single model is trained to first detect the spans of text that are the entity mentions and group them into coreference clusters. Those entity clusters are then labeled with the correct entity type and linked to each other by relations. Figure 1 shows an example of a document from the DocRED dataset and a graph of labeled entity clusters that are expected as an output from the model. We introduce the bidirectional memory-like dependency between tasks to address the drawbacks of pipeline-based methods and perform the joint task more accurately.



**Fig. 1.** Visualization of the document-level joint entity and relation extraction task based on the example taken from DocRED. Entity mentions originating from different entity clusters are distinguished by color.

Our contribution can be summarized as follows: (1) we introduce a new approach that solves multi-task learning problems by improving the architecture of the previously proposed pipeline-based method, introducing the memory module to provide bi-directional dependency between tasks (2) we provide evaluation results, which show that our method outperforms the pipeline-based methods and achieves state-of-the-art results on the BioCreative V CDR corpus (3) we propose a novel similarity classifier module solving distance learning problem for document-level joint entity and relation classification serving as a starting point for future work. The code of our solution is available at [https://github.com/kosciukiewicz/similarity\\_based\\_memory\\_re](https://github.com/kosciukiewicz/similarity_based_memory_re).

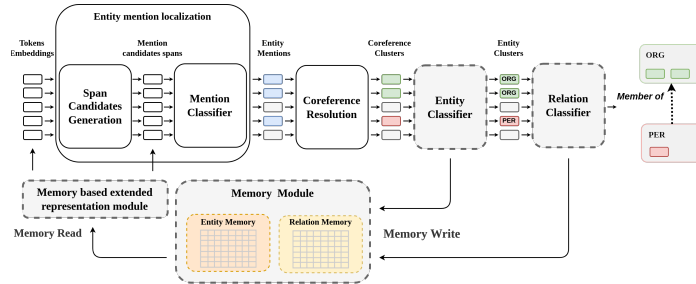
## 2 Related work

**Relation classification** The relation extraction task is commonly approached by using separately trained models for the Named Entity Recognition [10] to detect entities and then detect relations between them. The transformer-based architectures, pre-trained on large text corpora, such as BERT [4], have dominated the field i.e. Baldini Soares et al. [13] uses contextualized input embedding for the relation classification task.

**Joint entity and relation extraction** The early end-to-end solutions formulated task joint task as a sequence tagging based on BIO/BILOU scheme. These approaches include solving a table-filling problem proposed by Miwa et al. [11]. Several approaches tried to leverage multi-task learning abilities using attention-based [7] bi-directional LSTM sharing feature encoders between two

tasks to improve overall performance. The inability of the BIO/BILOU-based models to assign more than one tag to a token resulted in using the span-based method for joint entity and relation extraction proposed in Lee et al. [8]. Becoming a standard in recent years, this approach was further extended with graph-based methods like DyGIE++ [15] or memory models like TriMF [12] to enhance token span representation to an end-to-end approach for the joint task. **Document-level relation extraction** Although the DocRED [16] was originally introduced as relation classification benchmark, the opportunity arose to tackle a more complex joint entity and relation extraction pipelines consisting of mention detection, coreference resolution, entity classification, and relation classification. Since many relations link entities located in different sentences, considering inter-sentence reasoning is crucial to detect all information needed to perform all sub-tasks correctly. Eberts and Ulges [5] proposed JEREX - an end-to-end pipeline-based approach showing an advantage in joint training of all tasks rather than training each model separately. In recent work [2, 6], the problem is tackled using a sequence-to-sequence approach that outputs the extracted relation triples consisting of two related entities and relation type as text.

### 3 Approach



**Fig. 2.** The proposed architecture based on JEREX [5] enhanced with a feedback loop from entity and relation classifiers to the input of the mentioned classifier step. The novel part of the architecture is highlighted with a gray background and dashed borders.

Our document-level relation extraction framework is inspired by JEREX [5] which consists of four task-specific components: mention extraction ( $\mathcal{M}$ ), coreference resolution ( $\mathcal{C}$ ), entity extraction ( $\mathcal{E}$ ) and relation extraction ( $\mathcal{R}$ ). We change the original one-after-another pipeline architecture, introducing the memory module presented in Figure 2. The input representations of task-specific models are altered using the memory-based extended representation module that reads the memory using the Memory Read operation. The memory matrices  $\mathbf{M}_{\mathcal{E}}$  and  $\mathbf{M}_{\mathcal{R}}$ , are written by the entity and relation classifier, respectively. That feedback loop allows to share the information with previous steps extending their input by introducing a bi-directional dependency between tasks.

### 3.1 Memory reading

Similarly to TriMF [12], our approach memory reading is based on the attention mechanism that extends the input representation with the information read from memory. In our architecture, as shown in Figure 2 we extend both token embeddings  $\mathbf{X}_T$  and mention candidates span representations  $\mathbf{X}_S$ . For every input representation  $\mathbf{X}_i$  where  $i \in \{T, S\}$  and memory matrix  $\mathbf{M}_j$  where  $j \in \{\mathcal{E}, \mathcal{R}\}$ , the attention mechanism takes the representation  $\mathbf{X}_i \in \mathbb{R}^{n \times h}$  as keys and values, where  $n$  denotes the number of representations vectors and  $h$  is the embedding size.

As a query, the attention mechanism uses the memory matrix  $\mathbf{M}_j \in \mathbb{R}^{m \times s}$  where  $m$  denotes the number of memory slots and  $s$  is the size of the memory slot. To compute the attention weights vector  $\mathbf{a}_{i,j} \in \mathbb{R}^n$  we sum over the memory slots dimension as follows:

$$(\mathbf{a}_{i,j})^\top = \sum_k \text{softmax}(\mathbf{m}_j^{k,:} \mathbf{W}_{i,j}^{read} \mathbf{X}_i^\top) \quad (1)$$

where  $\mathbf{W}_{i,j}^{read} \in \mathbb{R}^{s \times h}$  is a learnable parameter matrix for the attention mechanism and  $\mathbf{m}_j^{k,:}$  is the  $k$ -th row of  $\mathbf{M}_j$ . The  $\mathbf{a}_{i,j}$  vector is then used to weight the  $\mathbf{X}_i$  to generate extended input representation  $\mathbf{X}'_{i,j}$ :

$$\mathbf{X}'_{i,j} = \text{diag}(\mathbf{a}_{i,j}) \mathbf{X}_i \quad (2)$$

For each input representation  $i$ , the memory reading operation creates two extended representations  $\mathbf{X}'_{i,\mathcal{E}}$  and  $\mathbf{X}'_{i,\mathcal{R}}$ , based on both memory matrices. The final extended representation is then calculated, using the element-wise mean of  $\mathbf{X}_i$ ,  $\mathbf{X}'_{i,\mathcal{E}}$  and  $\mathbf{X}'_{i,\mathcal{R}}$ :

### 3.2 Memory writing

Both memory matrices  $\mathbf{M}_{\mathcal{E}}$  and  $\mathbf{M}_{\mathcal{R}}$  store representations for entity and relation categories respectively. Values encoded in those matrices are written using the gradient of the loss function from the associated classifier – the entity classifier for  $\mathbf{M}_{\mathcal{E}}$  and the relation classifier for  $\mathbf{M}_{\mathcal{R}}$ . To make the stored representations more precise, the loss depends on the similarity between category embedding and the representation of the instance that belongs to that category according to the instance label. As a result, both entity and relation classifiers rely on similarity function  $S$  between input representation and suitable memory matrix. The probability distribution over entity types of entity  $e_i$  based on its representation vector  $\mathbf{x}_i^e$  is calculated as follows:

$$p(\mathbf{y}_e | e_i) = \text{softmax}(S(\mathbf{x}_i^e, \mathbf{M}_{\mathcal{E}})) \quad (3)$$

To get the existence probability over relation types for entity pair  $p_{i,j}$  represented by entity pair representation  $\mathbf{x}_{i,j}^p \in \mathbb{R}^h$  we used the sigmoid function:

$$p(\mathbf{y}_r | p_{i,j}) = \text{sigmoid}(S(\mathbf{x}_{i,j}^p, \mathbf{M}_{\mathcal{R}})) \quad (4)$$

We define  $S$  as bilinear similarity between instance representation  $\mathbf{x}$  and memory matrix  $\mathbf{M}$  as follows:

$$S(\mathbf{x}, \mathbf{M}) = S_{bilinear}(\mathbf{x}, \mathbf{M}; \mathbf{W}) = \mathbf{M}\mathbf{W}^\top \mathbf{x} \quad (5)$$

where  $\mathbf{W}$  is a learnable parameter matrix. For both entity and relation classifiers, separate learnable bilinear similarity weight matrices are used:  $\mathbf{W}_\mathcal{E}^{write} \in \mathbb{R}^{h_e \times s_\mathcal{E}}$  and  $\mathbf{W}_\mathcal{R}^{write} \in \mathbb{R}^{h_p \times s_\mathcal{R}}$  where  $h_e$  and  $h_p$  denote entity and entity pair representation sizes respectively.  $s_\mathcal{E}$  and  $s_\mathcal{R}$  denote the memory slot size of the entity and relation memory matrices. In our approach number of slots for the memory matrices are equal to the number of types in associated classifiers.

### 3.3 Training

Finally, our model is trained optimizing the joint loss  $\mathcal{L}^{joint}$  which contains the same four, sub-tasks related, loss  $\mathcal{L}^j$  weighted with fixed, task-related weight value  $\beta_j$  as in JEREX [5]:

$$\mathcal{L}^{joint} = \beta_{\mathcal{M}}\mathcal{L}^{\mathcal{M}} + \beta_{\mathcal{C}}\mathcal{L}^{\mathcal{C}} + \beta_{\mathcal{E}}\mathcal{L}^{\mathcal{E}} + \beta_{\mathcal{R}}\mathcal{L}^{\mathcal{R}}. \quad (6)$$

We also include the two-stage training approach proposed in TriMF [12], tuning the *memory warm-up proportion* during the hyperparameter search.

## 4 Experiments

**Datasets** We compare the proposed similarity-based memory learning framework to the existing approaches using DocRED [16] dataset which contains over 5000 human-annotated documents from Wikipedia and Wikidata. By design, DocRED dataset was intended to be used as a relation classification benchmark but its hierarchical annotations are perfectly suitable for joint task evaluation. For train, dev, and test split we follow the one provided in JEREX [5]. According to recent work [14], DocRED consists of a significant number of false negative examples. We used dataset splits provided with Re-DocRED [14] which is a re-annotated version of the DocRED dataset. We also provide results on one area-specific corpus annotated in a similar manner as DocRED - BioCreative V CDR [9] that contains 1500 abstracts from PubMed articles. Following the prior work [3, 6] we used the original train, dev, and test set split provided with the CDR corpus.

**Training** As a pretrained text encoder we used  $BERT_{BASE}$  [4]. For the domain-specific BioCreative V CDR dataset we used  $SciBERT_{BASE}$  [1] which was trained on scientific papers from Semantic Scholar. All classifiers and memory module parameters were initialized randomly. During training, we used batch size 2, AdamW optimizer with learning rate set to  $5e-5$  with linear warm-up for 10% of training steps and linear decay to 0. The stopping criteria for training were set to 20 epochs for all experiments.

**Evaluation** During the evaluation we used the strict scenario that assumes the prediction is considered correct only if all subtasks-related predictions are correct. We evaluated our method using micro-averaged F1-score. In Section 5 we reported F1-score for a final model evaluated on the test split. As the final model we selected the one that achieved the best F1-score measured on the dev split based on 5 independent runs using different random seeds. Our evaluation technique follows the one proposed in [5, 6].

**Hyperparameters** All hyperparameters like embedding sizes or multi-task loss weights were adopted from the original work [5] for better direct comparison. Our approach introduces new hyperparameters for which we conducted grid search on the dev split to find the best value. That includes hyperparameters such as *memory warm-up proportion* [12], memory read gradient, number and types of memory modules, and finally the size of memory slots.

## 5 Results

**Table 1.** Comparison (F1-score) of our method on the relation extraction task with existing end-to-end systems. \* - results from original publications.

Model	CDR	DocRED	Re-DocRED
JEREX [5]	42.88	*40.38	<b>45.56</b>
seq2rel [6]	*40.20	*38.20	—
<i>ours</i>	<b>43.75</b>	<b>40.42</b>	44.37
JEREX <sub>pre-training</sub>	-	41.27	45.81
<i>ours</i> <sub>pre-training</sub>	-	<b>41.75</b>	<b>45.96</b>

In Table 1 we present a comparison between our approach and existing end-to-end methods on 3 benchmark datasets for joint entity and relation extraction. The provided metric values show that our approach outperforms existing methods on CDR by about 0.9 percent points (pp.), achieving state-of-the-art results. Our method achieves similar results on DocRED and is outperformed by JEREX architecture on Re-DocRED dataset. We argue that the memory warm-up proportion value (0.4) is too small to properly initialize memories with accurate category representation. On the other hand increasing the memory warm-up steps leaves no time to properly train memory read modules. To address this issue we conducted experiments on pre-trained architecture using distantly annotated corpus of DocRED dataset to initialize memory matrices. We did the same pre-training for JEREX and the results show that our approach outperforms the original architecture by up to 0.48 pp. on both DocRED-based datasets.

For the direct comparison with the original architecture we evaluated our memory-enhanced approach with two relation classifiers modules proposed in [5]. Results presented in Table 2 show that our method improves the Global Relation Classifier (*GRC*) on every dataset by up to 1.70 pp. We also tested the performance of our method without the memory module - only with distance-based classifiers. Based on the results in Table 2, including a memory module

**Table 2.** Comparison (F1-score) between our architecture including memory reading module with JEREX using different relation classifier components - Global (*GRC*) and Multi-Instance (*MRC*). \* - results from original publications.

Model	CDR	DocRED	Re-DocRED
<i>GRC</i>			
JEREX [5]	42.04	*37.98	43.46
<i>ours</i>	42.04	37.76	43.64
<i>ours</i> + memory	<b>42.18</b>	<b>39.68</b>	<b>44.77</b>
<i>MRC</i>			
JEREX [5]	42.88	*40.38	<b>45.56</b>
<i>ours</i>	43.12	<b>40.68</b>	44.93
<i>ours</i> + memory	<b>43.75</b>	40.42	44.37

with a feedback loop between tasks, in most cases, improved the final results regardless of the *GRC* or *MRC* module.

## 6 Conclusions and future work

In this paper, we proposed a novel approach for multi-task learning for document-level joint entity and relation extraction tasks. By including memory-like extensions creating a feedback loop between the tasks, we addressed the issues present in the previous architectures. Empirical results show the superiority of our method in performance over other document-level relation extraction methods, achieving state-of-the-art results on BioCreative V CDR corpus. One of the possible directions for future work is further development of the memory module by using different memory read vectors for more meaningful input encoding in enhanced representation module or improving the content written to memory by replacing the bi-linear similarity classifier with different distance-based scoring functions or proposing a different method of writing to memory.

## 7 Acknowledgements

The research was conducted under the Implementation Doctorate programme of Polish Ministry of Science and Higher Education and also partially funded by Department of Artificial Intelligence, Wroclaw Tech and by the European Union under the Horizon Europe grant OMINO (grant number 101086321). It was also partially co-funded by the European Regional Development Fund within Measure 1.1. “Enterprise R&D Projects”, Sub-measure 1.1.1. “Industrial research and development by companies” as part of The Operational Programme Smart Growth 2014-2020, support contract no. POIR.01.01.01-00-0876/20-00.

## References

1. Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Lan-

- guage Processing and the 9th International Joint Conference on Natural Language Processing. ACL (2019)
2. Cabot, P.L.H., Navigli, R.: Rebel: Relation extraction by end-to-end language generation. In: Findings of the Association for Computational Linguistics: EMNLP (2021)
  3. Christopoulou, E., Miwa, M., Ananiadou, S.: Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. ACL (2019)
  4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2019)
  5. Eberts, M., Ulges, A.: An end-to-end model for entity-level relation extraction using multi-instance learning. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (2021)
  6. Giorgi, J., Bader, G., Wang, B.: A sequence-to-sequence approach for document-level relation extraction. In: Proceedings of the 21st Workshop on Biomedical Language Processing. pp. 10–25 (2022)
  7. Katiyar, A., Cardie, C.: 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 (2017)
  8. Lee, K., He, L., Lewis, M., Zettlemoyer, L.: End-to-end neural coreference resolution. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (2017)
  9. Li, J., Sun, Y., Johnson, R.J., Sciaky, D., Wei, C.H., Leaman, R., Davis, A.P., Mattingly, C.J., Wieggers, T.C., Lu, Z.: Biocreative v cdr task corpus: a resource for chemical disease relation extraction (2016)
  10. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. *IEEE Trans. on Knowl. and Data Eng.* (2022)
  11. Miwa, M., Sasaki, Y.: Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (2014)
  12. Shen, Y., Ma, X., Tang, Y., Lu, W.: A trigger-sense memory flow framework for joint entity and relation extraction. In: Proceedings of the web conference (2021)
  13. Soares, L.B., Fitzgerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: Distributional similarity for relation learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
  14. Tan, Q., Xu, L., Bing, L., Ng, H.T.: Revisiting docred—addressing the overlooked false negative problem in relation extraction. *arXiv preprint arXiv:2205.12696* (2022)
  15. Wadden, D., Wennberg, U., Luan, Y., Hajishirzi, H.: Entity, relation, and event extraction with contextualized span representations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (2019)
  16. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., Sun, M.: DocRED: A large-scale document-level relation extraction dataset. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)