Improving Relation Extraction by Sequence-to-sequence-based Dependency Parsing Pre-training

Masaki Asada[1]
[1]Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Japan
masaki.asada@aist.go.jp

Makoto Miwa[1,2]
[2]Toyota Technological Institute, Japan
makoto-miwa@toyota-ti.ac.jp

Abstract

Relation extraction is a crucial natural language processing task that extracts relational triplets from raw text. Syntactic dependencies information has shown its effectiveness for relation extraction tasks. However, in most existing studies, dependency information is used only for traditional encoder-only-based relation extraction, not for generative sequence-to-sequence (seq2seq)-based relation extraction. In this study, we propose a syntax-aware seq2seq pre-trained model for seq2seq-based relation extraction. The model incorporates dependency information into a seq2seq pre-trained language model by continual pre-training with a seq2seq-based dependency parsing task. Experimental results on two widely used relation extraction benchmark datasets show that dependency parsing pre-training can improve the relation extraction performance[1].

1 Introduction

Information extraction is the task of identifying both entities and their semantic relationships from raw texts. Recent studies have shown that generative language models can perform this task as a seq2seq task by outputting linearized strings encoding entity pairs and their relations (Paolini et al., 2021; Huguet Cabot and Navigli, 2021; Wadhwa et al., 2023). These methods achieved SOTA or near-SOTA results on several relation extraction benchmark datasets.

Tian et al. (2022) showed that employing dependency syntax information for pre-training is effective on the relation extraction in encoder-only models such as BERT (Devlin et al., 2019). He and Choi (2023) tackled the tasks of POS-tagging, constituency parsing, and dependency parsing with the seq2seq model, achieving SOTA performance.

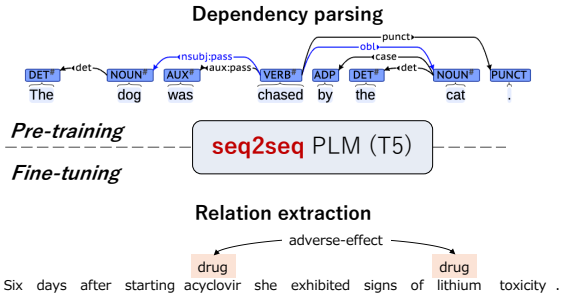

Figure 1: Overview of the syntax-aware seq2seq pre-trained model and its application to seq2seq-based end-to-end relation extraction.

However, there have been no methods that perform the dependency parsing task in the form of seq2seq as pre-training, and it has not been discussed whether dependency parsing is effective as a pre-training for performance improvement of seq2seq-based relation extraction tasks.

We propose a syntax-aware seq2seq pre-trained model for relation extraction. Specifically, we perform the seq2seq-based dependency parsing as continual pre-training from a publicly available checkpoint to incorporate dependency information into a seq2seq-based pre-trained language model. We investigate the effect of the pre-training with dependency parsing on downstream relation extraction tasks. Our contribution is two folds:

- We perform a seq2seq dependency parsing task as a continual pre-training to obtain a syntax-aware seq2seq pre-trained model.
- We show the effectiveness of the continual pre-training with a seq2seq dependency parsing task on two widely used relation extraction benchmark datasets: CONLL04 and ADE.

[1]The code is available on https://github.com/aistairc/DepParsingRE

Proceedings of the 31st International Conference on Computational Linguistics, pages 7099–7105
January 19–24, 2025.

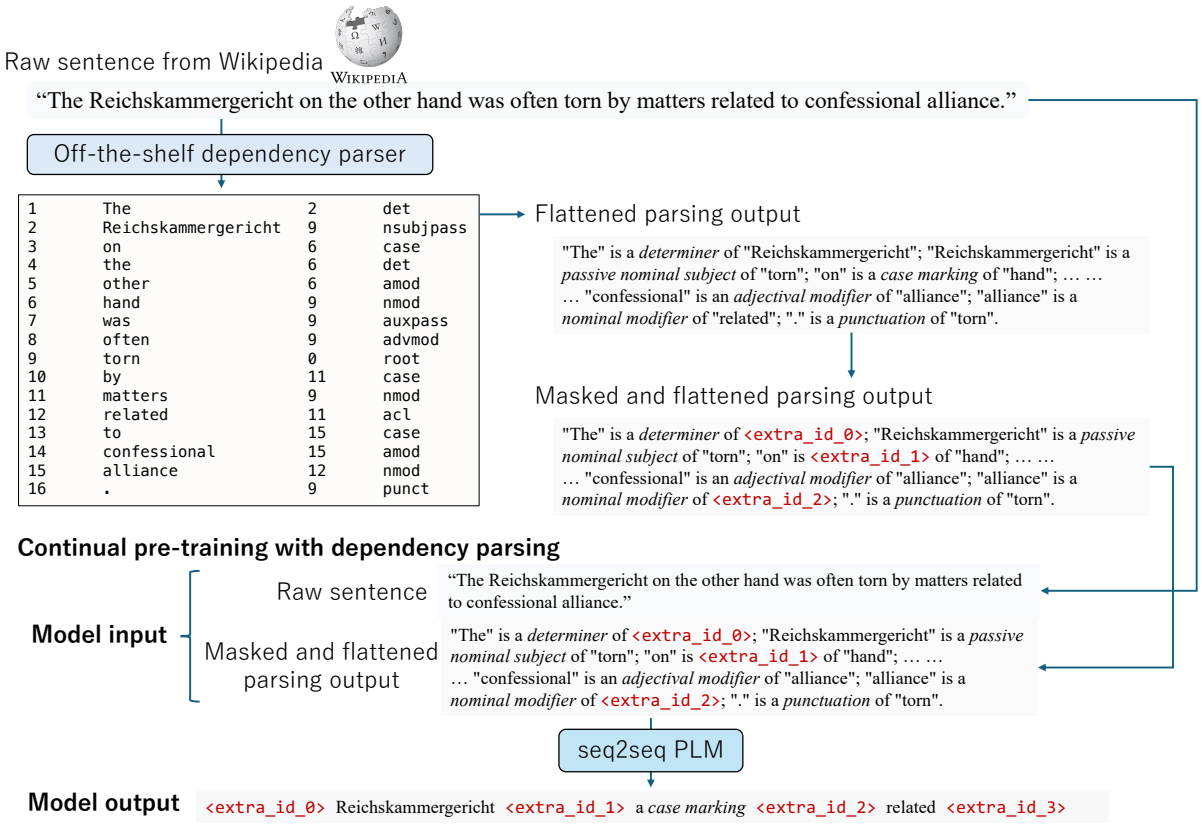

Figure 2: Pre-training with dependency parsing

2 Related Work

2.1 Seq2seq Relation Extraction

Several relation extraction approaches have been recently proposed to address the task using seq2seq generative models to output string encodings of target relational triples. Paolini et al. (2021) proposed a framework that formulated many structured prediction tasks, including relation extraction, relation classification, and semantic role labeling as seq2seq tasks where they decode outputs into structured information. Huguet Cabot and Navigli (2021) proposed REBEL, which extended this line of work by training BART (Lewis et al., 2020) specifically for relation extraction using a unique triplet linearization scheme. Wadhwa et al. (2023) investigated the use of large language models including GPT-3 (Brown et al., 2020) for relation extraction by training Flan-T5 (Chung et al., 2022) with Chain-of-Thought style explanations automatically generated by GPT-3. This method achieved new SOTA results.

2.2 Seq2seq Dependency Parsing

He and Choi (2023) aimed to unleash the potential of seq2seq models for sequence tagging and structure parsing, such as POS-tagging, constituency parsing, and dependency parsing, by proposing three novel linearization schemas and corresponding constrained decoding methods. Although seq2seq dependency parsing methods have been studied in recent years (Li et al., 2018; He and Choi, 2023), the dependency parsing task has not been used in the pre-training of seq2seq language models.

2.3 Dependency Tree Information and Relation Extraction

Several studies (Miwa and Bansal, 2016; Tsujimura et al., 2020) showed dependency tree substructure information is helpful for relation extraction tasks. Tian et al. (2022) showed that utilizing dependency syntax information for pre-training is effective on the relation extraction in encoder-only models. However, it has not been discussed whether dependency parsing as a pre-training is effective in the performance improvement of the seq2seq-based relation extraction task.

3 Method

In this study, we propose a novel seq2seq pre-trained model that captures the dependency structure of sentences for relation extraction by continual pre-training with a dependency parsing task. First, we apply an off-the-shelf dependency parser to raw sentences to obtain the word dependencies, and the output of the parser is converted into a flattened sequence in natural language. Then, we apply span masking to the resulting flattened sequence and perform a continual pre-training task to generate masked spans. Figure 2 shows the overview of our pre-training approach.

3.1 Dependency Parsing

We apply an off-the-shelf dependency parser to a large amount of raw text to perform sentence splitting and obtain dependency trees. Inspired by prior work (Tian et al., 2022), we convert the outputs of the dependency parser into a linearized sequence format that is closer to natural language to make the model easier to learn the structure of dependency trees in the pre-training stage. We express the flattened dependency parsing output by writing down the dependency targets and relations for all words into natural language, without using transitions by arc-standard system (Nivre, 2004) as shown in Table 1. This formatted sequence is created by applying the template *"x" is r of "y"* where y and r are dependency target word and relation of the word x. The output is finalized by joining all such sentences with a semicolon. For the expressions of the dependency relations, we follow the names defined by Universal Dependencies (Nivre et al., 2017), i.e., `nsubjpass` is converted into *passive nominal subject*.

3.2 Span Masking

We prepare the span-based masked data from the resulting dependency parsing outputs for continual pre-training so that seq2seq models can learn dependency structures. The description of dependency on each word is randomly masked according to the mask probability p. The parts that can be masked are the dependency target word, e.g., *"The" is a determiner of* ***<extra_id_0>*** or the dependency relation, e.g., *"chanceries" is* ***<extra_id_1>*** *of "became"*, where <*extra_id_**>> means a special masking token of T5 model. In the dependency parsing pre-training, the T5 model takes the concatenation of the raw sentence and the flattened parsing output with masks as input and predicts the masked spans. Since a single input usually contains multiple masked spans, the model predicts multiple spans together, e.g., <*extra_id_0*> *XXX* <*extra_id_1*> *YYY* <*extra_id_2*> as shown in Figure 2.

Raw sentence
The two chanceries became combined in 1502.
Flattened dependency parsing output
"The" is a determiner of "chanceries"; *"two" is a numeric modifier of "chanceries";* *"chanceries" is a nominal subject of "became";* *"became" is a root of "root";* *"combined" is an open clausal complement of "became";* *"in" is a case marking of "1502";* *"1502" is a modifier of nominal of "became";* *"." is a punctuation of "became".*
With span masks
"The" is a determiner of ***<extra_id_0>****;* *"two" is a numeric modifier of "chanceries";* *"chanceries" is* ***<extra_id_1>*** *of "became";* *"became" is a root of "root";* *"combined" is* ***<extra_id_2>*** *of "became";* *"in" is a case marking of "1502";* *"1502" is a nominal subject of of "became";* *"." is a punctuation of "became".*

Table 1: Input sentence, flattened dependency parsing output, and the output with span masks. Line breaks are inserted after the semicolons for ease of reading.

4 Experimental Settings

4.1 Pre-training

We used 1.9M input sentences from the English Wikipedia data dump with the version of `20220301.en` from Huggingface datasets[2]. The input sentences are truncated with a maximum sequence length of 256, and around 0.5B tokens are used for our pre-training. In obtaining dependency relations, we use Berkeley Neural Parser[3] (Kitaev and Klein, 2018) trained on English Penn Treebank (PTB) (Marcus et al., 1993) to automatically parse the Wikipedia data into constituency trees and then convert them into dependency trees by the Stanford Dependency converter[4] (Manning et al., 2014) fol-

[2] https://huggingface.co/datasets/wikipedia

[3] https://github.com/nikitakit/self-attentive-parser

[4] https://stanfordnlp.github.io/CoreNLP/

	Params	CONLL04			ADE		
		P	R	F	P	R	F [%]
TANL (Paolini et al., 2021)	220M	-	-	71.4†	-	-	80.61
TANL (multi data) (Paolini et al., 2021)	220M	-	-	72.6†	-	-	80.00
REBEL (Huguet Cabot and Navigli, 2021)	460M	75.22	69.01	71.97	80.80	82.62	81.69
REBEL + pre-training (Huguet Cabot and Navigli, 2021)	460M	75.59	75.12	**75.35**	81.45	83.07	82.21
Flan T5 (Wadhwa et al., 2023)	760M	-	-	75.28†	-	-	83.15
T5	760M	75.78	68.96	72.21	82.70	81.08	81.88
T5 + Dependency Parsing	760M	79.47	71.56	75.31	84.20	82.43	**83.31**

Table 2: Performance comparison of seq2seq-based models on relation extraction datasets. We exclude methods that utilize extremely large LMs, such as the GPT family, for fair comparison. † indicates the explicit use of train+validation set for training. P, R, and F indicate Precision, Recall, and micro-averaged F1-score, respectively.

lowing Tian et al. (2022). We set the probability of span masking p to 30%. The detailed settings of pre-training are shown in Appendix B.1.

4.2 Fine-tuning on Relation Extraction Tasks

We followed the same encoding/decoding schema as Paolini et al. (2021) for fine-tuning seq2seq-based relation extraction. We show an example of the input/output of the model below, where entities are enclosed in brackets, and their types and relationships are listed with '|' as a separator.

Input: Six days after starting acyclovir she exhibited signs of lithium toxicity.

Output: Six days after starting [acyclovir | drug] she exhibited signs of [[lithium | drug] toxicity | disease | effect = acyclovir | effect = lithium].

We trained and evaluated our model on the widely used relation extraction datasets. The statistics of datasets are shown in Appendix A. We used the micro-F1 score for the evaluation metrics.

CONLL04 CONLL04 (Roth and Yih, 2004) is composed of sentences from news articles, annotated with four entity types (person, organization, location, and other) and five relation types (kill, work for, organization based in, live in, and located in). To compare with previous work, we use the test split from Gupta et al. (2016), and the same validation set as Eberts and Ulges (2020), although we do not include the validation set at the final training.

ADE ADE (Gurulingappa et al., 2012) is a dataset on the biomedical domain, for which Adverse-Effects from drugs are annotated as pairs of drug and adverse-effect. The dataset provides 10-fold of train and test splits.

The detailed settings of fine-tuning are shown in Appendix B.2.

5 Results

5.1 Relation Extraction

Table 2 shows the performance comparison of seq2seq-based models on relation extraction. Regarding the existing baselines, the comparable performance between REBEL with 460M parameters and T5 with 760M parameters can be attributed to REBEL using BART-large as its LM backbone, given that BART-large and T5-large differ in their pre-training tasks and corpora.

CONLL04 We observed the additional pre-training with dependency parsing improves the F-score by 3.10 percentage points (pp) compared to the vanilla T5-large model. When compared to other methods, our T5 with dependency parsing pre-trained model showed comparable performance to REBEL + pre-training, which is a method that automatically created relation extraction datasets from Wikipedia and used it for pre-training, and Flan T5, which is a T5 model with additional instruction fine-tuning.

ADE We found that the dependency parsing pre-training improved the F-score of the vanilla T5 by 1.43 pp. Our proposed model showed the SOTA performance on the ADE dataset among seq2seq-based relation extraction methods.

5.2 Analysis

Table 3 compares the performance of relation extraction tasks under different pretraining settings.

	CONLL04			ADE		
	P	R	F	P	R	F [%]
T5	75.78	68.96	72.21	82.70	81.08	81.88
+ our pre-training (word masking)	77.80	68.95	73.11	82.94	81.08	82.00
+ our pre-training (word and tag masking)	79.47	71.56	**75.31**	84.20	82.43	**83.31**

Table 3: Performance comparison of T5 and our models. "word masking" refers to a setting where only the target words of dependencies are masked, while "word and tag masking" allows for masking either the target words of dependencies or the dependency relation tags.

The results indicate that the model achieves higher F-scores when both dependency target words and dependency relation tags are included as candidates for masking, compared to when only dependency target words are masked. This underscores the importance of learning both the target words and the dependency relations for the seq2seq model.

6 Conclusion

This work proposes the seq2seq-based dependency parsing task as a continual pre-training from T5 checkpoints for a syntax-aware pre-trained seq2seq model. We evaluate the obtained model on two widely used relation extraction benchmark datasets. Experimental results show that dependency parsing pre-training can improve the relation extraction performance, and the proposed method showed SOTA or SOTA-comparable results among seq2seq-based approaches.

For future work, we would like to investigate upstream tasks other than dependency parsing, such as part-of-speech tagging, named entity tagging, and constituency parsing, and their combinations for pre-training of seq2seq models.

Limitations

This paper shows that using the seq2seq-based dependency parsing as a continual pre-training task from T5 checkpoints is effective for the end-to-end relation extraction task. However, there is still room to be validated for pre-training with dependency parsing. First, we have not discussed how the continual pre-training affects the knowledge originally contained in the T5 model for general-purpose use other than relation extraction. Second, in this study, we adopted the T5-large model, which is a relatively small-size seq2seq PLM. We have not validated models with a larger size of nearly 10B.

Ethical Considerations

This paper utilizes standard publicly available pre-training corpus: Wikipedia, and relation extraction benchmarks: CONLL04 and ADE datasets. This paper solely focuses on technical improvements to seq2seq-based relation extraction tasks. Any sensitive data, conducting human studies, or developing applications that could raise ethical flags are not reported in this paper.

Acknowledgement

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *Proceedings of the 24th European Conference on Artificial Intelligence*.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics*, 45(5):885–892.

Han He and Jinho D. Choi. 2023. Unleashing the true potential of sequence-to-sequence models for sequence tagging and structure parsing. *Transactions of the Association for Computational Linguistics*, 11:582–599.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Makoto Miwa and Mohit Bansal. 2016. 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 (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57, Barcelona, Spain. Association for Computational Linguistics.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Improving relation extraction through syntax-induced pre-training with dependency masking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1875–1886, Dublin, Ireland. Association for Computational Linguistics.

Tomoki Tsujimura, Makoto Miwa, and Yutaka Sasaki. 2020. Automatic detection of important tokens on dependency trees for relation classification. *The Association for Natural Language Processing*.

	Entity Types	Relation Types	# of relation triplets Train	Val.	Test
CONLL04	4	5	922	231	288
ADE	2	1	4,272	-	-

Table 4: Statistics of the two relation extraction datasets

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

A Dataset Statistics

Statistics of the relation extraction datasets are reported in Table 4. The ADE dataset provides 10-fold of train and test splits.

B Implementation Details

B.1 Pre-training

Pre-training through dependency parsing is conducted starting from the parameters of the released T5-large model, using distributed data-parallel training with 16 NVIDIA V100 GPUs. The total batch size is 32. All source texts and target texts are padded to maximum lengths of 256 and 128 respectively. Adafactor (Shazeer and Stern, 2018) was adopted as the optimizer, with the learning rate set to 1e-04. The learning rate linearly decays after 1,000 steps of warm-up.

B.2 Fine-tuning

For fine-tuning, we followed the seq2seq-based relation extraction approach by Paolini et al. (2021), training the model that underwent pre-training through dependency parsing. For both CONLL04 and ADE datasets, the source text is padded to a maximum length of 256, and training was conducted using 8 NVIDIA A100 GPUs. The total batch size is 80. The learning rate is set to 7e-03 for CONLL04 and 5e-03 for ADE, with the number of training epochs set to 40 for CONLL and 15 for ADE. For both datasets, the warm-up period is set to the first 10% of the total training steps, after which the learning rate linearly decays. During inference, a beam size of 5 is adopted.