Silver Syntax Pre-training for Cross-Domain Relation Extraction

Elisa Bassignana● **Filip Ginter**💬 **Sampo Pyysalo**💬
Rob van der Goot● **Barbara Plank**●▲
●Department of Computer Science, IT University of Copenhagen, Denmark
💬TurkuNLP, Department of Computing, University of Turku, Finland
▲MaiNLP, Center for Information and Language Processing, LMU Munich, Germany
elba@itu.dk

Abstract

Relation Extraction (RE) remains a challenging task, especially when considering realistic out-of-domain evaluations. One of the main reasons for this is the limited training size of current RE datasets: obtaining high-quality (manually annotated) data is extremely expensive and cannot realistically be repeated for each new domain. An intermediate training step on data from related tasks has shown to be beneficial across many NLP tasks. However, this setup still requires supplementary annotated data, which is often not available. In this paper, we investigate intermediate pre-training specifically for RE. We exploit the affinity between syntactic structure and semantic RE, and identify the syntactic relations which are closely related to RE by being on the shortest dependency path between two entities. We then take advantage of the high accuracy of current syntactic parsers in order to automatically obtain large amounts of low-cost pre-training data. By pre-training our RE model on the relevant syntactic relations, we are able to outperform the baseline in five out of six cross-domain setups, *without* any additional annotated data.

1 Introduction

Relation Extraction (RE) is the task of extracting structured knowledge, often in the form of triplets, from unstructured text. Despite the increasing attention this task received in recent years, the performance obtained so far are very low (Popovic and Färber, 2022). This happens in particular when considering realistic scenarios which include out-of-domain setups, and deal with the whole task—in contrast to the simplified Relation Classification which assumes that the correct entity pairs are given (Han et al., 2018; Baldini Soares et al., 2019; Gao et al., 2019). One main challenge of RE and other related Information Extraction tasks is the "domain-specificity": Depending on the text domain, the type of information to extract changes. For example, while in the news domain we can find entities like *person* and *city*, and relations like *city of birth* (Zhang et al., 2017), in scientific texts, we can find information about *metrics*, *tasks* and *comparisons* between computational models (Luan et al., 2018). While high-quality, domain-specific data for fine-tuning the RE models would be ideal, as for many other NLP tasks, annotating data is expensive and time-consuming.[1] A recent approach that leads to improved performance on a variety of NLP tasks is intermediate task training. It consists of a step of training on one or more NLP tasks between the general language model pre-training and the specific end task fine-tuning (STILT, Supplementary Training on Intermediate Labeled-data Tasks; Phang et al., 2018). However, STILT assumes the availability of additional high quality training data, annotated for a related task.

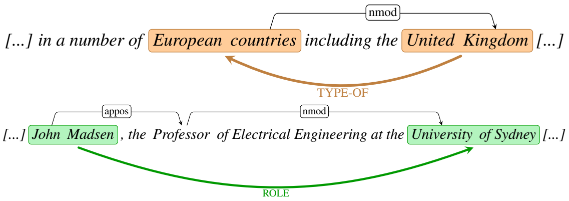
nmod
[...] in a number of European countries including the United Kingdom [...]
TYPE-OF
appos
nmod
[...] John Madsen , the Professor of Electrical Engineering at the University of Sydney [...]
ROLE

Figure 1: **Syntactic and Semantic Structures Affinity.** Shortest dependency path (above), and semantic relation (below) between two semantic entities.

In this paper, we explore intermediate pre-training specifically for cross-domain RE and look for alternatives which avoid the need of external manually annotated datasets to pre-train the model on. In particular, we analyze the affinity between syntactic structure and semantic relations, by considering the shortest dependency path between two entities (Bunescu and Mooney, 2005; Fundel et al., 2006; Björne et al., 2009; Liu et al., 2015). We replace the traditional intermediate pre-training step

[1] For example, Bassignana and Plank, 2022 report a cost of 19K USD (≈ 1$ per annotated relation) and seven months of annotation work for an RE dataset including 5.3K sentences.

Findings of the Association for Computational Linguistics: ACL 2023, pages 6984–6993
July 9-14, 2023 ©2023 Association for Computational Linguistics

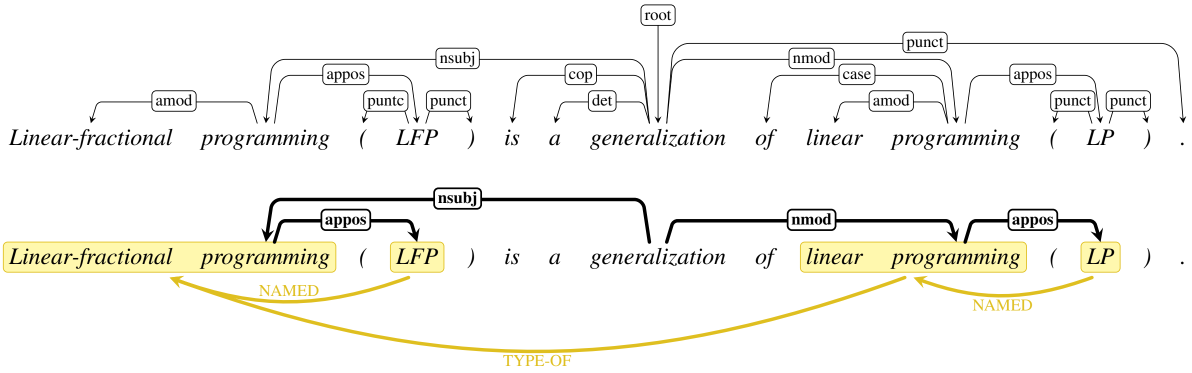
root
nsubj
punct
nmod
appos
cop
case
appos
amod
puntc
punct
det
amod
punct
punct
Linear-fractional programming (LFP) is a generalization of linear programming (LP) .
nsubj
appos
nmod
appos
Linear-fractional programming (LFP) is a generalization of linear programming (LP) .
NAMED
NAMED
TYPE-OF

Figure 2: **Pre-training Example**. Given the dependency tree (above), we filter in for pre-training only the UD labels which are on the shortest dependency path between two semantic entities (below).

on additional annotated data, with a *syntax pre-training* step on silver data. We exploit the high accuracy of current syntax parsers, for obtaining large amount of low-cost pre-training data. The use of syntax has a long tradition in RE (Zhang et al., 2006; Qian et al., 2008; Nguyen et al., 2009; Peng et al., 2015). Recently, work has started to infuse syntax during language model pre-training (Sachan et al., 2021) showing benefits for RE as well. We instead investigate dependency information as silver data in intermediate training, which is more efficient. To the best of our knowledge, the use of syntax in intermediate pre-training for RE is novel. We aim to answer the following research questions: ① Does syntax help RE via intermediate pre-training (fast and cheap approach)? and ② How does it compare with pre-training on additional labeled RE data (expensive)? We release our model and experiments.[2]

2 Syntax Pre-training for RE

Syntactic parsing is a structured prediction task aiming to extract the syntactic structure of text, most commonly in the form of a tree. RE is also a structured prediction task, but with the aim of extracting the semantics expressed in a text in the form of triplets—entity A, entity B, and the semantic relation between them.[3] We exploit the affinity of these two structures by considering the shortest dependency path between two (semantic) entities (see Figure 1).

The idea we follow in this work is to pre-train an RE baseline model over the syntactic relations—Universal Dependency (UD) labels—which most frequently appear on the shortest dependency paths between two entities (black bold arrows in Figure 2). We assume these labels to be the most relevant with respect to the final target task of RE. In order to feed the individual UD relations into the RE baseline (model details in Section 3.1) we treat them similarly as the semantic connections. In respect to Figure 2, we can formalize the semantic relations as the following triplets:

- `NAMED(LFP,Linear-fractional programming)`
- `TYPE-OF(linear programming,Linear-fractional programming)`
- `NAMED(LP,linear programming)`.

Accordingly, we define the syntax pre-training instances as:

- `appos(programming,LFP)`
- `nsubj(generalization,programming)`
- `nmod(generalization,programming)`
- `appos(programming,LP)`.

In the next section we describe the detailed training process.

3 Experiments

3.1 Setup

Data In order to evaluate the robustness of our method over out-of-domain distributions, we experiment with CrossRE (Bassignana and Plank, 2022),[4] a recently published multi-domain dataset. CrossRE includes 17 relation types spanning over six diverse text domains: news, politics, natural science, music, literature and artificial intelligence (AI). The dataset was annotated on top of a Named

[2] `https://github.com/mainlp/syntax-pre-training-for-RE`

[3] In this project, we follow previous work, and assume gold entities, leaving end-to-end RE for future work.

[4] Released with a GNU General Public License v3.0.

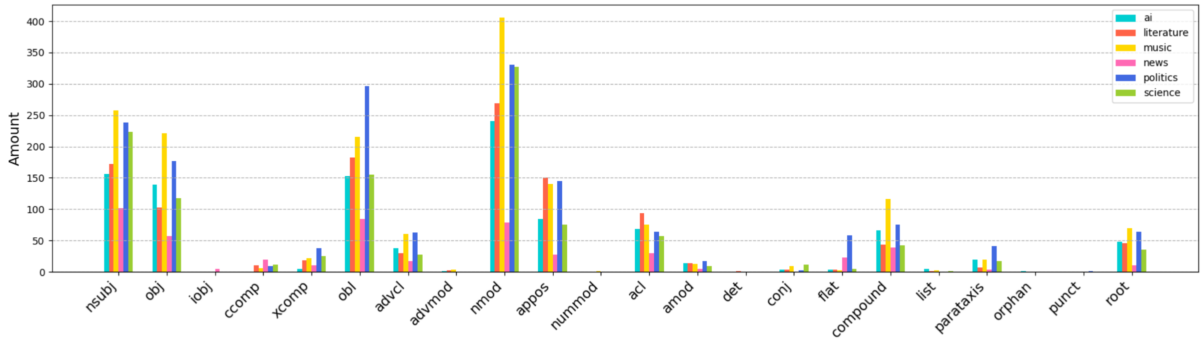

Figure 3: **UD Label Distribution Over the Shortest Dependency Paths.** Statistics of the UD labels which are on the shortest dependency path between two entities over the six train sets of CrossRE (Bassignana and Plank, 2022).

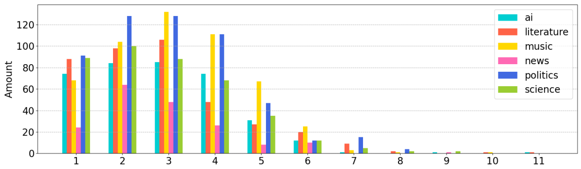

Figure 4: **Shortest Dependency Path Length.** Statistics of the shortest dependency path length between two entities over the train sets of CrossRE (Bassignana and Plank, 2022).

Entity Recognition dataset—CrossNER (Liu et al., 2021)—which comes with an unlabeled domain-related corpora.[5] We used the latter for the *syntax pre-training* phase.

UD Label Selection In order to select the UD labels which most frequently appear on the shortest dependency path between two semantic entities, we parsed the training portions of CrossRE. Our analysis combines RE annotations and syntactically parsed data. We observe that the syntactic distance between two entities is often higher than one (see Figure 4), meaning that the shortest dependency path between two entities includes multiple dependencies—in the examples in Figure 1, the one above has distance one, the one below has distance two. However, the shortest dependency paths contain an high frequency of just a few UD labels (see Figure 3) which we use for *syntax pre-training*: `nsubj`, `obj`, `obl`, `nmod`, `appos`. See Appendix A for additional data analysis.

Model Our RE model follows the current state-of-the-art architecture by Baldini Soares et al., 2019 which augments the sentence with four entity markers e_1^{start}, e_1^{end}, e_2^{start}, e_2^{end} before feeding it into a pre-trained encoder (BERT; Devlin et al., 2019). The classification is then made by a 1-layer feed-forward neural network over the concatenation of the start markers $[\hat{s}_{e_1^{start}}, \hat{s}_{e_2^{start}}]$. We run our experiments over five random seeds and report the average performance. See Appendix B for reproducibility and hyperparameters settings of our model.

Training The training of our RE model is divided into two phases. In the first one—which we are going to call *syntax pre-training*—we use the unlabeled corpora from CrossNER for pre-training the baseline model over the *RE-relevant* UD labels. To do so, ① we sample an equal amount of sentences from each domain[6] (details in Section 4), and ② use the MaChAmp toolkit (van der Goot et al., 2021) for inferring the syntactic tree of each of them. We apply an additional sub-step for disentangling the `conj` dependency, as illustrated in Appendix C. Then, ③ we filer in only the `nsubj`, `obj`, `obl`, `nmod`, and `appos` UD labels and ④ feed those connections to the RE model (as explained in the previous section). Within the RE model architecture described above, each triplet corresponds to one instance. In this phase, in order to assure more variety, we randomly select a maximum of five triplets from each pre-train sentence.

In the second training phase—the *fine-tuning* one—we replace the classification head (i.e. the feed-forward layer) with a new one, and individually train six copies of the model over the six train sets of CrossRE. Note that the encoder is fine-tuned in both training phases. Finally, we test each model on in- and out-of-domain setups.

[5] Released with an MIT License.

[6] Regarding the news domain, which does not have a corresponding unlabeled corpus available, we sampled from the train set of CrossNER which is not included in CrossRE.

					TEST			
	TRAIN	news	politics	science	music	literature	AI	avg.
BASELINE	news	10.98	1.32	1.24	1.01	1.49	1.42	**2.91**
	politics	16.07	11.30	6.74	7.24	7.29	5.54	9.03
	science	6.54	5.95	8.57	7.13	6.65	7.29	7.02
	music	3.99	9.91	9.22	19.01	10.43	8.53	10.18
	literature	11.30	9.60	9.79	12.49	17.17	9.79	11.69
	AI	6.58	7.42	11.03	7.11	6.15	15.57	8.98
SYNTAX	news	6.67	1.15	0.72	0.61	1.13	0.75	1.84
	politics	13.72	12.09	7.47	7.15	7.78	6.24	**9.08**
	science	8.46	7.08	8.69	8.19	7.52	8.91	**8.14**
	music	3.35	10.65	9.35	18.63	11.62	10.34	**10.66**
	literature	11.85	9.84	10.35	13.58	18.64	9.94	**12.37**
	AI	8.87	8.59	11.87	8.29	7.68	15.93	**10.21**
SCIERC	news	11.88	2.30	2.09	1.13	1.82	2.16	3.56
	politics	14.25	13.55	6.52	7.12	7.42	7.07	9.32
	science	8.27	10.31	13.59	9.09	7.78	11.11	10.03
	music	5.41	11.84	10.85	21.39	12.26	11.22	12.16
	literature	12.36	8.05	8.87	13.13	16.44	9.40	11.37
	AI	11.00	10.12	14.03	8.93	8.50	18.89	11.91

Table 1: **Performance Scores.** Macro-F1 scores of the baseline model, compared with the proposed *syntax pre-training* approach, and—as comparison—with the traditional pre-training over the manually annotated SciERC dataset (Luan et al., 2018).

3.2 Results

Table 1 reports the results of our cross-domain experiments in terms of Macro-F1. We compare our proposed approach which adopts *syntax pre-training* with the zero-shot baseline model.[7] Five out of six models outperform the average of the baseline evaluation, including in- and out-of-domain assessments. The average improvement—obtained without any additional annotated RE data—is 0.71, which considering the low score range given by the challenging dataset (with limited train sets, see dataset size in Appendix D), and the cross-domain setup, is considerable. The model fine-tuned on the news domain is the only one not outperforming the baseline. However, the performance scores on this domain are already extremely low for the baseline, because news comes from a different data source with respect to the other domains, has a considerable smaller train set, and present a sparse relation types distribution, making it a bad candidate for transferring to other domains (Bassignana and Plank, 2022).

As comparison, we report the scores obtained with the traditional intermediate pre-training which includes additional annotated data. We pre-train the language encoder on SciERC (Luan et al., 2018), a manually annotated dataset for RE. SciERC contains seven relation types, of which three overlap with the CrossRE relation set. In this setup, the improvement over the baseline includes the news, but not the literature domain. Nevertheless, while the gain is on average slightly higher with respect to the proposed *syntax pre-training* approach, it comes at a much higher annotation cost.

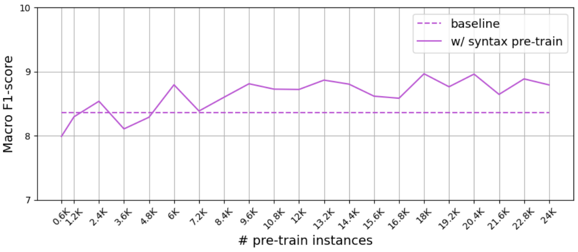

Figure 5: **Pre-train Data Quantity Analysis.** Average (dev) performance of the six models when pre-trained over increasing amounts of syntactic instances.

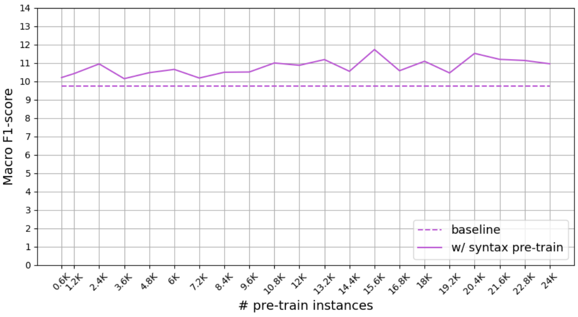

Figure 6: **Per-Domain Pre-train Data Quantity Analysis.** Individual (dev) performance of the model fine-tuned on AI when pre-trained over increasing amounts of syntactic instances.

4 Pre-training Data Quantity Analysis

We inspect the optimal quantity of syntactic data to pre-train our RE model on by fine-tuning this hyperparameter over the dev sets of CrossRE. The plot in Figure 5 reports the average performance of the six models when pre-trained on increasing amounts of syntactic dependencies.[8] Starting from 8.4K instances onward, the performance stabilizes above the baseline. We select the peak (20.4K, albeit results are similar between 18-20.4K) for reporting our test set results in Table 1. While we are interested in the robustness of our method across multiple domains, and therefore consider the average (Figure 5), domain-optima could be achieved by examining individual domain performance. As example, we report in Figure 6 the plot relative to the model fine-tuned on AI, which is the one obtain-

[7] While utilizing the model implementation by Bassignana and Plank, 2022, our score range is lower because we include the *no-relation* case, while they assume gold entity pairs.

[8] Pre-training performance in Appendix E.

ing the highest gain. The model fine-tuned on AI generally gains a lot from the *syntax pre-training* step, with its peak on 15.6K pre-training instances.

5 Conclusion

We introduce *syntax pre-training* for RE as an alternative to the traditional intermediate training which uses additional manually annotated data. We pre-train our RE model over silver UD labels which most frequently connect the semantic entities via the shortest dependency path. We test the proposed method over CrossRE and outperform the baseline in five out of six cross-domain setups. Pre-training over a manually annotated dataset, in comparison, only slightly increases our scores in five out of six evaluations, but at a much higher cost.

Limitations

While we already manage to outperform the baseline, the pre-training data quantity is relatively small (∼20K instances). Given the computational cost of training 30 models—six train sets, over five random seeds each—and testing them within in- and cross- domain setups, we break the inspection of the optimal pre-training data amount at 24K instances. However we do not exclude that more pre-training instances would be even more beneficial for improving even more over the baseline.

Related to computation cost constrains, we test our *syntax pre-training* approach over one set of UD labels only (`nsubj`, `obj`, `obl`, `nmod`, `appos`). Different sets could be investigated, e.g. including `acl` and `compound`, which present a lower, but still considerable amount of instances (see Figure 3).

Finally, while approaching RE by assuming that the gold entities are given is a common area of research, we leave for future work the inspection of the proposed method over end-to-end RE.

Acknowledgments

We thank the NLPnorth and the MaiNLP groups for feedback on an earlier version of this paper, and TurkuNLP for hosting EB for a research stay.

EB and BP are supported by the Independent Research Fund Denmark (Danmarks Frie Forskningsfond; DFF) Sapere Aude grant 9063-00077B. BP is in parts supported by the European Research Council (ERC) (grant agreement No. 101043235). FG and SP were supported by the Academy of Finland.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Elisa Bassignana and Barbara Plank. 2022. CrossRE: A cross-domain dataset for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3592–3604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 10–18, Boulder, Colorado. Association for Computational Linguistics.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2006. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 285–290, Beijing, China. Association for Computational Linguistics.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13452–13460.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1378–1387, Singapore. Association for Computational Linguistics.

Yifan Peng, Samir Gupta, Cathy Wu, and Vijay Shanker. 2015. An extended dependency graph for relation extraction in biomedical texts. In *Proceedings of BioNLP 15*, pages 21–30, Beijing, China. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Nicholas Popovic and Michael Färber. 2022. Few-shot document-level relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5733–5746, Seattle, United States. Association for Computational Linguistics.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 697–704, Manchester, UK. Coling 2008 Organizing Committee.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multitask learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 288–295, New York City, USA. Association for Computational Linguistics.

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*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A UD Analysis for RE

We inspect the same statistics as Figure 3 and Figure 4—UD labels on the shortest dependency paths, and shortest dependency path lengths respectively—but instead of at domain level, at semantic relation type level. Table 2 and Table 3 report this analysis, revealing similar trends over the 17 types.

B Reproducibility

We report in Table 4 the hyperparameter setting of our RE model (see Section 3.1). All experiments were ran on an NVIDIA® A100 SXM4 40 GB GPU and an AMD EPYC™ 7662 64-Core CPU. Within this computation infrastructure the baseline converges in $\sim$ 7 minutes. The the *syntax pre-training* step takes $\sim$ 10 minutes, to which we have to add $\sim$ 7 minutes in order to obtain the complete training time.

We train MaChAmp v0.4 on the English Web Treebank v2.10 with XLM-R large (Conneau et al., 2020) as language model with all default hyperparameters of MaChAmp.

		rel-to	artifact	cause-eff	compare	gen-aff	named	opposite	origin	part-of	physical	role	social	temporal	topic	type-of	usage	win-def
Universal Dependencies	nsubj	89	106	2	12	120	54	61	53	75	115	248	33	54	10	18	30	68
	obj	78	51	1	6	76	36	48	41	83	55	129	9	48	17	14	37	86
	iobj	0	0	0	0	0	1	0	0	0	1	3	0	0	0	0	0	0
	ccomp	5	7	0	4	7	10	8	2	2	2	9	0	0	0	0	0	0
	xcomp	6	9	0	3	15	5	5	9	5	11	17	1	16	3	1	2	10
	obl	88	62	5	14	92	53	25	44	77	202	224	19	121	17	26	6	11
	advcl	10	9	4	8	47	21	19	10	18	14	41	3	15	2	6	7	2
	advmod	0	3	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
	nmod	100	140	2	12	181	57	47	58	148	276	386	29	72	43	35	19	48
	appos	26	89	0	2	85	108	11	23	41	72	112	9	12	6	6	1	20
	nummod	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	acl	40	24	0	0	39	30	10	25	48	33	74	0	11	24	2	13	15
	amod	5	1	0	2	31	5	3	3	5	2	3	0	3	2	0	3	4
	det	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	conj	1	4	0	0	3	1	0	1	2	3	11	0	1	1	0	0	0
	flat	2	3	0	0	1	12	8	0	2	11	37	8	7	1	0	0	3
	compound	29	24	0	5	70	27	5	7	54	53	57	2	9	2	5	10	22
	list	0	1	0	0	2	2	0	0	0	0	2	0	2	0	0	0	0
	parataxis	5	7	0	0	30	14	0	0	14	5	17	1	8	1	5	0	1
	orphan	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	punct	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2: **UD Label Distribution Over the Shortest Dependency Paths per Relation Type.** Statistics of the UD labels which are on the shortest dependency path between two entities divided by the 17 relation types of CrossRE (Bassignana and Plank, 2022).

		related-to	artifact	cause-eff	compare	gen-aff	named	opposite	origin	part-of	physical	role	social	temporal	topic	type-of	usage	win-def
Shortest Dependency Path Length	1	3	43	0	0	73	92	1	19	26	72	64	0	5	7	17	0	12
	2	29	43	0	3	59	23	25	28	51	85	127	10	41	6	11	6	31
	3	36	71	2	1	33	14	21	29	57	75	136	17	30	14	3	14	34
	4	42	24	2	4	52	20	13	12	37	41	104	7	28	6	17	12	17
	5	17	12	0	5	36	12	14	8	18	28	33	5	10	8	2	4	3
	6	9	7	0	4	18	6	4	5	8	12	10	0	2	1	2	1	2
	7	4	0	0	0	4	3	0	0	2	6	4	0	5	0	0	0	5
	8	0	0	0	0	0	3	1	0	0	1	1	0	3	0	0	0	0
	9	0	1	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0
	10	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0

Table 3: **Shortest Dependency Path Length per Relation Type.** Statistics of the shortest dependency path length between two semantic entities divided by the 17 relation types of CrossRE (Bassignana and Plank, 2022).

Parameter	**Value**
Encoder	`bert-base-cased`
Classifier	1-layer FFNN
Loss	Cross Entropy
Optimizer	Adam optimizer
Batch size	12, 24
Learning rate	$1e^{-5}$ (pre-train)
Learning rate	$2e^{-5}$ (fine-tuning)
Seeds	4012, 5096, 8878, 8857, 9908

Table 4: **Hyperparameters Setting.** Model details for reproducibility of the baseline.

C Handling of `Conj`

In UD, the first element in a conjuncted list governs all other elements of the list via a `conj` dependency and represents the list syntactically w.r.t. the remainder of the sentence. CrossRE (Bassignana and Plank, 2022) relations, on the other hand, directly link the two entities involved in the semantic structure. To account for this difference, we propagate the conjunction dependencies in order to reflect the semantic relations, as shown in Figure 7.

	SENTENCES				RELATIONS			
	train	dev	test	**tot.**	train	dev	test	**tot.**
news	164	350	400	914	175	300	396	871
politics	101	350	400	851	502	1,616	1,831	3,949
science	103	351	400	854	355	1,340	1,393	3,088
music	100	350	399	849	496	1,861	2,333	4,690
literature	100	400	416	916	397	1,539	1,591	3,527
AI	100	350	431	881	350	1,006	1,127	2,483
tot.	668	2,151	2,446	**5,265**	2,275	7,662	8,671	**18,608**

Table 5: **CrossRE Statistics.** Number of sentences and number of relations for each domain of CrossRE (Bassignana and Plank, 2022).

D CrossRE Size

We report in Table 5 the dataset statistics of CrossRE (Bassignana and Plank, 2022) including the number of sentences and of relations.

E Syntax Pre-training Performance

Figure 8 reports the performance of the RE model during the *syntax pre-training* phase, over increasing amounts of pre-training dependency instances. The scores are computed on a set including 600 sentences (100 per domain) not overlapping with the train set used in the syntax pre-training phase.

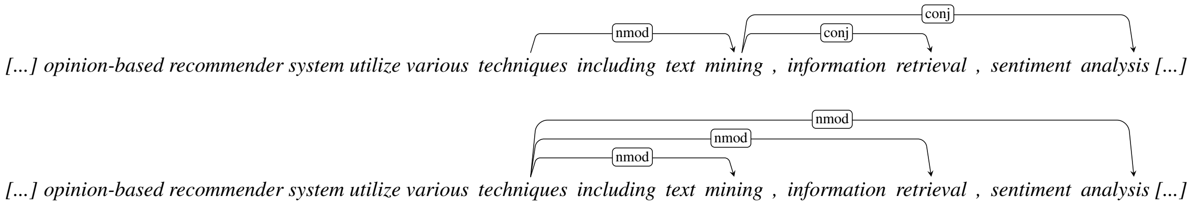
conj
nmod
conj
[...] opinion-based recommender system utilize various techniques including text mining , information retrieval , sentiment analysis [...]
nmod
nmod
nmod
[...] opinion-based recommender system utilize various techniques including text mining , information retrieval , sentiment analysis [...]

Figure 7: **Example of `conj` Alteration**. Original UD dependencies (above), and disentangled conjunction dependencies reflecting the semantic relation annotations (below).

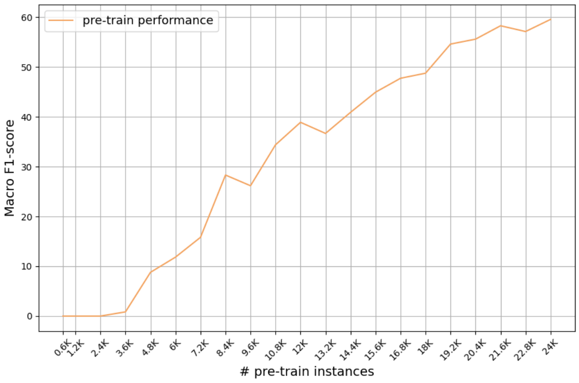

Figure 8: **Pre-train Performance.** Pre-train performance of the RE model over increasing amounts of dependency instances

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section "Limitation" after section 5.

☒ A2. Did you discuss any potential risks of your work?
We do not see any potential risk in our work: using an existing dataset, an existing model architecture.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
-

B ☑ Did you use or create scientific artifacts?

Section 3.

☑ B1. Did you cite the creators of artifacts you used?
Section 3.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
I used already published data. In the paper we refer to the original dataset paper.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3, and reference to original dataset paper.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3, Section 4, Appendix D, and reference to original dataset paper.

C ☑ Did you run computational experiments?

Section 3.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3, Section 4, and Appendix B.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. -

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

-

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. -

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.