Improving Distantly Supervised Document-Level Relation Extraction Through Natural Language Inference

Clara Vania **Grace E. Lee*** **Andrea Pierleoni**
Amazon Alexa
{vaniclar, apierleoni}@amazon.co.uk
grace.lee2@thomsonreuters.com

Abstract

The distant supervision (DS) paradigm has been widely used for relation extraction (RE) to alleviate the need for expensive annotations. However, it suffers from noisy labels, which leads to worse performance than models trained on human-annotated data, even when trained using hundreds of times more data. We present a systematic study on the use of natural language inference (NLI) to improve distantly supervised document-level RE. We apply NLI in three scenarios: (i) as a filter for denoising DS labels, (ii) as a filter for model prediction, and (iii) as a standalone RE model. Our results show that NLI filtering consistently improves performance, reducing the performance gap with a model trained on human-annotated data by 2.3 F1.

1 Introduction

Relation extraction (RE) is the task of identifying relations between two entities in natural language text. It has an important role in many NLP applications, such as knowledge base population and question answering. Existing work on RE has been focused mostly on extraction within a sentence (Mintz et al., 2009; Zhang et al., 2017; Han et al., 2018). However, sentence-level RE has one major limitation: it is not designed to extract relational facts expressed in multiple sentences.[1] To address this, recent work has explored models which use document-level context to extract both intra- and inter-sentence relations from text (Li et al., 2020; Xu et al., 2021; Eberts and Ulges, 2021)

Currently, high-performance RE models require large-scale human-annotated data, which is expensive and does not scale to a large number of relations or new domains. To reduce the reliance on human-annotated data, Mintz et al. (2009) introduce the distant supervision (DS) approach, which assumes that if two entities are connected through a relation in a knowledge base, sentences that mention the two entities express that relation. While this assumption allows the creation of large-scale training data without expensive human annotation, it also produces many noisy labels (Riedel et al., 2010).[2] As a result, the performance of models trained on DS datasets is considerably lower (~5%) than models trained on human-annotated datasets.

This paper aims to reduce the performance gap between models trained on DS versus annotated data through natural language inference (NLI). NLI, also known as *textual entailment*, is the task of determining whether a premise entails a hypothesis. Recently, Sainz et al. (2021) used an NLI model as a standalone RE model and demonstrated its effectiveness for zero-shot and few-shot sentence-level RE. In line with their work, we investigate if NLI can also benefit document-level RE in this paper. Specifically, we apply NLI for document-level RE in three scenarios: (i) as a filter for denoising DS labels, (ii) as a filter for model prediction, and (iii) as a standalone RE model.

We experiment with DocRED (Yao et al., 2019), the largest document-level RE dataset to date. It consists of both DS and human-annotated datasets, which is ideal for our study. Across all scenarios, we find that NLI is especially effective when it is used as a filter; we observe improvement up to 2.3 F1, reducing the gap with a model trained on annotated data from 5.3 to 3.0 F1. However, the gains are less significant when the model has access to human-annotated data. Finally, we highlight the importance of having high-quality entity type information when using NLI as a standalone RE model.

* Work completed at Amazon Alexa. The author now works at Thomson Reuters.

[1] According to Yao et al. (2019), at least 40.7% facts in Wikipedia can only be extracted from multiple sentences.

[2] For document-level RE, Yao et al. (2019) report 41% and 61% incorrect labels for intra- and inter-sentence relations in DS, respectively.

Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, pages 14 - 20
July 14, 2022

2 NLI for RE

We first describe the approach by Sainz et al. (2021), which uses an NLI model as a standalone model for sentence-level RE.

Let p be an input text containing two entity mentions m_1 and m_2. We take p as the premise and generate the hypothesis h by verbalizing each relation r using a template t, m_1, and m_2. For example, the relation "*capital of*" can be verbalized using the template "$\{m_1\}$ is the capital of $\{m_2\}$". One relation can be verbalized using multiple templates, leading to multiple hypotheses. To avoid mismatch between the entity types and the relation, a set of allowed types for the first and the second entities is created for each relation, e.g., the relation "*date of birth*" should involve a PERSON and a DATE entities. We use a function f_r to determine whether a relation $r \in R$ matches the given entity types, e_1 and e_2:

$$f_r(e_1, e_2) = \begin{cases} 1 & e_1 \in E_{r1} \land e_2 \in E_{r2} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where E_{r1} and E_{r2} are the set of allowed types for the first and the second entities in r. We then compute the probability of each relation r as:

$$P_r(p, m_1, m_2) = f(e_1, e_2) \max_{t \in T_r} P_{NLI}(p, h|t, m_1, m_2) \quad (2)$$

where P_{NLI} is the entailment probability of (p, h) given by the NLI model, and T_r is the set of templates for relation r, and h is the hypothesis generated using a template t and the two entity mentions, m_1 and m_2. In practice, we only need to run NLI inference for relation with $f_r(e_1, e_2) = 1$. To identify cases when no relation exists between m_1 and m_2, we apply a threshold $\mathcal{T}$ to Eq. 2. If none of the relations surpasses $\mathcal{T}$, then we assume there is no relation between the two mentions, otherwise we return the relation with the highest entailment probability:

$$\hat{r} = \arg\max_{r \in R} P_r(p, m_1, m_2). \quad (3)$$

Adapting to Document-Level RE For our experiments with document-level RE, we adapt the same setup as Sainz et al. (2021) by treating the whole document context as the premise. We apply NLI in three scenarios: (i) as a filter to for denoising DS labels (**pre-filter**), (ii) as a filter for model predictions (**post-filter**), and (iii) as a standalone RE model. In the pre-filtering scenario, we verbalize the labels (relations) identified using the DS assumption and remove all labels that do not surpass the threshold $\mathcal{T}$ from the DS dataset. Similarly, in the post-filtering scenario, we verbalize the relations predicted by an RE model and remove those which do not surpass $\mathcal{T}$. In both scenarios, we do not need to generate candidate relations (Eq. 1) since they are provided by the DS labels or the RE model predictions. Unlike Sainz et al. (2021) which chooses *one* relation label that maximizes the probability of the hypothesis (Eq. 3), we use *all* relation labels that have entailment probability above $\mathcal{T}$.[3] In our experiments, we set $\mathcal{T} = 0.5$, i.e., taking all relations that the NLI model predicts as entailment. Additionally, since the DS dataset is known to be noisy, for the pre-filtering scenario, we also experiment with higher thresholds to study the effect of using more strict filters on the RE performance.

We experiment with two types of NLI models: a model that is not trained specifically for RE (zero-shot NLI) and a model that is fine-tuned using a small number of human-annotated RE examples (few-shot NLI). The zero-shot NLI model simulates a case when we do not have any annotations, while the few-shot NLI model simulates a case when we have a small budget for annotations. We fine-tune the NLI model for a binary entailment task (*entail* or *not entail*). Since DocRED annotations do not contain negative examples (*no-relation* label), we generate the non-entail examples for NLI as follows. First, we train a model using the DS dataset and generate predictions for the human-annotated training data. We then use the model's incorrect predictions as the non-entail examples. We use a maximum $N = \{10, 100\}$ examples per relation.

3 Experiments

Dataset We experiment with DocRED (Yao et al., 2019), a document-level RE dataset created from Wikipedia articles aligned with Wikidata. It covers six entity types (ORG, LOC, PER, TIME, NUM, MISC) and 96 relation types. DocRED contains $101,873$ DS training documents and $5,051$ human-annotated documents, split into training $(3,053)$,

[3]The setup of Sainz et al. (2021) most likely influenced by their experimental dataset, TACRED (Zhang et al., 2017), which only allows one relation per mention pair. On the other hand, DocRED annotations may have multiple relations per entity pair.

development (998), and testing (1,000). [4]

RE Model For our document-level RE model, we use JEREX (Eberts and Ulges, 2021) which obtains comparable performance with the state-of-the-art SSAN (Xu et al., 2021) model when using `bert-base-case` encoder. The model has four main components (entity mention localization, coreference resolution, entity classification, relation classification), which share the same encoder and mention representations, and are trained jointly. For the relation classifier module, we use the multi-instance version, which predicts relation on the mention-level. JEREX is originally designed for end-to-end RE without the need for entity information. However, since our main focus is on the RE side, we use its standard RE pipeline, which assumes that entity clusters are given.

NLI Model We use a pretrained document-level NLI model based on DeBERTaV3 (He et al., 2021)[5], which was trained on 1.3M premise-hypothesis pairs from 8 datasets: MNLI (Williams et al., 2018), FEVER-NLI (Nie et al., 2019), NLI dataset from Parrish et al. (2021), and DocNLI (Yin et al., 2021) (which is curated from ANLI (Nie et al., 2020), SQuAD (Rajpurkar et al., 2016), DUC2001[6], CNN/DailyMail (Nallapati et al., 2016), and Curation (Curation, 2020)). The model was trained for a binary entailment task.

Training and Optimization For training JEREX models, we use the default hyperparameters of Eberts and Ulges (2021). We use a maximum of 10 epochs for training with the DS dataset and 40 epochs for training with the human-annotated dataset. For NLI fine-tuning, we use a maximum of 10 epochs for the few-shot setting and one epoch when using the full annotated data. We tune the learning rate $\in \{1e{-}5, 2e{-}5, 3e{-}5\}$, with a batch size of 8 and gradient accumulation steps of 4. Each model is trained using a single V100 GPU with 16GB memory. We train each model with three random restarts and report the average performance.

[4]We use the revised version of DocRED development set with 998 documents after two documents were removed because they overlap with the annotated training data.

[5]`https://huggingface.co/MoritzLaurer/DeBERTa-v3-base-mnli-fever-docnli-ling-2c`

[6]`https://www-nlpir.nist.gov/projects/duc/guidelines/2001.html`

Threshold	zero-shot	10-shot	100-shot	full
low (0.5)	73.4	71.1	66.0	65.1
med (0.95)	68.6	70.1	56.4	48.4
high (0.99)	59.0	69.1	38.8	12.3

Table 1: Percentages of triples left in the DS data after pre-filtering with NLI.

Model	Precision	Recall	F1	IgnF1
Training with annotated data only (supervised)				
BERT Base†	-	-	58.6	56.3
SSAN Biaffine†	-	-	59.2	57.0
JEREX	64.5	54.8	59.2	57.4
Training with DS data only (weakly supervised)				
JEREX	51.5	**56.5**	53.9	51.0
+ pre-filter (low)	61.3	51.8	56.1	54.0
+ pre-filter (med)	62.4	50.3	55.7	53.7
+ pre-filter (high)	**65.7**	46.2	54.3	52.6
+ post-filter	60.8	52.3	**56.2**	**54.1**
+ double-filter	64.0	50.0	56.1	54.2

Table 2: Results on DocRED development set when using zero-shot NLI models. Results with † are from Xu et al. (2021). IgnF1: F1 score that ignores triples occur in the annotated training data.

4 Results and Analysis

Zero-shot NLI Table 1 shows the percentages of triples left in the DS dataset (out of ~1.5M instances) after pre-filtering with different thresholds $\mathcal{T}$ (for other thresholds, see Appendix A). For the zero-shot NLI, setting $\mathcal{T}$ to the lowest value (0.5) leaves us with 73.4% of the original DS triples, while setting it to the maximum value (0.99) leaves us with 59.0% of the original DS triples.

Table 2 reports our main RE results. Our baseline is a JEREX model trained with the DS dataset. To understand how far NLI can help in reducing the gap between models trained using the DS (*weakly supervised*) vs. human-annotated (*supervised*) datasets, we also provide results of supervised models using BERT base, JEREX, SSAN (Xu et al., 2021). All of the models use the same BERT base encoder (Devlin et al., 2019).

We find that NLI improves RE performance in both pre-filter and post-filter scenarios. Post-filtering with NLI achieves the best performance with 56.2 F1, reducing the gap with the supervised model by 2.3 F1. Looking into the other metrics, it is evident that NLI filtering yields RE models with higher precision but lower recall. We observe that our most aggressive pre-filtering (*high*) outper-

Model	Precision	Recall	F1	IgnF1
10-shot NLI				
JEREX	65.5	56.2	60.5	58.6
+ pre-filter (low)	64.3	58.5	**61.2**	**59.7**
+ pre-filter (high)	61.9	**59.6**	60.7	58.6
+ post-filter	**69.0**	52.7	59.8	58.2
+ double-filter	66.1	55.8	60.5	58.8
100-shot NLI				
JEREX	66.3	57.8	61.7	59.8
+ pre-filter (low)	65.5	**59.3**	**62.2**	**60.4**
+ pre-filter (med)	66.2	56.9	61.2	59.4
+ pre-filter (high)	67.3	54.6	60.3	58.7
+ post-filter	**70.3**	53.3	60.6	59.1
+ double-filter	69.9	53.9	60.8	59.3
Training with DS + full annotated data				
JEREX	68.0	**58.3**	62.7	60.9
+ pre-filter (low)	71.3	57.8	**63.8**	**62.3**
+ pre-filter (med)	70.5	56.7	62.9	61.4
+ pre-filter (high)	64.4	46.7	54.2	52.5
+ post-filter	71.0	54.1	61.4	59.9
+ double-filter	**73.4**	54.0	62.2	60.9

Table 3: Results on DocRED development set when using fine-tuned RE and NLI models.

forms the precision of the supervised model. This result suggests that pre-filtering is especially useful for applications where having high precision is preferable to recall. We also experiment with the *double-filter* scenario, where we apply both our best pre-filter (low) and post-filter. We find it has minimal effect on the model performance.

Few-shot NLI This scenario assumes that a small human-annotated dataset is available, so in the next set of experiments, all RE models are trained using the DS dataset and then fine-tuned using the small annotated dataset.[7] Unlike NLI fine-tuning, where we limit the maximum number of examples per relation when fine-tuning the RE models, we use all annotations in the document since we want the model to learn all and not just the subset of correct triples. We fine-tune the RE models using 427 and 1,761 annotated documents for the 10-shot and the 100-shot NLI settings, respectively.

As shown in Table 3, in the few-shot settings, we can still see improvement by using NLI as a pre-filter. However, the improvements are not as large as in the DS-only training.[8] We also see 1.2 F1 improvements when using the full annotated data (~3k documents) for fine-tuning the NLI and RE model.

[7]The DS training followed by fine-tuning setup yields the best model performance on DocRED (Xu et al., 2021).

[8]We only experiment with *low* and *high* for the 10-shot experiments since the *medium* filtering yield very similar training data distribution (Table 1).

NLI Model	Precision	Recall	F1	IgnF1
Coarse-grained types				
Zero-shot	3.1	68.0	5.9	5.2
10-shot	2.5	68.4	4.8	4.2
100-shot	2.3	66.6	4.4	3.8
Full-data	2.4	68.2	4.7	4.1
Fine-grained types				
Zero-shot	20.4	27.8	23.5	20.5
10-shot	15.4	28.4	20.0	16.9
100-shot	15.3	26.5	19.4	16.5
Full-data	16.6	27.6	20.7	17.7

Table 4: Results on DocRED development set when using NLI as a standalone RE model.

NLI as a standalone RE model We utilize the entity type information in the DocRED annotated training data to create the list of allowed entity types for each relation. However, we find that this strategy still leads us to mismatch types between the relation and entity, which might be due to several reasons. First, DocRED entities are annotated with coarse-grained types (Section 3), which might confuse the model when learning about relations that exist between entities. For instance, frequent location relations such as P17 (*country*) require the tail entity to be a country. However, with the generic LOC type and sometimes similar NLI template (e.g. "$\{m_1\}$ is located in $\{m_2\}$"), other types of locations, such as cities, can also fit the slot for m_2 and be inferred as correct by the NLI model. We also find that the MISC type is especially ambiguous since it is allowed in almost all relations. Second, DocRED relations are annotated on entity-level, where one entity can have multiple mentions with different types, e.g., the entity *Finland* has mentions *Finland* (LOC) as well as *Finnish* (MISC). To alleviate this, we only add entity types to a relation if they co-occur more than 100 times in the data. In addition, we also experiment using ~500 fine-grained entity types using ReFinED (Ayoola et al., 2022), which currently obtain state-of-the-art performance on several entity linking datasets.

Table 4 presents our results. We observe that using coarse-grained entity type information leads to poor model performance. In particular, we find that the model overpredicts the relations, as shown by the high recall. Using finer-grained types improves performance up to 23.5 F1, but it is still

NLI Model	Training	F1	IgnF1
Zero-shot	Annotated only	59.5	57.5
	DS only	52.9	49.8
	DS + NLI	55.6	53.4
Few-shot	10-shot	59.3	57.4
	10-shot + NLI	61.1	58.8
	100-shot	61.7	59.7
	100-shot + NLI	61.8	59.9
Full-data	DS + Annotated	62.0	60.0
	DS + Annotated + NLI	**63.4**	**61.5**

Table 5: Results on DocRED test set.

far below the performance of a model specifically trained for RE. This result suggests that when the NLI model is provided with a set of noisy candidate relations, it predicts many of them as correct. On the other hand, when the set of candidate relations is less noisy (given by the DS labels or RE model predictions), the NLI model performs well and can improve RE performance.

Results on Test Set We validate our result by running our overall best strategy, pre-filtering by NLI ($\mathcal{T} = 0.5$) on the test set. Table 5 shows a similar pattern as observed in the development data: NLI filtering consistently improves performance in all settings. We only report F1 and IgnF1 since DocRED CodaLab output does not provide precision and recall numbers.

5 Conclusion

In this paper, we presented a systematic study on the use of NLI for distantly supervised document-level RE, focusing on the case when human-annotated data is not available. Our results demonstrate that NLI is most effective when used as a pre-filter to denoise DS labels. In the absence of human annotations, we show that NLI filtering reduces the gap with a model trained on human-annotated data by 2.3 F1. We also show that NLI filtering still benefits the RE model (+1.1 F1) when we have small human-annotated data. Our experiment on using NLI as a standalone model for document-level RE leads to worse performance than using it as a pre-filter, suggesting that using NLI directly as an RE model for document-level is more challenging than sentence-level RE.

For future work, we plan to explore other strategies to better leverage the entity type information for RE with NLI and investigate if document-level NLI is also more challenging than sentence-level NLI. Another potential direction is to experiment with other DS techniques, such as integrating a denoising module to the RE model (Xiao et al., 2020) or using DS-trained models as a DS filter (Zhou and Chen, 2021).

Acknowledgements

We thank Tom Ayoola, Shubhi Tyagi, Siffi Singh, Marco Damonte, and the anonymous reviewers for helpful discussion of this work and comments on previous drafts of this paper.

References

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, Seattle, Washington. Association for Computational Linguistics.

Curation. 2020. Curation corpus base.

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. 2021. 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*, pages 3650–3660, Online. 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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Bo Li, Wei Ye, Zhonghao Sheng, Rui Xie, Xiangyu Xi, and Shikun Zhang. 2020. Graph enhanced dual attention network for document-level relation extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1551–1560, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Chaojun Xiao, Yuan Yao, Ruobing Xie, Xu Han, Zhiyuan Liu, Maosong Sun, Fen Lin, and Leyu Lin. 2020. Denoising relation extraction from document-level distant supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3683–3688, Online. Association for Computational Linguistics.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *AAAI*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. 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.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Pre-filtering with NLI

Threshold	zero-shot	10-shot	100-shot	full
0.5	73.4	71.1	66.0	65.1
0.7	72.6	70.8	64.9	63.7
0.9	70.8	70.4	60.9	56.2
0.95	68.6	70.1	56.4	48.4
0.97	66.1	69.9	52.4	40.0
0.99	59.0	69.1	38.8	12.3

Table 6: Percentages of triples left in the DS data after pre-filtering with NLI with different threshold values.

B DocRED NLI Templates

Relation	Templates
applies to jurisdiction	{head} rules {tail}.
	{head} represents {tail}.
	{head} works for the {tail} government.
author	{head} is written by {tail}.
	{head} is a story by {tail}.
	{tail} is the author of {head}.
	{tail} wrote {head}.
award received	{head} received {tail}.
	{head} won {tail}.
	{head} was a recipient of {tail}.
	{head} was awarded {tail}.
basin country	{head} is located near {tail}.
	{tail} is located in {head}.
capital of	{head} is the capital of {tail}.
	{tail}'s capital is {head}.
capital	{head}'s capital is {tail}.
	{tail} is the capital of {head}.
cast member	{head}'s cast includes {tail}.
	{tail} starred in {head}.
	{tail} appeared in {head}.
continent	{head} is located in {tail}.
country of citizenship	{head} country of citizenship is {tail}.
	{head} is from {tail}.
country	{head} is located in {tail}.
creator	{head} is created by {tail}.
	{tail} is the creator of {tail}.
date of birth	{head} was born {tail}.
date of death	{head} died {tail}.
director	{head} is a movie directed by {tail}.
	{head} is a game directed by {tail}.
	{tail} is the director of {head}.

Table 7: Examples of DocRED NLI Templates. Full templates can be found in the supplementary materials.