An Improved Baseline for Sentence-level Relation Extraction

Wenxuan Zhou
University of Southern California
zhouwenx@usc.edu

Muhao Chen
University of Southern California
muhaoche@usc.edu

Abstract

Sentence-level relation extraction (RE) aims at identifying the relationship between two entities in a sentence. Many efforts have been devoted to this problem, while the best performing methods are still far from perfect. In this paper, we revisit two problems that affect the performance of existing RE models, namely ENTITY REPRESENTATION and NOISY OR ILL-DEFINED LABELS. Our improved RE baseline, incorporated with entity representations with typed markers, achieves an F_1 of 74.6% on TACRED, significantly outperforms previous SOTA methods. Furthermore, the presented new baseline achieves an F_1 of 91.1% on the refined Re-TACRED dataset, demonstrating that the pretrained language models (PLMs) achieve high performance on this task. We release our code[1] to the community for future research.

1 Introduction

As one of the fundamental information extraction (IE) tasks, relation extraction (RE) aims at identifying the relationship(s) between two entities in a given piece of text from a pre-defined set of relationships of interest. For example, given the sentence "Bill Gates founded Microsoft together with his friend Paul Allen in 1975" and an entity pair ("Bill Gates", "Microsoft"), the RE model is expected to predict the relation `ORG:FOUNDED_BY`. On this task, SOTA models based on PLMs (Devlin et al., 2019; Joshi et al., 2020) have gained significant success.

Recent work on sentence-level RE can be divided into two lines. One focuses on injecting external knowledge into PLMs. Methods of such, including ERNIE (Zhang et al., 2019) and KnowBERT (Peters et al., 2019), take entity embedding pretrained from knowledge graphs as inputs to the Transformer. Similarly, K-Adapter (Wang et al., 2020) introduces a plug-in neural adaptor that injects factual and linguistic knowledge into the language model. LUKE (Yamada et al., 2020) further extends the pretraining objective of masked language modeling to entities and proposes an entity-aware self-attention mechanism. The other line of work focuses on continually pretraining PLMs on text with linked entities using relation-oriented objectives. Specifically, BERT-MTB (Baldini Soares et al., 2019) proposes a matching-the-blanks objective that decides whether two relation instances share the same entities. Despite extensively studied, existing RE models still perform far from perfect. On the commonly-used benchmark TACRED (Zhang et al., 2017), the SOTA F_1 result only increases from 70.1% ($\text{BERT}_{\text{LARGE}}$) to 72.7% (LUKE) after applying PLMs to this task. It is unclear what building block is missing to constitute a promising RE system.

In this work, we discuss two obstacles that have hindered the performance of existing RE models. First, the RE task provides a structured input of both the raw texts and *side information* of the entities, such as entity names, spans, and types (typically provided by NER models), which are shown important to the performance of RE models (Peng et al., 2020). However, existing methods fall short of representing the entity information comprehensively in the text, leading to limited characterization of the entities. Second, human-labeled RE datasets (e.g., TACRED), may contain a large portion of noisy or ill-defined labels, causing the model performance to be misestimated. Alt et al. (2020) relabeled the development and test set of TACRED and found that 6.62% of labels are incorrect. Stoica et al. (2021) refined many ill-defined relation types and further re-annotated the TACRED dataset using an improved annotation strategy to

[1] https://github.com/wzhouad/RE_improved_baseline

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 161–168
November 20–23, 2022.

ensure high-quality labels. To this end, we propose an improved RE baseline, where we introduce the typed entity marker to sentence-level RE, which leads to promising improvement of performance over existing RE models.

We evaluate our model on TACRED (Zhang et al., 2017), TACREV (Alt et al., 2020), and Re-TACRED (Stoica et al., 2021). Using RoBERTa (Liu et al., 2019) as the backbone, our improved baseline model achieves an F_1 of 74.6% and 83.2% on TACRED and TACREV, respectively, significantly outperforming various SOTA RE models. Particularly, our baseline model achieves an F_1 of 91.1% on Re-TACRED, demonstrating that PLMs can achieve much better results on RE than shown in previous work.[2]

2 Method

In this section, we first formally define the relation extraction task in Sec. 2.1, and then present our model architecture and entity representation techniques in Sec. 2.2 and Sec. 2.3.

2.1 Problem Definition

In this paper, we focus on sentence-level RE. Specifically, given a sentence $\boldsymbol{x}$ mentioning an entity pair (e_s, e_o), referred as the subject and object entities, respectively, the task of sentence-level RE is to predict the relationship r between e_s and e_o from $\mathcal{R} \cup \{\text{NA}\}$, where $\mathcal{R}$ is a pre-defined set of relationships of interest. If the text does not express any relation from $\mathcal{R}$, the entity pair will be accordingly labeled NA.

2.2 Model Architecture

Our RE classifier is an extension of previous PLM-based RE models (Baldini Soares et al., 2019). Given the input sentence $\boldsymbol{x}$, we first mark the entity spans and entity types using techniques presented in Sec. 2.3, then feed the processed sentence into a PLM to get its contextual embedding. Finally, we feed the hidden states of the subject and object entities in the language model's last layer, i.e., $\boldsymbol{h}_{\text{subj}}$ and $\boldsymbol{h}_{\text{obj}}$, into the softmax classifier:

$$\boldsymbol{z} = \text{ReLU}\left(\boldsymbol{W}_{\text{proj}}\left[\boldsymbol{h}_{\text{subj}}, \boldsymbol{h}_{\text{obj}}\right]\right),$$

$$\text{P}(r) = \frac{\exp(\boldsymbol{W}_r \boldsymbol{z} + \boldsymbol{b}_r)}{\sum_{r' \in \mathcal{R} \cup \{\text{NA}\}} \exp(\boldsymbol{W}_{r'} \boldsymbol{z} + \boldsymbol{b}_{r'})},$$

where $\boldsymbol{W}_{\text{proj}} \in \mathbb{R}^{2d \times d}$, $\boldsymbol{W}_r, \boldsymbol{W}_{r'} \in \mathbb{R}^d, \boldsymbol{b}_r, \boldsymbol{b}_{r'} \in \mathbb{R}$ are model parameters. In inference, the classifier returns the relationship with the maximum probability as the predicted relationship.

2.3 Entity Representation

For sentence-level RE, the names, spans, and NER types of subject and object entities are provided in the structured input. Such composite entity information provides useful clues to the relation types. For example, the relationship `ORG:FOUNDED_BY` is more likely to hold when entity types of subject and object are `ORGANIZATION` and `PERSON`, respectively, and is less likely for instances where the entity types do not match. The entity information needs to be represented in the input text, allowing it to be captured by the PLMs. Such techniques have been studied in previous work (Zhang et al., 2017; Baldini Soares et al., 2019; Wang et al., 2020), while many of them fall short of capturing all types of the provided information. In this paper, we re-evaluate existing entity representation techniques and also seek to propose a better one. We evaluate the following techniques:

- **Entity mask** (Zhang et al., 2017). This technique introduces new special tokens `[SUBJ-`*`TYPE`*`]` or `[OBJ-`*`TYPE`*`]` to mask the subject or object entities in the original text, where *`TYPE`* is substituted with the respective entity type. This technique was originally proposed in the PA-LSTM model (Zhang et al., 2017), and was later adopted by PLMs such as SpanBERT (Joshi et al., 2020). Zhang et al. (2017) claim that this technique prevents the RE model from over-fitting specific entity names, leading to more generalizable inference.
- **Entity marker** (Zhang et al., 2019; Baldini Soares et al., 2019). This technique introduces special tokens pairs `[E1]`, `[/E1]` and `[E2]`, `[/E2]` to enclose the subject and object entities, therefore modifying the input text to the format of "`[E1]` SUBJ `[/E1]` ... `[E2]` OBJ `[/E2]`"[3].

[2] This work first appeared as a technical report on arXiv in Feb 2021 (Zhou and Chen, 2021). Since then, the proposed techniques have been incorporated into several follow-up works (Chen et al., 2022; Wang et al., 2022b,a; Lu et al., 2022; Han et al., 2021; Kulkarni et al., 2022) that are published before this version of the paper.

[3] SUBJ and OBJ are respectively the original token spans of subject and object entities.

- **Entity marker (punct)** (Wang et al., 2020; Zhou et al., 2021). This technique is a variant of the previous technique that encloses entity spans using punctuation. It modifies the input text to "@ SUBJ @ ... # OBJ #". The main difference from the previous technique is that this one does not introduce new special tokens into the model's reserved vocabulary.
- **Typed entity marker** (Zhong and Chen, 2021). This technique further incorporates the NER types into entity markers. It introduces new special tokens ⟨S:*TYPE*⟩, ⟨/S:*TYPE*⟩, ⟨O:*TYPE*⟩, ⟨/O:*TYPE*⟩, where *TYPE* is the corresponding NER type given by a named entity tagger. The input text is accordingly modified to "⟨S:*TYPE*⟩ SUBJ ⟨/S:*TYPE*⟩ ... ⟨O:*TYPE*⟩ OBJ ⟨/O:*TYPE*⟩".
- **Typed entity marker (punct)**. We propose a variant of the typed entity marker technique that marks the entity span and entity types without introducing new special tokens. This is to enclose the subject and object entities with "@" and "#", respectively. We also represent the subject and object entity types using their label text, which is prepended to the entity spans and is enclosed by "*" for subjects or "∧" for objects. The modified text is "@ * *subj-type* * SUBJ @ ... # ∧ *obj-type* ∧ OBJ # ", where *subj-type* and *obj-type* is the label text of NER types.

The embedding of all new special tokens is randomly initialized and updated during fine-tuning.

3 Experiments

In this section, we evaluate the proposed techniques based on widely used RE benchmarks. The evaluation starts by first identifying the best-performing entity representation technique (Sec. 3.2), which is further incorporated into our improved RE baseline to be compared against prior SOTA methods (Sec. 3.3). Due to space limits, we study in the Appendix of how the incorporated techniques lead to varied generalizability on unseen entities (Appx. B) and how they perform under annotation errors (Appx. C).

3.1 Preliminaries

Datasets. The datasets we have used in the experiments include three versions of TACRED: the original TACRED (Zhang et al., 2017), TACREV (Alt et al., 2020), and Re-TACRED (Stoica et al., 2021). Alt et al. (2020) observed that the TACRED dataset contains about 6.62% noisily-labeled instances and relabeled the development and test set. Stoica et al. (2021) further refined the label definitions in TACRED and relabeled the whole dataset. We provide the statistics of the datasets in Appx. A.

Compared methods. We compare with the following methods. **PA-LSTM** (Zhang et al., 2017) adopts bi-directional LSTM (Hochreiter and Schmidhuber, 1997) and positional-aware attention (Bahdanau et al., 2015) to encode the text into an embedding, which is then fed into a softmax layer to predict the relation. **C-GCN** (Zhang et al., 2018) is a graph-based model, which feeds the pruned dependency tree of the sentence into the graph convolutional network (Kipf and Welling, 2017) to obtain the representation of entities. **SpanBERT** (Joshi et al., 2020) is a PLM based on the Transformer (Vaswani et al., 2017). It extends BERT (Devlin et al., 2019) by incorporating a training objective of span prediction and achieves improved performance on RE. **KnowBERT** (Peters et al., 2019) jointly trains a language model and an entity linker, which allows the subtokens to attend to entity embedding that is pretrained on knowledge bases. **LUKE** (Yamada et al., 2020) pretrains the language model on both large text corpora and knowledge graphs. It adds frequent entities into the vocabulary and proposes an entity-aware self-attention mechanism.

Model configurations. For the compared methods, we rerun their officially released code using the recommended hyperparameters in their papers. Our model is implemented based on HuggingFace's Transformers (Wolf et al., 2020). Our model is optimized with Adam (Kingma and Ba, 2015) using the learning rate of 5e−5 on $\text{BERT}_{\text{BASE}}$, and 3e−5 on $\text{BERT}_{\text{LARGE}}$ and $\text{RoBERTa}_{\text{LARGE}}$, with a linear warm-up (Goyal et al., 2017) of for the first 10% steps followed by a linear learning rate decay to 0. We use a batch size of 64 and fine-tune the model for 5 epochs on all datasets. For all experiments, we report the median F_1 of 5 runs of training using different random seeds.

3.2 Analysis on Entity Representation

We first provide an analysis on different entity representation techniques. In this analysis, we use the base and large versions of BERT (Devlin et al., 2019) and the large version of RoBERTa (Liu et al., 2019) as the encoder. Tab. 1 shows the perfor-

Method	Input Example	$\text{BERT}_{\text{BASE}}$	$\text{BERT}_{\text{LARGE}}$	$\text{RoBERTa}_{\text{LARGE}}$
Entity mask	`[SUBJ-PERSON]` was born in `[OBJ-CITY]`.	69.6	70.6	60.9
Entity marker	`[E1]` Bill `[/E1]` was born in `[E2]` Seattle `[/E2]`.	68.4	69.7	70.7
Entity marker (punct)	@ Bill @ was born in # Seattle #.	68.7	69.8	71.4
Typed entity marker	⟨`S:PERSON`⟩ Bill ⟨`/S:PERSON`⟩ was born in ⟨`O:CITY`⟩ Seattle ⟨`/O:CITY`⟩.	**71.5**	**72.9**	71.0
Typed entity marker (punct)	@ * person * Bill @ was born in # ∧ city ∧ Seattle #.	70.9	72.7	**74.6**

Table 1: Test $\boldsymbol{F_1}$ (in %) of different entity representation techniques on TACRED. For each technique, we also provide the processed input of an example text *"Bill was born in Seattle"*. Typed entity markers (original and punct) significantly outperforms others.

mance of the PLMs incorporated with different entity representation techniques. For each technique, we also provide an example of the processed text. We have several observations from the results. First, the typed entity marker and its variants outperform untyped entity representation techniques by a notable margin. Especially, the RoBERTa model achieves an F_1 score of 74.6% using the typed entity marker (punct), which is significantly higher than the SOTA result of 72.7% by LUKE (Yamada et al., 2020). This shows that representing all categories of entity information is helpful to the RE task. It also shows that keeping entity names in the input improves the performance of RE models. Second, symbols used in entity markers have an obvious impact on the performance of RE models. Although the original and *punct* versions of entity representation techniques represent the same categories of entity information, they do lead to a difference in model performance. Particularly, introducing new special tokens hinders the model performance drastically on RoBERTa. On $\text{RoBERTa}_{\text{LARGE}}$, the entity marker underperforms the entity marker (punct) by 0.7%, the typed entity marker underperforms the typed entity marker (punct) by 3.6%, while the entity mask gets a much worse result of 60.9%.

3.3 Comparison with Prior Methods

The prior experiment has found $\text{RoBERTa}_{\text{LARGE}}$ with the typed entity marker (punct) to be the best-performing RE model. We now compare our improved baseline with methods in prior studies.

The experimental results are shown in Tab. 2. We evaluate all methods on TACRED, TACREV, and Re-TACRED. Incorporated with the typed entity marker (punct) and using $\text{RoBERTa}_{\text{LARGE}}$ as the backbone, our improved baseline model achieves new SOTA results over previous methods on all datasets. However, we observe that on Re-TACRED, the gain from the typed entity marker is much smaller compared to TACRED and TACREV, decreasing from $3.1 - 3.9\%$ and $2.0 - 3.4\%$ to $0.2 - 0.8\%$ of F_1. This observation could be attributed to the high noise rate in TACRED, in which the noisy labels are biased towards the side information of entities.

Model	**TACRED** Test F_1	**TACREV** Test F_1	**Re-TACRED** Test F_1
Sequence-based Models			
PA-LSTM (Zhang et al., 2017)	65.1	73.3‡	79.4†
C-GCN (Zhang et al., 2018)	66.3	74.6‡	80.3†
Transformer-based Models			
$\text{BERT}_{\text{BASE}}$ + entity marker	68.4	77.2	87.7
$\text{BERT}_{\text{LARGE}}$ + entity marker	69.7	77.9	89.2
$\text{RoBERTa}_{\text{LARGE}}$ + entity marker	70.7	81.2	90.5
SpanBERT (Joshi et al., 2020)	70.8	78.0*	85.3†
KnowBERT (Peters et al., 2019)	71.5	79.3*	-
LUKE (Yamada et al., 2020)	72.7	80.6‡	90.3‡
Improved RE baseline			
$\text{BERT}_{\text{BASE}}$ + typed entity marker	71.5	79.3	87.9
$\text{BERT}_{\text{LARGE}}$ + typed entity marker	72.9	81.3	89.7
$\text{RoBERTa}_{\text{LARGE}}$ + typed entity marker (punct)	**74.6**	**83.2**	**91.1**

Table 2: F_1 (in %) on the test sets. * marks re-implemented results from Alt et al. (2020). † marks re-implemented results from Stoica et al. (2021). ‡ marks our re-implemented results.

To assess how the presented techniques contribute to robustness and generalizability of RE, we provide more analyses on varied generalizability on unseen entities (Appx. B) and the performance under annotation errors (Appx. C) in the Appendix.

4 Conclusion

In this paper, we present a simple yet strong RE baseline that offers new SOTA performance, along with a comprehensive study to understand its prediction generalizability and robustness. Specifically, we revisit two technical problems in sentence-level RE, namely *entity representation* and *noisy or ill-defined labels*. We propose an improved entity

representation technique, which significantly outperforms existing sentence-level RE models. Especially, our improved RE baseline achieves an F_1 score of 91.1% on the Re-TACRED dataset, showing that PLMs already achieve satisfactory performance on this task. We hope the proposed techniques and analyses can benefit future research on RE.

Acknowledgement

We appreciate the reviewers for their insightful comments and suggestions. This work supported by the National Science Foundation of United States Grant IIS 2105329, and a Cisco Research Award.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

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.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, pages 2778–2788.

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.

Priya Goyal, P. Dollár, Ross B. Girshick, P. Noordhuis, L. Wesolowski, Aapo Kyrola, Andrew Tulloch, Y. Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*.

Thomas Kipf and M. Welling. 2017. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2022. Learning rich representation of keyphrases from text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 891–906, Seattle, United States. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Keming Lu, I Hsu, Wenxuan Zhou, Mingyu Derek Ma, Muhao Chen, et al. 2022. Summarization as indirect supervision for relation extraction. *arXiv preprint arXiv:2205.09837*.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672, Online. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. Association for Computational Linguistics.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, and Bryan Hooi. 2022a. GraphCache: Message passing as caching for sentence-level relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1698–1708, Seattle, United States. Association for Computational Linguistics.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022b. Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3071–3081, Seattle, United States. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. 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.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for joint entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *arXiv preprint arXiv:2102.01373v1*.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Dataset Statistics

Dataset	# train	# dev	# test	# classes
TACRED	68124	22631	15509	42
TACREV	68124	22631	15509	42
Re-TACRED	58465	19584	13418	40

Table 3: Statistics of datasets.

The statistics of the datasets are shown in Tab. 3.

B Analysis on Unseen Entities

Some previous work (Zhang et al., 2018; Joshi et al., 2020) claims that entity names may leak superficial clues of the relation types, allowing heuristics to hack the benchmark. They show that neural RE models can achieve high evaluation results only based on the subject and object entity names even without putting them into the original sentence. They also suggest that RE models trained without entity masks may not generalize well to unseen entities. However, as the provided NER types in RE datasets are usually coarse-grained, using entity masks may lose the meaningful information of entities. Using entity masks also contradicts later studies' advocacy of injecting entity knowledge into RE models (Zhang et al., 2019; Peters et al., 2019; Wang et al., 2020). If RE models should not consider entity names, it is unreasonable to suppose that they can be improved by external knowledge graphs.

To evaluate whether the RE model trained without entity mask can generalize to unseen entities, we propose a *filtered* evaluation setting. Specifically, we remove all test instances containing entities from the training set of TACRED, TACREV, and Re-TACRED. This results in *filtered test sets* of 4,599 instances on TACRED and TACREV, and 3815 instances on Re-TACRED. These filtered test sets only contain instances with unseen entities during training.

We present the evaluation results on the filtered test set in Tab. 4. We compare the performance of models with entity mask or typed entity marker representations, between which the only difference lies in whether to include entity names in entity representations or not. Note that as the label distributions of the original and filtered test set are different, their results are not directly comparable. Still, the *typed entity marker* consistently outperforms the *entity mask* on all encoders and datasets,

Model	TACRED Test F_1	TACREV Test F_1	Re-TACRED Test F_1
$\text{BERT}_{\text{BASE}}$ + entity mask	75.2	82.7	83.8
$\text{BERT}_{\text{BASE}}$ + typed entity marker	75.8	83.7	87.0
$\text{BERT}_{\text{LARGE}}$ + entity mask	75.8	83.7	85.6
$\text{BERT}_{\text{LARGE}}$ + typed entity marker	77.0	85.3	89.8
$\text{RoBERTa}_{\text{LARGE}}$ + entity mask	69.4	78.8	82.2
$\text{RoBERTa}_{\text{LARGE}}$ + typed entity marker (punct)	78.7	86.9	91.7

Table 4: Test F_1 on the filtered test sets. The typed entity marker consistently outperforms the entity mask, showing that knowledge from entity names can generalize to unseen entities.

Model	$\text{BERT}_{\text{BASE}}$	$\text{BERT}_{\text{LARGE}}$	$\text{RoBERTa}_{\text{LARGE}}$
Entity marker	83.8	86.0	88.6
Typed entity marker (punct for RoBERTa)	84.3	87.5	89.4
Gain	+0.5	+1.5	+0.8
Gain on TACRED	+3.1	+3.2	+3.9
Gain on TACREV	+2.1	+3.4	+2.0

Table 5: Test F_1 on the clean test set of TACRED. The gain on the clean test set is smaller than on TACRED and TACREV.

which shows that RE models can learn from entity names and generalize to unseen entities. Our finding is consistent with Peng et al. (2020), whose work suggests that entity names can provide semantically richer information than entity types to improve the RE model.

C Analysis on Annotation Errors

Our model achieves a smaller performance gain on Re-TACRED compared to TACRED and TACREV. We find that this difference can be mainly attributed to the annotation errors in their evaluation sets. Specifically, we create a clean TACRED test set by pruning all instances in the TACRED test set, of which the annotated relation is different in the Re-TACRED test set. The remaining instances are considered clean. Note that as the label sets of TACRED and Re-TACRED are different, instances of some classes cannot be found in Re-TACRED and are thus completely pruned. We train the model on the original (noisy) training set and show the results on the clean test set in Tab. 5. We observe a similar drop of performance gain on the clean TACRED test set. It shows that the annotation errors in TACRED and TACREV can lead to overestimation of the performance of models depending on the side information of entities. We hypothesize

that in data annotation, much noise may be created as some annotators label the relation only based on the two entities without reading the whole sentence. Therefore, integrating NER types into the entity representation can brings larger performance gain. Overall, this experiment shows that the evaluation sets of both TACRED and TACREV are biased and unreliable. We recommend future work on sentence-level RE should use Re-TACRED as the evaluation benchmark.