Learning from Context or Names? An Empirical Study on Neural Relation Extraction - Appendix -

Hao Peng[1*], **Tianyu Gao**[2*], **Xu Han**[1], **Yankai Lin**[3], **Peng Li**[3], **Zhiyuan Liu**[1†], **Maosong Sun**[1], **Jie Zhou**[3]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Princeton University, Princeton, NJ, USA
[3]Pattern Recognition Center, WeChat AI, Tencent Inc, China
{h-peng17,hanxu17}@mails.tsinghua.edu.cn, tianyug@princeton.edu

A Pre-training Details

Pre-training Dataset We construct a dataset for pre-training following the method in the paper. We use Wikipedia articles as corpus and Wikidata (Vrandečić and Krötzsch, 2014) as the knowledge graph. Firstly, We use anchors to link entity mentions in Wikipedia corpus with entities in Wikidata. Then, in order to link more unanchored entity mentions, we adopt `spaCy`[1] to find all possible entity mentions, and link them to entities in Wikidata via name matching. Finally, we get a pre-training dataset containing 744 relations and 867,278 sentences. We release this dataset together with our source code at our GitHub repository[2].

We also use this dataset for MTB, which is slightly different from the original paper (Baldini Soares et al., 2019). The original MTB takes all entity pairs into consideration, even if they do not have a relationship in Wikidata. Using the above dataset means that we filter out these entity pairs. We do this out of training efficiency, for those entity pairs that do not have a relation are likely to express little relational information, and thus contribute little to the pre-training.

Data Sampling Strategy For MTB (Baldini Soares et al., 2019), we follow the same sampling strategy as in the original paper. For pre-training our contrastive model, we regard sentences labeled with the same relation as a "bag". Any sentence pair whose sentences are in the same bag is treated as a positive pair and as a negative pair otherwise. So there will be a large amount of possible positive samples and negative samples. We dynamically sample positive pairs of a relation with respect to the number of sentences in the bag.

Parameter	MTB	CP
Learning Rate	3×10^{-5}	3×10^{-5}
Batch Size	256	2048
Sentence Length	64	64
$P_{\texttt{BLANK}}$	0.7	0.7

Table 1: Hyperparameters for pre-training models. $P_{\texttt{BLANK}}$ corresponds to the probability of replacing entities with `[BLANK]`.

Hyperparameters We use Huggingface's `Transformers`[3] to implement models for both pre-training and fine-tuning and use AdamW (Loshchilov and Hutter, 2019) for optimization. For most pre-training hyperparameters, we select the same values as Baldini Soares et al. (2019). We search hyperparameter batch size in {256, 2048} and $P_{\texttt{BLANK}}$ in {0.3, 0.7}. For MTB, batch size N means that a batch contains $2N$ sentences, which form $N/2$ positive pairs and $N/2$ negative pairs. For CP, batch size N means that a batch contains $2N$ sentences, which form N positive pairs. For negative samples, we pair the sentence in each pair with sentences in other pairs.

We set hyperparameters according to results on supervised RE dataset TACRED (micro F_1). Table 1 shows hyperparameters for pre-training MTB and our contrastive model (CP). The batch size of our implemented MTB is different from that in Baldini Soares et al. (2019), because in our experiments, MTB with a batch size of 256 performs better on TACRED than the batch size of 2048.

Pre-training Efficiency MTB and our contrastive model have the same architecture as $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019), so they both hold 110M parameters approximately. We use four

[*] Equal contribution
[†] Corresponding author e-mail: liuzy@tsinghua.edu.cn
[1] https://spacy.io/
[2] https://github.com/thunlp/RE-Context-or-Names
[3] https://github.com/huggingface/transformers

Dataset	Train	Dev	Test
TACRED	68,124	22,631	15,509
SemEval	6,507	1,493	2,717
Wiki80	39,200	5,600	11,200
ChemProt	4,169	2,427	3,469
FewRel	44,800	11,200	14,000

Table 2: Numbers of instances in train / dev / test splits for different RE datasets.

Dataset	1%	10%	100%
TACRED	703	6,833	68,124
SemEval	73	660	6,507
Wiki80	400	3,920	3,9200
ChemProt	49	423	4,169

Table 3: Numbers of training instances in supervised RE datasets under different proportion settings.

Nvidia 2080Ti GPUs to pre-train models. Pre-training MTB takes $30,000$ training steps and approximately 24 hours. Pre-training our model takes $3,500$ training steps and approximately 12 hours.

B RE Fine-tuning

RE Datasets We download TACRED from LDC[4], Wiki80, SemEval from OpenNRE[5], ChemProt from sciBert[6], and FewRel from FewRel[7]. Table 2 shows detailed statistics for each dataset and Table 3 demonstrates the sizes of training data for different supervised RE datasets in 1%, 10% and 100% settings. For 1% and 10% settings, we randomly sample 1% and 10% training data for each relation (so the total training instances for 1% / 10% settings are not exactly 1% / 10% of the total training instances in the original datasets). As shown in the table, the numbers of training instances in SemEval and ChemProt for 1% setting are extremely small, which explains the abnormal performance.

Hyperparameters Table 4 shows hyperparameters when finetuning on different RE tasks for BERT, MTB and CP. For CNN, we train the model by SGD with a learning rate of 0.5, a batch size of 160 and a hidden size of 230. For few-shot RE, we use the recommended hyperparameters in FewRel[8].

[4] https://catalog.ldc.upenn.edu/LDC2018T24

[5] https://github.com/thunlp/OpenNRE

[6] https://github.com/allenai/scibert

[7] https://github.com/thunlp/fewrel

[8] https://github.com/thunlp/FewRel

Parameter	Supervised RE	Few-Shot RE
Learning Rate	3×10^{-5}	2×10^{-5}
Batch Size	64	4
Epoch	6	10
Sentence Length	100	128
Hidden Size	768	768

Table 4: Hyperparameters for fine-tuning on relation extraction tasks (BERT, MTB and CP).

Multiple Trial Settings For all the results on supervised RE, we run each experiment 5 times using 5 different seeds $(42, 43, 44, 45, 46)$ and select the median of 5 results as the final reported number. For few-shot RE, as the model varies little with different seeds and it is evaluated in a sampling manner, we just run one trial with 10000 evaluation episodes, which is large enough for the result to converge. We report accuracy (proportion of correct instances in all instances) for Wiki80 and FewRel, and micro F_1[9] for all the other datasets.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL*, pages 2895–2905.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR 2019*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Proceedings of CACM*, 57(10):78–85.

[9] https://en.wikipedia.org/wiki/F1_score