Rescue Implicit and Long-tail Cases: Nearest Neighbor Relation Extraction

Zhen Wan [*1] **Qianying Liu** [*1]
Zhuoyuan Mao[1] **Fei Cheng**[1] **Sadao Kurohashi**[1] **Jiwei Li**[2]
[1] Kyoto University, Japan
[2] Zhejiang University, China
{zhenwan, ying, zhuoyuanmao}@nlp.ist.i.kyoto-u.ac.jp
{feicheng, kuro}@i.kyoto-u.ac.jp
{jiwei_li}@zju.edu.cn

Abstract

Relation extraction (RE) has achieved remarkable progress with the help of pre-trained language models. However, existing RE models are usually incapable of handling two situations: implicit expressions and long-tail relation types, caused by language complexity and data sparsity. In this paper, we introduce a simple enhancement of RE using k nearest neighbors (kNN-RE). kNN-RE allows the model to consult training relations at test time through a nearest-neighbor search and provides a simple yet effective means to tackle the two issues above. Additionally, we observe that kNN-RE serves as an effective way to leverage distant supervision (DS) data for RE. Experimental results show that the proposed kNN-RE achieves state-of-the-art performances on a variety of supervised RE datasets, i.e., ACE05, SciERC, and Wiki80, along with outperforming the best model to date on the i2b2 and Wiki80 datasets in the setting of allowing using DS. Our code and models are available at: https://github.com/YukinoWan/kNN-RE.

1 Introduction

Relation extraction (RE) aims to identify the relationship between entities mentioned in a sentence, and is beneficial to a variety of downstream tasks such as question answering and knowledge base population. Recent studies (Zhang et al., 2020; Zeng et al., 2020; Lin et al., 2020; Wang and Lu, 2020; Cheng et al., 2020; Zhong and Chen, 2021) in supervised RE take advantage of pre-trained language models (PLMs) and achieve SOTA performances by fine-tuning PLMs with a relation classifier. However, we observe that existing RE models are usually incapable of handling two RE-specific situations : **implicit expressions** and **long-tail relation types**.

Implicit expression refers to the situation where a relation is expressed as the underlying message that is not explicitly stated or shown. For example, for the relation "sibling to", a common expression can be "*He* has a brother *James*", while an implicit expression could be "He is the youngest son of Liones, comparing with *Samuel Liones* and *Henry Liones*." In the latter case, the relation "sibling to" between "*Samuel Liones*" and "*Henry Liones*" is not directly expressed but could be inferred from them both are brothers of the same person. Such underlying message can easily confuse the relation classifier. The problem of **long-tail relation types** is caused by data sparsity in training. For example, the widely used supervised RE dataset TACRED (Zhang et al., 2017) includes 41 relation types. The most frequent type "per:title" has 3,862 training examples, while over 22 types have less than 300 examples. The majority types can easily dominate model predictions and lead to low performance on long-tail types.

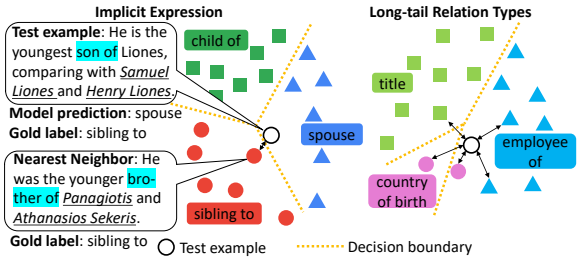

Figure 1: **Left**: the retrieved example has a similar structure but with the phrase "younger brother", it becomes easier to infer. **Right**: Referring to the gold labels of nearest neighbors can reduce the bias. Highlighted words may directly influence on the relation prediction.

Inspired by recent studies (Khandelwal et al., 2020; Guu et al., 2020; Meng et al., 2021) using kNN to retrieve diverse expressions for language generation tasks, we introduce a simple but effective kNN-RE framework to address above-mentioned two problems. Specifically, we store the training examples as the memory by a vanilla RE model and consult the stored memory at test time

[*] This denotes equal contribution.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1731–1738
December 7-11, 2022

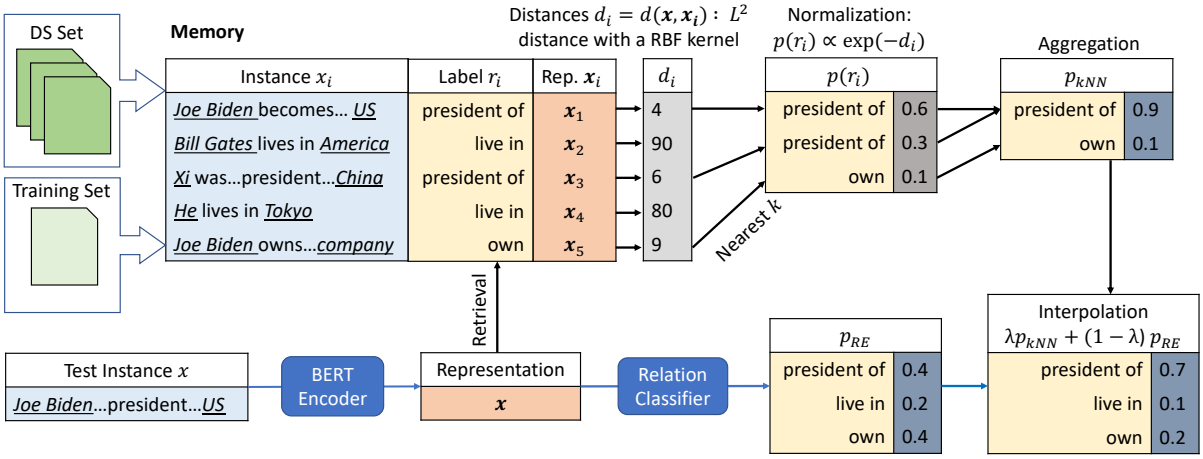

Figure 2: **An illustration of kNN-RE**. The memory is constructed with each pair of relation representations (Rep.) and relation labels from training set or DS set. For inference, the blue line denotes the workflow for vanilla RE and the black line denotes the workflow for kNN.

through a nearest-neighbor search. As shown in Figure 1, for an **implicit expression**, the expression "son of" may mislead to an incorrect prediction while its retrieved nearest neighbor contains a direct expression "brother of", which is a more explicit expression of the gold label "sibling to". The prediction of **long-tail** examples, as shown in Figure 1, is usually biased toward the majority class. Nearest neighbor retrieval provides direct guidance to the prediction by referring to the labels of its nearest neighbors in the training set, and thus can significantly reduce the imbalanced classification.

Additionally, we observe that kNN-RE serves as an efficient way to leverage distant supervision (DS) data for RE. DS augments labeled RE datasets by matching knowledge base (KB) relation triplets and raw text entity pairs in a weak-supervision fashion (Mintz et al., 2009; Lin et al., 2016; Vashishth et al., 2018; Chen et al., 2021). Recent studies (Baldini Soares et al., 2019; Ormándi et al., 2021; Peng et al., 2020; Wan et al., 2022), which apply PLMs to the DS labeled data to improve supervised RE, require heavy computation due to the fact that they require pre-training on DS data, whose size is usually dozens of times that of supervised datasets. To address this issue, we propose a lightweight method to leverage DS data to benefit supervised RE by extending the construction of stored memory for kNN-RE to DS labeled data and outperforming the recent best pre-training method with no extra training.

In summary, we propose kNN-RE: a flexible kNN framework to solve the RE task. We conduct the experiments for kNN-RE with three different memory settings: training, DS, and the combination of training and DS. The results show that our kNN-RE with the training memory obtains a 0.84%-1.15% absolute F1 improvement on five datasets and achieves state-of-the-art (SOTA) F1 scores on three of them (ACE05, SciERC and Wiki80). In the DS setup, kNN-RE outperforms SOTA DS pre-training methods on two datasets (i2b2, Wiki20) significantly without extra training.

Dataset	# Rel.	# Train	# Dev	# Test
ACE05	6	4,788	1,131	1,151
Wiki80	80	45,330	5,070	5,600
TACRED	41	68,124	22,631	15,509
i2b2 2010VA	8	3,020	111	6,147
SciERC	7	1,861	275	551
Wiki20m	80	303K	-	-
MIMIC-III	8	36K	-	-

Table 1: **Statistics of datasets**. Rel. denotes relation types.

2 Methodology

2.1 Background: Vanilla RE model

For the vanilla RE model, We follow the recent SOTA method PURE (Zhong and Chen, 2021). To encode an input example to a fixed-length representation by fine-tuning PLMs such as BERT (Devlin et al., 2019), PURE adds extra marker tokens to highlight the head and tail entities and their types.

Specifically, given an example x: "*He* has a brother *James*.", the input sequence is "[CLS] [H_PER] *He* [/H_PER] has a brother [T_PER] *James* [/T_PER]. [SEP]" where "PER" is the entity type if

Methods	ACE05	Wiki80	TACRED	i2b2 2010VA	SciERC
			Baselines		
(Peng et al., 2019)	-	-	-	**76.2**†	-
(Han et al., 2019)	-	86.61	-	-	-
(Zhou and Chen, 2021)	-	-	**71.5**	-	-
CP (Peng et al., 2020)♣	-	87.50	-	72.84	-
PURE (Zhong and Chen, 2021)	74.00	86.70	69.42	72.28	68.45
			Ours (Best k, λ)		
kNN only: Train memory	**75.07** $(4, 1.0)$	87.35 $(4, 1.0)$	70.21 $(8, 1.0)$	73.18 $(32, 1.0)$	68.58 $(64, 1.0)$
kNN-RE: Train memory	**75.07** $(4, 1.0)$	**87.54** $(4, 0.5)$	70.57 $(8, 0.4)$	73.38 $(32, 0.7)$	**69.47** $(64, 0.6)$
kNN-RE: DS memory ♣	-	87.79 $(256, 0.5)$	-	73.22 $(64, 0.3)$	-
kNN-RE: Combined memory ♣	-	**88.32** $(\alpha = 0.5)$	-	73.67 $(\alpha = 0.6)$	-

Table 2: **Main Results of kNN-RE with different memory settings on five datasets**. ♣ denotes the methods using DS set. †: SOTA i2b2 2010VA adopts specific encoding. "kNN only" means only using $p_{kNN}(y|x)$ and is described by $\lambda = 1$ in the parameters. "Combined" means the combination of both memories by: $\alpha p_{kNN-RE}(Train) + (1 - \alpha)p_{kNN-RE}(DS)$, where $p_{kNN-RE}(Train)$ and $p_{kNN-RE}(DS)$ is computed by Equation 2 corresponding to the "Train memory" and "DS memory.", and k, λ are given by the best setting of each single memory.

Methods	Wiki80	i2b2 2010VA
	Baselines	
CP (Peng et al., 2020)♣	87.32	75.62
PURE (Zhong and Chen, 2021)	85.78	73.45
	Ours (Best k, λ)	
kNN only: Train memory	86.70 $(4, 1.0)$	74.70 $(32, 1.0)$
kNN-RE: Train memory	**87.12** $(4, 0.5)$	**75.20** $(32, 0.7)$
kNN-RE: DS memory♣	88.20$(256, 0.5)$	76.80 $(64, 0.3)$
kNN-RE: Combined memory♣	**88.54** $(\alpha = 0.5)$	**78.25** $(\alpha = 0.6)$

Table 3: **Results on development set.** . ♣ denotes the methods using DS set.

provided. Denote the n-th hidden representation of the BERT encoder as $\mathbf{h}_n$. Assuming i and j are the indices of two beginning entity markers [H_PER] and [T_PER], we define the relation representation as $\mathbf{x} = \mathbf{h}_i \oplus \mathbf{h}_j$ where $\oplus$ stands for concatenation. Subsequently, this representation is fed into a linear layer to generate the probability distribution $p_{RE}(y|x)$ for predicting the relation type.

2.2 Proposed Method: kNN-RE

Training Memory Construction For the i-th training example (x_i, r_i), we construct the *key-value* pair $(\mathbf{x_i}, r_i)$ where the *key* $\mathbf{x_i}$ is the relation representation obtained from the vanilla RE model and the *value* r_i denotes the labeled relation type. The memory $(\mathcal{K}, \mathcal{V}) = \{(\mathbf{x_i}, r_i)|(x_i, r_i) \in \mathcal{D}\}$ is thus the set of all *key-value* pairs constructed from all the labeled examples in the training set $\mathcal{D}$.

DS Memory Construction In this paper, with the awareness of the unique feature of RE to generate abundant labeled data by DS, we extend our method by leveraging DS examples for memory construction. Similar to training memory construction, we build *key-value* pairs for all the DS labeled examples with the vanilla RE model.

Inference Given the test example x, the RE model outputs its relation representation $\mathbf{x}$ and generate the relation distribution $p_{RE}(y|x)$ between two mentioned entities. We then query the memory with $\mathbf{x}$ to retrieve its k nearest neighbors $\mathcal{N}$ according to a distance function $d(.,.)$ by L^2 distance with the KBF kernel. We weight retrieved examples by a softmax function on the negative distance and make an aggregation on the labeled relation types to predict a relation distribution $p_{kNN}(y|x)$:

$$p_{kNN}(y|x) \propto \sum_{(\mathbf{x_i}, r_i) \in \mathcal{N}} \mathbb{1}_{y=r_i} \frac{exp(-d(\mathbf{x}, \mathbf{x_i}))}{\mathcal{T}} \quad (1)$$

where $\mathcal{T}$ denotes a scaling temperature. Finally, we interpolate the RE model distribution $p_{RE}(y|x)$ and kNN distribution $p_{kNN}(y|x)$ to produce the final overall distribution:

$$p_{kNN-RE}(y|x) = \lambda p_{kNN}(y|x) + (1 - \lambda)p_{RE}(y|x) \quad (2)$$

where λ is a hyperparameter.

3 Experiment settings

Supervised Datasets We evaluate our proposed method on five popular RE datasets. Table 1 shows the statistics. ACE05 and TACRED datasets are built over an assortment of newswire and online text. Wiki80 (Han et al., 2019) is derived from Wikipedia crossing various domains. The i2b2

Test example 1	$\mathbf{p}_{\text{RE}}$
He was the captain of ... team that won the *1950 World Cup* after beating ... in the final round match known as the "*Maracanazo*".	part of: 0.3 (√) participant of: 0.7
Retrieved nearest neighbors	**Gold label** $\mathbf{d}_i \rightarrow \mathbf{p}(\mathbf{r}_i)$
Turkish Cypriots are the minority of the island with *Turkish Settlers* from Turkey and ...	part of $1 \rightarrow 0.9$
It was succeeded as Danish representative at *2006 Contest* by *Sidsel Ben Semmane* with ...	participant of $5 \rightarrow 0.03$
Test example 2	$\mathbf{p}_{\text{RE}}$
South Air Command of the *Indian Air Force* is headquartered in the city	subsidiary: 0.478 (√) has part: 0.496
Retrieved nearest neighbors	**Gold label** $\mathbf{d}_i \rightarrow \mathbf{p}(\mathbf{r}_i)$
The artwork depicts the Christian *Holy Family* of *Mary*, Joseph, and the infant Jesus, in an enclosed garden, symbolizing Mary 's virginity.	has part $2 \rightarrow 0.6$
Fidobank (formerly *SEB Bank*) is a bank of Ukraine that until 2012 belonged to the Swedish *SEB Group*.	subsidiary $3 \rightarrow 0.3$

Table 4: **Two implicit test examples from Wiki80**.

Relation type	# Train	# Test	PURE	kNN-RE
per:charges	280	103	75.14	**77.23** (+2.09)
org:founded by	268	68	64.06	**64.12** (+0.06)
per:siblings	250	55	71.84	**72.90** (+1.06)
per:s_a	229	30	64.00	**66.67** (+2.67)
per:c_of_d	227	28	25.00	**30.30** (+5.30)
org:founded	166	37	80.25	**82.05** (+1.80)
per:religion	153	47	65.82	**68.24** (+2.42)
org:p/r_a	125	10	47.62	**50.00** (+3.38)
org:n_of_e/m	121	19	73.68	**74.29** (+0.61)
per:s_o_b	61	8	**66.67**	61.54 (-5.13)

Table 5: **Results on long-tail relation types**. s_a: schools attended, c_of_d: city of death, p/r_a: political/religious affiliation, n_of_e/m: number of employees/members, s_o_b: stateorprovince of birth,.

2010VA dataset collects medical reports while SciERC (Luan et al., 2018) collects AI paper abstracts and annotated relations, specially for scientific knowledge graph construction.

DS Datasets We evaluate our DS memory construction method on two supervised datasets Wiki80 and i2b2 2010VA. For i2b2 2010VA, we generate DS examples from MIMIC-III based on triplets extracted from the training set. For Wiki80 dataset, as it is derived from the Wikidata KB, we leverage the existing DS dataset Wiki20m derived from the same KB which leads to shared relation types.

Refer to Appendix B for implementation details.

4 Results and Analysis

4.1 Main results

Table 2 compares our proposed kNN-RE with previous SOTA methods. For training memory construction, we can observe that: (1) Our approach outperforms vanilla RE model PURE on all datasets and achieves new SOTA performances on ACE05, Wiki80 and SciERC (2) By comparing PURE and "kNN only", we find that the kNN prediction itself has a performance improvement over the vanilla RE model from $+0.13$ on SciERC to $+1.07$ on ACE05. This leads to a large λ for kNN. For DS and combined memory construction, we can observe that: (1) The best λ becomes smaller while k becomes larger which is reasonable as DS suffers from the noise but still benefit the model inference. (2) Compared with the previous best pretraining method CP (Peng et al., 2020) that requires huge computation, our kNN-RE on DS achieves higher performance without extra training and is further improved by combining with the training memory to achieve a 1.62 F1 score improvement over PURE on the Wiki80.

Besides, we also compare performances on the development set of two datasets as shown in Table 3, the experiment results emphasize the consistent improvement of our proposed methods.

4.2 Analysis

Case Study for Implicit Expressions We select two typical test examples to better illustrate the amendment by kNN retrieval as shown in Table 4. For the first example, the implicit relation between "*maracanazo*" and "*1950 world cup*" need to be inferred by other contextual information and the RE model makes an incorrect prediction as competition usually belongs to the object of the relation "participant of" as the second retrieved examples. However, the nearest example contains a simpler expression and rectifies the prediction. Refer to Appendix A for visualized analysis.

For the second example, the implicit expression leads to another confusing relation "subsidiary" while the nearest example captures the same structure that contains an "of" between two entities and makes the final prediction easier.

Performance on Long-tail Relation Types We check the performance kNN-RE with training memory on several most long-tail relation types from the TACRED dataset and show in table 5. Note that all long-tail relation types benefit from the effectiveness of kNN prediction except for "stateorprovince of birth", which contains only 8 test examples leading to an unconvincing performance.

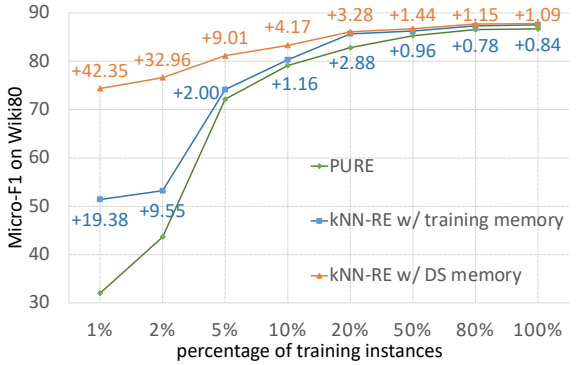

Figure 3: **Analyzing retrieval ability on Wiki80**.

Retrieval Ability in Low-Resource Scenario We also check the retrieval ability by varying the percentage of the training set to constraint the representation quality in the memory (Figure 3). We can observe that with the decreasing number of the training examples, our kNN-RE (training) tends to achieve greater improvement even the training memory is also limited by the low resource. Surprisingly, our kNN-RE (DS) achieves the F1-score of 74.31 (an improvement gap of 42.35 over PURE) with only 1% training examples provided, which indicates that the model can still retrieve accurate nearest neighbors from the DS memory. We believe this is due to the modern PLMs have learned robust representations during pre-training.

5 Conclusion

We propose kNN-RE: a flexible kNN framework with different memory settings for solving implicit expression and long-tail relation issues in RE. The results show that our kNN-RE with training memory outperforms vanilla RE model and achieves SOTA F1 scores on three datasets. In the DS setup, kNN-RE also outperforms SOTA DS pre-training methods significantly without extra training.

Limitations

In this paper, we use kNN-based strategy in the inference stage to address the language complexity and data sparsity problem. It is more challenging for a model to learn the characteristics of these examples. While our approach is light-weighted and flexible, it cannot directly help the model to improve the classification of the implicit expression examples or long-tail relation examples types during the training stage. The representations of these examples remain coarse-grained. Incorporating the kNN manner strategies in the training stage by providing additional nearest neighbor references for the model could help the model learn better representations of the examples, which we leave as future work.

Acknowledgements

This work is partially supported by JST SPRING Grant No.JPMJSP2110, MHLW PRISM Grant Number 21AC5001, KAKENHI Number 21H00308 and KAKENHI Number 22J13719, Japan.

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.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Tao Chen, Haizhou Shi, Siliang Tang, Zhigang Chen, Fei Wu, and Yueting Zhuang. 2021. CIL: Contrastive instance learning framework for distantly supervised relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6191–6200, Online. Association for Computational Linguistics.

Fei Cheng, Masayuki Asahara, Ichiro Kobayashi, and Sadao Kurohashi. 2020. Dynamically updating event representations for temporal relation classification with multi-category learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1352–1357, 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.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-

augmented language model pre-training. *CoRR*, abs/2002.08909.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. 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): System Demonstrations*, pages 169–174, Hong Kong, China. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

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.

Yuxian Meng, Shi Zong, Xiaoya Li, Xiaofei Sun, Tianwei Zhang, Fei Wu, and Jiwei Li. 2021. GNN-LM: language modeling based on global contexts via GNN. *CoRR*, abs/2110.08743.

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.

Róbert Ormándi, Mohammad Saleh, Erin Winter, and Vinay Rao. 2021. Webred: Effective pretraining and finetuning for relation extraction on the web. *CoRR*, abs/2102.09681.

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.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 58–65. Association for Computational Linguistics.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RESIDE: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266, Brussels, Belgium. Association for Computational Linguistics.

Zhen Wan, Fei Cheng, Qianying Liu, Zhuoyuan Mao, Haiyue Song, and Sadao Kurohashi. 2022. Relation extraction with weighted contrastive pre-training on distant supervision.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9507–9514.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Minimize exposure bias of Seq2Seq models in joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, 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.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *CoRR*, abs/2102.01373.

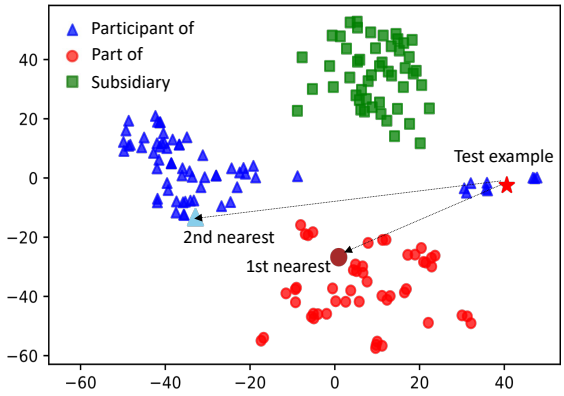

Figure 4: **The implicit test example in t-SNE**. Note that t-SNE only visualizes the distribution not the exact distance.

A T-SNE Visualization

As shown in figure 4, the test example with the gold label "part of" is incorrectly classified to another relation "participant of". However, with the help of the 1st nearest example closer to the "part of" clustering, kNN makes a correct prediction.

B Implementation Details

During the construction of DS data for i2b2 2010VA, we use the preprocessing tool NLTK to split raw corpora into sentences. We use *bert-base-uncased* (Devlin et al., 2019) as the base encoders for ACE05, Wiki80 and TACRED for a fair comparison with previous work. We also use *scibert-scivocab-uncased* (Beltagy et al., 2019) as the base encoder for SciERC and *BLUEBERT* (Peng et al., 2019) for i2b2 2010VA as in-domain PLMs are more effective.

For the baseline PURE (Zhong and Chen, 2021), we follow their single-sentence to keep consistency among datasets as Wiki80 and TACRED are both sentence-level RE datasets. For another baseline CP (Peng et al., 2020), we modify their official implementations to pre-train on our DS set. For our kNN-RE, we choose hyperparameters k , $\mathcal{T}$ and λ by greedy search where k is the power of 2 from 2 to 256, λ and $\mathcal{T}$ both increase from 0 to 1. All experiment results are the average of 3 to 5 times running.

We used 2 NVIDIA RTX3090 for training.