Few-shot Named Entity Recognition with Supported and Dependent Label Representations

Yasuhide Miura
Fujifilm Corporation
yasuhide.a.miura@fujifilm.com

Takumi Takahashi
Fujifilm Corporation
takumi.c.takahashi@fujifilm.com

Abstract

We explore the problem of few-shot named entity recognition (NER) by introducing two ideas to improve label representations. Recently, the use of token representations with a distance metric has been shown to be effective in few-shot NER, and we take an approach to use label representations along with token representations. Firstly, we add *support examples* to a label name (e.g., "person; example: Federic Krupp, Gao, Honecker, Bush, Deverow") when obtaining a label representation. Secondly, we estimate a transition score among labels with a bilinear function among label representations. The proposed approach is evaluated on 4 open few-shot NER datasets and we found that the approach can improve the performance of one-stage few-shot NER.

1 Introduction

The advance of large language models (e.g., BERT, GPT) has brought some of natural language understanding tasks to be tackled with few training samples. One such task that has especially gathered an attention from researchers is named entity recognition (NER) where a simple nearest neighbor classification using an NER model and a distance metric has shown to achieve a moderate performance in a few-shot setting (Yang and Katiyar, 2020). Ma et al. (2022a) proposed a related but slightly different approach where they prepare an additional BERT to encode labels into representations.

We extend an idea to use label representations to improve few-shot NER. A simple approach to obtain label representations is to encode just label names (Ma et al., 2022a), and we add randomly sampled label examples to label names and encode the combined label names and examples to improve label representations. Figure 1 illustrates our approach to use label examples as the supports of label names. In this approach, an input text and all labels are encoded with a dual encoder architecture. For each token, similarities against all labels are calculated to decide a label. The extension to add label examples may seem like a naive approach to improve label representations but this approach follows previous findings to obtain fine-grained label representations. Firstly, in the context of zero-shot NER, Aly et al. (2021) explored the effectiveness of using label descriptions to encode labels. They have found that the use of label name is a strong baseline to represent a label and a label description can further improve the performance of zero-shot NER depending on its quality. One downside of using label descriptions is that fine-grained descriptions are not always available in an NER dataset. Secondly, an approach to use multiple examples to estimate a label is a well-known approach of Prototypical Networks (Snell et al., 2017). In a typical Prototypical Networks setting, a label prototype can be represented as an average of multiple examples.

We further extend the approach to use label representations in few-shot NER by estimating a label dependency between two labels. A straightforward approach to model label dependencies in NER is to add a Conditional Random Field (CRF, Lafferty et al., 2001) layer after a token encoder (Lample et al., 2016). However, this CRF layer is known to be difficult to transfer since it directly learns a $K \times K$ transition matrix over K labels (Yang and Katiyar, 2020; Hou et al., 2020). We estimate a transition score between two labels with a trainable function which maps two label representations into a single scalar score. We show that this estimation works quite effectively in the dual encoder architecture.

In summary, the contributions of this paper are the followings:

1. We propose an approach to sample *support examples* to improve label representations for few-shot NER and confirm its effectiveness on 4 few-shot NER datasets.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 385–391

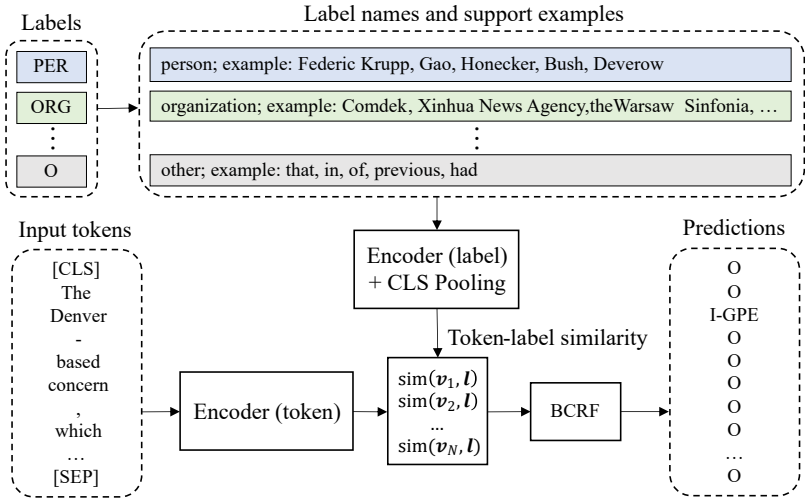

Figure 1: The overview of our approach to encode label representations by adding randomly sampled *support examples*. *Encoder* is a large language model such as BERT, *sim* is a similarity function among representations, $\boldsymbol{v}_n$ is the token representation of n-th token, $\boldsymbol{l}$ is label representations and *BCRF* is Bilinear-trainsition CRF (§3.3).

2. We define a trainable function between two label representation to estimate a transition score of the two labels and show its transfer capability in a few-shot setting.

2 Related Work

2.1 One-stage Few-shot NER

The use of distance metric has shown to be effective in a few-shot setting, where Wiseman and Stratos (2019) and Yang and Katiyar (2020) found that a nearest neighbor search among token representations can be a promising approach for few-shot NER. Fritzler et al. (2019) and Hou et al. (2020) have explored Prototypical Networks to model token-level entity prototypes in few-shot NER. These ideas are further investigated to train a model with a contrastive learning objective (Das et al., 2022). Like these approaches, our approach is in the paradigm of one-stage few-shot NER where named entities are recognized simply as the labels of input tokens.

2.2 Two-stage Few-shot NER

Recently, the paradigm of two-stage few-shot NER (Wang et al., 2022a) where entity spans are extracted in the first stage and their types are recognized in the second stage are investigated to extend the one-stage few-shot NER. In this paradigm, span or entity prototypes are defined (Wang et al., 2022b; Ji et al., 2022; Ma et al., 2022b; Wang et al., 2022a) to achieve stronger performances with the complexity of an additional stage.

2.3 Label Representation in NER

The use of label description has been also explored in a low-resource NER. Aly et al. (2021) explored the effect of label description in a zero-shot NER and Wang et al. (2021) has utilized label descriptions along with entity representations in a few-shot and a zero-shot NER. Ma et al. (2022a) has shown that simply using label names is quite effective in few-shot NER. Our approach extends these ideas to use support samples to improve label representation in few-shot NER.

3 Method

3.1 Dual Encoder Model

We followed the approach of dual encoders that was taken by Aly et al. (2021) and Ma et al. (2022a) as the base architecture of our model. As shown in Figure 1, we prepare an encoder for an input tokens and an encoder for labels. Given input tokens $\boldsymbol{u}_I$, the tokens are encoded with a language model $\boldsymbol{v} = \mathrm{LM}_{\mathrm{token}}(\boldsymbol{u}_I)$. The tokens of given labels $\boldsymbol{u}_L$ are similarly encoded with another language model $\boldsymbol{m} = \mathrm{LM}_{\mathrm{label}}(\boldsymbol{u}_L)$, and the representations of CLS[1] are pooled as label representations $\boldsymbol{l}$. For each token representation $\boldsymbol{v}_n$, similarities against all label representations are calculated with a similarity function as $\boldsymbol{o}_n = \mathrm{sim}(\boldsymbol{v}_n, \boldsymbol{l})$. These token-label similarities are used to calculate a loss against true labels. As done in previous studies, we first pre-finetune this model on a large scale NER

[1]A special token of a language model that is prepended to an input text.

Algorithm 1 Support example sampling

Require: input_text x, label y, training_texts $\boldsymbol{X}$, # of example n

$S \leftarrow \phi$
while $|S| < n$ **do**
  // Sample a text including y from $\boldsymbol{X}$
  $x_y \leftarrow \text{sample\_text}(\boldsymbol{X}, y)$
  if $x_y \neq x$ **then**
    if $y \neq \text{O}$ **then**
      $s_y \leftarrow \text{sample\_entity}(x_y)$
    else
      $s_y \leftarrow \text{sample\_word}(x_y)$
    end if
    $S \leftarrow S \uplus \{s_y\}$
  end if
end while
return S

dataset (e.g., OntoNotes 5.0) and then fine-tune it on a few-shot NER dataset.

3.2 Support Example Sampling

The dual encoder model encodes label tokens to obtain label representations. Our idea improves label representations by extending label tokens with support examples. Algorithm 1 shows processes to sample support examples S for input text x and label y. In the case of *PER* label in Figure 1, $n = 5$ examples of *Federic Krupp*, *Gao*, *Honecker*, *Bush* and *Deverow* are sampled from entire training data $\boldsymbol{X}$[2]. These examples are then combined with the label name *person* with the fixed text snippet of "; example:"[3]. One exceptional label that needs to be considered in this sampling is *O* label. Since *O* is not a label for named entity and does not have an entity boundary, we decided to samples a non-entity word from text x_y.

3.3 Estimation of Transition Score

Lample et al. (2016) have shown that a CRF layer can be added to a token encoder to improve NER. However, the few-shot transfer of this CRF layer is known to be difficult since the prediction score is defined as $s(\boldsymbol{x}, \boldsymbol{y}) = \sum_i A_{y_i,y_{i+1}} + \sum_i P_{i,y_i}$ where i is a token index, $\boldsymbol{A}$ is a transition matrix and $\boldsymbol{P}$ is an emission matrix. A typical approach to realize $\boldsymbol{A}$ is to prepare a trainable $K \times K$ matrix when there are K labels. We estimate a transition score among two labels $A_{y_i,y_{i+1}}$ simply with a bilinear function as

$$A_{y_i,y_{i+1}} = \boldsymbol{l}_i^T W \boldsymbol{l}_{i+1} + b \quad (1)$$

where W is a trainable weight matrix of size $D \times D$, b is a bias and D is the embedding size of the label encoder. We call a score estimated with this approach **B**ilinear-transition **CRF** (BCRF) score. The estimation of a transition score has been investigated in a more resource rich setting in Hu et al. (2020). Our BCRF score takes a simple estimation approach since we focus on a resource poor few-shot setting.

4 Experiment

4.1 Datasets and Baselines

We evaluate the effectiveness of our approach using 4 datasets: Few-NERD (Ding et al., 2021), CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), WNUT-2017 (Derczynski et al., 2017) and i2b2-2014 (Stubbs and Özlem Uzuner, 2015). Few-NERD is a large-scale dataset specialized for the evaluation of few-shot NER. CoNLL-2003, WNUT-2017 and i2b2-2014 are datasets that were used in a few-shot domain transfer setting in previous studies (Yang and Katiyar, 2020; Ma et al., 2022a; Das et al., 2022; Ji et al., 2022).

We compare our approach against various state-of-the-art one-stage baselines (§2.1) and two-stage baselines (§2.2). For the one-stage baselines, we compare against ProtoBERT (Snell et al., 2017), StructShot (Yang and Katiyar, 2020), LabelSem (Ma et al., 2022a) and CONTaiNER (Das et al., 2022). For the two-stage baselines, we compare against ESD (Wang et al., 2022b), MAML-ProtoNet (Ma et al., 2022b), EPNet (Ji et al., 2022) and SpanProto (Wang et al., 2022a). For the scores of ProtoBERT, StructShot and CONTaiNER, we refer to the values reported in Das et al. (2022). For the scores of LabelSem, ESD, MAML-ProtoNet, EPNet and SpanProto, we refer to the values reported in the original papers.

4.2 Model Configuration

We first pre-finetuned the dual encoder model (§3.1) on a large-scale NER dataset. The training section is used for Few-NERD and OntoNotes 5.0 (Weischedel et al., 2013) is used for CoNLL-2003, WNUT-2017 and i2b2-2014. BERT base (cased) is used as language models and dot product is used as the similarity function of the model. IO scheme is

[2]This sampling is done for every training batch. This random process is important for the dual encoder model to avoid overfitting to certain examples.

[3]In 1-shot NER, this algorithm can fail to sample examples. In such case, we used the text snippet of "; example: none".

		INTRA				INTER			
		5-way		10-way		5-way		10-way	
Stage	Model	1~2-S	5~10-S	1~2-S	5~10-S	1~2-S	5~10-S	1~2-S	5~10-S
one	ProtoBERT	23.45	41.93	19.76	34.61	44.44	58.80	39.09	53.97
	StructShot	35.92	38.83	25.38	26.39	57.33	57.16	49.46	49.39
	CONTaiNER	40.40	53.71	33.82	47.51	56.10	61.90	48.36	57.13
	DualEnc++$_{proposed}$	50.81	64.20	**46.89**	58.64	63.98	72.04	62.31	69.95
	-BCRF	49.51	61.09	43.90	54.86	61.56	69.88	58.90	67.10
	-SupEx	49.71	63.95	46.26	58.41	62.76	72.06	61.62	69.95
	-SupEx, +BCRF[R]	17.14	39.55	12.51	29.39	23.07	51.83	17.06	42.27
two	ESD	36.08	52.14	30.00	42.15	59.29	69.06	52.16	64.00
	MAML-ProtoNet	52.04	63.23	43.50	56.84	68.77	71.62	63.26	68.32
	EPNET	43.36	58.85	36.41	46.40	62.49	65.24	54.39	62.37
	SpanProto	**54.49**	**73.10**	45.39	**64.63**	**73.36**	**82.68**	**66.26**	**78.69**

Table 1: The episode evaluation F_1 scores on Few-NERD over 5000 episodes. The shaded models are the proposed model with alternative configurations (§5): -SupEx is without support examples, -BCRF is trained on cross-entropy loss, +BCRF[R] is trained on BCRF with randomly initialized label embeddings. The bold values are the best scores and the underlined values are the second-best scores for each N-way K-shot setting.

		1-shot			5-shot		
Stage	Model	CoNLL	WNUT	i2b2	CoNLL	WNUT	i2b2
one	ProtoBERT	$49.9_{\pm 8.6}$	$17.4_{\pm 4.9}$	$13.4_{\pm 3.0}$	$61.3_{\pm 9.1}$	$22.8_{\pm 4.5}$	$17.9_{\pm 1.8}$
	StructShot	$62.4_{\pm 10.5}$	$24.2_{\pm 8.0}$	$21.4_{\pm 3.5}$	$74.8_{\pm 2.4}$	$30.4_{\pm 6.5}$	$30.3_{\pm 2.1}$
	CONTaiNER	$61.2_{\pm 10.7}$	$27.5_{\pm 1.9}$	$21.5_{\pm 1.7}$	$75.8_{\pm 2.7}$	$32.5_{\pm 3.8}$	$36.7_{\pm 2.1}$
	LabelSem	$(68.4_{\pm 6.7})$	$(38.3_{\pm 1.7})$	$(61.9_{\pm 4.3})$	$(76.6_{\pm 2.1})$	$(40.8_{\pm 2.1})$	$(76.8_{\pm 2.0})$
	DualEnc++$_{proposed}$	$\mathbf{71.0}_{\pm 3.8}$	$\mathbf{36.1}_{\pm 4.9}$	$\mathbf{44.4}_{\pm 5.1}$	$74.8_{\pm 5.1}$	$\mathbf{40.3}_{\pm 2.3}$	$\mathbf{46.5}_{\pm 4.9}$
two	EPNet	$64.8_{\pm 10.4}$	$32.3_{\pm 4.8}$	$27.5_{\pm 4.6}$	$\mathbf{78.8}_{\pm 2.7}$	$38.4_{\pm 5.2}$	$44.9_{\pm 2.7}$

Table 2: The F_1 scores on CoNLL-2003, WNUT-2017 and i2b2-2014. The scores of the proposed models are average with standard deviation on 10 different K-shot samples. The bold values are the best scores and the underlined values are the second-best scores for each K-shot setting with the greedy sampling algorithm (Yang and Katiyar, 2020). The values with parenthesis are the scores with the downsampling algorithm (Hou et al., 2020).

used as the tagging scheme of NER. The Viterbi algorithm is used to decide the best label sequence as in the decoding process of CRF. The further detail of the training configuration, label configuration and dataset statistics are shown in §A.1, §A.2 and §A.3, respectively.

4.3 Evaluation

Table 1 shows the result on Few-NERD over INTRA and INTER configurations. In Few-NERD, coarse-grained entity types are shared (INTER) or not shared (INTRA) among the training data and the test data. DualEnc++ has shown best scores on all 8 settings against one-stage previous models. For the comparison against two-stages models, DualEnc++ has shown best or second-best scores on 5 settings. Table 2 shows the results on CoNLL-2003, WNUT-2017 and i2b2-2014.

5 Discussions

5.1 The Effects of Support Examples and BCRF

We examined the effects of two ideas with an ablation study on Few-NERD. The -SupEx and -BCRF scores in Table 1 shows the result of ablation study. BCRF and support examples have shown effective on all 8 settings. We further confirmed the performance of BCRF without the label encoder by using randomly initialized label embeddings as in Hu et al. (2020) (-SupEx, +BCRF[R]). The low performance of this setting indicates the strength of BCRF combined with the label encoder.

5.2 The Effects of n Support Examples

The labels encoder encodes n support examples to obtain label representations. In the experiment (§4.3), we chose $n = 5$ so that our approach can consider enough examples in the 5~10 shot settings of Few-NERD. We have additionally tried $n = 1, 3$ on Few-NERD and found the result to be quite stable regardless of the value of n. The standard deviation of F_1 score was largest on INTRA 10-way 1~2 shot with the value of 47.27 ± 0.33. The more detailed effect of the number of support examples can be confirmed in §A.4.

6 Conclusion

We proposed two ideas to improve label representations that can be effective for few-shot NER. These ideas have shown effectiveness to achieve strong performances compared against previous one-stage approaches and comparable performances to some of two-stage approaches. As future work, we would like to explore whether these ideas can be applied to span representations which have shown superiority compared to simpler token representations that we have explored in this study.

Limitations

Our approach has shown strong performances on 4 widely used few-shot NER datasets. Additional datasets and transfer settings have been tested in previous studies (Fritzler et al., 2019; Yang and Katiyar, 2020; Wang et al., 2021; Ma et al., 2022a; Das et al., 2022; Ma et al., 2022b; Ji et al., 2022) and our approach can be suboptimal on them. The result of few-shot domain transfer settings in CoNLL-2003, WNUT-2017 and i2b2-2014 depends on randomly sampled few-shot samples. Since these random samples differ among our approach and previous studies, the comparison is not a fair comparison in an exact manner. This variance in random samples is alleviated in Few-NERD since the episode evaluation use pre-sampled 5000 episodes. The evaluation of our approach requires certain amount of computational resources to run, especially in Few-NERD. Even though a single episode evaluation can be done quite quickly (e.g., 3 minutes), the full evaluation on Few-NERD will take $3 \times 5000 \times 8$ minutes $\approx$ 2000 hours on single gpu.

Ethics Statement

The language resources used in our paper are all publicly available from the corresponding websites. The licenses of the resources are: CC BY-SA 4.0 for Few-NERD, LDC User Agreement for Non-Members[4] for OntoNotes 5.0, CoNLL-2003 license[5] for CoNLL-2003, CC-BY 4.0 for WNUT-2017 and i2b2 Data Use Agreement[6] for i2b2-2014. The resources consist of online encyclopedia (Few-NERD), newswire (CoNLL-2003, OntoNotes 5.0), broadcast news (OntoNotes 5.0), broadcast conversation (OntoNotes 5.0), telephone conversation (OntoNotes 5.0), web data (OntoNotes 5.0), social media (WNUT-2017) and clinical narratives (i2b2-2014). Protected health information in the clinical narratives are de-identified and we have made the agreement with the data provider on a research and development use of them.

[4] https://catalog.ldc.upenn.edu/LDC2013T19

[5] https://www.clips.uantwerpen.be/conll2003/ner/

[6] https://n2c2.dbmi.hms.harvard.edu/data-sets

References

Rami Aly, Andreas Vlachos, and Ryan McDonald. 2021. Leveraging type descriptions for zero-shot named entity recognition and classification. 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 1516–1528, Online. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. 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 3198–3213, Online. Association for Computational Linguistics.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 993–1000, New York, NY, USA. Association for Computing Machinery.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

Zechuan Hu, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. An investigation of potential function designs for neural CRF. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2600–2609, Online. Association for Computational Linguistics.

Bin Ji, Shasha Li, Shaoduo Gan, Jie Yu, Jun Ma, Huijun Liu, and Jing Yang. 2022. Few-shot named entity recognition with entity-level prototypical network enhanced by dispersedly distributed prototypes. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1842–1854, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022a. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971, Dublin, Ireland. Association for Computational Linguistics.

Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. 2022b. Decomposed meta-learning for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1584–1596, Dublin, Ireland. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022a. SpanProto: A two-stage span-based prototypical network for few-shot named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3476, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022b. An enhanced span-based decomposition method for few-shot sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5012–5024, Seattle, United States. Association for Computational Linguistics.

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021. Learning from language description: Low-shot named entity recognition via decomposed framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1618–1630, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Hovy Eduard, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

A Appendix

A.1 Training Configuration

The model is pre-finetuned for 3 epochs and is further fine-tuned on a few-shot dataset for 100 epochs. We optimized the model with support examples (§3.2) and BCRF scores (§3.3) using AdamW (Loshchilov and Hutter, 2019). The learning rate of the optimization is set to $1e^{-5}$ following Ma et al. (2022a) with linear warmup throughout pre-finetuning and fine-tuning for BERT-base

Dataset	#train	#dev	#test
OntoNotes 5.0	59.9K	8.5K	8.3K
Few-NERD	131.8K	18.8K	37.6K
CoNLL-2003	14.0K	3.3K	3.5K
WNUT-2017	3.4K	0.8K	1.1K
i2b2-2014	51.5K	23.2K	48.5K

Table 3: The number of sentences included in the datasets of the experiment (§4).

(cased). Transformers library[7] and PyTorch[8] are used to implement the proposed model. The number of support examples is set to $n = 5$. NVIDIA Tesla V100 with 32GB memory is used to train and evaluate the proposed model. The training time of the proposed model is short: 1–8 minutes for a 100 epochs fine-tuning on a target dataset.

A.2 Label Configuration

The model uses label names along with support examples to obtain label representations (§3.2). We used the label names defined in Ma et al. (2022a) for CoNLL-2003, WNUT-2017 and i2b2-2014. For example, "person" is used as the label name of "PER" in CoNLL-2003. We combined the coarse type and the fine type of a named entity with hyphen in Few-NERD which is available in Table 8 of Ding et al. (2021). For example, "Location-GPE" is used for the named entity with the coarse type of "Location" and the fine type of "GPE". We additionally prepared "start of sentence" and "end of sentence" label names for BCRF which are used in the first token of a sentence and the last token of a sentence, respectively.

A.3 Dataset Statistics

Table 3 shows the number of sentences included in OntoNotes 5.0, Few-NERD, CoNLL-2003, WNUT-2017 and i2b2-3014. For OntoNotes 5.0, we used the splits of CoNLL-2012[9] following the setting of Yang and Katiyar (2020). The language of the datasets is English for all datasets. All datasets are designed to evaluate NER, and Few-NERD is specifically designed for few-shot settings. Note that the actual training splits of the experiment (§4) are samples of the training split in Table 3.

[7] https://huggingface.co/transformers
[8] https://pytorch.org/
[9] http://conll.cemantix.org/2012/data.html

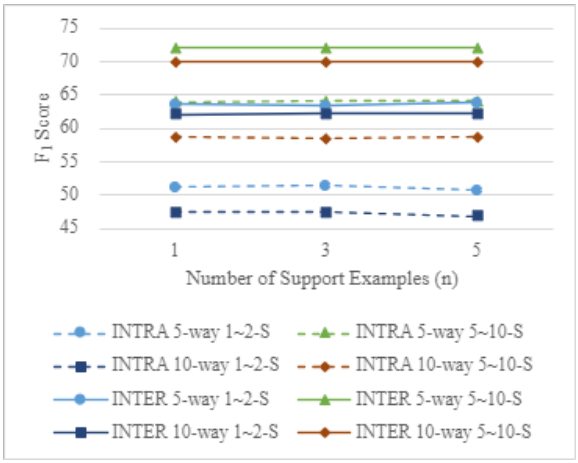

Figure 2: The F_1 scores on Few-NERD with varying number of support examples.

A.4 The Number of Support Examples and Its Effects

Figure 2 shows the changes in F_1 score when the number of support examples are in $n = 1, 3, 5$. The performance is quite stable regardless of the value of n in our approach, and the standard deviation of F_1 score was largest on INTRA 10-way 1~2 with the value of 47.27 ± 0.33 and was smallest on INTRA 10-way 5~10 shot with the value of 58.62 ± 0.03.