Reconciliation of Pre-trained Models and Prototypical Neural Networks in Few-shot Named Entity Recognition

Youcheng Huang[♠♡], **Wenqiang Lei**[♠†], **Jie Fu**[♣], **Jiancheng Lv**[♠]
♠ College of Computer Science, Sichuan University
♣ Beijing Academy of Artificial Intelligence
♡ laerster@gmail.com † wenqianglei@gmail.com

Abstract

Incorporating large-scale pre-trained models with the prototypical neural networks is a *de-facto* paradigm in few-shot named entity recognition. Existing methods, unfortunately, are not aware of the fact that embeddings from pre-trained models contain a prominently large amount of information regarding word ***frequencies***, biasing prototypical neural networks against learning word ***entities***. This discrepancy constrains the two models' synergy. Thus, we propose a one-line-code normalization method to reconcile such a mismatch with empirical and theoretical grounds. Our experiments based on nine benchmark datasets show the superiority of our method over the counterpart models and are comparable to the state-of-the-art methods. In addition to the model enhancement, our work also provides an analytical viewpoint for addressing the general problems in few-shot name entity recognition or other tasks that rely on pre-trained models or prototypical neural networks.[1]

1 Introduction

Named entity recognition (NER) is a classical task in natural language processing (NLP) which aims to automatically identify entities in the plain text by classifying each word to a set of pre-defined entities, *e.g.* "person/location", or to the "others" (no-entity) (Yadav and Bethard, 2019). As a crucial sub-component of many language understanding tasks, NER has been widely adopted to different applications, *e.g.* news (Sang and De Meulder, 2003) and the medical (Stubbs and Uzuner, 2015).

Neural networks (NNs) have achieved great success in NER (Lample et al., 2016). However, NNs face the adaptation challenge (Wilson and Cook, 2020) as words in different entities can change to a great extent (Yang and Katiyar, 2020), *e.g.* "Mr. Bush" in the "person" *v.s.* "budgets" in the "money", and obtaining sufficient annotations of new entities can be expensive (Ding et al., 2021). Few-shot NER, a cost-efficient solution, aims at training a model to be aware of unseen entities given few labeled examples (Huang et al., 2021). Few-shot NER has received a rising interest in the NLP community, where new datasets (Ding et al., 2021) and methods (Das et al., 2022; Yang and Katiyar, 2020; Tong et al., 2021) have been constantly proposed.

Low-dim manifold encodes more adaptive information (Wang et al., 2018). Prototypical neural networks (PNNs) (Snell et al., 2017) learn an embedding space where the same-entity datapoints are clustered around a center, called the prototype, and distances between the query data to all prototypes represent its entity probabilities. In addition to using an embedding network, PNNs calculate the prototypes and distances *via* a *non-parameteric* algorithm, gaining popularity for the flexibility and low computing cost (Wang et al., 2020). A supplementary enhancement will be using embeddings from large-scale pre-trained models (PTMs), like BERT (Devlin et al., 2019), to provide extra knowledge that helps PNNs' learning of entities. As such, incorporating PTMs with PNNs has become a de-facto paradigm for few-shot NER that achieves competitive results to state-of-the-arts (Ding et al., 2021; Huang et al., 2021; Bao et al., 2020). Related works consider NER-specific properties (Tong et al., 2021) or new learning algorithms (Das et al., 2022; Yang and Katiyar, 2020) to enhance the model, but they tend not to examine the coordinating effects between PTMs and PNNs in terms of the information contained in embeddings.

It should be reminded that PNNs calculate distances between word embeddings and prototypes to represent entity probabilities. However, PTMs embeddings may not effectively provide entity information as they prominently contain information on word frequencies (Mu and Viswanath, 2018; Li

† Correspondence to Wenqiang Lei.

[1]Our code is available at https://github.com/HamLaertes/EMNLP_2022_Reconciliation

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 1793–1807

et al., 2020b), and we find frequencies are shallow statistics that can cause a loss of in-depth and useful entity-denoting information. By probing into PNNs, we find that words tend to be classified to the entity centred with words of higher frequencies. Therefore, the distance measure is biased towards focusing on frequencies. Such a bias can cause the over-fitting of the PNNs and the unreliability on classifying new entities. As a consequence, when frequencies are changed on a new corpus, the distances can no longer effectively represent the entity probabilities.

Form a mathematical view, the biased distance is mainly caused by the varying prototype $\ell 2$-norms. However, we argue that those $\ell 2$-norms contribute little to but actually undermine the correct classification. We propose to normalize all prototypes to unit vectors as a simple yet effective remedy to reconcile PNNs and PTMs for few-shot NER. Our experiments on nine few-shot NER datasets (Huang et al., 2021; Ding et al., 2021) demonstrate the effectiveness of our one-line-code remedy. The normalized PNNs achieve competitive results compared to the state-of-the-art methods while retaining all the PNNs' advantages, such as easy implementation and low computation cost. We also demonstrate normalization can make PNNs learn more effectively about correctly classifying entities, and conduct ablation studies on different normalization strategies.

Our study on reconciling PTMs and PNNs, and the promising performance of the simple normalization method may inspire new research motivations to the few-shot NER, as well as other fields that involve the use of PTMs /or PNNs.

2 Background and Related Works

2.1 Few-shot Classification and Embedding-based Classifiers

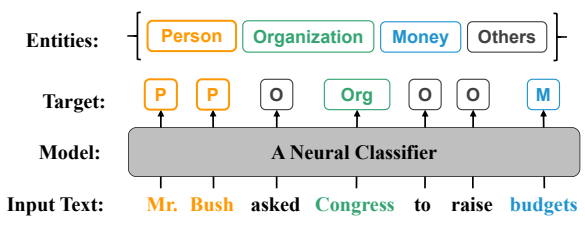

Figure 1: An example of the input and output of NER.

Named entity recognition can be formalized as the word classification (Figure 1). For few-shot classification (FSC), "K-way N-shot" describes the task setting: after the training, the model needs to classify the query data to K training-unseen classes, given N labeled examples per class. The core issue in FSC is the unreliable empirical loss minimization: as the labeled data is extremely limited during testing, the loss defined on new classes will result in a sub-optimal solution that may lead to undesired performance (Wang et al., 2020).

To tackle this issue, researchers seek solutions with the embedding-based methods (Wang et al., 2020; Koch et al., 2015; Vinyals et al., 2016; Sung et al., 2018; Snell et al., 2017). Specifically, an embedding network projects datapoints to a low-dim manifold that contains some general features shared among training and testing classes. On the embedding space, to train only a small classifier for new classes consumes fewer data and can achieve equivalently good results. The recent embedding-based classifiers with meta-learning (Hochreiter et al., 2001) divides the training data into several "episodes" mimicking the "K-way N-shot" testing format. Such a method is popularly known for its effectiveness in FSC.

2.2 Prototypical Neural Network

PNNs (Snell et al., 2017) assume in the embedding space, the same-class datapoints are clustered around class centers, called the prototypes, and the distances between datapoints to prototypes represent their class probabilities. Based on this assumption, PNNs need only calculate: 1) the prototypes using the embedded labeled data, and 2) the distances between the embedded query data and prototypes to conduct the classification. The detailed discussions about PNNs will be presented in section 3 and 5. Utilizing large-scale PTMs as the embedding networks, PNNs can achieve competitive results in various natural language FSC tasks (Ding et al., 2021; Holla et al., 2020; Huang et al., 2021; Bao et al., 2020).

In NER, recent works consider a bunch of methods to enhance the coordinating usage of PTMs and PNNs, including in-domain pre-training (Huang et al., 2021), NER specific properties (Tong et al., 2021), and sophisticated learning algorithm (Das et al., 2022; Yang and Katiyar, 2020). However, to best of our knowledge, little has been explored for the correct combination of PTMs and PNNs. There have been works that find both the small-scale (Mikolov et al., 2013; Pennington et al., 2014) and recent large-scale (Devlin et al., 2019; Liu

et al., 2019) PTMs have limitations in representing diverse language semantics (Mu and Viswanath, 2018; Yang et al., 2018; Gao et al., 2018; Li et al., 2020b). Such limitations may prevent PNNs from correctly adopting entity information, reducing the possibility of getting optimal results.

3 Distance in Prototypical Neural Networks

In this section, we describe PNNs' feed-forward propagation from the mathematical viewpoint focusing on the PNNs' distance function. In K-way N-shot, let $\mathbb{S}_k$ denote the small support set containing N labeled examples with the class k. PNNs calculate the prototype of each class through mean-aggregating the embedded support examples:

$$\mathbf{c}_k = \frac{1}{|\mathbb{S}_k|} \sum_{(\mathbf{x}_i \in \mathbb{S}_k)} f_\phi(\mathbf{x}_i) \tag{1}$$

where f_ϕ is the embedding network. The class probabilities of a query data $\mathbf{x}$ are given by a distance function d following a softmax operation:

$$p_\phi(\mathbf{y} = k \mid \mathbf{x}) = \frac{exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_k)}{\sum_{k'} exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_{k'})} \tag{2}$$

Theorem 1. *Assume data embeddings of the support and query set are independent and identically distributed. Let $\mathbf{c}_k$ be the class prototype calculated by an aggregation function $proto(\cdot) : \prod_{i=1}^{N} \mathbf{H}_i \mapsto \mathbf{h} \in \mathbf{H}$, the problem: $min_{proto(\cdot)} J$, where J is the classification loss, achieves minimization given by $proto(\cdot)$ being the arithmetic mean.*

Corollary 1.1. *Based on the support set, PNNs estimate a Gaussian distribution $\mathbf{N}_k(\mathbf{c}_k, \sigma^2)$ for the embeddings in the class k (σ is a constant vector). The corresponding choice of the Bregman divergence d should be the squared Euclidean distance.*

Proofs are provided in the Appendix B. While d is proposed to be any Bregman Divergence (Banerjee et al., 2005; Snell et al., 2017), we prove the optimal distance function should be the squared Euclidean distance: $\|\mathbf{z} - \mathbf{z}'\|^2$. [2]

PNNs consider that distances between embeddings and prototypes represent the entity probabilities. Therefore, we count on the distance to capture the sharing entity information between the word and the prototypes. Factorizing the distance $(-\|f_\phi(\mathbf{x}) - \mathbf{c}_k\|^2)$ to $(-\|f_\phi(\mathbf{x})\|^2 + 2f_\phi(\mathbf{x})^T\mathbf{c}_k - \|\mathbf{c}_k\|^2)$, the entity probabilities are not only proportional to the dot production, but are also reversely proportional to the two $\ell 2$-norms. While $\|f_\phi(\mathbf{x})\|^2$ represents query data information, different $\|\mathbf{c}_k\|^2$ implies part of the probabilities are priorly determined, and the word is more likely to be classified to the entity that has the smaller prototype $\ell 2$-norm. Unfortunately, because of the representation degeneration of PTMs, these priorly determined probabilities tend to introduce non-entity information, and bias the PNNs' distance towards frequencies.

4 Representation Degeneration of Pre-trained Models

In this section, we introduce the concept of representation degeneration in PTMs and explain its associated effects to PNNs. Small-scale PTMs, like GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013), are argued by researchers as low-capacity models for representing the richness in semantics of natural languages (Yang et al., 2018; Zhao et al., 2018). Both theoretical (Gao et al., 2018) and empirical (Mu and Viswanath, 2018) results in literature have proven: the learned word embeddings contain substantial non-semantic statistics information, *i.e.* the frequencies of the words, causing a lower performance on various downstream tasks, like the task of word classification (Mu and Viswanath, 2018).

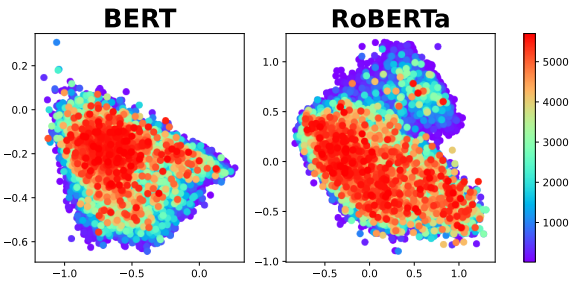

Figure 2: The first two coefficients of PCA analysis on the word embeddings. Color represents frequencies. The deep colors are clustered.

Recent Transformer (Vaswani et al., 2017)-based large-scale PTMs (Devlin et al., 2019; Liu et al., 2019) are groundbreaking in modeling natural language. However, we are concerned that the learned embeddings might also contain the information regarding the non-semantic word frequencies. In line

[2] According to corollary 1.1, PNNs require the embeddings to follow Gaussian distribution. Similarly, works (Yang et al., 2020; Hu et al., 2022) empirically follow the assumption and propose corresponding embedding post-processions to achieve performance gains.

with (Mu and Viswanath, 2018), we use the online statistics data[3], get the word embeddings from BERT and RoBERTa, and do principal component analysis (PCA) to extract the first two coefficients, and plot them on point diagrams. Figure 2 displays the result. The results show both models' embeddings have correlation to frequencies.

In addition, (Li et al., 2020b) finds in the embedding space, word embedding $\ell 2$-norms are inversely proportional to their frequencies. As in PNNs, the prototypes are the mean-aggregation of the words in the support set. Therefore, the prototype $\ell 2$-norms are also correlated to word frequencies as well as the priorly determined probabilities we find in section 3. However, we hypothesize that word frequencies are shallow statistics that are irrelevant to word entities, and the priorly determined probabilities represent little entity information.

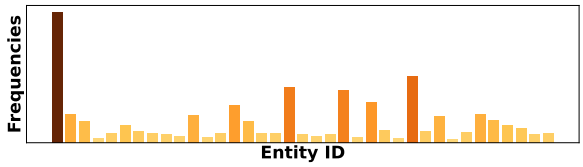

Figure 3: A bar chart displaying the mean word frequencies of different entities. The deeper the color, the larger the mean frequency.

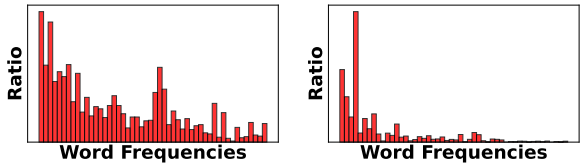

Figure 4: A histogram displaying the word frequencies in two entities.

We empirically demonstrate the irrelevance between entities and frequencies in this section. We will demonstrate the irrelevance between prototype $\ell 2$-norms and entities in the next section. In a few-shot NER dataset (Ding et al., 2021), we count the mean word frequencies of different entities and the frequencies of each word in two random sampled entities.[4] Figure 3 and Figure 4 display the results. Frequencies can be similar among different entities yet distinct in the same entity. Same as the analysis in the next section, we suppose that this irrelevance introduces non-entity information into PNNs probabilities, and biases the PNNs distance towards focusing on frequencies.

5 Distance Bias of Prototypical Neural Networks

In section 3, we have shown that PNNs have a priori on the distances between word embeddings and different entities: embeddings are more likely to be close to the entity that has a smaller prototype $\ell 2$-norm, and the word is more likely to be classified to that entity. However, in section 4, we argue this priori will introduce non-entity information that confuses the calculation of probabilities in PNNs. We have shown frequencies and entities are irrelevant. In the following two figures, we further show the prototype $\ell 2$-norms vary in a manner that is also irrelevant to entities.

Figure 5 displays the average prototype l2-norms of all classes. The $\ell 2$-norms vary greatly among different classes (min=7.25, max=17.13, coefficient of variation=0.202). In Figure 6, the blue column represents the largest class-prototype $\ell 2$-norm, the orange one the smallest and the green one the average. Even within the same class, the prototypes $\ell 2$-norms demonstrate large variance due to the contrasting difference among episodes.

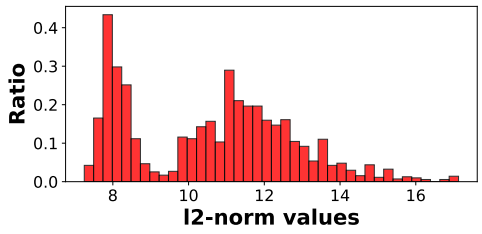

Figure 5: A histogram displays the average prototypes $\ell 2$-norm of all classes.

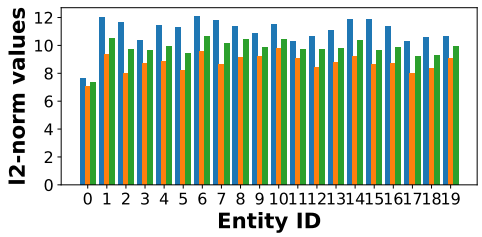

Figure 6: Max(Blue)/Avg.(Green)/Min(Orange) prototypes $\ell 2$-norms within a same class.

Distances between prototypes and word embeddings should represent entity probabilities. Unfor-

[3]Data are taken from the Corpus of Contemporary American English (COCA) that provides 60000 English words with frequencies (COCA_60000).

[4]The words frequencies are counted on the first 2.5 million sentences in BookCorpus (Zhu et al., 2015) processed by HuggingFace (Wolf et al., 2020).

tunately, with respect to the above problem, the distances in the original PNNs are biased towards frequencies instead of being entity-oriented. As a result, PNNs tend to overfit the training data and be trained with unreliable loss minimization.

5.1 The Overfitting Problem

In this section, we aim to account for the overfitting problem caused by the biased distance. Let $\mathbb{S}_u$ be the embeddings of few-labeled data set and $\mathbb{Q}_u$ be the embeddings of the query data set.

Theorem 2. *PNNs learn on a Markov Chain: $\mathbb{S}_u \rightarrow \mathbb{Q}_u$, and maximizes the information bound on the mutual information between $\mathbb{S}_u$ and $\mathbb{Q}_u$.*

Corollary 2.1. *Let $\mathbb{S}_u^g$ be unknown embeddings that the Markov chain: $\mathbb{S}_u^g \rightarrow \mathbb{Q}_u$ holds according to entity information. The integrated Markove chain becomes: $\mathbb{S}_u^g \rightarrow \mathbb{S}_u \rightarrow \mathbb{Q}_u$, and PNNs will overfit the words frequencies information in $\mathbb{S}_u$.*

Proofs are provided in the Appendix A-B. PNNs learn to maximize the information bound of the mutual information between the support and query data, where the information bound is modeled by the frequency-related distances. However, it is because frequencies are irrelevant to entities. Thus, frequency-related distances will confuse PNNs with incorrect evidences, *i.e.* word frequencies, when connecting labeled and query data, preventing PNNs from learning meaningful entity information. As the frequencies can change randomly on new classes, the distances can no longer correctly model the entity probabilities on a new testing data.

5.2 Unreliable Empirical Loss Minimization

In this section, we provide a further explanation to the problem of unreliable empirical loss minimization of training PNNs with biased distances. Given a hypothesis space $\mathcal{H}$ and its element h[5], we aim at minimizing the expected loss to find the optimal solution for a given task:

$$R(h) = \int \ell(h(\mathbf{x}_i, y_i)) dp(\mathbf{x}, y) \tag{3}$$

Noted that $p(\mathbf{x}, y)$ is unknown and we use the empirical loss in practical as a proxy for $R(h)$:

$$R_I(h) = \frac{1}{I} \sum_{i=1}^{I} \ell(h(\mathbf{x}_i, y_i)) \tag{4}$$

[5] $\mathcal{H}$ can be the all potential parameters of a given network structure and h can be an arbitrary parameter.

Let $h^* = argmin_{h \in \mathcal{H}} R(h)$ be the hypothesis that minimizes the expected loss and $h_I = argmin_{h \in \mathcal{H}} R_I(h)$ be the hypothesis that minimizes the empirical loss. The approximation error $[R(h_I) - R(h^*)]$ quantifies the degree of closeness to the optimal result h_I. Noting that the frequency information guides the loss minimization during training PNNs as analyzed in section 5.1. Due to the uncertainty of word frequencies, a good approximation on the training data can have a large approximation error on the testing, which can jeopardize PNNs testing performance.

Moreover, the labeled examples for each episode are limited to N-shot, where data in each episode is not likely to cover many words. As such, the frequencies of the words and prototype $\ell 2$-norms can vary among episodes, resulting in unstable training with low efficiency in model learning and lowering the testing performance.

6 Normalizing the Prototypes

In this section, we aim to provide a solution to the above-mentioned problems through a normalizing method. Varying $\ell 2$-norms mainly causes frequency-biased distances and the above two problems. As a result, we consider normalizing the prototypes to $\ell 2$-norm-invariant vectors. Earlier works in Computer Vision find normalizing both prototypes and the query data embeddings can achieve better and more stable results (Gidaris and Komodakis, 2018). However, we do not normalize the query data embeddings, because word embeddings represent more detail and other useful information that may be eliminated by the normalization.

Representing high-level entity information, prototypes should not be priorly distinguished from each other. Furthermore, observing the following evidence, we argue that prototype $\ell 2$-norms have limited contribution to the correct classification.

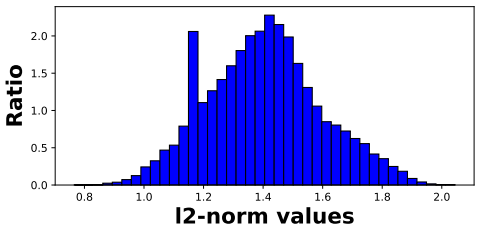

Figure 7: A histogram displays the $\ell 2$-norms of the pre-trained classifier in BERT.

In both the BERT's pre-training (1) and the original PNNs (2), we find the $\ell 2$-norms of class fea-

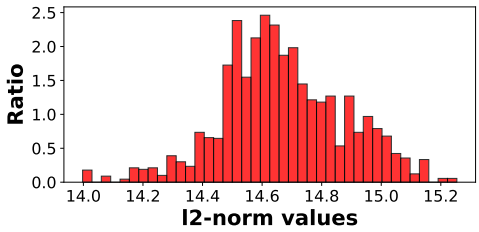

Figure 8: A histogram displays the average prototypes $\ell 2$-norms of all classes after training.

tures play limited roles to the correct classification.

(1) In the BERT's pre-training that predicts a word by its context, the $\ell 2$-norms of the words features, *i.e.* rows of the classifier, show subtle variance. Figure 7 presents the $\ell 2$-norms of the classifier rows: min=0.766, max=2.045, coefficient of variation=0.138.

(2) Without any intervention to the original PNNs, after the training, the prototype $\ell 2$-norms vary much less compared to the original, *i.e.* after the training: (min=14.00, max=15.25, coefficient of variation=0.014) compared to the original :(min=7.25, max=17.13, coefficient of variation=0.202), and Figure 8 compared to Figure 5.

Based on the above analysis, we propose to normalize the prototypes to unit vectors before calculating the class probabilities.

Algorithm 1 Normalizing the Prototypes

*** Pseudo-code in PyTorch ***
import torch.nn.functional as F
$\boldsymbol{C}$ = Calculate Prototype ($\mathbb{S}$) $\in \mathbb{R}^{k \times h}$
*** The Normalization ***
$\boldsymbol{C}$ = F.normalize ($\boldsymbol{C}$, dim=-1)
... the same as the original PNNs ...

Connection to the Adaptive Loss: Different data may associate with different difficulties to be classified. Adaptive loss is proposed to be able to change dynamically in magnitude so as to capture the difficult ones (Han et al., 2021; Oreshkin et al., 2018; Li et al., 2020a). Humans are prone to processing high-frequency words as reported in psychological studies (Brysbaert et al., 2018). Applying this psychological finding to the named entity recognition in natural language processing, we postulate that if a word appears more frequently, its entity should be easier to be classified. To this end, PNNs well adapt to task difficulty through the frequency-related embedding $\ell 2$-norms of the query data.

7 Experiments & Results

To demonstrate the effectiveness of our normalized PNNs, we conduct experiments on nine few-shot named entity recognition datasets proposed by (Huang et al., 2021) and (Ding et al., 2021).

Datasets: Being a classical and basic natural language understanding task, dozens of supervised NER datasets have been proposed, including WikiGold (Balasuriya et al., 2009), CoNLL 2003 (Sang and De Meulder, 2003), WNUT 2017 (Derczynski et al., 2017), MIT Movie (Liu et al., 2013b), MIT Restaurant (Liu et al., 2013a), SNIPS (Coucke et al., 2018), ATIS (Hakkani-Tür et al., 2016), Multiwoz (Budzianowski et al., 2018). Based on these datasets, researchers (Huang et al., 2021) re-structure them to the "K-way N-shot" few-shot setting into a comprehensive few-shot NER benchmark. However, except for the formatting change of data, the simple and direct re-structuring shall lose track of some critical NER properties, such as the task-difficulty differences between the fine-grained and coarse-grained entities (Ding et al., 2021). Therefore, a new expert and challenging dataset has been proposed as a benchmark in few-shot NER (Ding et al., 2021).

Experimental Settings: Without special notations, we basically follow the original implementations in the two open sources[6][7], including models, training/testing pipelines, hyper-parameters, and sampled episodes. We report results using the standard evaluation metrics: micro averaged F1 score. We re-run all the experiments of the origin PNNs to examine the performance improvements by our normalization method based on the same hardware device. We add early stop constraints when reproducing results of (Huang et al., 2021)) and relocate the comparable results from the peer models (Das et al., 2022; Ding et al., 2021). [8] All the experiments are conducted on a single 3090Ti GPU.

Comparison to the State-Of-The-Art Methods: We compare the normalized PNN ($\text{Proto}_{\text{ours}}$) to four advanced methods on Few-NERD. "Struct" and "NNShot" are proposed by (Yang and Katiyar, 2020). "NNShot" classifies the query data

[6] https://github.com/thunlp/Few-NERD
[7] https://github.com/few-shot-NER-benchmark
[8] The replicated performances are inferior to the reported results in the related works, so we use the reported results for a standard reference.

Table 1: The performance State-of-the-art models and our method on FEW-NERD.

Model	FEW-NERD(INTRA) F1 scores				Avg.
	5 way 1~2 shot	5 way 5~10 shot	10 way 1~2 shot	10 way 5~10 shot	
Struct (EMNLP 2020)	30.21	38.00	21.03	26.42	28.92
NNShot (EMNLP 2020)	25.75	36.18	18.27	27.38	26.90
CONTaiNER (ACL 2020)	**40.43**	53.70	**33.84**	47.49	**43.87**
+Viterbi (ACL 2020)	**40.43**	53.71	33.82	47.51	43.86
Proto (Neurips 2017)	20.76	42.54	15.05	35.40	28.43
$\text{Proto}_{\text{ours}}$*	36.83	**54.62**	30.06	**47.61**	42.28
Model	**FEW-NERD(INTER) F1 scores**				**Avg.**
	5 way 1~2 shot	**5 way 5~10 shot**	**10 way 1~2 shot**	**10 way 5~10 shot**	
Struct (EMNLP 2020)	51.88	57.32	43.34	49.57	50.53
NNShot (EMNLP 2020)	47.24	55.64	38.87	49.57	47.83
CONTaiNER (ACL 2020)	55.95	61.83	48.35	57.12	55.81
+Viterbi (ACL 2020)	**56.10**	61.90	**48.36**	57.13	55.87
Proto (Neurips 2017)	38.83	58.79	32.34	52.92	45.72
$\text{Proto}_{\text{ours}}$*	54.35	**66.93**	47.32	**61.50**	**57.52**

* We change the learning rate from 1e-4 to 1e-5. We lower the learning rate because normalized PNN converges too rapidly to be tested on dev set (given the same evaluation steps) before it overfits the training set.

to its nearest data entity in the embedding space, and "Struct" further leverages the Viterbi decoding (Forney, 1973) to produce the final results. "CONTaiNER" as well as the Viterbi enhanced version are proposed by (Das et al., 2022). It utilizes contrastive learning to differentiate word entities. And unlike PNN, "NNShot" and "Struct", "CONTaiNER" will be fine-tuned on the new entities using the limited labeled examples.

We briefly introduce the main characteristic of Few-NERD: it defines entity types from two perspectives called the fine-grained (INTER) and coarse-grained (INTRA). Under the fine-grained definition, different entities can share more abstract similarities. For example, entities "Island" and "Mountain" are both "Location", and entities "Director" and "Athlete" are both "Person". Under the coarse-grained definition, entities have more differences, such as "Location" v.s. "Person" and "Event" v.s. "Organization". If the training classes contain "Island", the model can easily identify the entity "Mountain" at the testing because they share the same "Location" information. Therefore, training on the fine-grained set is less challenging for NER on new testing entities.

Table 1 reports our normalized PNNs on Few-NERD as well as the results of state-of-the-art models, and the original PNNs for comparisons. Compared with the original PNNs, the normalization achieves at least 8.14% performance gain (largest: 16.07% and average: 12.82%). The sophisticated contrastive learning-based CONTaiNER outperforms our method in certain settings. On average, our model is slightly superior (49.84% ($\text{Proto}_{\text{ours}}$) v.s. 49.80%). Besides, CONTaiNER needs to be fine-tuned on the testing data in order to alleviate the differences between training and testing entities, which can account for its superior performance on the coarse-grained (INTRA) set. It should be noted that our normalization method shows competitive performances yet maintains the PNNs' advantages, *i.e.* the low computation cost and easy implementation. In addition, our model achieves the highest average F1 scores (57.52% ($\text{Proto}_{\text{ours}}$)) on the fine-grained (INTER) set, demonstrating its superiority in a more practical setting (Ding et al., 2021).

Incorporation with the Data-Driven Pre-training: (Huang et al., 2021) proposes two pre-training techniques called noisy supervised pre-training (NSP) and self-training (ST). NSP utilizes the large-scale noisy labeled entities on Wikipedia to pre-train the models, while ST utilizes an NER system (teacher model) to label large-scale unlabeled datasets to pre-train the student models. Both the techniques seek extra supervisions to help the model tackle the challenges of the few-shot classification. (Huang et al., 2021) chooses two baselines: the linear classification (LC) and PNNs. And on ten re-structured few-shot NER datasets, they compare the performances of the two baselines as well as the two baselines plus the two pre-training techniques. They report the best performance is achieved by the combination of "LC+NSP+ST".

Because the processed datasets "I2B2" and

Table 2: The performance on benchmark datasets proposed by (Huang et al., 2021).

Model	Datasets (5-shot) F1 scors								Avg.
	CoNLL	WikiGold	WNUT17	MIT Movie	MIT Restaurant	SNIPS	ATIS	Multiwoz	
Proto*	58.22	47.58	20.51	29.94	43.65	56.96	73.82	23.74	44.30
$\text{Proto}_{\text{ours}}$*	58.70	55.69	28.46	50.01	51.34	76.69	87.41	27.78	54.51
Proto+NSP*	62.92	63.33	33.87	35.25	44.15	51.66	74.58	40.52	50.79
$\text{Proto}_{\text{ours}}$+NSP*	**66.50**	67.63	**37.75**	51.32	**54.98**	**83.17**	**90.47**	**47.26**	**62.39**
LC+NSP+ST**	65.4	**68.4**	37.6	**55.9**	51.3	83.0	**90.5**	45.1	62.12

* For meaningful comparison and to calculate the performance gains, we re-run the baseline models "Proto" and "$\text{Proto}_{\text{ours}}$" with the same setting. To reduce the time cost, we add the early stop constraints, i.e. stop the training if a continual 5 epochs training does not improve the dev-set F1 scores.

** The replicated results in [*] are lower than the reported results in the original paper. Therefore, we directly copy the results in the original paper as a comparison for demonstrating our method's effectiveness.

"Onto" are not open-sourced by (Huang et al., 2021), we conduct the experiments on the other eight datasets. For more details of the datasets, please refer to (Huang et al., 2021).

Table 2 reports the results on the eight datasets. Results vary among different datasets, but the normalized PNNs consistently outperform the original PNNs (min:0.48%, max:20.07%, average:10.20%). In certain datasets, normalized PNNs achieves extremely close even higher results than the original PNNs plus a pre-training method that is expensive in time cost (Proto+NSP). Furthermore, higher performance gains are obtained when incorporating the normalized PNNs with the NSP technique (Proto+NSP +6.48% v.s. $\text{Proto}_{\text{ours}}$+NSP +7.88%). Our results show that the classical PNNs combined with the simple normalization and NSP can achieve the best results on the eight few-shot NER datasets (the open sources do not provide the ST checkpoints for PNN). This finding is innovative compared to the results in (Huang et al., 2021).

Effective Learning: Figure 9 (in Appendix C) visualizes the training and dev F1 scores on two settings of Few-NERD, including the original and the normalized PNNs (the * mark denotes that we set the learning rate to $1e^{-}5$ as the same as our experimental settings). Comparing the red with blue lines (with the same learning rate), normalized PNNs can fit the training data in a faster mode yet can achieve higher Dev F1 scores. Comparing the red with green lines, setting the learning rate to $1e^{-4}$ and without normalizing, PNNs learn unstably and more significantly overfit the training data (in INTER 5 way 5~5 shot, dev F1 scores decreases before increasing, and in INTRA 10 way 1~2 shot, the increasing of training F1 scores results in decreasing of dev F1 scores).

Ablation Studies: Based on our analysis in section 6, we only normalize the prototypes and leave the query data embeddings unchanged. We conduct ablation studies about the normalization strategies on Few-NERD as shown in Table 3 in Appendix C. Proto_{AB1} means we normalize only the query data embeddings and leave the prototypes unchanged, and Proto_{AB2} means we normalize both the prototypes and the query data embeddings. We provide four sub-cases for ablation studies. All cases report substantial performance decrease.

8 Conclusion

We examine the synergistic effects of the large-scale PTMs and the classical PNNs in the few-shot NER. Our theoretical analysis of PNNs shows that PNNs' distances that represent the query data's entity probabilities are partly priorily determined in terms of the prototype $\ell 2$-norms. However, on the embeddings of the PTMs, we empirically verify that embedding $\ell 2$-norms contain little entity information, being a type of PTMs' representation degeneration. Furthermore, we show that such representation degeneration makes PNNs' distance biased towards frequencies instead of entity-denoting. This distance bias prevents PNNs from learning useful entity information and causes PNNs to overfit the training corpus and become unreliable on new entities. Therefore, We propose a one-line-code normalization remedy to reconcile PTMs and PNNs for few-shot NER. The experimental results based on nine datasets suggest that the normalized PNNs proposed in this work achieve significant performance improvements over the original PNNs and get competitive results compared with the latest sophisticated methods while maintaining PNNs' all advantages, such as easy implementation and low computation cost. Considering the promising results and the innovation in normalizing the existing models, our results and analysis may be an interest of reference study for researchers and practitioners

working with few-shot NER or other relevant tasks that involve the use of PTMs or PNNs.

9 Limitations

There are certain limitations in this paper. While our theoretical analysis about PNNs and the concept of PTMs' representation degeneration are not limited to the few-shot named-entity recognition, our focused problem, *e.g.* PNNs' distance is biased towards frequencies, is based on the fact that the greatly varied word frequencies represent limited entity information. It is possible that in other tasks, the corpus frequencies can represent semantic features, or the frequencies change much less. Our normalization remedy, therefore, cannot be directly applied to those tasks. Also, representation degeneration is a crucial intrinsic problem of large-scale PTMs. Our focused aspects, *e.g.* frequencies and entity information, is one type of practical issue. We argue that such intrinsic problems can result in different practical issues affecting other NLP tasks beyond this current work's scope.

Acknowledgement

This work is supported by National Key R&D Program of China (2020AAA0105200).

References

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 2009 workshop on the people's web meets NLP: Collaboratively constructed semantic resources (People's Web)*, pages 10–18.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. 2005. Clustering with bregman divergences. *Journal of machine learning research*, 6(10).

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *International Conference on Learning Representations*.

Marc Brysbaert, Paweł Mandera, and Emmanuel Keuleers. 2018. The word frequency effect in word processing: An updated review. *Current Directions in Psychological Science*, 27(1):45–50.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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.

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.

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.

G.D. Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2018. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Spyros Gidaris and Nikos Komodakis. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719.

Jiale Han, Bo Cheng, and Wei Lu. 2021. Exploring task difficulty for few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods*

in Natural Language Processing, pages 2605–2616, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. 2001. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4517–4533.

Yuqing Hu, Stéphane Pateux, and Vincent Gripon. 2022. Squeezing backbone feature distributions to the max for efficient few-shot learning. *Algorithms*, 15(5):147.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423.

Gregory Koch et al. 2015. Siamese neural networks for one-shot image recognition.

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.

Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. 2020a. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12576–12584.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020b. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013a. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and Jim Glass. 2013b. Query understanding enhanced by hierarchical parsing structures. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–77. IEEE.

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*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Erik 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.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

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.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Meihan Tong, Shuai Wang, Bin Xu, Yixin Cao, Minghui Liu, Lei Hou, and Juanzi Li. 2021. Learning from miscellaneous other-class words for few-shot named entity recognition. 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 6236–6247, Online. Association for Computational Linguistics.

Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807.

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.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. 2018. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 402–410.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.

Garrett Wilson and Diane J Cook. 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Vikas Yadav and Steven Bethard. 2019. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*.

Shuo Yang, Lu Liu, and Min Xu. 2020. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*.

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.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. 2018. Breaking the softmax bottleneck: A high-rank rnn language model. In *International Conference on Learning Representations*.

Yue Zhao, Deli Zhao, Shaohua Wan, and Bo Zhang. 2018. Softmax supervision with isotropic normalization.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Bregman Divergence

Definition 1 (Bregman (1967); Censor and Zenios (1998)). *Let $\phi : \mathcal{S} \mapsto \mathbb{R}$, $\mathcal{S} = dom(\phi)$ be a strictly convex function defined on a convex set $\mathcal{S} \subseteq \mathbb{R}^d$ such that ϕ is differentiable on $ri(S)$, assumed to be nonempty. The Bregman divergence $d_\phi : \mathcal{S} \times ri(\mathcal{S}) \mapsto [0, \infty)$ is defined as*

$$d_\phi = \phi(x) - \phi(y) - \langle x - y, \nabla\phi(y)\rangle \quad (5)$$

where $\nabla\phi(y)$ represents the gradient vector of ϕ evaluated at y.

Proposition 1 (Banerjee (2005)). *Let X be a random variable that take values in $\mathcal{X} = \{x_i\}_{i=1}^n \subset \mathcal{S} \subseteq \mathcal{R}^d$ following a positive probability measure ν such that $E_\nu[X] \in ri(\mathcal{S})$. Given a Bregman divergence $d_\phi : \mathcal{S} \times ri(\mathcal{S}) \mapsto [0, \infty)$, the problem*

$$\min_{s \in ri(\mathcal{S})} E_\nu[d_\phi(X, s)] \quad (6)$$

has a unique minimizer given by $s^\dagger = \mu = E_\nu[X]$.

Theorem 3 (Banerjee (2005)). *Let $p_{(\psi,\theta)}$ be the probability density function of a regular exponential family distribution. Let ϕ be the conjugate function of ψ so that $(int(dom(\phi)), \phi)$ is the Legendre dual of (Θ, Ψ). Let $\theta \in \Theta$ be the natural parameter and $\mu \in int(dom(\phi))$be the corresponding expectation parameter. Let d_ϕ be the Bregman divergence derived from ϕ. Then $p_{(\psi,\theta)}$ can be uniquely expressed as*

$$p_{(\psi,\theta)}(x) = exp(-d_\phi(x, \mu))b_\phi(x), \quad \forall x \in dom(\phi) \quad (7)$$

where $b_\phi : dom(\phi) \mapsto \mathbb{R}_+$ is a uniquely determined function.

B Prototypical Neural Networks

Algorithm 2 K-way N-shot Prototypical Neural Network

Input: An episode E_i containing: support data $\mathbb{S}_u$ and query data $\mathbb{Q}_u$.
Output: The loss J for the episode E_i.
#Calculating prototypes on $\mathbb{S}_u$
C = NewEmptyList(Length=K)
for $k = 1$ **to** K **do**
  $\mathbf{c}_k = \frac{1}{N_k} \sum_{(\mathbf{x}^i, \mathbf{y}^i == k)} f_{enc}(\mathbf{x}^i)$
end for
#Classification on $\mathbb{Q}_u$ and calculating the loss J
J = NewEmptyList(Length=0)
for $k = 1$ **to** K **do**
  for $(\mathbf{x}^i, \mathbf{y}^i == k)$ in $\mathbb{Q}_u$ **do**
    $Q(\hat{\mathbf{y}}^i == k|\mathbf{x}^i) = \frac{exp(-d_\phi(f_{enc}(\mathbf{x}^i), \mathbf{c}_k))}{\sum_{k'=1}^{K} exp(-d_\phi(f_{enc}(\mathbf{x}^i), \mathbf{c}_{k'}))}$
    J.Add(CrossEntropyLoss($\hat{\mathbf{y}}^i$, $\mathbf{y}^i$))
  end for
end for
J = Mean(J)

Remark. *Prototype calculation and query data classification are independent but have the same goal of minimizing the classifying loss.*

Theorem. *Assume data embeddings of the support and query data are independent and identically distributed. Let* $\mathbf{c}_k$ *be the class prototype calculated by an aggregation function* $proto(\cdot) : \prod_{i=1}^{N} \mathbf{H}_i \mapsto \mathbf{h} \in \mathbf{H}$*, the problem*

$$\min_{proto(\cdot)} J$$

,where J *is the classifying loss, achieves minimization given by* $proto(\cdot)$ *being the arithmetic mean.*

Proof. In the above Remark, we argue the prototype calculation should also minimize the classifying loss while the query data is unseen. As the optimal prototypes should minimize the classification loss on query data, and the support and query data are independent and identically distributed, we let the support data be the agency of the query data. Therefore, the optimal prototype should minimize the classification loss on support data.

Let us consider the m^{th} class, the corresponding cross-entropy loss is:

$$\begin{aligned} J_m &= -\sum_i \log \frac{exp(-d_\phi(f_{enc}(\mathbf{x}^i), \mathbf{c}_m))}{\sum_{k'=1}^{K} exp(-d_\phi(f_{enc}(\mathbf{x}^i), \mathbf{c}_{k'}))} \\ &= -\sum_i [-d_\phi(f_{enc}(\mathbf{x}^i), \mathbf{c}_m) \\ &\quad - \log \sum_{k'=1}^{K} exp(-d_\phi(f_{enc}(\mathbf{x}^i, \mathbf{c}_{k'})))] \\ &= \sum_i d_\phi(f_{enc}(\mathbf{x}^i), \mathbf{c}_m) \\ &\quad + \sum_i \log \sum_{k'=1}^{K} exp(-d_\phi(f_{enc}(\mathbf{x}^i, \mathbf{c}_{k'}))) \end{aligned} \tag{8}$$

where $\mathbf{x}^i$ is the support data with the class m, $\mathbf{c}_m$ and $\mathbf{x}_{k'}$ be the m^{th} and k'^{th} class prototype. As we aim to find the optimal $\mathbf{c}_m$, we take the derivative of J_m respect to $\mathbf{c}_m$:

$$\begin{aligned} \frac{\partial J_m}{\partial \mathbf{c}_m} &= \frac{\partial \sum_i d_\phi(\mathbf{h}^i, \mathbf{c}_m)}{\partial \mathbf{c}_m} \\ &\quad + \frac{\partial \sum_i \log \sum_{k'} exp(-d_\phi(\mathbf{h}^i, \mathbf{c}_{k'}))}{\partial \mathbf{c}_m} \\ &= \frac{\partial \sum_i d_\phi(\mathbf{h}^i, \mathbf{c}_m)}{\partial \mathbf{c}_m} \\ &\quad + \sum\nolimits_i \frac{\partial(-d_\phi(\mathbf{h}^i, \mathbf{c}_m))/\partial \mathbf{c}_m}{\sum_{k'} exp(-d_\phi(\mathbf{h}^i, \mathbf{c}_{k'}))} \\ &= \sum\nolimits_i (1 - \frac{1}{\sum_{k'} exp(-d_\phi(\mathbf{h}^i, \mathbf{c}_{k'}))}) \\ &\quad \times \partial(d_\phi(\mathbf{h}^i, \mathbf{c}_m))/\partial \mathbf{c}_m \end{aligned} \tag{9}$$

where $\mathbf{h}^i = f_{enc}(\mathbf{x}^i)$. As d_ϕ is a Bregman Divergence, according to Proposition 1, we have $\frac{\partial E_\nu[d_\phi(\mathcal{H}, s)]}{\partial s} = 0$ if and only if $s = E_\nu[\mathcal{H}]$. If we use α to normalize the weight of Equation 9 to have $\sum_i \alpha(1 - \frac{exp(-d_\phi(\mathbf{h}^i, \mathbf{c}_m))}{\sum_{k'} exp(-d_\phi(\mathbf{h}^i, \mathbf{c}_{k'}))}) = 1$, then the optimized $\mathbf{c}_m$ can be calculated as:

$$\mathbf{c}_m = \sum\nolimits_i \alpha(1 - \frac{1}{\sum_{k'} exp(-d_\phi(\mathbf{h}^i, \mathbf{c}_{k'}))})\mathbf{h}^i \tag{10}$$

The Equation 10 show the optimized $\mathbf{c}_m$ should be the arithmetic mean of the support data embeddings minus the category confidences. But the category confidences correspond to the probability normalization of *Softmax*. If we ignore this, the optimal prototype calculation is the arithmetic mean.

□

Corollary. *Based on the support data, PNNs estimate a Gaussian distribution* $\mathbf{N}_k(\mathbf{c}_k, \sigma^2)$ *for embeddings in class* k*, where* σ *is a constant vector.*

And the corresponding choice of the Bregman divergence d should be the squared Euclidean distance.

Proof. According to (Banerjee et al., 2005), for the d-dimension spherical Gaussian distribution, the parameter formula is:

$$p(\mathbf{x};\theta) = \frac{1}{\sqrt{(2\pi\sigma^2)}} exp(-\frac{\|\mathbf{x}-a\|^2}{2\sigma^2}) \tag{11}$$

$$\mu = a \tag{12}$$

$$\phi(\mu) = \frac{1}{2\sigma^2}\mu \tag{13}$$

$$d_\phi(\mathbf{x},\mu) = \frac{1}{2\sigma^2}\|\mathbf{x}-\mu\|^2 \tag{14}$$

The μ in PNNs is the prototypes, i.e. the arithmetic mean of sampled observations, and it exactly estimates the parameter in Gaussian distribution. Therefore, the optimal prototype calculation results in estimating a Gaussian distribution for each class. On a Gaussian distribution where σ is a constant, d_ϕ corresponds to the squared Euclidean distance. □

Theorem. *PNNs learn on a Markov Chain: $\mathbb{S}_u \rightarrow \mathbb{Q}_u$, and maximizes the information bound on the mutual information between $\mathbb{S}_u$ and $\mathbb{Q}_u$.*

Proof. According to the Theorem 3, a Bregman divergence and a Distribution are connected:

$$\log(P_{(\psi,\theta)}(\mathbf{h})) = -d_\phi(\mathbf{h},\mu) + \phi(\mathbf{h}) + log(p_0(\mathbf{h})) \tag{15}$$

when $P_{(\psi,\theta)}$ is the Gaussian distribution, we have $\phi(\mathbf{h}) = \frac{1}{2\sigma^2}\|\mathbf{h}\|_2$ and p_0 is uniquely determined.

PNNs calculate the distance between $\mathbf{h}$ and μ, which can be viewed as the probability of observing $\mathbf{h}$ given μ. This relationship between the support and query data implies the Markov Chain: $\mathbb{S}_u \rightarrow \mathbb{Q}_u$, for observing the query data is dependent on the support data.

In the right of Equation 15, $-d_\phi(\mathbf{h},\mu)$ can be viewed as the probability of observing $\mathbf{h}$ given μ, and the rest $\phi(\mathbf{h}) + log(p_0(\mathbf{h}))$ can be viewed as the probability of observing $\mathbf{h}$ unknown μ: $p(\mathbf{h})$. The first term $p(\mathbf{h} \mid \mu)$ is inversely proportional to $\|\mathbf{h}\|^2$, while the second $p(\mathbf{h})$ is proportional to $\|\mathbf{h}\|^2$. PNNs maximize $-d_\phi(\mathbf{h},\mu)$, resulting in the implicit minimizing of $p(\mathbf{h})$. Integratedly, the learnt probability $P_{(\psi,\theta)}(\mathbf{h})$ is proportional to $\frac{p(\mathbf{h}|\mu)}{p(\mathbf{h})}$. Substitute this back to the loss:

$$\begin{aligned}
J &= -\mathop{\mathbb{E}}_{\mathcal{H}} \log \left[\frac{\frac{p(\mathbf{h}^k|\mu^k)}{p(\mathbf{h}^k)}}{\frac{p(\mathbf{h}^k|\mu^k)}{p(\mathbf{h}^k)} + \sum_{k'\neq k} \frac{p(\mathbf{h}^k|\mu^{k'})}{p(\mathbf{h}^k)}} \right] \\
&= \mathop{\mathbb{E}}_{\mathcal{H}} \log \left[1 + \frac{p(\mathbf{h}^k)}{p(\mathbf{h}^k \mid \mu^k)} \sum_{k'\neq k} \frac{p(\mathbf{h}^k \mid \mu^{k'})}{p(\mathbf{h}^k)} \right] \\
&\approx \mathop{\mathbb{E}}_{\mathcal{H}} \log \\
&\left[1 + \frac{p(\mathbf{h}^k)}{p(\mathbf{h}^k \mid \mu^k)} (K-1) \mathop{\mathbb{E}}_{\mu^{k'}} \frac{p(\mathbf{h}^k \mid \mu^{k'})}{p(\mathbf{h}^k)} \right] \\
&= \mathop{\mathbb{E}}_{\mathcal{H}} \log \left[1 + \frac{p(\mathbf{h}^k)}{p(\mathbf{h}^k \mid \mu^k)} (K-1) \right] \\
&\geq = \mathop{\mathbb{E}}_{\mathcal{H}} \log \left[\frac{p(\mathbf{h}^k)}{p(\mathbf{h}^k \mid \mu^k)} K \right] \\
&= -I(\mathbf{h}^k, \mu^k) + \log(K)
\end{aligned} \tag{16}$$

The results show $I(\mathbf{h}^k, \mu^k) \geq \log(K) - J$, which means that PNNs minimize the classification loss to maximize the information bound on the mutual information between rvh and μ, and integratedly, between the support and query data. We notice the above detail proof follows the same mathematical process in the works on contrastive learning (Van den Oord et al., 2018)

□

Corollary. *Let $\mathbb{S}_u^g$ be unknown embeddings that the Markov chain: $\mathbb{S}_u^g \rightarrow \mathbb{Q}_u$ holds according to entity information. The integrated Markove chain becomes: $\mathbb{S}_u^g \rightarrow \mathbb{S}_u \rightarrow \mathbb{Q}_u$, and PNNs will overfit the words frequencies information in $\mathbb{S}_u$.*

Proof. In the Markov Chain: $\mathbb{S}_u^g \rightarrow \mathbb{S}_u \rightarrow \mathbb{Q}_u$, using the data processing inequality, [9] we have:

$$I(\mathbb{S}_u, \mathbb{Q}_u) \geq I(\mathbb{S}_u^g, \mathbb{Q}_u) \tag{17}$$

The learnt extra information $I(\mathbb{S}_u, \mathbb{Q}_u) - I(\mathbb{S}_u^g, \mathbb{Q}_u) \geq 0$ represents PNN's overfitting to $\mathbb{S}_u$'s words frequencies introduced by the frequency-related distances.

□

[9] http://www.scholarpedia.org/article/Mutual_information

C Effective Learning and Ablation Studies

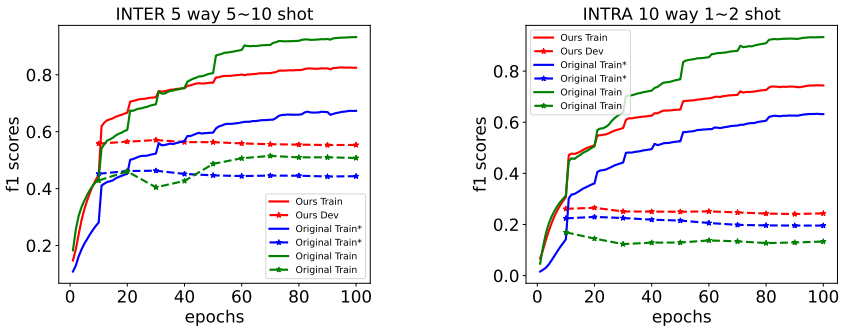

Figure 9: Training and Dev F1 Scores on Few-NERD of two cases.

Table 3: Ablation studies of our method on FEW-NERD.

Model	**FEW-NERD(INTRA) F1 scores**			
	5 way 1~2 shot	**5 way 5~10 shot**	**10 way 1~2 shot**	**10 way 5~10 shot**
$\text{Proto}_{\text{ours}}$	**36.83**	**54.62**	**30.06**	**47.61**
Proto_{AB1}	/	/		1.04
Proto_{AB2}	/	9.89	/	/
Model	**FEW-NERD(INTER) F1 scores**			
	5 way 1~2 shot	**5 way 5~10 shot**	**10 way 1~2 shot**	**10 way 5~10 shot**
$\text{Proto}_{\text{ours}}$	**54.35**	**66.93**	**47.32**	**61.50**
Proto_{AB1}	9.24	/	/	/
Proto_{AB2}	/	/	1.56	/