Label Semantics for Few Shot Named Entity Recognition

Jie Ma[1] **Miguel Ballesteros**[1] **Srikanth Doss**[1] **Rishita Anubhai**[1]
Sunil Mallya[1*] **Yaser Al-Onaizan**[1*] **Dan Roth**[1,2]
[1]AWS AI Labs
[2]Computer and Information Science, University of Pennsylvania
{jieman, ballemig, srikad, ranubhai, drot}@amazon.com
mallya16@gmail.com, onaizan2000@yahoo.com

Abstract

We study the problem of few shot learning for named entity recognition. Specifically, we leverage the semantic information in the names of the labels as a way of giving the model additional signal and enriched priors. We propose a neural architecture that consists of two BERT encoders, one to encode the document and its tokens and another one to encode each of the labels in natural language format. Our model learns to match the representations of named entities computed by the first encoder with label representations computed by the second encoder. The label semantics signal is shown to support improved state-of-the-art results in multiple few shot NER benchmarks and on-par performance in standard benchmarks. Our model is especially effective in low resource settings.

1 Introduction

Named entity recognition (NER) seeks to locate named entity spans in unstructured text and classify them into pre-defined categories such as PERSON, LOCATION and ORGANIZATION (Tjong Kim Sang and De Meulder, 2003a). As a fundamental natural language understanding task, NER often serves as an upstream component for more complex tasks such as question answering (Mollá et al., 2006), relation extraction (Chan and Roth, 2011) and coreference resolution (Clark and Manning, 2015). However, building an accurate NER system has traditionally required large amounts of high quality annotated in-domain data (Lison et al., 2020; Chen et al., 2020). This usually involves well defined annotation guidelines and training of annotators, which requires rich domain knowledge and can be prohibitively expensive (Huang et al., 2020).

Few shot learning (FSL) (Vinyals et al., 2017; Finn et al., 2017; Snell et al., 2017) aims at performing a task using only very few annotated examples (i.e. support set).

Similarity-based methods, such as prototypical networks, are extensively studied and show great success for FSL (Vinyals et al., 2017; Snell et al., 2017; Yu et al., 2018a; Hou et al., 2020). The core idea is to classify input examples from a new domain based on their similarities with representations of each class in the support set. These methods do not utilize the semantics of label names and usually represent labels by directly averaging the embedding of support set examples, oversimplifying the learning of label representations. The main premise of our work is that label names carry meaning that our models can induce from data; the labels are themselves words that appear in text in various contexts and are thus semantically related to other words that appear in text, and this relatedness can be leveraged. For example, the representation of "Lionel Messi" is more similar to that of PERSON than to the representations of LOCATION or DATE when similar priors are used for labels and words or phrases.

In this work, we propose a neural architecture that uses two separate BERT-based encoders (Devlin et al., 2019) to leverage semantics of label names for NER.[1] One encoder (a) is used to encode the document and its words while the other encoder (b) is used to encode label names (e.g. PERSON, LOCATION etc.). The model is trained to match word representations from encoder (a) with label representations from encoder (b), and assign a label for each word by maximizing the

[*]Work done while at AWS AI Labs.

[1]Our model is similar to the two-tower model widely adopted in question answering (Karpukhin et al., 2020), recommender systems (Wang et al., 2021) and entity linking (Logeswaran et al., 2019; Vyas and Ballesteros, 2020).

Findings of the Association for Computational Linguistics: ACL 2022, pages 1956 - 1971

similarity. We also experiment by replacing the BERT label encoder with GloVe embeddings (Pennington et al., 2014) as a simplified architecture.

We report experimental results in multiple NER datasets from different domains. We summarize our contribution as follows:

- We propose a simple and effective model architecture that leverages label semantics for NER.
- We show that the proposed model is particularly effective in low resource settings and gives on-par results with the state-of-the-art models in high resource settings.
- We achieve a new state-of-the-art in multiple few shot NER benchmarks. Specifically, our model outperforms prior work by 1.2 to 6.6 F1 points on CoNLL'03, WNUT'17, JNLPBA, NCBI-disease and I2B2'14 datasets on various few shot shots settings (§3.6).
- We show that the proposed model is robust to variations of label names and that it is able to differentiate semantically similar labels.

2 Model

We present our NER model. As shown in Figure 1, it consists of two BERT-based encoders where one encoder is used to encode the document and its tokens and the other to encode labels. We formalize the differences between datasets used in our experimentation (§2.1), then present how two BERT-based encoders (and the modification with GloVe-based encoder for labels) are used to leverage semantics in labels for NER (§2.2). Finally we discuss the training procedure (§2.3) and how labels are represented (§2.4).

2.1 Source and Target Datasets

For few shot NER, we use a setup similar to meta-learning. We first train our models on **source datasets** $\{\mathcal{D}_1^S, \mathcal{D}_2^S, ...\}$, then evaluate the model on unseen few shot **target datasets** $\{\mathcal{D}_1^T, \mathcal{D}_2^T, ...\}$ with or without finetuning. Each target dataset only contains a few examples and a different taxonomy of labels compared to the source datasets.

2.2 Architecture

We use two BERT-based encoders as shown in Figure 1: a BERT document encoder and a BERT label encoder (we also experiment with GloVe embeddings as label encoder, described in §3.5). Like the traditional NER models (Carreras et al., 2003; Collobert et al., 2011; Lample et al., 2016, inter alia), we predict the label of each token with BIO scheme.[2] For each token we get an embedding e from the first BERT document encoder. For the unique set of labels $\mathcal{L}_\mathcal{D}$ associated with dataset $\mathcal{D}$, we apply three steps to get the representations: First, we manually convert the label names to their natural language forms, e.g. "PER" to "person", "ORG" to "organization" etc. Second, we convert each of the label names to BIO scheme, in the form of natural language, e.g. "person" to "begin person" or "inside person". Finally, we use the second BERT label encoder to embed each of the labels in natural language BIO scheme. We compute the BERT [CLS] token embedding as the representation for the corresponding label. We form a label vector $\boldsymbol{b}$ of all label embeddings b_i for all i in $\{1, 2, ..., 2 \times N_L - 1\}$ [3]. The label encoder acts like a lookup table for label embeddings. Finally, to find the most appropriate label for this token, we use:

$$y = \arg\max_i \ \text{softmax}(e \cdot \boldsymbol{b})$$

2.3 Training

Comparing with prior work on neural architectures for NER, our model does not require a new randomly initialized top layer classifier for a new dataset with new unseen label names. Instead, we generate label representations from the BERT label encoder. We hypothesize that this is beneficial because it prevents the model from forgetting priors since no parameters are dropped or randomly initialized for different datasets.

We propose a simple two stage training procedure. In the first stage, we pre-finetune our model on the mix of all source datasets (which usually have different label set taxonomies), then we fine-

[2] Each token is predicted as B-entity_type, I-entity_type or O, indicating the token is at the beginning, inside or outside of the entity_type.

[3] Each of the N_L labels are converted to BIO scheme except "O"/"other", thus it is $2 \times N_L - 1$ embeddings in total.

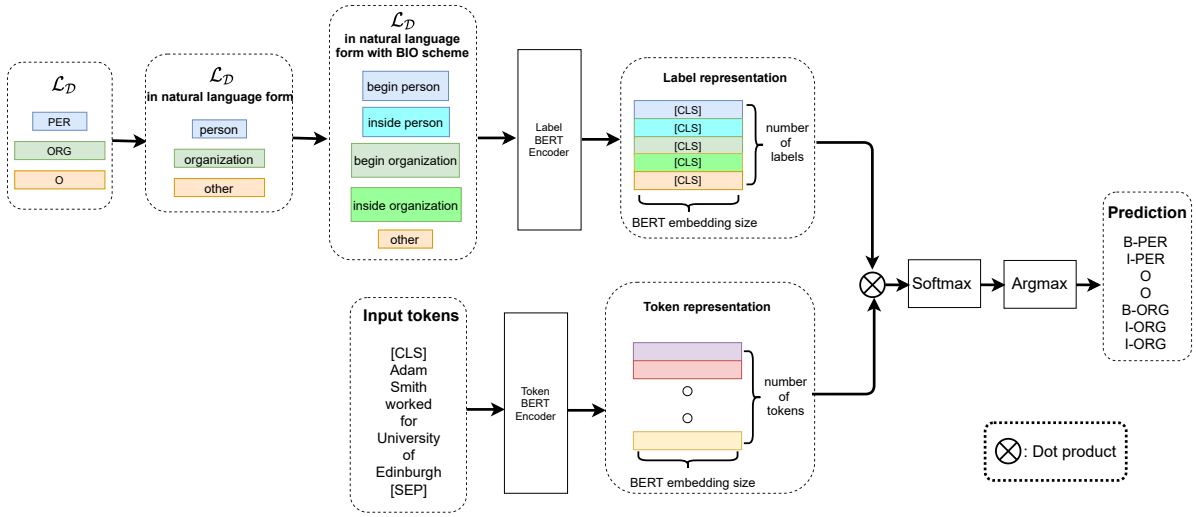

Figure 1: The architecture of our NER model. The diagram shows how representation of labels and tokens are produced, and how we use them to calculate final model prediction. The top part of the figure shows how labels are encoded; the bottom part of the figure shows how sentence are encoded.

tune the trained model on the target dataset. This process is also known as pre-finetuning (Aghajanyan et al., 2021) and finetuning. For scenarios where no source datasets are available, we simply skip the first stage. During model training time, both encoders are updated for every iteration at both stages, which helps to align the token embedding space and the label embedding space.

During inference time, the learned label encoder is only required to produce label representations once. This is because the label representations may be cached and the label encoder is no longer needed to recompute representations. Our model is therefore not introducing additional memory overhead (since label encoder is removed) or latency overhead (since label representation is cached).

2.4 Label Representation

Given that our label encoder is based on BERT and contains the priors from pretraining, our architecture allows any textual form as input for the generation of label representations. In order to make our results comparable with previous studies, we use only the natural language form of label names for our primary results. We discuss more label representations in Appendix E.

3 Experiments

We evaluate our model and we compare it against existing few shot methods in two scenarios: high resource and low resource (few shot). In both cases, we assume there is a source dataset (which may be a set) with abundant data, and our goal is to maximize model performance on unseen target datasets which follow different taxonomies from the source dataset.

3.1 Datasets

We perform experiments on 6 NER datasets from 5 different domains: OntoNotes 5.0 (Weischedel et al., 2013) (Mixed), CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003a) (News), WNUT-2017 (Derczynski et al., 2017) (Social), JNLPBA (Collier and Kim, 2004) (Biology), NCBI-disease (Dogan et al., 2014) (Biology) and I2B2-2014 (Stubbs and Uzuner, 2015) (Medical). In all our experiments and following the definition in 2.1, we treat OntoNotes as the **source dataset** and all other as **target datasets**.[4]

3.2 Settings and Evaluation

In this Section, we present the different experiments, and how do we carry out the evaluation.

High Resource: Given a target dataset, we simply take all available data and evaluate on the standard held-out test set.

[4]We use train/dev/test split released from CoNLL-2012 shared task: https://cemantix.org/conll/2012/data.html.

		1 Shot	5 Shot	20 Shot	50 Shot	Full Dataset
CoNLL-2003	TransferBERT	44.8 ±15.0	66.9 ±6.7	77.5 ±1.2	82.0 ±1.1	91.3 ±0.2
	Prototypical Network	7.5 ±2.6	11.5 ±5.6	18.6 ±7.5	16.3 ±2.7	N/A
	WPN-CRF	56.26 ±9.1	67.7 ±4.4	67.4 ±2.0	69.0 ±1.7	N/A
	Struct NN shot	63.7 ±3.7	70.0 ±3.0	73.1 ±1.9	75.7 ±1.8	N/A
	TANL	54.7 ±9.4	65.6 ±3.8	71.0 ±2.4	74.4 ±1.9	**91.7** ±0.4
	Our model - GloVe	63.1 ±6.9	73.5 ±2.4	78.3 ±1.1	82.0 ±1.5	91.6 ±0.2
	Our model - BERT	**68.4** ±6.7	**76.6** ±2.1	**79.7** ±1.1	**83.1** ±1.2	91.5 ±0.2
WNUT-2017	TransferBERT	27.6 ±6.8	35.2 ±3.4	40.9 ±1.6	42.5 ±1.2	44.0 ±0.2
	Prototypical Network	1.7 ±1.2	2.1 ±1.0	2.7 ±1.6	3.5 ±1.7	N/A
	WPN-CRF	23.1 ±2.8	29.9 ±3.2	32.9 ±1.2	33.2 ±1.1	N/A
	Struct NN shot	31.1 ±6.4	33.2 ±2.0	30.8 ±2.2	31.8 ±1.8	N/A
	TANL	25.6 ±6.3	33.3 ±4.4	34.1 ±2.1	34.4 ±2.4	45.2 ±0.6
	Our model - GloVe	36.6 ±2.4	39.6 ±1.9	42.5 ±1.3	43.0 ±1.1	**45.7** ±0.6
	Our model - BERT	**38.3** ±1.7	**40.8** ±2.1	**42.7** ±1.1	**43.3** ±0.8	45.0 ±0.6
JNLPBA	TransferBERT	26.6 ±7.8	40.3 ±2.8	53.2 ±2.9	59.7 ±1.3	71.0 ±0.5
	Prototypical Network	2.1 ±1.5	4.0 ±3.2	6.8 ±3.6	5.7 ±3.0	N/A
	WPN-CRF	6.5 ±5.0	10.3 ±5.7	10.3 ±4.9	9.4 ±2.7	N/A
	Struct NN shot	15.9 ±5.3	19.2 ±2.9	23.1 ±2.1	26.8 ±0.7	N/A
	TANL	32.4 ±4.0	41.1 ±5.0	51.7 ±2.6	58.8 ±0.6	**74.3** ±0.2
	Our model - GloVe	25.4 ±6.1	39.7 ±2.3	52.3 ±3.1	59.3 ±1.4	71.8 ±0.3
	Our model -BERT	**32.7** ±3.0	**43.15** ±2.4	**53.8** ±2.7	**59.8** ±1.3	71.0 ±0.5
NCBI-disease	TransferBERT	16.8 ±9.5	24.1 ±6.3	43.0 ±5.0	56.7 ±3.0	84.5 ±0.9
	Prototypical Network	12.2 ±8.7	12.5 ±9.6	14.0 ±11.6	10.8 ±7.3	N/A
	WPN-CRF	5.5 ±4.8	6.8 ±9.1	3.5 ±5.4	5.7 ±5.3	N/A
	Struct NN shot	18.5 ±5.6	20.6 ±5.2	27.6 ±2.4	36.7 ±5.0	N/A
	TANL	15.8 ±4.0	21.0 ±6.2	26.0 ±3.9	40.9 ±4.2	85.8 ±0.9
	Our model - GloVe	15.1 ±8.7	26.2 ±6.1	44.6 ±4.2	56.8 ±3.1	**86.7** ±0.6
	Our model - BERT	**30.7** ±9.1	**34.9** ±4.9	**50.9** ±3.3	**60.5** ±2.2	85.0 ±0.6
I2B2-2014	TransferBERT	58.4 ±5.7	75.2 ±1.9	86.2 ±0.9	90.3 ±0.4	93.0 ±0.1
	Prototypical Network	2.1 ±0.7	2.2 ±0.4	2.6 ±0.4	2.7 ±0.1	N/A
	WPN-CRF	10.0 ±2.5	13.1 ±3.3	13.9 ±2.1	13.3 ±2.1	N/A
	Struct NN shot	46.7 ±6.4	59.1 ±1.9	67.4 ±1.3	72.4 ±0.6	N/A
	TANL	47.1 ±5.2	65.1 ±2.9	80.7 ±1.2	87.0 ±0.3	92.0 ±0.1
	Our model - GloVe	58.2 ±5.8	75.5 ±2.3	85.6 ±1.0	90.5 ±0.3	**93.5** ±0.1
	Our model - BERT	**61.9** ±4.3	**76.8** ±2.0	**86.7** ±0.8	**90.5** ±0.4	93.2 ±0.3

Table 1: Results on held out test sets of all datasets. "Our model - GloVe": this refers to our model with GloVe label encoder. "Our model - BERT": this refers to our model with BERT label encoder. All numbers indicate micro F1 scores unless noted otherwise. Results for low resource settings are average of 10 runs with different support set sampling. Results for high resource setting are average of 5 runs with different random seeds. For some baselines we cannot run the released implementation from originally papers due to GPU out of memory and they are marked as N/A. We visualize the results with bar chart in Appendix D.

Low Resource: Given a target dataset, we downsample the data (at sentence level) in the train split to construct a K-shot support set. This simulates the low resource scenario where only a few training examples are available in the target dataset. The definition of a K-shot support set is that it contains exact K examples for each of the labels. However, unlike the text classification task where each sentence is associated with one label, in the NER task multiple named entities may co-occur in the same sentence. We cannot guarantee that the support set contains exact K named entities for each label after downsampling. We therefore define the proxy for K-shot support set similar as the one by Hou et al. (2020), with the following two criteria: 1) Each label in the target dataset (except "O") has at least

K corresponding named entities in the support set; 2) At least one of the labels in the target dataset will have less than K named entities in the support set if any sentence is removed.[5] We apply the same downsampling algorithm as in (Hou et al., 2020) for the support set. More details can be found in Appendix B.

To evaluate the model performance in the K-shot support set, most prior work (Hou et al., 2020; Athiwaratkun et al., 2020; Fritzler et al., 2019) followed the few-shot classification setup, where test sets are also downsampled to K-shot subsets (query set) such that each entity labels are evenly distributed. The model is trained and evaluated on multiple support datasets and query set pairs, and final model performance is reported with average of scores on each query set. However, we argue that in real world cases, entity labels have certain distribution corresponding to the domain, downsampled K-shot query set does not reflect this real distribution. Therefore instead of evaluating on the downsampled query set, we directly evaluate the model in the full test split from the target dataset. This also improves comparability and replicability of our results since the same test set is used across and in prior work (even in papers that are not focused on few-shot experiments).

Evaluation To thoroughly test our model, we evaluate it with 1-shot, 5-shot, 20-shot, 50-shot (low resource) and also the full dataset (high resource) settings. Following prior work (Tjong Kim Sang and De Meulder, 2003b), we use micro F1 score as metric. For low resource settings, we repeat the experiments 10 times with randomly sampled support sets. For high resource setting, we repeat the experiments 5 times with different random seeds. In all cases, we report average micro F1 with standard deviation. Table 2 shows an overview of dataset statistics.

3.3 Baselines

TransferBERT trains the same NER model in (Devlin et al., 2019) by pre-finetuning on a source dataset then finetuning on a target dataset. **Prototypical Network** (Snell et al., 2017) approaches NER as a token level classification task. It assigns label for each token based on similarities between candidate token and tokens in few shot support set. **WPN-CRF** (Fritzler et al., 2019) pretrains a prototypical network with source dataset and evaluate it on target dataset without finetuning. It uses a conditional random field (CRF) (Huang et al., 2015) to output the final labels of the sentence. **Struct NN shot** (Yang and Katiyar, 2020) finds nearest token in support set for a given candidate token and assign it the same label as its nearest neighbor. **TANL** (Paolini et al., 2021) forms NER as sequence to sequence. The model is trained to generate the original input text with entities being decorated in a bracket.[6]

3.4 Hyperparameters

We use English cased BERT-base (Devlin et al., 2019) as contextual embedder for all baseline models and our model, except for TANL where T5-base is used.[7] We use Adam optimizer (Kingma and Ba, 2014) to train our model with a learning rate of 1×10^{-5} and batch size of 10. We pre-finetune our model on the source dataset (Ontonotes) for 3 epochs and continue finetuning on target datasets for 200 epochs for both high resource and low resource settings. We pick the last epoch as the final model. For label names, we manually expand all shortcut names into full natural language names (e.g. "PER" to "person", "LOC" to "location") and lower case all names. Textual forms for all datasets can be found in Appendix A.2. We run all experiments on NVIDIA V100 GPU.[8]

3.5 GloVe as Label Encoder

We experiment with GloVe embeddings (Pennington et al., 2014) as the label encoder.[9] In this case,

[5] We count at named entity level instead of token level. For example, "Lionel Messi" is counted as one occurrence for PERSON entity. However, Hou et al. (2020) counted it as one occurrence for "B-PERSON" (for token "Lionel") and one occurrence for "I-PERSON" (for token "Messi").

[6] We are not able to include (Hou et al., 2020) as a baseline as we are not able to reproduce the model with their published repository, even on a machine with 40GB of GPU memory. We also cannot compare with the published results due to the differences in the following settings: (1) we are evaluating our model on full test splits while Hou et al. carry out an episodic evaluation) and (2) We use more datasets (from different domains).

[7] We use the checkpoint released for BERT-base: https://github.com/google-research/bert, and checkpoints released in Hugging Face for T5-base: https://huggingface.co/t5-base

[8] More details about hardware in Appendix C.

[9] We use 300 dimensional GloVe that is pretrained on Wikipedia and Gigaword 5 corpus released here: https:

our model has no extra parameters compared to other baselines. As in the case with BERT, the vectors are updated throughout the training. Given that there is no [CLS] token available, we apply max pooling on all the GloVe embeddings corresponding to each label token. If the label consists only of one token, max pooling will return the actual GloVe embedding for the token as the label representation.

Dataset	Support Set Shot			
	1	5	20	50
CoNLL'03	3.6	12.3	38.5	102.5
WNUT'17	13.4	44.6	143.6	366.3
JNLPBA	6.8	27.5	99.2	241.2
NCBI	1.8	3.7	14.5	37.2
I2B2'14	155.4	613.4	2339.4	5888.1

Table 2: Number of sentences in support set with different shots for all target datasets. Numbers are averaged across 10 different random samplings. NCBI refers to NCBI-disease dataset. More details are reported in Appendix A.1.

3.6 Results

We summarize experiment results in Table 1. As shown, our model outperforms all previous methods in low resource settings. In extreme low resource scenarios (1 and 5 shot), our model performs significantly better than previous methods by a margin of 6.6 F1 and 4.8 F1 on average in 1 shot and 5 shot, respectively. This indicates that our model can leverage semantics in label names effectively to improve accuracy when data is extremely scarce. However, we also notice that when the target data size increases, the improvement of our model becomes smaller. This suggests that with more training examples, the model relies less on semantics of labels.

In a high resource setting, we find that our model achieves the same level of performance as other baselines, except for JNLPBA dataset where our model is 3.3 F1 behind TANL.[10] This model is based on T5-base which is pretrained on a much larger unannotated dataset, and with different objectives, than our BERT-base encoders.

We also note that when label names in the target dataset are similar to the source ones, few shot models have a much smaller gap with their high resource counterparts, compared to when source and target label names are totally different. Specifically, CoNLL-2003, WNUT-2017 and I2B2 have more similar label names with Ontonotes (the source data), and our model can achieve 84%, 91% and 83% of the score of the high resource model performance with only 5 shot. While for JNLPBA and NCBI-disease, where the label names are totally different from source data, our model can only achieve 61% and 41% of the score of the high resource model performance with 5 shot.

4 Analysis

Here, we show how semantics in label names help in low resource scenarios and how our model benefits from pre-finetuning stage.

Entity Types	Original Labels		Renamed Labels
	0 shot	1 shot	0 shot
PER	92.3	90.3	85.4
LOC	70.9	61.2	54.8
ORG	50.3	59.7	58.4
MISC	0.5	47.5	6.8

Table 3: F1 for 0 and 1 shot performance on CoNLL-2003 development set.

4.1 Impact of the Label Encoder

We hypothesize that encoding label names with a label encoder (either BERT or GloVe) leverages prior knowledge from the pretraining phase and uses it as inductive bias. In addition, by performing pre-finetuning on the source dataset, we are not only aligning the embedding space between labels and tokens in the vocabulary, but also updating the label encoder to produce useful label representations in the source dataset.

To further strengthen our hypothesis (besides what is presented in Table 1), we show results in zero shot settings. Specifically, we pre-finetune a model on the source dataset (Ontonotes) and directly test it on CoNLL-2003 without updating its parameters. We also rename the labels to avoid

//nlp.stanford.edu/projects/glove/

[10] We cannot run released implementation of three baselines (marked as N/A in Table 1) due to GPU out of memory even with 40GB of GPU memory.

overlapping of label names between source and target datasets while still retaining the semantics.[11] Particularly, during evaluation we rename "PER" to "individual", "LOC" to "geographical area" and "ORG" to "corporation". "MISC" stays the same since it does not overlap with any of the Ontonotes labels. The results are shown in Table 3.

With original label names, the zero shot performance of our model is comparable to 1 shot performance for all entity types with the exception of "MISC". Even with the renamed labels that do not have any overlap with the source dataset, the zero shot performance still remains comparable with 1 shot. This seems to validate our hypothesis that the model is able to leverage prior knowledge.

4.2 Semantics of Label Names

To demonstrate the impact of semantics of label names, we carry out experiments with our model on target datasets with the following variations of label names: (1) original label names (which is simply our experimental setup as in the experiments above, where we use the natural language form of the label names), (2) meaningless label names and (3) misleading label names.

We compare our model with the TransferBERT baseline, since it is the counterpart of our model without label semantics. We pre-finetune our model on Ontonotes as previous experiments. Results on CoNLL2003 and JNLPBA are shown in Figure 2.[12]

Meaningless labels We simply use "label 1", "label 2" etc., as input representation for label names, which simulates the case where there is no more semantics information in the form than the fact that they are different labels and they have some sort of ordering. This evaluates the few shot model performance when meaningless (or shallow in semantics, just a differentiation of label indices) inputs are given. Comparing to the original label names, the results drop in 1 and 5 shot settings, then gradually converged to the original label performance as the training data size increases. This shows that label semantics is critical for extreme low resource scenarios (1 and 5 shot).

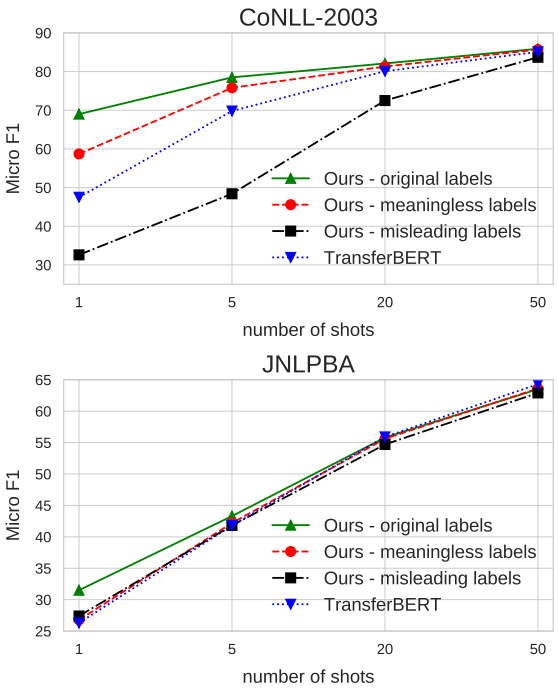

Figure 2: Model performance on meaningless and misleading laberls. Micro F1 is reported on the development data.

Misleading labels We randomly swap the natural language form between labels. For example, in CoNLL2003 dataset, we assign "location" for "PER", "person" for "ORG", "organization" for "MISC" and "miscellaneous" for "PER".[13] The performance drops are larger for CoNLL2003 than the ones in JNLPBA. We hypothesize that since CoNLL2003 label set is closer to Ontonotes, there is stronger prior knowledge incorporated in the label encoder from the pre-finetuning phase. Also, we find that more supervised examples are required to correct such wrong strong prior information. JNLPBA needs 5 shot data to achieve the same performance with original labels and misleading labels, but CoNLL2003 needs 50 shot data to match the performance. This indicates that our model is misled by the labels when the number of training examples is small, which indicates that the label semantics signal is critical in few shot settings.

[11] Ontonotes has both "LOC" and "GPE" labels, however, the definition of label "GPE" in Ontonotes is much closer to "LOC" in CoNLL2003. Therefore, we use "GPE" instead of "LOC" for zero shot experiments.

[12] We present experiments with contextualized label representations in appendix E.

[13] For each run we randomly assign different misleading label names, and we report results averaging 10 different runs.

4.3 Impact of Pre-finetuning

Our model does not require a new randomly initialized top layer classifier for a new dataset, we hypothesize that it can prevent the model from forgetting learned prior knowledge from the pre-finetuning stage thus benefits the low resource scenarios, where prior knowledge is critical. To validate it, we compare 1-shot results on target datasets with and without pre-finetuning stage, as shown in Table 4. First, when pre-finetuning stage is eliminated, performance of both our model and TransferBERT drop significantly, indicating that prior knowledge from pre-finetuning stage is critical in low resource settings. Second, our model outperforms TransferBERT significantly when pre-finetuning stage is included, however, the performance is similar between our model and TransferBERT when it is excluded. This suggests that our model is highly effective in leveraging knowledge learned from the pre-finetuning stage.

Datasets	Pre-finetune on Ontonotes		No pre-finetune	
	Transfer-BERT	Ours	Transfer-BERT	Ours
CoNLL'03	47.5	69.0	9.0	10.7
WNUT'17	35.6	48.2	4.0	5.7
JNLPBA	26.3	31.5	14.8	19.5
NCBI	15.1	31.3	12.5	13.9
I2B2'14	56.9	60.1	47.5	46.8

Table 4: 1-shot performance on development set of corresponding datasets. Micro F1 is reported. NCBI refers to NCBI-disease dataset.

5 Related Work

Few Shot Learning: Meta learning is widely studied for the problem of few shot learning, aiming to quickly adapt a model to new tasks based on tasks learned in an earlier stage. Recent research (Snell et al., 2017; Vinyals et al., 2017; Sung et al., 2017) mostly focused on metric-based methods. Snell et al. (2017) learns a prototype representation for each class and classify test data based on their similarities with prototypes. These methods have been successfully adapted to NLP tasks such as classification (Yu et al., 2018b; Bao et al., 2019), relation classification (Han et al., 2018) and NER (Fritzler et al., 2019; Yang and Katiyar, 2020). However, all these methods do not directly leverage the semantics of label names.

Label Semantics: Earlier work has shown the ability to perform zero- and few-shot learning by exploiting the semantic of labels in text classification tasks (Chang et al., 2008; Luo et al., 2021). Zhou et al. (2018) study zero-shot fine-type NER with label semantics by automatically reading from Wikipedia via a linking approach, but assumes that the mentions of the entities are given. Paolini et al. (2021) and Athiwaratkun et al. (2020) approach NER as a generation task and predict named entities in augmented (or decorated) languages. Cui et al. (2021) reformulate NER as a cloze task and use sequence to sequence models to fill named entities in pre-defined templates. Both of these two methods suffer from long inference time due to an autoregressive decoder. Hou et al. (2020) leverage label semantics in Task-Adaptive Projection Network (TapNet), where the core idea is to learn a projection function that separates words that have different labels in the projected space. In contrast, our model learns to align token representations with label representations. Hou et al. (2020) only uses label representations as a reference to guide the learning of the projection function, and in their case label representations are computed once. Our label representations are updated with every update while training.

6 Conclusion

We propose a neural architecture that leverages semantics of label names for Named Entity Recognition. Our model significantly outperforms the state-of-the-art few shot NER baselines on low resource settings, and performs on-par in the high resource setting. We perform extensive experiments to show that the label encoder incorporates strong prior knowledge from BERT and a dataset (source dataset) used in a pre-finetuning stage. We demonstrate that the semantics of label names in target datasets are critical to retrieve the prior knowledge. We also show that our model is robust to variation of label names and that it is able to differentiate between semantically closed labels.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *CoRR*, abs/2101.11038.

Ben Athiwaratkun, Cicero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2019. Few-shot text classification with distributional signatures. *CoRR*, abs/1908.06039.

Xavier Carreras, Lluís Màrquez, and Lluís Padró. 2003. Learning a perceptron-based named entity chunker via online recognition feedback. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 156–159.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*.

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. 2020. Local additivity based data augmentation for semi-supervised NER. *CoRR*, abs/2010.01677.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China. Association for Computational Linguistics.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. *CoRR*, abs/2106.01760.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. *CoRR*, abs/2105.07464.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *CoRR*, abs/1810.10147.

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.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. *CoRR*, abs/2012.14978.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR*, abs/2004.04906.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1518–1533, Online. Association for Computational Linguistics.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *FINDINGS*.

Diego Mollá, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification. *J. of Biomedical Informatics*, 58(S):S20–S29.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2017. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003a. 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.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003b. 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 - Volume 4*, CONLL '03, page 142–147, USA. Association for Computational Linguistics.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2017. Matching networks for one shot learning.

Yogarshi Vyas and Miguel Ballesteros. 2020. Linking entities to unseen knowledge bases with arbitrary schemas. *CoRR*, abs/2010.11333.

Tian Wang, Yuri M. Brovman, and Sriganesh Madhvanath. 2021. Personalized embedding-based e-commerce recommendations at ebay. *CoRR*, abs/2102.06156.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018a. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018b. Diverse few-shot text classification with multiple metrics. *CoRR*, abs/1805.07513.

Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. Zero-shot open entity typing as type-compatible grounding. In *EMNLP*.

A Datasets Details

A.1 Statistics

Table 5 shows the statistics of original datasets we use in the main experiments.

Dataset	Domain	# Sent	# Labels
Ontonotes	Mix	76,714	18
CoNLL'03	News	20,744	4
WNUT'07	Social	5,690	6
JNLPBA	Bio	22,402	5
NCBI-disease	Bio	7,287	1
I2B2'14	Medical	75,330	23

Table 5: Original dataset statistics.

A.2 Label Names

Table 6 shows the original label names in each dataset and corresponding natural language forms we use in our experiments.

Dataset	Original Labels	Natural Language
CoNLL'03	PER	person
	LOC	location
	ORG	organization
	MISC	miscellaneous
Ontonotes	CARDINAL	cardinal
	DATE	date
	EVENT	event
	FAC	facility
	GPE	geographical social political entity
	LANGUAGE	language
	LAW	law
	LOC	location
	MONEY	money
	NORP	nationality religion political
	ORDINAL	ordinal
	ORG	organization
	PERCENT	percent
	PERSON	person
	PRODUCT	product
	QUANTITY	quantity
	TIME	time
	WORK_OF_ART	work of art
WNUT'17	corporation	corporation
	creative-work	creative work
	group	group
	location	location
	person	person
	product	product
JNLPBA	DNA	DNA
	RNA	RNA
	cell_line	cell line
	cell_type	cell type
	protein	protein
NCBI-disease	Disease	disease
I2B2'14	AGE	age
	BIOID	biometric ID
	CITY	city
	COUNTRY	country
	DATE	date
	DEVICE	device
	DOCTOR	doctor
	EMAIL	email
	FAX	fax
	HEALTHPLAN	health plan number
	HOSPITAL	hospital
	IDNUM	ID number
	LOCATION_OTHER	location
	MEDICALRECORD	medical record
	ORGANIZATION	organization
	PATIENT	patient
	PHONE	phone number
	PROFESSION	profession
	STATE	state
	STREET	street
	URL	url
	USERNAME	username
	ZIP	zip code

Table 6: Original label names and their corresponding natural language formats.

B Support Set Sampling Algorithm

Algorithm 1 Support set sampling

Require: # shot K, dataset $\mathcal{D}$, labels $\mathcal{L}_\mathcal{D}$
Initialize support set $\mathcal{S}$={}, Count$_{\ell_i}$=0 ($\forall \ell_i \in \mathcal{L}_\mathcal{D}$)
for ℓ in $\mathcal{L}_\mathcal{D}$ **do**
  while Count$_\ell$ < K **do**
    Randomly pick $(\boldsymbol{t}, \boldsymbol{y})$ from $\mathcal{D} \setminus \mathcal{S}$ that $\boldsymbol{y}$ include ℓ
    $\mathcal{S} \leftarrow \mathcal{S} \cup (\boldsymbol{t}, \boldsymbol{y})$
    Update all Count$_{\ell_i}$ ($\forall \ell_i \in \mathcal{L}_\mathcal{D}$)
  end while
end for
for $(\boldsymbol{t}, \boldsymbol{y})$ in $\mathcal{S}$ **do**
  $\mathcal{S} = \mathcal{S} \setminus (\boldsymbol{t}, \boldsymbol{y})$
  Update all Count$_{\ell_i}$ ($\forall \ell_i \in \mathcal{L}_\mathcal{D}$)
  if Any Count$_{\ell_i}$ < K **then**
    $\mathcal{S} = \mathcal{S} \cup (\boldsymbol{t}, \boldsymbol{y})$
    Update all Count$_{\ell_i}$ ($\forall \ell_i \in \mathcal{L}_\mathcal{D}$)
  end if
end for

C Hardware for Experiments

We provide details about hardware we use to produce numbers for each baseline models. We run experiments for Struct NN shot model on NVIDIA V100 GPU with 32GB of memory, while for all other models (including baselines and our models) we use NVIDIA V100 GPU with 16GB of memory.

D Visualization of Results

We visualize the results in Table 1 with bar chart, as shown in Figure 3.

E Contextualized Label Representations

In this experiment, we compute contextualized label representations by randomly selecting a sentence from the support set that contains an entity of the type, and replace that entity with the label name in the sentence. We encode this sentence with the label encoder and compute the average pooling as the label representation. The label names used are in their natural language form with BIO schemes per 2.2. We depict this process in Figure 4. At inference time, to avoid biasing toward any particular sentence, we randomly choose 10 sentences from the support set for each label and average their representations as the final label representations.[14]

[14]When there are less than 10 sentences for a given label in the support set, we use all the available sentences. Sentences are selected once then fixed. We also experimented by randomly choosing one fixed sentence for both training and inference from the support set, but preliminary results show it is worse than our current method.

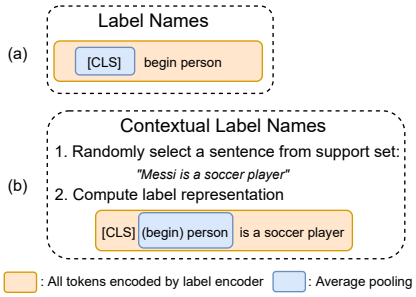

Figure 4: Differences between contextualized label representations and label representations in isolation.

We perform experiments on FEW-NERD dataset (Ding et al., 2021).[15] This dataset consists of 8 coarse-grained and 66 fine-grained entity types in hierarchy. The fine-grained entity types under the same coarse-grained type are semantically close.

Results are shown in Table 7 and Appendix E. In the following, we show 1-shot results under "Person" coarse-grained type for FEW-NERD dataset.[16] By using contextual label names, we observe a decrease in model performance by 3.5 F1 points on FEW-NERD, compared to when only label names are used. This suggests that the trained label encoder is capable of capturing critical semantics with only label names, even without contexts to help distinguish semantically close labels.

Datasets	Model	
	Ours	Ours + context
CoNLL'03	69.0±6.9	70.8±4.1
WNUT17	48.2±1.7	51.8±1.8
JNLPBA	31.5±2.9	30.1±3.2
FEW-NERD-Person	32.5±8.1	29.0±7.1

Table 7: 1-shot micro F1 on development set across various datasets and models. Ours: Our model with label names. Ours+context: Our model with contextual label names. Numbers are averaged across 10 different random samplings.

[15]As in the other experiments, we pre-finetune all models on Ontonotes then continue finetuning on target datasets.

[16]Fine-grained entity types under "Person" are: Actor, Artist/author, Athlete, Director, Politician, Scholar and Soldier.

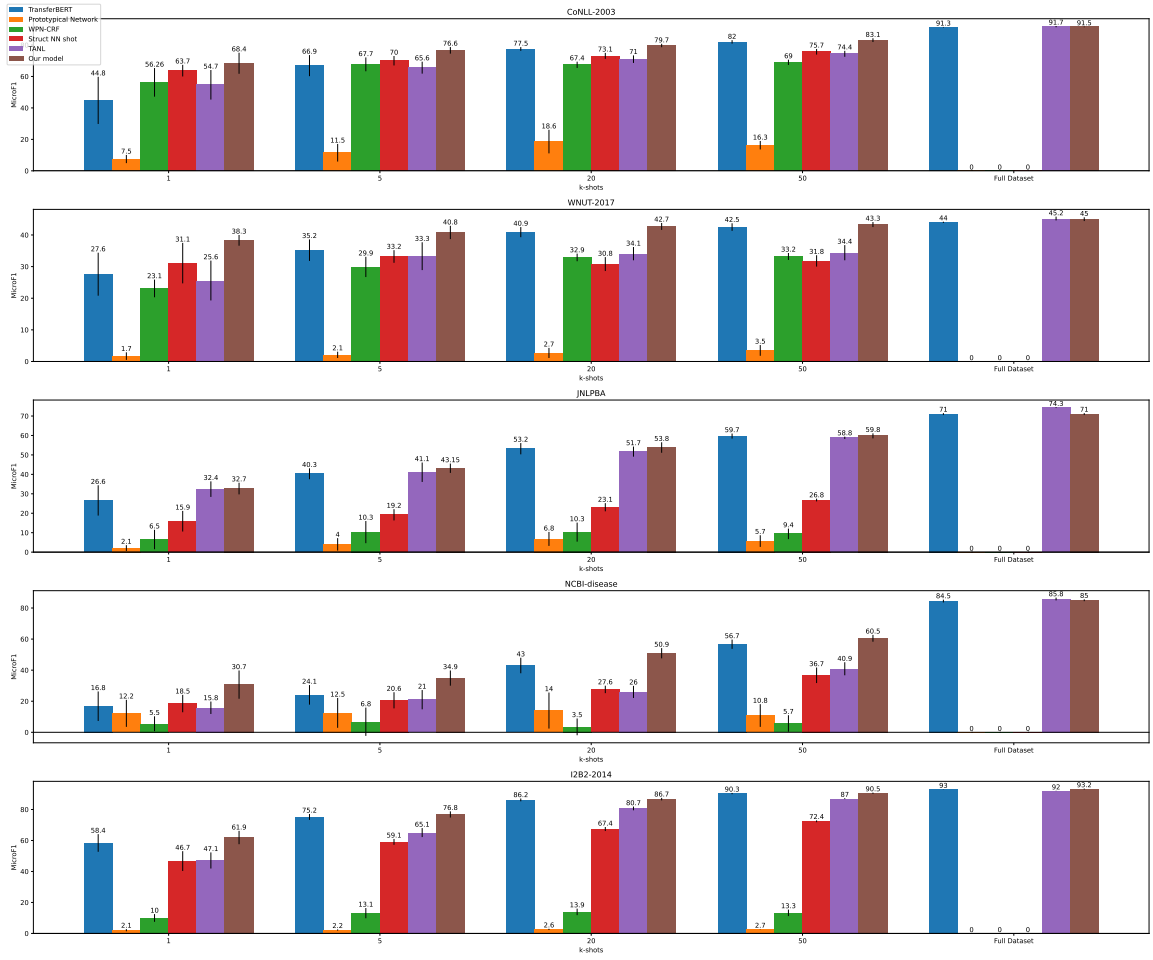

Figure 3: Visualization of the results in Table 1. Results on test set of all datasets. All numbers indicate micro F1 scores except noted otherwise. Results for low resource settings are average of 10 runs with different support set sampling. Results for high resource setting are average of 5 runs with different random seeds. For some baselines we cannot run the released implementation from originally papers due to GPU out of memory and they are marked as 0.

E.1 Additional Experiment 1

We present additional experiments on contextual label representations. We will first introduce more details on the FEW-NERD dataset, then describe methods we explore to contextualize labels, finally we will show experiment results. To validate whether contextual label representation can improve model performance in scenarios where labels are semantically close, we perform experiments on one additional dataset: FEW-NERD (Ding et al., 2021). FEW-NERD is a human annotated NER dataset that consists of 188,238 sentences. It has a hierarchy of 8 coarse-grained and 66 fine-grained entity types. The fine-grained entity types under each coarse-grained type are usually semantically close. All sentences are sourced from Wikipedia. We use train/dev/test split from the original dataset distribution.

We select "Person" and "Art" coarse-grained entity types for the experiments, because we think fine-grained entity types under them have closest semantic similarities. Specifically, we take one coarse-grained entity type at a time, and remove all entity annotations that do not belong to it, on train, dev and test split. After removal, comparing with the original dataset, the resulting dataset has much more sentences with no annotation than sentences that have at least one annotations. To mitigate this entity distribution shifting, we randomly remove sentences that do not contain any annotations, such that the resulting dataset has the same percentage of sentences with annotations as the original dataset. We perform this process on "Person" and "Art" types and result in two datasets called "FEW-NERD-Person" and "FEW-NERD-Art". The statistics for these two datasets are shown in Table 8. The original entity types and their corresponding natural language format are shown in Table 9

Dataset	Original Labels	Natural Language
FEW-NERD-Person	person-actor	actor
	person-artist/author	artist author
	person-athlete	athlete
	person-director	director
	person-politician	politician
	person-scholar	scholar
	person-soldier	soldier
FEW-NERD-Art	art-broadcastprogram	broadcast-program
	art-film	film
	art-music	music
	art-painting	painting
	art-writtenart	written art

Table 9: Original label names and their corresponding natural language formats for FEW-NERD-Person and FEW-NERD-Art datasets.

E.2 Additional Experiment 2

In this experiment, we replace the entity in the selected sentence with different texts rather than label names.

We experiment with various schemes for the new span and use the following terminology to describe them. *TOKEN* refers to the original token that is replaced. *LABEL* refers to the label name that the token is annotated with. *BIO-TAG* refers to the natural BIO tag that the token is annotated with. For the example illustrated in Figure 4, *TOKEN* corresponds to "Messi", *LABEL* corresponds to "person", *BIO-TAG* corresponds to "begin". We hypothesize that the *TOKEN* gives natural context to the labels since it is unmodified sentence, *LABEL* captures the semantic information in label names and *BIO-TAG* helps differentiate the B and I chunks for the label. In addition, we experiment to replace the entity with "[MASK]" token to make the label reprensetation close to BERT pretraining inputs. The various schemes are illustrated with example in Figure 5.

Dataset	# Labels	Support Set Shot				Dev
		1	5	20	50	
FEW-NERD-Person	7	19.0	66.7	212.7	508.9	4437.0
FEW-NERD-Art	5	41.5	123.5	412.2	2569.0	1364.0

Table 8: Number of sentences in support set and dev set for FEW-NERD-Person and FEW-NERD-Art datasets. Numbers are averaged across 10 different random samplings.

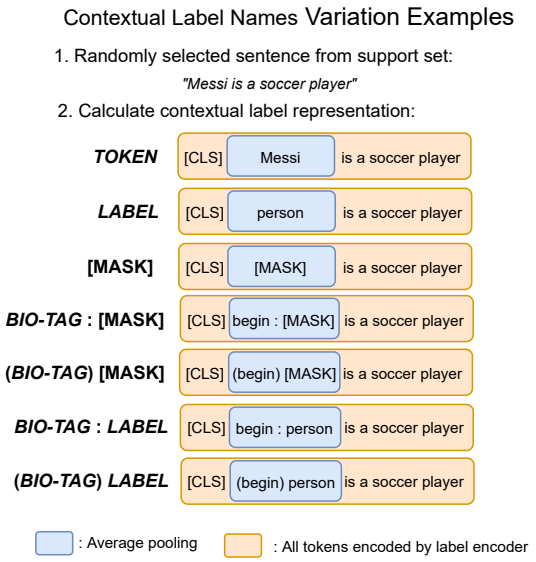

Figure 5: Example for contextual label representation.

E.3 Results

The results from various schemes of the new span is compared with TransferBERT and our model which encodes label names only. This is summarized in Table 10.

TOKEN scheme is the simplest way to get a contextualized representation of a label where we pool the representations of all the tokens annotated with the label. Although performance of this scheme is better than TransferBERT, comparing with other schemes, we see that this model performs poorly. Here no new information is added to the model and the text that the label encoder and document encoder encodes is similar. In order to provide our model prior knowledge about the label name from BERT encoder, we use *LABEL* scheme. We see that this scheme performs better than *TOKEN* across datasets suggesting that the prior knowledge about label semantics helps to improve performance.

One limitation with *LABEL* scheme is that the replaced token is same for both B and I chunks in BIO scheme. For example, to get contextualized representation for B-PER in the document "Lionel Messi is a soccer player", the document will be transformed to "person person is a soccer player", where B and I chunks are confused. "*BIO-TAG* : *LABEL*" scheme addresses this by prefixing the natural language BIO chunk name to the label name. We see improvements in performance compared with *LABEL* scheme.

When we incorporate the "[MASK]" token from BERT pretraining, we find that this does not perform as well as other schemes that contains label names. This further prove that semantics in label names is critical.

		1 Shot	5 Shot	20 Shot	50 Shot
CoNLL03	TransferBERT	47.6 ±15.5	69.9 ±6.0	80.1 ±1.7	85.1 ±1.1
	Ours, label name only	69.0 ±6.9	**78.6** ±1.8	**82.1** ±1.5	**85.9** ±1.2
	TOKEN	60.1 ±16.8	75.0 ±4.2	80.0 ±1.8	84.3 ±1.1
	LABEL	61.4 ±12.7	74.2 ±2.9	80.4 ±1.9	84.6 ±1.2
	[MASK]	61.2 ±6.1	72.9 ±5.8	81.5 ±2.2	85.3 ±0.9
	BIO-TAG : [MASK]	60.8 ±15.4	74.5 ±5.6	81.3 ±1.5	85.2 ±0.8
	(*BIO-TAG*) [MASK]	66.8 ±6.7	74.6 ±7.0	81.6 ±1.8	85.3 ±1.0
	BIO-TAG : *LABEL*	69.2 ±6.4	76.1 ±2.1	80.8 ±1.9	84.9 ±1.1
	(*BIO-TAG*) *LABEL*	**70.8** ±4.2	76.5 ±1.6	81.2 ±2.0	84.7 ±1.1
WNUT17	TransferBERT	35.6 ±11.2	44.7 ±5.6	50.3 ±1.7	51.7 ±1.9
	Ours, label name only	48.3 ±1.7	51.2 ±1.4	53.2 ±1.1	54.1 ±1.3
	TOKEN	42.8 ±12.3	49.9 ±1.9	53.1 ±1.8	53.9 ±1.8
	LABEL	48.9 ±3.0	51.4 ±2.1	53.0 ±1.6	53.9 ±1.5
	[MASK]	45.0 ±3.5	47.1 ±2.2	50.2 ±2.3	51.9 ±1.6
	BIO-TAG : [MASK]	46.8 ±2.8	49.6 ±1.7	51.3 ±2.8	52.7 ±1.0
	(*BIO-TAG*) [MASK]	45.6 ±4.8	48.5 ±2.6	51.2 ±2.7	52.6 ±1.7
	BIO-TAG : *LABEL*	51.2 ±2.2	52.6 ±1.8	53.6 ±1.4	**54.8** ±0.6
	(*BIO-TAG*) *LABEL*	**51.9** ±1.8	**52.3** ±1.2	**53.7** ±1.5	54.0 ±1.3
NCBI-diseas	TransferBERT	15.1 ±9.4	19.5 ±6.0	37.0 ±4.1	51.2 ±4.1
	Ours, label name only	**31.4** ±9.2	**30.2** ±4.3	**45.8** ±3.4	**57.3** ±2.6
	TOKEN	18.7 ±10.3	22.5 ±6.4	40.9 ±5.6	53.8 ±4.1
	LABEL	26.9 ±8.3	28.7 ±4.2	40.2 ±3.7	52.3 ±2.9
	[MASK]	18.1 ±9.6	22.2 ±4.0	38.2 ±5.3	53.0 ±4.0
	BIO-TAG : [MASK]	17.7 ±10.0	22.3 ±4.2	40.0 ±4.5	52.1 ±3.7
	(*BIO-TAG*) [MASK]	17.5 ±11.5	23.6 ±4.1	38.8 ±4.7	51.9 ±4.0
	BIO-TAG : *LABEL*	26.8 ±7.4	26.2 ±3.8	42.0 ±4.1	54.4 ±3.4
	(*BIO-TAG*) *LABEL*	26.8 ±9.2	26.7 ±3.3	43.9 ±3.8	54.6 ±3.3
JNLPBA	TransferBERT	26.3 ±8.0	41.8 ±3.0	**55.9** ±3.5	**64.3** ±1.3
	Ours, label name only	**31.5** ±3.0	**43.3** ±2.8	55.8 ±3.4	63.6 ±1.0
	TOKEN	29.0 ±6.5	43.2 ±2.4	**55.9** ±3.6	63.8 ±1.2
	LABEL	28.4 ±4.3	40.8 ±2.5	54.3 ±3.4	62.5 ±1.3
	[MASK]	25.4 ±6.5	36.5 ±2.2	51.0 ±3.7	60.2 ±1.5
	BIO-TAG : [MASK]	24.9 ±5.1	36.0 ±2.5	50.5 ±4.2	60.5 ±1.7
	(*BIO-TAG*) [MASK]	24.8 ±6.5	37.1 ±2.9	50.4 ±4.1	60.3 ±1.7
	BIO-TAG : *LABEL*	30.4 ±4.6	41.9 ±2.5	55.5 ±3.3	62.9 ±1.1
	(*BIO-TAG*) *LABEL*	30.1 ±3.2	41.4 ±2.2	55.1 ±3.2	62.8 ±1.5
FN-Person	TransferBERT	13.2 ±5.0	24.0 ±7.4	48.7 ±3.4	66.9 ±3.0
	Ours, label name only	**32.5** ±8.1	**51.0** ±7.0	**66.2** ±2.0	**72.0** ±0.7
	(*BIO-TAG*) *LABEL*	29.0 ±7.2	50.6 ±6.3	**66.2** ±2.0	71.2 ±0.9
FN-Art	TransferBERT	19.4 ±10.9	43.1 ±9.8	69.5 ±1.7	98.9 ±0.3
	Ours, label name only	**44.5** ±8.8	**56.3** ±4.6	**70.5** ±1.8	**99.1** ±0.1
	(*BIO-TAG*) *LABEL*	41.3 ±10.8	56.0 ±3.8	69.4 ±2.0	98.9 ±0.2

Table 10: Results on development set across all datasets. FN-Person = FEW-NERD-Person. FN-Art = FEW-NERD-Art. All numbers indicate micro F1 scores and are average of 10 runs with different support set sampling.