CONTAINER: Few-Shot Named Entity Recognition via Contrastive Learning

Anonymous ACL submission

Abstract

Named Entity Recognition (NER) in Few-Shot setting is imperative for entity tagging in low resource domains. Existing approaches only learn class-specific semantic features and intermediate representations from source domains. This affects generalizability to unseen target domains, resulting in suboptimal performances. To this end, we present CONTAINER, a novel contrastive learning technique that optimizes the inter-token distribution distance for Few-Shot NER. Instead of optimizing class-specific attributes, CONTAINER optimizes a generalized objective of differentiating between token categories based on their Gaussian-distributed embeddings. This effectively alleviates overfitting issues originating from training domains. Our experiments in several traditional test domains (OntoNotes, CoNLL’03, WNUT ’17, GUM) and a new large scale Few-Shot NER dataset (Few-NERD) demonstrate that on average, CONTAINER outperforms previous methods by 3%-13% absolute F1 points while showing consistent performance trends, even in challenging scenarios where previous approaches could not achieve appreciable performance.

1 Introduction

Named Entity Recognition (NER) is a fundamental NLU task that recognizes mention spans in unstructured text and categorizes them into a pre-defined set of entity classes. Recent deep-learning based approaches (Huang et al., 2015; Ma and Hovy, 2016; Lample et al., 2016; Peters et al., 2018; Devlin et al., 2018) have achieved impressive performance. As these supervised NER models require large-scale human-annotated datasets, few-shot techniques that can effectively perform NER in resource constraint settings have recently garnered a lot of attention.

Few-shot learning involves learning unseen classes from very few labeled examples (Fei-Fei et al., 2006; Lake et al., 2011; Bao et al., 2020).

Figure 1: Contrastive learning dynamics of a token (Islands) with all other tokens in an example sentence from GUM (Zeldes, 2017). CONTAINER decreases the embedding distance between tokens of the same category (PLACE) while increasing the distance between different categories (QTY. and O).

Meta-learning methods such as Prototypical Networks (Snell et al., 2017) focus on how to learn (Vinyals et al., 2016; Bao et al., 2020) from this limited data. Fritzler et al. (2019) and Hou et al. (2020) also used Prototypical Networks for few-shot NER. Yang and Katiyar (2020), on the other hand, proposed a supervised NER model that learns class-specific features and extends the intermediate representations to unseen domains.

Few-shot NER poses some unique challenges that make it significantly more difficult than other few-shot learning tasks. First, as a sequence labeling task, NER requires label assignment according to the concordant context as well as the dependencies within the labels (Lample et al., 2016; Yang and Katiyar, 2020). Second, in NER, tokens that do not refer to any defined set of entities are labeled as Outside (O). Consequently, an "O" token in the training entity set may correspond to a valid target entity in test set. This challenges the notion of entity examples being clustered around a single prototype which is the key idea of Prototypical Networks. As for Nearest Neighbor based methods such as Yang and Katiyar (2020), these methods are initially "pretrained" with the objective of source class-specific supervision. As a result, the trained weights will be closely tied to the source classes and the network will project training set O-tokens so that they get clustered in embed-
We also test our model in a very useful attribute of our model that none of the recent large dataset Few-NERD (Ding et al., 2021) in various datasets (CoNLL ’03, OntoNotes 5.0, WNUT ’17, GUM) have their distinctive tag-set distributions. Thus, sampling test episodes from the actual test data perturbs the true distribution that may not represent the actual performance. Consequently, Yang and Katiyar (2020) proposed to sample multiple support sets from the original development set for prediction in
Figure 2: Illustration of our proposed CONTAI\textsc{NER} framework based on Contrastive Learning over Gaussian Embeddings: (i) Training in source domains using training NER labels \textsc{PER} and \textsc{DATE}, (ii) Fine-tuning to target domains using target NER labels \textsc{ORG} and \textsc{LOCATION}, (iii) Assigning labels to test samples via Nearest Neighbor support set labels.

The original test set. Performance over multiple support sets are averaged to report final performance. We also use this evaluation strategy for these traditional NER datasets.

3 Method

CONTAI\textsc{NER} utilizes contrastive learning to optimize distributional divergence between different token entity representations. Instead of focusing on label specific attributes, this contradistinction explicitly trains the model to distinguish between different categories of tokens which results in better generalization. Furthermore, modeling embedding distribution instead of traditional point representation effectively lets CONTAI\textsc{NER} capture the embedding uncertainties originating from sample scarcity in target domain. Finally, this contrastive learning technique lets us carefully finetune our model even with an extremely small number of samples without overfitting which is imperative for domain adaptation.

As demonstrated in Figure 2, we first train our model in source domains. Next, we finetune model representations using few-sample support sets to adapt it to target domains. The training and finetuning of CONTAI\textsc{NER} is illustrated in Algorithm 1. Finally, we use an instance level nearest neighbor classifier for inference in test sets.

3.1 Model

Figure 2 shows the key components of our model. To generate contextualized representation of sentence tokens, CONTAI\textsc{NER} incorporates a pre-trained language model encoder \textsc{PLM}. For proper comparison against existing approaches, we use BERT \cite{devlin2018bert} as our \textsc{PLM} encoder. Thus given a sequence of \( n \) tokens \([x_1, x_2, \ldots, x_n]\), we take the final hidden layer output of the \textsc{PLM} as the intermediate representations \( h_i \in \mathbb{R}^l \).

\[
[h_1, h_2, \ldots, h_n] = \text{PLM}([x_1, x_2, \ldots, x_n]) \quad (1)
\]

These intermediate representations are then channeled through simple projection layer for generating the embedding. Unlike SimCLR \cite{chen2020simple} that uses projected point embedding for contrastive learning, we assume that token embeddings follow Gaussian distributions. Specifically, we employ projection network \( f_\mu \) and \( f_\Sigma \) for producing Gaussian distribution parameters:

\[
\mu_i = f_\mu(h_i), \quad \Sigma_i = \text{ELU}(f_\Sigma(h_i)) + (1+\epsilon) \quad (2)
\]

where \( \mu_i \in \mathbb{R}^l, \Sigma_i \in \mathbb{R}^{l \times l} \) represents mean and diagonal covariance of the Gaussian Embedding respectively; \( f_\mu \) and \( f_\Sigma \) are implemented as ReLU followed by single layer networks; ELU for exponential linear unit; and \( \epsilon \approx e^{-14} \) for numerical stability.
3.2 Training in Source Domain

For calculating the contrastive loss, we consider the KL-divergence between all valid token pairs in the sampled batch. Two tokens $x_p$ and $x_q$ are considered as positive examples if they have the same label $y_p = y_q$. Given their Gaussian Embeddings $N(\mu_p, \Sigma_p)$ and $N(\mu_q, \Sigma_q)$, we can calculate their KL-divergence as following:

$$D_{KL}[N_q||N_p] = D_{KL}[N(\mu_q, \Sigma_q)||N(\mu_p, \Sigma_p)]$$

$$= \frac{1}{2} \left( \text{Tr}(\Sigma_p^{-1} \Sigma_q) - I + \log \frac{\Sigma_p}{\Sigma_q} \right) + (\mu_p - \mu_q)^T \Sigma_p^{-1} (\mu_p - \mu_q) \quad (3)$$

Both directions of the KL-divergence are calculated since it is not symmetric.

$$d(p, q) = \frac{1}{2} \left( D_{KL}[N_q||N_p] + D_{KL}[N_p||N_q] \right) \quad (4)$$

We first train our model in resource rich source domain having training data $X_t$. At each training step, we randomly sample a batch of sequences (without replacement) $X \in X_t$ from the training set having batch size of $b$. For each $(x_i, y_i) \in X$, we obtain its Gaussian Embedding $N(\mu_i, \Sigma_i)$ by channeling the corresponding token sequence through the model (Algorithm 1: Line 3-6).

We find in-batch positive samples $X_t'$ for sample $p$ and subsequently calculate the Gaussian embedding loss of $x_p$ with respect to that of all other valid tokens in the batch:

$$X_p = \{ (x_q, y_q) \in X_t' \mid y_p = y_q, p \neq q \} \quad (5)$$

$$\ell(p) = -\log \frac{\sum_{(x_q, y_q) \in X_p} \exp(-d(p, q))}{\sum_{(x_q, y_q) \in X_t', p \neq q} \exp(-d(p, q))} \quad (6)$$

In this way we can calculate the distributional divergence of all the token pairs in the batch (Algorithm 1: Line 7-10). We do not scale the contrastive loss by any normalization factor as proposed by Chen et al. (2020) since we did not find it to be beneficial for optimization.

3.3 Finetuning to Target Domain using Support Set

After training in source domains, we finetune our model using a small number of support samples for unseen classes in the target domains. We follow a similar procedure as in training stage. When multiple few-shot samples (e.g., 5-shot) are available for the target classes, the model can effectively adapt to the new domain by optimizing KL-divergence of Gaussian Embeddings as in Eq. 4.

By contrast, for the 1-shot case, it turns out challenging for models to adapt to the target class distribution. If the model has no prior knowledge about target classes (either from direct training or indirectly from source domain training where the target class entities are marked as O-type), a single example might not be sufficient to deduce the variance of the target class distribution. Thus, for 1-shot scenario, we optimize $d'(p, q) = ||\mu_p - \mu_q||_2^2$, the squared Euclidean distance between mean of the embedding distributions, essentially treating them as point embeddings. When the model has direct or indirect prior knowledge about the target classes involved, we still optimize the KL-divergence of the distributions similar to the 5-shot scenario. In Appendix Table 8, we demonstrate that optimizing with squared Euclidean distance gives us slightly better performance in 1-shot scenario. Nevertheless, in all cases with 5-shot support set, optimizing the KL-divergence between the Gaussian Embeddings gives us the best result.

**Early Stopping**  Finetuning a model with a small support set runs the risk of overfitting. Thus an

\begin{algorithm}
\caption{Training and Finetuning of CON-TaINER}
\begin{algorithmic}[1]
\Require Training data $X_t$, Support Data $X_{sup}$, Train loss function $d_t$, Finetune loss function $d_{ft}$, $f_{\mu}, f_{\Sigma}$, PLM
\Function{Training}{1} // training in source domain
\State 2: for sampled (w/o replacement) mini-batch $X \in X_t$ do
\State 3: for all $i \equiv (x_i, y_i) \in X$ do
\State 4: $\mu_i = f_{\mu}(PLM(x_i))$ //Eq. 1
\State 5: $\Sigma_i = ELU(f_{\Sigma}(PLM(x_i))) + (1 + \epsilon)$ //Eq. 2
\State 6: end for
\State 7: for all $i \equiv (x_i, y_i) \in X$ do
\State 8: Calculate $\ell(i)$ as in Eq. 5 and 6
\State 9: end for
\State 10: $L_t = \frac{1}{|X|} \sum_{i \in X} \ell(i)$
\State 11: update $f_{\mu}, f_{\Sigma}$, PLM by backpropagation to reduce $L_t$
\State 12: end for
\State 13: if finetuning to target domain
\State 14: $L_{prev} = \infty$
\State 15: $L_{fit} = L_{prev} - 1$ //Stable Initialization
\State 16: while $L_{fit} < L_{prev}$ do
\State 17: $L_{prev} = L_{fit}$
\State 18: for all $i \equiv (x_i, y_i) \in X_{sup}$ do
\State 19: Calculate $\mu_i$ and $\Sigma_i$ using Eq. 1, 2 //Line 4.5
\State 20: end for
\State 21: for all $i \equiv (x_i, y_i) \in X_{sup}$ do
\State 22: Calculate $\ell(i)$ as in Eq. 5 and 6
\State 23: end for
\State 24: $L_{fit} = \frac{1}{|X_{sup}|} \sum_{i \in X_{sup}} \ell(i)$
\State 25: update $f_{\mu}, f_{\Sigma}$, PLM by backpropagation to reduce $L_{fit}$
\State 26: end while
\State 27: return PLM and discard $f_{\mu}, f_{\Sigma}$
\end{algorithmic}
\end{algorithm}
effective early stopping scheme needs to be maintained in the finetuning stage. Unfortunately, without access to a held-out validation set due to data scarcity in the target domain, we cannot keep tabs on the saturation point where we need to stop finetuning. To alleviate this, we rely on the calculated contrastive loss as our early stopping criteria. We continue finetuning our model until the loss starts to increase. (Algorithm 1: Line 16-17, 24).

3.4 Instance Level Nearest Neighbor Inference

After training and finetuning CONAI-NER, we extract the pretrained language model encoder \( \text{PLM} \) for inference. Similar to SimCLR (Chen et al., 2020), we found that representations before the projection layers actually contain more information than the final output representation which contributes to better performance, so \( f_u \) and \( f_v \) projection heads are not used for inference. We thus calculate the representations of the test data from \( \text{PLM} \) and find nearest neighbor support set representation for inference (Wang et al., 2019; Yang and Katiyar, 2020).

The PLM representations \( h_j^\text{sup} \) of each of the support token \( (x_j^\text{sup}, y_j^\text{sup}) \in X^\text{sup} \) can be calculated as in Eq. 1. Similarly for test data \( X^\text{test} \), we get the PLM representations \( h_i^\text{test} \) where \( x_i^\text{test} \in X^\text{test} \). Here we assign \( x_i^\text{test} \) the same label as the support token that is nearest in the PLM representation space:

\[
y_i^\text{test} = \arg \min_{y_j^\text{sup}} \| h_i^\text{test} - h_j^\text{sup} \|_2^2
\]

Viterbi Decoding Most previous works (Hou et al., 2020; Yang and Katiyar, 2020; Ding et al., 2021) noticed a performance improvement by using CRFs (Lafferty et al., 2001). Thus we also employ Viterbi decoding in the inference stage with an abstract transition distribution as in StructShot (Yang and Katiyar, 2020). For the transition probabilities, the transition between three abstract tags O, I, and I-other is estimated by counting their occurrences in the training set. Then for the target domain tag-set, these transition probabilities are evenly distributed into corresponding target distributions. The emission probabilities are calculated from Nearest Neighbor Inference stage.

Comparing domain transfer results (Table 2) against other tasks (Table 1, 4, 5) we find that, interestingly, if there is no significant domain shift involved in the test data, contrastive learning allows CONAI-NER to automatically extract label dependencies, obviating the requirement of extra Viterbi decoding stage.

4 Experiment Setups

Datasets For evaluating the Few-Shot NER capabilities, we use datasets across different domains: General (OntoNotes 5.0 (Weischedel et al., 2013)), Medical (I2B2 (Stubbs and Uzuner, 2015)), News (CoNLL’03 (Sang and De Meulder, 2003)), Social (WNUT’17 (Derczynski et al., 2017)). We also test on GUM (Zeldes, 2017) that represents a wide variety of texts: interviews, news articles, instrumental texts, and travel guides. The miscellany of domains makes it an extremely challenging dataset to work on. Ding et al. (2021) argue that the distribution of these datasets may not be suitable for proper representation of Few-Shot capability. Thus, they proposed a new large-scale dataset Few-NERD that contains 66 fine-grained entities across 8 coarse-grained entities, significantly richer than previous datasets (4-18 entities). A summary of these datasets is given in Table 3.

Table 3: Summary Statistics of Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th># Classes</th>
<th># Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>OntoNotes</td>
<td>General</td>
<td>18</td>
<td>76K</td>
</tr>
<tr>
<td>I2B’14</td>
<td>Medical</td>
<td>23</td>
<td>140K</td>
</tr>
<tr>
<td>CoNLL’03</td>
<td>News</td>
<td>4</td>
<td>20K</td>
</tr>
<tr>
<td>WNUT’17</td>
<td>Social</td>
<td>6</td>
<td>5K</td>
</tr>
<tr>
<td>GUM</td>
<td>Mixed</td>
<td>11</td>
<td>3.5K</td>
</tr>
<tr>
<td>FEW-NERD</td>
<td>Wikipedia</td>
<td>66</td>
<td>188K</td>
</tr>
</tbody>
</table>

Baselines We compare the performance of CONAI-NER with state-of-the-art Few-Shot NER models on different datasets across several settings. We first measure the model performance in traditional NER datasets in tag-set extension and domain transfer tasks as proposed in Yang and Katiyar (2020). We then evaluate our model in Few-NERD (Ding et al., 2021) dataset that is explicitly designed for Few-Shot NER and compare it against the Few-NERD leaderboard baselines. Similar to Ding et al. (2021), we take Prototypical Network based Protobert (Snell et al., 2017; Fritzler et al., 2019; Hou et al., 2020), nearest neighbor based NNShot, and additional Viterbi decoding based Structshot (Yang and Katiyar, 2020) as the main SOTA baselines.

4.1 Tag-set Extension Setting

A common use-case of Few-Shot NER is that new entity types may appear in the existing text-domain. Thus (Yang and Katiyar, 2020) proposed to evaluate tag-set extension capability in OntoNotes...
(Weischedel et al., 2013). The 18 existing entity classes are split into 3 groups: A, B, and C, each having six classes. Models are tested in each few sample support group being trained in the remaining two groups. During training, test group entities are replaced with O-tag. Since source and destination domains are the same, the training phase will induce some indirect information about unseen target entities. Consequently, during finetuning of CONTAiNER, we optimize the KL-divergence between output embeddings as in Eq. 4.

We use the same entity class splits as used by Yang and Katiyar (2020) and used bert-base-cased as the backbone encoder for all models. Since they could not share the sampled support set for licensing reasons, for a proper comparison, we sampled five sets of support samples for each group and averaged the results, as done by the authors. We show these results in Table 1.

### 4.2 Domain Transfer Setting

In this experiment, a model trained on a source domain is deployed to a previously unseen novel text-domain. Here we take OntoNotes (General) as our source text domain, and evaluate the Few-Shot performance in I2B2 (Medical), CoNLL (News), WNUT (Social) domains as in (Yang and Katiyar, 2020). Additionally, we also evaluate the performance in GUM (Zeldes, 2017) dataset due to its challenging nature. We show these results in Table 2. While all the other domains have almost no intersection with OntoNotes, the target entities in CoNLL are fully contained within OntoNotes entities, making it comparable to supervised learning.

### 4.3 Few-NERD Setting

For few-shot setting, Ding et al. (2021) proposed two different settings: Few-NERD (INTRA) and Few-NERD (INTER). In Few-NERD (INTRA) train, dev, and test sets are divided according to coarse-grained types. As a result, fine-grained entity types belonging to People, Art, Product, MISC coarse-grained types are put in the train set, Event, Building in dev set, and ORG, LOC in test set. So, there is no overlap between train, dev, test sets classes in terms of coarse-grained types. On the other hand, in Few-NERD (INTER) coarse-grained types are shared, although

<table>
<thead>
<tr>
<th>Model</th>
<th>Group A</th>
<th>Group B</th>
<th>Group C</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proto</strong></td>
<td>19.3 ± 3.9</td>
<td>22.7 ± 8.9</td>
<td>18.9 ± 7.9</td>
<td>20.3</td>
</tr>
<tr>
<td><strong>NNShot</strong></td>
<td>28.5 ± 9.2</td>
<td>27.3 ± 12.3</td>
<td>21.4 ± 9.9</td>
<td>25.7</td>
</tr>
<tr>
<td><strong>StructShot</strong></td>
<td>30.5 ± 12.3</td>
<td>28.8 ± 11.2</td>
<td>20.8 ± 9.9</td>
<td>26.7</td>
</tr>
<tr>
<td><strong>CONTAiNER</strong></td>
<td>32.2 ± 5.3</td>
<td>30.9 ± 11.6</td>
<td>32.9 ± 12.7</td>
<td>32.0</td>
</tr>
<tr>
<td>+ Viterbi</td>
<td>34.2 ± 5.1</td>
<td>30.9 ± 11.6</td>
<td>33.0 ± 12.8</td>
<td>32.1</td>
</tr>
</tbody>
</table>

Table 1: F1 scores in Tag Set Extension on OntoNotes. Group A, B, C are three disjoint sets of entity types. Results vary slightly from Yang and Katiyar (2020) as they used different support set (publicly unavailable) than ours.

<table>
<thead>
<tr>
<th>Model</th>
<th>Group A</th>
<th>Group B</th>
<th>Group C</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I2B2</strong></td>
<td>13.4 ± 3.0</td>
<td>49.9 ± 8.6</td>
<td>17.4 ± 4.9</td>
<td>24.6</td>
</tr>
<tr>
<td><strong>CoNLL</strong></td>
<td>15.3 ± 1.6</td>
<td>61.2 ± 10.4</td>
<td>22.7 ± 7.4</td>
<td>27.4</td>
</tr>
<tr>
<td><strong>WNUT</strong></td>
<td>21.4 ± 3.8</td>
<td>62.4 ± 10.5</td>
<td>24.2 ± 8.0</td>
<td>29.0</td>
</tr>
<tr>
<td><strong>GUM</strong></td>
<td>16.4 ± 1.7</td>
<td>57.8 ± 10.7</td>
<td>24.2 ± 2.9</td>
<td>29.1</td>
</tr>
<tr>
<td><strong>CONTAiNER</strong></td>
<td>21.5 ± 1.7</td>
<td>62.1 ± 10.7</td>
<td>27.5 ± 1.9</td>
<td>32.2</td>
</tr>
<tr>
<td>+ Viterbi</td>
<td>21.5 ± 1.7</td>
<td>62.1 ± 10.7</td>
<td>27.5 ± 1.9</td>
<td>32.2</td>
</tr>
</tbody>
</table>

Table 2: F1 scores in Domain Extension with OntoNotes as the source domain. Results vary slightly compared to Table 1: F1 scores in Tag Set Extension on OntoNotes. Group A, B, C are three disjoint sets of entity types. Results vary slightly from Yang and Katiyar (2020) as they used different support set (publicly unavailable) than ours.

<table>
<thead>
<tr>
<th>Model</th>
<th>1-shot 5-way</th>
<th>1-shot 10-way</th>
<th>5-shot 1-way</th>
<th>5-shot 10-way</th>
<th>Avg. 1-way</th>
<th>Avg. 10-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StructShot</strong></td>
<td>35.92 ± 2.0</td>
<td>38.83 ± 2.7</td>
<td>25.38 ± 2.1</td>
<td>26.39 ± 2.2</td>
<td>31.63 ± 1.8</td>
<td>31.63 ± 1.8</td>
</tr>
<tr>
<td><strong>ProtoBERT</strong></td>
<td>23.45 ± 1.3</td>
<td>41.93 ± 2.4</td>
<td>19.76 ± 1.3</td>
<td>34.61 ± 2.5</td>
<td>29.94 ± 1.6</td>
<td>29.94 ± 1.6</td>
</tr>
<tr>
<td><strong>NNShot</strong></td>
<td>31.01 ± 1.7</td>
<td>35.74 ± 2.9</td>
<td>21.88 ± 1.4</td>
<td>27.67 ± 1.8</td>
<td>29.08 ± 1.5</td>
<td>29.08 ± 1.5</td>
</tr>
<tr>
<td><strong>CONTAiNER</strong></td>
<td>40.43 ± 2.7</td>
<td>53.70 ± 2.2</td>
<td>33.84 ± 2.0</td>
<td>47.49 ± 2.7</td>
<td>43.87 ± 2.0</td>
<td>43.87 ± 2.0</td>
</tr>
<tr>
<td>+ Viterbi</td>
<td>40.40 ± 2.7</td>
<td>53.71 ± 2.2</td>
<td>33.82 ± 2.0</td>
<td>47.51 ± 2.7</td>
<td>43.86 ± 2.0</td>
<td>43.86 ± 2.0</td>
</tr>
</tbody>
</table>

Table 4: F1 scores in FEW-NERD (INTRA).

<table>
<thead>
<tr>
<th>Model</th>
<th>1-shot 5-way</th>
<th>1-shot 10-way</th>
<th>5-shot 1-way</th>
<th>5-shot 10-way</th>
<th>Avg. 1-way</th>
<th>Avg. 10-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StructShot</strong></td>
<td>57.33 ± 2.7</td>
<td>57.16 ± 2.4</td>
<td>49.46 ± 2.0</td>
<td>49.39 ± 2.1</td>
<td>53.34 ± 2.0</td>
<td>53.34 ± 2.0</td>
</tr>
<tr>
<td><strong>ProtoBERT</strong></td>
<td>44.44 ± 2.4</td>
<td>58.80 ± 2.7</td>
<td>39.09 ± 2.0</td>
<td>53.97 ± 2.5</td>
<td>49.08 ± 2.0</td>
<td>49.08 ± 2.0</td>
</tr>
<tr>
<td><strong>NNShot</strong></td>
<td>54.29 ± 2.0</td>
<td>50.56 ± 2.4</td>
<td>46.98 ± 2.0</td>
<td>50.00 ± 2.5</td>
<td>50.46 ± 2.0</td>
<td>50.46 ± 2.0</td>
</tr>
<tr>
<td><strong>CONTAiNER</strong></td>
<td>55.95 ± 2.7</td>
<td>61.83 ± 2.2</td>
<td>48.35 ± 2.0</td>
<td>57.12 ± 2.6</td>
<td>55.81 ± 2.0</td>
<td>55.81 ± 2.0</td>
</tr>
<tr>
<td>+ Viterbi</td>
<td>56.1 ± 2.7</td>
<td>61.90 ± 2.2</td>
<td>48.36 ± 2.0</td>
<td>57.13 ± 2.6</td>
<td>55.87 ± 2.0</td>
<td>55.87 ± 2.0</td>
</tr>
</tbody>
</table>

Table 5: F1 scores in FEW-NERD (INTER).
all the fine-grained types are mutually disjoint. Because of the restrictions of sharing coarse-grained types, Few-NERD (INTRA) is more challenging. Since few-shot performance of any model relies on the sampled support set, the authors also released train, dev, test split for both Few-NERD (INTRA) and Few-NERD (INTER). We evaluate our model performance using these provided dataset splits and compare the performance in the Few-NERD leaderboard. All models use bert-base-uncased as the backbone encoder. As shown in Table 4 and Table 5, CONTAiNER establishes new benchmark results in the leaderboard in both of these tests.

### 5 Results and Analysis

We prudently analyze different components of our model and justify the design choices in CONTAiNER. We also examine the results discussed in the previous section that gives some intuitions about few-shot NER in general.

#### 5.1 Overall Results

Table 1-5 demonstrates that overall, in every scenario CONTAiNER convincingly outperforms all other baseline approaches. This improvement is particularly noticeable in challenging scenarios, where all other baseline approaches perform poorly. For example, FEW-NERD (intra) (Table 4) is a challenging scenario where the coarse-grained entity types corresponding to train and test sets do not overlap. Here, other baseline approaches face substantial performance hit, whereas CONTAiNER still performs well. In tag-set extension (Table 1), we see a similar performance trend - CONTAiNER performs consistently well across the board. Likewise, in domain transfer to a very challenging unlabelled scenario CONT performs better than other baseline approaches. This improvement is particularly noticeable in challenging scenarios, where the coarse-grained entity types corresponding to train and test sets do not overlap.

#### 5.2 Training Objective

Traditional contrastive learners usually optimize cosine similarity of point embeddings (Chen et al., 2020). While this has proven to work well in image data, in more challenging NLU tasks like Few Shot NER, we find it to give subpar performance. We see an example scenario in Table 6, where we show the performance comparison of tag-set extension in Group A of OntoNotes. Basically, modeling class distribution via Gaussian Distribution allows us to capture the uncertainties in Few-Shot embedding, which is useful in this data scarce environment. This is one of the novel features of CONTAiNER that sets it apart from competition.

#### 5.3 Effect of Model Fine-tuning

Being a contrastive learner, CONTAiNER can take advantage of extremely small support set to refine its representations through fine-tuning. To closely examine the effects of fine-tuning, we conduct a case study with OntoNotes tag-extension task using PERSON, DATE, MONEY, LOC, FAC, PRODUCT target entities. From Table 7, we see that finetuning indeed improves few-shot performance. Besides, the effect of finetuning is even more marked in 5-shot case indi-

<table>
<thead>
<tr>
<th></th>
<th>Point Embedding</th>
<th>Gaussian Embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-shot</td>
<td>6.37</td>
<td>32.17</td>
</tr>
<tr>
<td>5-shot</td>
<td>25.47</td>
<td>51.20</td>
</tr>
</tbody>
</table>

Table 6: F1 scores comparison in OntoNotes Group A. Gaussian Embedding surpasses point embedding based optimization in all cases.

<table>
<thead>
<tr>
<th></th>
<th>W/O Finetuning</th>
<th>W/ Finetuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-shot</td>
<td>31.76</td>
<td>32.90</td>
</tr>
<tr>
<td>5-shot</td>
<td>56.99</td>
<td>61.48</td>
</tr>
</tbody>
</table>

Table 7: Comparison of F1-Scores with and without support set finetuning of CONTAiNER
547
cating that CONTAINER finetuning can make the
best use of few-samples available in target domain.

5.4 Modeling Label Dependencies
Analyzing the results, we observe that domain
transfer (Table 2) sees some good gains in perfor-
ance from using Viterbi decoding. In contrast,
tag-set extension (Table 1) and FEW-NERD (Ta-
ble 4.5) gets almost no improvement from Viterbi
decoding. This indicates an interesting property
of CONTAINER. During domain transfer the text
domains have no overlap in train and test set. So,
an extra Viterbi decoding actually provides addi-
tional information regarding the label dependen-
cies. On the other hand, the train and target text
domain have substantial overlap in both tagset ex-
tension and FEW-NERD. Thus CONTAINER can
indirectly learn the label dependencies through in-
batch contrastive learning. Consequently, unless
there is a marked shift in the target text domain,
we can achieve the best performance even without
employing additional Viterbi decoding.

6 Related Works

Meta Learning The idea of Few-shot learning
was popularized in computer vision through Match-
ing Networks (Vinyals et al., 2016). Subsequently,
Prototypical Network (Snell et al., 2017) was pro-
posed where class prototypical representations
were learned. Test samples are labelled accordin-
g to the nearest prototype. Later this technique
was proven successful in other domains as well.
Wang et al. (2019), on the other hand found sim-
ple feature transformations to be quite effective in
few shot image recognition. These metric learning
based approaches have also been deployed in differ-
ent NLP tasks (Geng et al., 2019; Bao et al., 2020;
Han et al., 2018; Fritzler et al., 2019).

Contrastive Learning Early progress was made
by contrasting positive against negative samples
(Hadsell et al., 2006; Dosovitskiy et al., 2014; Wu
et al., 2018). Chen et al. (2020) proposed SimCLR
by refining the idea of contrastive learning by utiliz-
ing modern image augmentation to learn robust sets
of features. Khosla et al. (2020) leveraged this to
boost supervised learning performance as well. In-
batch negative sampling has also been explored for
learning representation (Doersch and Zisserman,
2017; Ye et al., 2019). Storing instance class rep-
References


A Implementation Details

For all of our experiments, we chose the same hyperparameters as in Yang and Katiyar (2020). Across all our tests, we kept Gaussian Embedding dimension fixed to $l = 128$. In order to guarantee proper comparison against prior competitive approaches, we use the same backbone encoder for all methods in same tests, i.e. bert-base-cased was used for all methods in Tag-Set Extension and Domain Transfer tasks while bert-base-uncased was used for Few-NERD following the respective evaluation strategies. Finally, to observe the effect of Viterbi decoding on CONTAI NER output, we set the renormalizing temperature $\tau$ to 0.1. We will also release the full source code of CONTAI NER upon publication. All the model trainings were conducted using an RTX A6000 GPU.

B Fine-tuning Objective

During finetuning, if a model does not have any prior knowledge about the target classes, directly or indirectly, a 1-shot example may not give sufficient information about the target class distribution (i.e. the variance of the distribution). Consequently during finetuning, for 1-shot adaptation to new classes, optimizing euclidean distance of the mean embedding gives better performance. Nevertheless, for 5-shot cases, KL-divergence of the Gaussian Embedding always gives better performance indicating that it takes better advantage of multiple samples. We show this behavior in the best result of domain transfer task with WNUT in Table 8. Since this domain transfer task gives no prior information about target embeddings during training, optimizing KL-divergence in 1-shot fineutuning actually hurts performance a bit compared to euclidean fine-tuning. However, in 5-shot, KL-finetuning again gives superior performance as it can now adapt better to the novel target class distributions.

C NER Prediction Examples

Table 9 demonstrates some predictions with CONTAI NER and StructShot using PERSON, DATE, MONEY, LOC, FAC, PRODUCT as target few-shot entities while being trained on all other entity types in OntoNotes dataset.

<table>
<thead>
<tr>
<th></th>
<th>KL-Gaussian</th>
<th>Euclidean-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-shot</td>
<td>18.78</td>
<td>27.48</td>
</tr>
<tr>
<td>5-shot</td>
<td>32.50</td>
<td>31.12</td>
</tr>
</tbody>
</table>

Table 8: F1 scores comparison in Domain Transfer Task with WNUT with different finetune objectives. While optimizing the KL-divergence of the Gaussian Embedding gives superior result in 5-shot, optimizing Euclidean distance of the mean embeddings actually achieve better result in 1-shot.
<table>
<thead>
<tr>
<th>Gold</th>
<th>CONTAINER</th>
<th>StructShot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gold</strong></td>
<td><strong>CONTAINER</strong></td>
<td><strong>StructShot</strong></td>
</tr>
<tr>
<td>BMEC general director Dr. Johnsee Lee ___PER says that the ITRI 's four-year DATE R&amp;D program in biochip applications and technology is now in its second year DATE.</td>
<td>BMEC general director Dr. Johnsee Lee ___PER says that the ITRI 's four-year DATE R&amp;D program in biochip applications and technology is now in its second year DATE.</td>
<td>BMEC general director Dr. Johnsee Lee ___PER says that the ITRI 's four-year DATE R&amp;D program in biochip applications and technology is now in its second year.</td>
</tr>
<tr>
<td>DR. Chip Bio-technology was set up in September 1998 DATE.</td>
<td>DR. Chip Bio-technology was set up in September 1998 DATE.</td>
<td>DR. Chip Bio-technology PRODUCT was set up in September 1998.</td>
</tr>
<tr>
<td>Wang Shin - hwan ___PER notes that traditional bacterial and viral cultures take seven to ten days to prepare , and even with the newer molecular biology testing techniques it takes three days DATE to get a result.</td>
<td>Wang Shin - hwan ___PER notes that traditional bacterial and viral cultures take seven to ten days DATE to prepare , and even with the newer molecular biology testing techniques it takes three days DATE to get a result.</td>
<td>Wang Shin - hwan notes that traditional bacterial and viral cultures take seven to ten days DATE to prepare , and even with the newer molecular biology testing techniques it takes three days DATE to get a result.</td>
</tr>
<tr>
<td>Research program director Pan Chao - chi ___PER states that at present they are actively developing a &quot; fever chip &quot; with a wide range of applications.</td>
<td>Research program director Pan Chao - chi ___PER states that at present they are actively developing a &quot; fever chip &quot; with a wide range of applications.</td>
<td>Research program director Pan Chao - chi ___PER states that at present they are actively developing a &quot; fever chip &quot; with a wide range of applications.</td>
</tr>
<tr>
<td>Pan explains that in clinical practice , the causes of fever are difficult to quickly diagnose.</td>
<td>Pan explains that in clinical practice , the causes of fever are difficult to quickly diagnose.</td>
<td>Pan explains that in clinical practice , the causes of fever are difficult to quickly diagnose.</td>
</tr>
<tr>
<td>Jerry Huang ___PER, executive vice president of U - Vision Biotech ___PRODUCT reveals that U - Vision ___PRODUCT , which was set up in September 1999 ___DATE , has signed a contract with the US company Zen - Bio to jointly develop human adipocyte cDNA microarray chips.</td>
<td>Jerry Huang ___PER, executive vice president of U - Vision Biotech ___PRODUCT reveals that U - Vision ___PRODUCT , which was set up in September 1999 ___DATE , has signed a contract with the US company Zen - Bio to jointly develop human adipocyte cDNA microarray chips.</td>
<td>Jerry Huang , executive vice president of U - Vision Biotech ___PRODUCT reveals that U - Vision ___PRODUCT , which was set up in September 1999 ___DATE , has signed a contract with the US company Zen - Bio to jointly develop human adipocyte cDNA microarray chips.</td>
</tr>
<tr>
<td>Huang ___PER states that research in recent years DATE has revealed that adipocytes -LR fat cells -RR are active regulators of the energy balance in the body , and play an important role in disorders such as obesity , diabetes , osteoporosis and cardiovascular disease.</td>
<td>Huang ___PER states that research in recent years DATE has revealed that adipocytes -LR fat cells -RR are active regulators of the energy balance in the body , and play an important role in disorders such as obesity , diabetes , osteoporosis and cardiovascular disease.</td>
<td>Huang states that research in recent years has revealed that adipocytes -LR fat cells -RR are active regulators of the energy balance in the body , and play an important role in disorders such as obesity , diabetes , osteoporosis and cardiovascular disease.</td>
</tr>
<tr>
<td>Maybe a 30 year old DATE man &amp; a 15 year old DATE boy doesn’t qualify.</td>
<td>Maybe a 30 year old DATE man &amp; a 15 year old DATE boy doesn’t qualify.</td>
<td>Maybe a 30 year old DATE man &amp; a 15 year old DATE boy doesn’t qualify.</td>
</tr>
<tr>
<td>Jerry Huang ___PER, executive vice president of U - Vision Biotech ___PRODUCT reveals that U - Vision ___PRODUCT , which was set up in September 1999 ___DATE , has signed a contract with the US company Zen - Bio to jointly develop human adipocyte cDNA microarray chips.</td>
<td>Jerry Huang , executive vice president of U - Vision Biotech ___PRODUCT reveals that U - Vision ___PRODUCT , which was set up in September 1999 ___DATE , has signed a contract with the US company Zen - Bio to jointly develop human adipocyte cDNA microarray chips.</td>
<td>Jerry Huang , executive vice president of U - Vision Biotech ___PRODUCT reveals that U - Vision ___PRODUCT , which was set up in September 1999 ___DATE , has signed a contract with the US company Zen - Bio to jointly develop human adipocyte cDNA microarray chips.</td>
</tr>
<tr>
<td>Russian and Norwegian divers searched the fourth compartment of the wrecked submarine Kursk PRODUCT, Sunday DATE, but they found too much damage to proceed with the task of recovering bodies.</td>
<td>Russian and Norwegian divers searched the fourth compartment of the wrecked submarine Kursk PRODUCT, Sunday DATE, but they found too much damage to proceed with the task of recovering bodies.</td>
<td>Russian and Norwegian divers searched the fourth compartment of the wrecked submarine Kursk, Sunday DATE, but they found too much damage to proceed with the task of recovering bodies.</td>
</tr>
<tr>
<td>Today , the enterovirus chip is in the testing phase , and DR. Chip is collaborating with Taipei Veterans General Hospital to obtain samples with which to establish the accuracy of the chip.</td>
<td>Today DATE , the enterovirus chip is in the testing phase, and DR. Chip PRODUCT is collaborating with Taipei Veterans General Hospital FAC to obtain samples with which to establish the accuracy of the chip.</td>
<td>Today DATE , the enterovirus chip is in the testing phase, and DR. Chip PRODUCT is collaborating with Taipei Veterans General Hospital to obtain samples with which to establish the accuracy of the chip.</td>
</tr>
</tbody>
</table>
And I think perhaps no one more surprised than some of the people running those firms on Wall Street.

We’re all getting this news in from the speech that the Homeland Security Secretary Tom Ridge is expected to be delivering at the international press club around 1:00 Eastern at the top of the hour.

Yesterday American pilots mechanics approved their share $1.8 billion in labor concession.

<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Person</th>
<th>Event</th>
<th>Time</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yesterday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9: NER Prediction Examples from OntoNotes with PERSON, DATE, MONEY, LOC, FAC, PRODUCT as target few-shot entities