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. The source code of CONTAINER will be available at: https://github.com/ANONYMOUS/container.

1 Introduction

Named Entity Recognition (NER) is a fundamental NLU task that recognizes mention spans in unstructured text and categorizes them into a predefined set of entity classes. In spite of its challenging nature, 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.

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).

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). To avoid overfitting with the limited available data, meta-learning has been introduced to focus on how to learn (Vinyals et al., 2016; Bao et al., 2020). Snell et al. (2017) proposed Prototypical Networks to learn a metric space where the examples of a specific unknown class cluster around a single prototype. Although it was primarily deployed in computer vision, 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. Additionally, they employed a Viterbi decoding variant of their model as “StructShot”. 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, a token that is labeled as O in training entity set may correspond to a valid target entity in test set. For prototypical networks, this challenges the notion of entity exam-
samples being clustered around a single prototype. As for Nearest Neighbor based methods such as Yang and Katiyar (2020), they 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 embedding space. This will force the embeddings to drop a lot of useful features pertaining to its true target entity in the test set. Third, in few-shot setting, there are not enough samples from which we can select a validation set. This reduces the capability of hyperparameter tuning, which particularly affects template based methods where prompt selection is crucial for good performance (Cui et al., 2021). In fact, the absence of held-out validation set puts a lot of earlier few-shot works into question whether their strategy is truly “Few-Shot” (Perez et al., 2021).

To deal with these challenges, we present a novel approach, CONTAINER that harnesses the power of contrastive learning to solve Few-Shot NER. CONTAINER tries to decrease the distance of token embeddings of similar entities while increasing it for dissimilar ones (Figure 1). This enables CONTAINER to better capture the label dependencies. Also, since CONTAINER is trained with a generalized objective, it can effectively avoid the pitfalls of O-tokens that the prior methods struggle with. Lastly, CONTAINER does not require any dataset specific prompt or hyperparameter tuning. Standard settings used in prior works (Yang and Katiyar, 2020) works well across different domains in different evaluation settings.

Unlike traditional contrastive learners (Chen et al., 2020; Khosla et al., 2020) that optimize similarity objective between point embeddings, CONTAINER optimizes distributional divergence effectively modeling Gaussian Embeddings. While point embedding simply optimizes sample distances, Gaussian Embedding faces an additional constraint of maintaining class distribution through the variance estimation. Thus Gaussian Embedding explicitly models entity class distributions which not only promotes generalized feature representation but also helps in few-sample target domain adaptation. Previous works in Gaussian Embedding has also shown that mapping to a density captures representation uncertainties (Vilnis and McCallum, 2014) and expresses natural asymmetries (Qian et al., 2021) while showing better generalization requiring less data to achieve optimal performance (Bojchevski and Günne mann, 2017).

Inspired by these unique qualities of Gaussian Embedding, in this work we leverage Gaussian Embedding in contrastive learning for Few-Shot NER.

A nearest neighbor classification scheme during evaluation reveals that on average, CONTAINER significantly outperforms previous SOTA approaches in a wide range of tests by up to 13% absolute F1-points. In particular, we extensively test our model in both in-domain and out-of-domain experiments as proposed in Yang and Katiyar (2020) in various datasets (CoNLL ’03, OntoNotes 5.0, WNUT ’17, I2B2). We also test our model in a large dataset recently proposed for Few-Shot NER - Few-NERD (Ding et al., 2021) where CONTAINER outperforms all other SOTA approaches setting a new benchmark result in the leaderboard.

In summary, our contributions are as follows:
(1) We propose a novel Few-Shot NER approach CONTAINER that leverages contrastive learning to infer distributional distance of their Gaussian Embeddings. To the best of our knowledge we are the first to leverage Gaussian Embedding in contrastive learning for Named Entity Recognition.
(2) We demonstrate that CONTAINER representations are better suited for adaptation to unseen novel classes, even with a low number of support samples. (3) We extensively test CONTAINER in a wide range of experiments using several datasets and evaluation schemes. In almost every case, our model largely outperforms present SOTAs establishing new benchmark results.

2 Task Formulation

Given a sequence of n tokens \( \{x_1, x_2, \ldots, x_n\} \), NER aims to assign each token \( x_i \) to its corresponding tag label \( y_i \).

**Few-shot Setting** For Few-shot NER, a model is trained in a source domain with a tag-set \( \{C_{\text{s}}^i\} \) and tested in a data-scarce target domain with a tag-set \( \{C_{\text{d}}^j\} \) where \( i, j \) is index of different tags. Since \( \{C_{\text{s}}^i\} \cap \{C_{\text{d}}^j\} = \emptyset \), it is very challenging for models to generalize to unseen test tags. In an N-way K-shot setting, there are \( N \) tags in the target domain \( \{|C_{\text{d}}^j|\} = N \), and each tag is associated with a support set with \( K \) examples.

**Tagging Scheme** For fair comparison of CONTAINER against previous SOTA models, we follow an IO tagging scheme where I-type repres-
Figure 2: Illustration of our proposed CONTAI\textsuperscript{NER} framework based on Contrastive Learning over Gaussian Embeddings: (i) Training in source domains using training NER labels \textit{PER} and \textit{DATE}, (ii) Fine-tuning to target domains using target NER labels \textit{ORG} and \textit{LOCATION}, (iii) Assigning labels to test samples via Nearest Neighbor support set labels.

\textbf{Evaluation Scheme} To compare with SOTA models in Few-NERD leaderboard (Ding et al., 2021), we adopt \textit{episode evaluation} as done by the authors. Here, a model is assessed by calculating the micro-F1 score over multiple number of test episodes. Each episode consists of a \textit{K-shot} support set and a \textit{K-shot} unlabeled query (test) set to make predictions. While Few-NERD is explicitly designed for episode evaluation, traditional NER datasets (e.g., OntoNotes, CoNLL’03, 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 and use them for prediction in the original test set. We also use this evaluation strategy for these traditional NER datasets.

\textbf{3 Method}

CONTAI\textsuperscript{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. Furthermore, modeling Gaussian Embedding instead of traditional point representation effectively lets CONTAI\textsuperscript{NER} model the entity class distribution, which incites generalized representation of tokens. Finally, it lets us carefully finetune our model even with a 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\textsuperscript{NER} is illustrated in Algorithm 1. Finally, we use an \textit{instance level nearest neighbor classifier} for inference in test sets.

\textbf{3.1 Model}

Figure 2 shows the key components of our model. To generate contextualized representation of sentence tokens, CONTAI\textsuperscript{NER} incorporates a pre-trained language model encoder \textit{PLM}. For proper comparison against existing approaches, we use BERT (Devlin et al., 2018) as our \textit{PLM} encoder. Thus given a sequence of \textit{n} tokens \([x_1, x_2, \ldots, x_n]\), we take the final hidden layer output of the \textit{PLM} as the intermediate representations \(h_i \in \mathbb{R}'\).

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

These intermediate representations are then channeled through simple projection layer for generating the embedding. Unlike SimCLR (Chen et al., 2020) 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 \( \mu_i \in \mathbb{R}^d, \Sigma_i \in \mathbb{R}^{d \times d} \) represents mean and diagonal covariance (with nonzero elements only along the diagonal of the matrix) 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 \( \mathcal{N}(\mu_p, \Sigma_p) \) and \( \mathcal{N}(\mu_q, \Sigma_q) \), we can calculate their KL-divergence as following:

\[
D_{\text{KL}}[\mathcal{N}_q || \mathcal{N}_p] = D_{\text{KL}}[\mathcal{N}(\mu_q, \Sigma_q) || \mathcal{N}(\mu_p, \Sigma_p)]
\]

\[
= \frac{1}{2} \left[ \text{Tr}(\Sigma_p^{-1} \Sigma_q) + (\mu_p - \mu_q)^T \Sigma_p^{-1} (\mu_p - \mu_q) - l + \log \frac{|\Sigma_p|}{|\Sigma_q|} \right]
\]

(3)

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

\[
d(p, q) = \frac{1}{2} (D_{\text{KL}}[\mathcal{N}_q || \mathcal{N}_p] + D_{\text{KL}}[\mathcal{N}_p || \mathcal{N}_q])
\]

(4)

We first train our model in resource rich source domain having training data \( \mathcal{X}_0 \). At each training step, we randomly sample a batch of sequences (without replacement) \( \mathcal{X} \in \mathcal{X}_0 \) from the training set having batch size of \( b \). For each \( (x_i, y_i) \in \mathcal{X} \), we obtain its Gaussian Embedding \( \mathcal{N}(\mu_i, \Sigma_i) \) by channeling the corresponding token sequence through the model (Algorithm 1: Line 3-6). We find in-batch positive samples \( \mathcal{X}_p \) 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:

\[
\mathcal{X}_p = \{(x_q, y_q) \in \mathcal{X} \mid y_p = y_q, p \neq q \}
\]

(5)

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 target domain support samples following a similar procedure as in the training stage. As we have only a few samples for finetuning, we take them in a single batch.

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. In contrast, for 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. When the model has direct/indirect prior knowledge about the target classes involved, we still optimize the KL-divergence of the distributions similar to the 5-shot scenario.

We demonstrate in Table 7 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 with a small support set runs the risk of overfitting and 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 and use it as our early stopping criteria with a patience of 1. (Algorithm 1: Line 16-17, 24)
Algorithm 1 Training and Finetuning of CONTAiNER

Require: Training data $X_t$, Support Data $X_{sup}$, Train loss function $d_{tr}$, Finetune loss function $d_{ft}$, $f_s$, $f_c$, $PLM$
1: // training in source domain
2: for sampled (w/o replacement) minibatch $X \in X_t$ do
3: for all $i \equiv (x_i, y_i) \in X$ do
4: $\mu_i = f_s(PLM(x_i))$ //Eq. 1
5: $\Sigma_i = ELU(f_c(PLM(x_i))) + (1 + \epsilon)$ //Eq. 2
6: end for
7: for all $i \equiv (x_i, y_i) \in X$ do
8: Calculate $\ell(i)$ as in Eq. 5 and 6
9: end for
10: $L_{tr} = \frac{1}{|X|} \sum_{i \in X} \ell(i)$
11: update $f_\mu, f_c, PLM$ by backpropagation to reduce $L_{tr}$
12: end for
13: // finetuning to target domain
14: $L_{prev} = \infty$
15: $L_{fs} = L_{prev} - 1$ //Stable Initialization
16: while $L_{fs} < L_{prev}$ do
17: $L_{prev} = L_{fs}$
18: for all $i \equiv (x_i, y_i) \in X_{sup}$ do
19: Calculate $\mu_i$ and $\Sigma_i$ using Eq. 1, 2 //Line 4, 5
20: end for
21: for all $i \equiv (x_i, y_i) \in X_{sup}$ do
22: Calculate $\ell(i)$ as in Eq. 5 and 6
23: end for
24: $L_{fs} = \frac{1}{|X_{sup}|} \sum_{i \in X_{sup}} \ell(i)$
25: update $f_\mu, f_c, PLM$ by backpropagation to reduce $L_{fs}$
26: end while
27: return $PLM$ and discard $f_\mu, f_c$

3.4 Instance Level Nearest Neighbor Inference

After training and finetuning the network with train and support data respectively, we extract the pretrained language model encoder $PLM$ 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 $f_s$ and $f_c$ projection heads are not used for inference. We thus calculate the representations of the test data from $PLM$ and find nearest neighbor support set representation for inference (Wang et al., 2019; Yang and Katiyar, 2020).

The $PLM$ representations $h_{sup}^{(j)}$ of each of the support token $(x_{sup}^{(j)}, y_{sup}^{(j)}) \in X_{sup}$ can be calculated as in Eq. 1. Similarly for test data $X_{test}$, we get the $PLM$ representations $h_{test}^{(j)}$ where $x_{test}^{(j)} \in X_{test}$. Here we assign $x_{test}^{(j)}$ the same label as the support token that is nearest in the $PLM$ representation space:

$$y_{test}^j = \arg \min_{y_{sup}^k} \| h_{test}^j - h_{sup}^k \|_2$$ (7)
Table 1: F1 scores in Tag Set Extension on OntoNotes. Group A, B, C are three disjoint sets of entity types. Results vary slightly compared to Yang and Katiyar (2020) since they used different support set samples (publicly unavailable) than ours.

| Model          | Group A | Group B | Group C | Avg. | Group A | Group B | Group C | Avg. |
|----------------|---------|---------|---------|------|---------|---------|---------|------|
| Proto          | 19.3 ± 3.9 | 22.7 ± 8.9 | 18.9 ± 7.9 | 20.3 | 30.5 ± 3.5 | 38.7 ± 5.6 | 41.1 ± 3.3 | 36.7 |
| NNShot         | 28.5 ± 9.2 | 27.3 ± 12.3 | 21.4 ± 9.7 | 25.7 | 44.0 ± 2.1 | 51.6 ± 5.9 | 47.6 ± 2.8 | 47.7 |
| StructShot     | 30.5 ± 12.3 | 28.8 ± 11.2 | 20.8 ± 9.9 | 26.7 | 47.5 ± 4.0 | 53.0 ± 7.9 | 48.7 ± 2.7 | 49.8 |
| CONTaiNER      | 32.8 ± 5.3 | 30.9 ± 11.6 | 32.9 ± 12.7 | 32.0 | 51.2 ± 5.9 | 55.9 ± 6.2 | 61.5 ± 2.7 | 56.2 |
| + Viterbi      | 32.4 ± 5.1 | 30.9 ± 11.6 | 33.0 ± 12.8 | 32.1 | 51.2 ± 6.0 | 56.0 ± 6.2 | 61.5 ± 2.7 | 56.2 |

Table 2: F1 scores in Domain Extension with OntoNotes as the source domain. Results vary slightly compared to Yang and Katiyar (2020) since they used different support set samples (publicly unavailable) than ours.

| Model | 1-shot | 5-shot | 1-shot | 5-shot |
|-------|--------|--------|--------|--------|
|       | 12B2   | CoNLL  | WNUT   | GUM    | Avg.   | 12B2   | CoNLL  | WNUT   | GUM    | Avg.   |
| Proto | 13.4 ± 3.0 | 49.9 ± 8.6 | 17.4 ± 4.9 | 17.8 ± 3.5 | 24.6 | 17.9 ± 1.8 | 61.3 ± 9.1 | 22.8 ± 4.5 | 19.5 ± 3.4 | 30.4 |
| NNShot | 15.3 ± 1.6 | 61.2 ± 10.4 | 22.7 ± 7.4 | 10.5 ± 2.9 | 27.4 | 22.0 ± 1.5 | 74.1 ± 2.3 | 27.3 ± 5.4 | 15.9 ± 1.8 | 34.8 |
| StructShot | 21.4 ± 3.8 | 62.4 ± 10.5 | 24.2 ± 8.0 | 7.8 ± 2.1 | 29.0 | 30.3 ± 2.1 | 74.8 ± 2.4 | 30.4 ± 6.5 | 13.3 ± 1.3 | 37.2 |
| CONTaiNER | 16.4 ± 1.7 | 57.8 ± 10.7 | 24.2 ± 2.9 | 17.9 ± 1.8 | 29.1 | 24.1 ± 1.9 | 72.8 ± 2.0 | 27.7 ± 2.2 | 24.4 ± 2.2 | 37.3 |
| + Viterbi | 21.5 ± 1.7 | 61.2 ± 10.7 | 27.5 ± 1.9 | 18.5 ± 4.9 | 32.2 | 36.7 ± 2.1 | 75.8 ± 2.7 | 32.5 ± 3.8 | 25.2 ± 2.7 | 42.6 |

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 12B2 (Medical), CoNLL (News), WNUT (Social) domains as in (Yang and Katiyar, 2020). We also evaluate the performance in GUM (Zeldes, 2017) dataset due to its particularly challenging nature. We show these results in Table 2. While all the other domains have almost no intersection with OntoNotes, target entities in CoNLL are fully contained within OntoNotes entities, that makes 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 coarse grained types in dev set, and ORG, LOC in test set. So, there is no overlap between train, dev, test set classes in terms of coarse grained types. On the other hand, in Few-NERD (INTER) coarse grained types are shared, although 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
We prudently analyze different components of our model and justify the design choices made in the scheming of CONTAiNER. We also examine the results discussed in "Experiments" section that gives some intuitions about few-shot NER in general.

5 Results and Analysis

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

5.1 Overall Results

Table 1-4 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 3) is a challenging scenario where the coarse grained entity types corresponding to train and test sets do not overlap. As a result, other baseline approaches face a 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 unseen text domain like GUM (Zeldes, 2017), baseline models performs miserably; yet CONTAiNER manages to perform consistently outperforming SOTA models by a significant margin. Analyzing these results more closely, we notice that while CONTAiNER surpasses other baselines in almost every tests, more prominently in 5-shot cases. Evidently, CONTAiNER is able to make better use of multiple few-shot samples thanks to distribution modeling via contrastive Gaussian Embedding optimization. In this context, note that StructShot actually got marginally higher F1-score in 1-shot CoNLL domain adaptation and 1~2 shot FEW-NERD (INTER) cases. In CoNLL, the target classes are subsets of training classes, so supervised learning based feature extractors are expected to get an advantage in prediction. On the other hand, Ding et al. (2021) carefully tuned the hyperparameters for baselines like StructShot for best performance. We could also improve performance in a similar manner, however for uniformity of model across different few-shot settings, we use the same model architecture in every test. Nevertheless, CONTAiNER shows comparable performance even in these cases while significantly outperforming in every other test.

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, it gives subpar performance. We compare the performance of point embeddings with euclidean distance and cosine similarity to that of CONTAiNER using Gaussian Embedding and KL-divergence in OntoNotes tag-set extension. We report these performance in Table 8 in Appendix. Basically, Gaussian Embedding leads to learning generalized representation during training, which is more suitable for finetuning to few sample target domain. In Appendix E, we examine this aspect by comparing the t-SNE representations from point embedding and Gaussian Embedding.

5.3 Modeling Label Dependencies

Analyzing the results, we observe that domain transfer (Table 2) sees some good gains in performance from using Viterbi decoding. In contrast, tag-set extension (Table 1) and FEW-NERD (Table 3, 4) gets almost no improvement from using 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 additional information regarding the label dependencies, giving us some nice improvement. Otherwise, the train
and target domain have substantial overlap in both
tagset extension and FEW-NERD. Thus the model
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-
ning Networks (Vinyals et al., 2016). Subsequently,
Prototypical Network (Snell et al., 2017) was pro-
posed where class prototypical representations
were learned. Test samples are given labels accord-
ing to the nearest prototype. Later this technique
was proven successful in other domains as well.
Simple feature transformation has also been suc-
cessfully used in Few-Shot Learning. (Wang et al.,
2019; 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 with the
help of modern image augmentation techniques to
learn robust sets of features. Khosla et al. (2020)
leveraged this to boost supervised learning perfor-
ance as well. In-batch negative sampling has also
been explored for learning representation (Doer-
sch and Zisserman, 2017; Ye et al., 2019). Storing
instance class representation vectors is another pop-
ular direction (Wu et al., 2018; Zhuang et al., 2019;
Misra and Maaten, 2020).

Gaussian Embedding  Vilnis and McCallum
(2014) first explored the idea of learning word em-
beddings as Gaussian Distributions. Although the
authors used RANK-SVM based learning objective
instead of modern deep contextual modeling,
they found that embedding densities in a Gaussian
space enables natural representation of uncertainty
through variances. Later, Bojchevski and Günne-
mann (2017) leveraged Gaussian Embedding in
Graph representation. Besides state-of-the-art per-
formance, they found Gaussian Embedding to be
surprisingly effective in inductive learning, gen-
eralizing to unseen nodes with few training data.
Moreover, KL-divergence between Gaussian Em-
beddings allows explicit consideration of asym-
metric distance which better represents inclusion,
similarity or entailment (Qian et al., 2021) and
preserve the hierarchical structures among words
(Athiwaratkun and Wilson, 2018).

Few-Shot NER For few shot NER, Fritzler et al.
(2019) leveraged prototypical network (Snell et al.,
2017). Inspired by the potency of simple feature
extractors and nearest neighbor inference (Wang
et al., 2019; Wiseman and Stratos, 2019) in few-
Shot learning, Yang and Katiyar (2020) used super-
vised learner based feature extractors for Few-Shot
NER. Pairing it with abstract transition tag Viterbi
decoding, they achieved current SOTA result in
Few-Shot NER tasks. The role of data augmentation
in low-resource NER has also been explored
(Ding et al., 2020). Huang et al. (2020) on the
other hand proposed noisy supervised pre-training
which requires access to a large scale noisy NER
dataset such as WiNER (Ghaddar and Langlais,
2017) for the supervised pretraining. Acknowl-
edging the shortcomings and evaluation scheme
disparity in Few-Shot NER, Ding et al. (2021)
proposed a large scale dataset specifically designed for
this task. Wang et al. (2021b) explored model distil-
lation for Few-Shot NER. Prompt based techniques
have also surfaced in this domain (Cui et al., 2021;
Ma et al., 2021; Chen et al., 2021; Wang et al.,
2021a). However, the performance of these meth-
ods rely heavily on the chosen prompt. As denoted
by Cui et al. (2021), the performance delta can be
massive (upto 19% absolute F1 points) depending
on the prompt. Thus, in the absence of a large val-
ification set, their applicability becomes limited in
true few-shot learning (Perez et al., 2021).

7 Conclusion

We propose a contrastive learning based frame-
work CONTaNER that models Gaussian embed-
ding and optimizes inter token distribution distance.
This generalized objective helps us model a class
agnostic feature extractor that avoids the pitfalls of
prior Few-Shot NER methods. CONTaNER
can also take advantage of few-sample support data
to adapt to new target domains. Extensive evalu-
ations in multiple traditional and recent few-shot
NER datasets reveal that, CONTaNER consis-
tently outperforms prior SOTAs, even in challeng-
ing scenarios. While we investigate the efficacy of
distribution optimization based contrastive learning
in Few-Shot NER, it will be of particular interest
to investigate its potency in other domains as well.
8 Ethics Statement

With CONTAINER, we have achieved state-of-the-art Few-Shot NER performance leveraging Gaussian Embedding based contrastive learning. However, the overall performance is still quite low compared to supervised NER that takes advantage of the full training dataset. Consequently, it is still not ready for deployment in high-stake domains (e.g., Medical Domain, I2B2 dataset), leaving a lot of room for improvement in future research.

References

Ben Athiwaratkun and Andrew Gordon Wilson. 2018. Hierarchical density order embeddings. arXiv preprint arXiv:1804.09843.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In ICLR.

Aleksandar Bojchevski and Stephan Günnemann. 2017. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint arXiv:1707.03815.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In ICML.

Xiang Chen, Ningyu Zhang, Lei Li, Xin Xie, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Lightner: A lightweight generative framework with prompt-guided attention for low-resource ner. arXiv preprint arXiv:2109.00720.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart. arXiv preprint arXiv:2106.01760.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shaflag Joty, Luo Si, and Chunyan Miao. 2020. Daga: Data augmentation with a generation approach for low-resource tagging tasks. arXiv preprint arXiv:2011.01549.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. 2021. Few-nert: A few-shot named entity recognition dataset. arXiv preprint arXiv:2105.07464.

Carl Doersch and Andrew Zisserman. 2017. Multi-task self-supervised visual learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 2051–2060.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. 2014. Discriminative unsupervised feature learning with convolutional neural networks. Advances in neural information processing systems, 27:766–774.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence, 28(4):594–611.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pages 993–1000.

Ruiyi Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. arXiv preprint arXiv:1902.10482.

Abbas Ghaddar and Philippe Langlais. 2017. Winer: A wikipedia annotated corpus for named entity recognition. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 413–422.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), volume 2, pages 1735–1742. IEEE.

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. arXiv preprint arXiv: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. arXiv preprint arXiv:2006.05702.

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. arXiv preprint arXiv:2012.14978.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991.

Pranay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. arXiv preprint arXiv:2004.11362.
Oriol Vinyals, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. 2011. One shot learning of simple visual concepts. In Proceedings of the annual meeting of the cognitive science society.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01340.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Template-free prompt tuning for few-shot ner. arXiv preprint arXiv:2109.13532.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354.

Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6707–6717.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. arXiv preprint arXiv:2105.11447.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Chen Qian, Fuli Feng, Lijie Wen, and Tat-Seng Chua. 2021. Conceptualized and contextualized gaussian embedding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 13683–13691.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050.

Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. arXiv preprint arXiv:1703.05175.

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

Luke Vilnis and Andrew McCallum. 2014. Word representations via gaussian embedding. arXiv preprint arXiv:1412.6623.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. NeurIPS.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. 2019. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. arXiv preprint arXiv:1911.04623.

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021a. Learning from language description: Low-shot named entity recognition via decomposed framework. arXiv preprint arXiv:2109.05357.

Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2021b. Meta self-training for few-shot neural sequence labeling. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 1737–1747.

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.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. arXiv preprint arXiv:1906.04225.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3733–3742.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. arXiv preprint arXiv:2010.02405.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6210–6219.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. Language Resources and Evaluation.

Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. 2019. Local aggregation for unsupervised learning of visual embeddings. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6002–6012.
A Implementation Details

For all of our experiments in CONT\(\text{AI}\)NER, 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 back-bone encoder for all methods in same tests, i.e. \texttt{bert-base-cased} was used for all methods in Tag-Set Extension and Domain Transfer tasks while \texttt{bert-base-uncased} was used for Few-\text{NERD} following the respective evaluation strategies. Finally, to observe the effect of Viterbi decoding on CONT\(\text{AI}\)NER output, we set the normalization temperature \(\tau\) to 0.1.

Using an RTX A6000, we trained the network on OntoNotes dataset for 30 minutes. The finetuning stage requires less than a minute due to the small number of samples.

B Datasets

A summary statistics of the datasets used in our evaluation is given below in Table 5

| Dataset     | Domain     | # Class | # Sent |
|-------------|------------|---------|--------|
| OntoNotes   | General    | 18      | 76K    |
| I2B2’14     | Medical    | 23      | 140K   |
| CoNLL’03    | News       | 4       | 20K    |
| WNUT’17     | Social     | 6       | 5K     |
| GUM         | Mixed      | 11      | 3.5K   |
| FEW-NERD    | Wikipedia  | 66      | 188K   |

Table 5: Summary Statistics of Datasets

C Effect of Model Fine-tuning

Being a contrastive learner, CONT\(\text{AI}\)NER 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 \texttt{PERSON}, \texttt{DATE}, \texttt{MONEY}, \texttt{LOC}, \texttt{FAC}, \texttt{PRODUCT} target entities.

| W/O Finetuning | W/ Finetuning |
|----------------|--------------|
| 1-shot         | 31.76        |
| 5-shot         | 56.99        |

Table 6: Comparison of F1-Scores with and without support set finetuning of CONT\(\text{AI}\)NER

As shown in Table 6, we see that finetuning indeed improves few-shot performance. Besides, the effect of finetuning is even more marked in 5-shot prediction indicating that CONT\(\text{AI}\)NER finetuning process can make the best use of few-samples available in target domain.

D 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 7. Since this domain transfer task gives no prior information about target embeddings during training, optimizing KL-divergence in 1-shot finetuning 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.

| W/O Finetuning | W/ Finetuning |
|----------------|--------------|
| 1-shot         | 32.50        |
| 5-shot         | 32.50        |

Table 7: 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. Note that in both cases the model is trained on out-of-domain data using KL-Gaussian.

E t-SNE Visualization: Point Embedding vs. Gaussian Embedding

Figure 3 offers a deep dive into how Gaussian Embedding improves generalization and takes better advantage of few shot support set for target domain adaptation. Here we compare the t-SNE visualization of support set and test set of a sample few-shot scenario in OntoNotes tag set extension task. In Figure 3 (a) we can see that point embedding
Figure 3: t-SNE visualization of support set and test set representations in a sample few-shot task in OntoNotes tag extension. We show both support and test set representation here before and after finetuning. Prior to finetuning, (a) contrastive learner with point embedding and Euclidean distance objective gives intermixed class representations; (b) Gaussian Embedding with KL-divergence generates clusters for different unseen classes. After finetuning, (c) point embedding overfits the support examples which further intermingles different class examples; (d) Gaussian Embedding with KL-divergence cleans up the clusters offering better separation between different classes, which results in higher F1-score.
Table 8: OntoNotes Tag Set extension mean-F1 score comparison between Point Embedding (with Euclidean distance and cosine similarity) and Gaussian Embedding (KL-divergence).

| Model                              | 1-shot     | 5-shot     |
|------------------------------------|------------|------------|
|                                    | Group A    | Group B    | Group C    | Avg. | Group A    | Group B    | Group C    | Avg.   |
| Point Embedding + Cosine           | 7.73       | 11.27      | 15.57      | 11.52 | 17.33      | 30.08      | 22.51      | 23.31 |
| Point Embedding + Euclidean        | 14.96      | 13.67      | 11.12      | 13.25 | 25.35      | 41.56      | 43.11      | 36.67 |
| Gaussian Embedding + KL-div.       | 32.2       | 30.9       | 32.9       | 32.0  | 51.2       | 55.9       | 61.5       | 56.2  |

G Embedding Quality: Before vs. After Projection

Table 9: Comparison of F1-Scores on OntoNotes Group A before and after the projection layer of CONTaiNER

|                  | Before Projection | After Projection |
|------------------|-------------------|------------------|
| 1-shot           | 32.17             | 29.21            |
| 5-shot           | 51.19             | 49.78            |

As explained in Section 3.4, the representation before the projection layer contains more information than that of after. In Table 9, we compare the performance of representations before and after the Gaussian projection layer. From the results it is evident that, representation before the projection indeed achieves higher performance, which also supports the findings of (Chen et al., 2020). This is because the representation after the projection head is directly adjacent to the contrastive objective, which causes information loss in this layer. Consequently, the representation before projection achieves better performance.

H NER Prediction Examples

Table 10 demonstrates some predictions with CONTaiNER 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. A quick look at these qualitative examples reveal that StructShot often fails to distinguish between non-entity and entity tokens. Moreover, it also misclassifies non-entity tokens as one of the target classes. CONTaiNER on the other hand has lower misclassifications and better entity detection indicating its stability and higher performance.
| Gold | CONTAINER | StructShot |
|------|-----------|------------|
| BMEC general director Dr. Johnsee Lee\_PER says that the ITRI’s four-year DATE R&D program in biochip applications and technology is now in its second year DATE. | BMEC general director Dr. Johnsee Lee\_PER says that the ITRI’s four-year DATE R&D program in biochip applications and technology is now in its second year DATE. | BMEC general director Dr. Johnsee Lee\_PER says that the ITRI’s four-year DATE R&D program in biochip applications and technology is now in its second year. |
| DR. Chip Bio-technology was set up in September 1998 DATE. | DR. Chip Bio-technology was set up in September 1998 DATE. | DR. Chip Bio-technology PRODUCT was set up in September 1998. |
| 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. | 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. | 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. |
| Research program director Pan Chao - chi\_PER states that at present they are actively developing a "fever chip" with a wide range of applications. | Research program director Pan Chao - chi\_PER states that at present they are actively developing a "fever chip" with a wide range of applications. | Research program director Pan Chao - chi\_PER states that at present they are actively developing a "fever chip" with a wide range of applications. |
| Pan explains that in clinical practice, the causes of fever are difficult to quickly diagnose. | Pan explains that in clinical practice, the causes of fever are difficult to quickly diagnose. | Pan explains that in clinical practice, the causes of fever are difficult to quickly diagnose. |
| Jerry Huang\_PER, executive vice president of U - Vision Biotech, reveals that U - Vision, 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. | Jerry Huang\_PER, executive vice president of U - Vision Biotech, reveals that U - Vision, 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. | Jerry Huang\_PER, executive vice president of U - Vision Biotech, reveals that U - Vision, 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. |
| 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. | 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. | 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. |
| Maybe a 30 year old DATE man & a 15 year old DATE boy doesn’t qualify. | Maybe a 30 year old DATE man & a 15 year old DATE boy doesn’t qualify. | Maybe a 30 year old DATE man & a 15 year old DATE boy doesn’t qualify. |
| After Tom DeLay\_PER was zapped, Charles Colson\_PER became DeLay’s personal guru. | After Tom DeLay\_PER was zapped, Charles Colson\_PER became DeLay’s personal guru. | After Tom DeLay was zapped, Charles Colson\_PER became DeLay’s personal guru. |
| She does not sit still or lay still for you to change her Pampers\_PRODUCT. | She does not sit still or lay still for you to change her Pampers\_PRODUCT. | She does not sit still or lay still for you to change her Pampers\_PRODUCT. |
| 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. | 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. | 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. |
| Zinni\_PER testifying after the attack on the “USS Cole”\_PRODUCT – Aden never had a specific terrorist threat. | Zinni\_PER testifying after the attack on the “USS Cole”\_PRODUCT – Aden never had a specific terrorist threat. | Zinni\_PER testifying after the attack on the “USS Cole”\_PRODUCT – Aden never had a specific terrorist threat. |
| 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. | Today, 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. | Today, 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. |
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 10: NER Prediction Examples from OntoNotes with PERSON, DATE, MONEY, LOC, FAC, PRODUCT as target few-shot entities