A Streamlined Span-based Factorization Method for Few Shot Named Entity Recognition

WenJie Xu[1,*], JianQuan OuYang[2,*]
Xiangtan University
xuwenjie444@gmail.com, oyjq@xtu.edu.cn

Abstract

Few-shot named entity recognition (NER) is a challenging task that aims to recognize new named entities with only a limited amount of labeled examples. In this paper, we introduce SSF, which is a streamlined span-based factorization method that addresses the problem of few-shot NER. Our approach formulates few-shot NER as a span-level alignment problem between query and support instances. To achieve this goal, SSF decomposes the span-level alignment problem into several refined span-level procedures. The proposed approach encompasses several key modules such as the Span Boosting Module, Span Prototypical Module, Span Alignment Module, and Span Optimization Module. Our experimental results demonstrate a significant improvement over the previous state-of-the-art performance. Specifically, compared to previous methods, our proposed approach achieves an average F1 score improvement of 12 points on the FewNERD dataset and 10 points on the SNIPS dataset. Moreover, our approach has surpassed the latest state-of-the-art performance on both datasets.

Keywords: Named Entity Recognition, Few-Shot Named Entity Recognition

1. Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP). It aims to identify and categorize named entities in text, such as person names, organizations, locations, dates, and other entities. Prior approaches (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Peters et al., 2017) have introduced various deep neural architectures that demonstrate promising results. However, these approaches require a large amount of labeled data, which is laborious and time-consuming to collect. As a result, the challenge of few-shot NER (Ding et al., 2021; Ziyadi et al., 2020; Hou et al., 2020) has emerged. Few-shot NER involves learning to identify and categorize unseen entity classes from a limited number of labeled examples. This task has attracted significant attention from the research community in recent years.

Prompt-based techniques (Chen et al., 2022; Cui et al., 2021) have been implemented to address the challenges of few-shot NER. These techniques have shown remarkable potential in exploiting the knowledge embedded in pre-trained language models (PLMs). In contrast to conventional fine-tuning approaches, prompt-based methods have shown superior performance in both cross-domain and few-shot tasks. However, a drawback of these methods is that their effectiveness is highly dependent on the quality and design of the prompts chosen, which can affect their stability. As a result, the application of prompt-based techniques to few-shot learning is challenging without a sufficiently large validation dataset.

Previous studies on few-shot NER have primarily used token-level metric learning (Snell et al., 2017; Fritzler et al., 2019; Yang and Katiyar, 2020; Hou et al., 2020). Under this method, each query token compares with each entity class prototype or support example token to label it based on their distance. Although effective in some cases, this approach tends to overlook the integrity of named entities that might include multiple tokens that form a text span instead of a single word. Furthermore, defining prototypes for the 'O' class, which represents non-entities, is often challenging because of the large vocabulary's high frequency of common words that usually don't share common features. As a result, inaccuracies in few-shot NER can occur when attempting to accurately identify and classify noisy prototypes.

To address these limitations, we propose SSF, a streamlined span-based factorization method that formulates few-shot NER as a span-level alignment problem between the support set and the query set.

Specifically, the SSF methodology factorizes the span alignment problem into four main span-level procedures. One of the main modules in the SSF methodology is the Span Boosting Module, which enhances the span representation by combining information from contiguous spans within the same sentence and the interplay between the query and support set using Localized Span Attention and Trans-Span Attention. Another module in the SSF methodology is the Span Prototypical Module, which consolidates the span vectors for each class in the support set and forms them into a prototype

* Corresponding author

LREC-COLING 2024, pages 1673–1683
20-25 May, 2024.

representation. The Span Alignment Module in the SSF methodology aligns the enhanced span representation from the Span Boosting Module with the prototype representation from the Span Prototypical Module. To address conflicts between predicted spans in the alignment paradigm at the span-level, the SSF methodology includes the Span Optimization Module, which utilizes the Adaptive Soft-Beam Non-Maximum Suppression algorithm (ASBNMS). We conducted extensive experiments on two widely recognized benchmarks: FewNERD (Ding et al., 2021) and SNIPS (Coucke et al., 2018). The experimental results show a significant improvement in performance compared to the previous state-of-the-art methods. Our proposed approach achieves an average F1 score improvement of 12 points on the FewNERD dataset and 10 points on the SNIPS dataset. In summary, our main contributions are:

- We proposed the SSF methodology, which is a streamlined span-based factorization method that solves the problem of few-shot NER using four main modules.
- We presented the adaptive soft-beam non-maximum suppression algorithm to address conflicts in span predictions.
- Our methodology achieves unprecedented levels of performance compared to previous state-of-the-art approaches, as demonstrated by extensive experimentation on two widely recognised benchmarks.

2. Task Formulation

Given a sequence $X = \{x_1, x_2, \dots, x_n\}$ with N tokens, NER aims to assign each token x_i a corresponding label $y_i \in \mathcal{Y} \cup \{O\}$, where $\mathcal{Y}$ is the entity type set and O denotes the non-entity label. This paper focuses on the standard N-way K-shot setting for the few-shot NER task, as outlined in Ding et al. (2021). An illustrative example of a 2-way 1-shot episode is provided in Table 1. During the training phase, we construct episodes denoted by $\mathcal{E}_{train} = \{(\mathcal{S}_{train}, \mathcal{Q}_{train}, \mathcal{Y}_{train})\}$ utilizing labeled data from the source domain. The support set $\mathcal{S}_{train} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N\times K}$ comprises N×K labeled examples, while the query set $\mathcal{Q}_{train} = \{x^{(j)}, y^{(j)}\}_{j=1}^{N\times K'}$ unlabeled examples. Entity classes are denoted by $\mathcal{Y}_{train}$ with a cardinality of N. During the testing phase, we evaluate our model's ability to generalize to novel domains by constructing new episodes $\mathcal{S}_{new} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N\times K}$ in a similar fashion to the training data. In the few-shot NER task, we aim to equip our trained model with the capability to leverage the support set $\mathcal{S}_{new} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N\times K}$ of a new episode $(\mathcal{S}_{new}, \mathcal{Q}_{new}, \mathcal{Y}_{new}) \in \mathcal{E}_{new}$ to predict labels for the query set $\mathcal{Q}_{new} = \{x^{(j)}\}_{j=1}^{N\times K'}$. Here, $\mathcal{Y}_{new}$ denotes the set of entity classes associated with $\mathcal{S}_{new}$ and $\mathcal{Q}_{new}$ with a cardinality of N. Notably, the entity classes in $\mathcal{Y}_{train}$ and $\mathcal{Y}_{new}$ are disjoint. $\forall\ \mathcal{Y}_{train}, \mathcal{Y}_{new}, \mathcal{Y}_{train} \cap \mathcal{Y}_{new} = \emptyset$.

Target Types $\mathcal{Y}$	[person-actor], [art-film]
Support set $\mathcal{S}$	(1) *Brad Pitt*[person-actor] is an accomplished and talented film actor. (2) *Titanic*[art-film] is a classic and beloved romantic drama film.
Query Set $\mathcal{Q}$	Tom Cruise starred in Top Gun, a classic '80s action movie.
Expected output	*Tom Cruise*[person-actor] starred in *Top Gun*[art-film] , a classic '80s action movie.

Table 1: An example of the 2-way 1-shot setting where different entity classes are distinguished by contrasting colors.

3. Methodology

Our SSF framework aims to resolve the challenge of aligning spans in few-shot NER by decomposing the problem into a series of targeted procedures designed to achieve precise span matches. Figure 1 provides a graphical representation of the innovative architecture underlying the SSF framework.

3.1. Span Initialization Module

This module is designed to generate span representations for a given task $\mathcal{E} = \{(\mathcal{S}, \mathcal{Q}, \mathcal{Y})\}$. To achieve this, we employ BERT (Devlin et al., 2019) as our encoder and use the function f_θ to obtain contextualized representations $h = \{h_i\}_{i=1}^{L}$ for all tokens in a sentence $x = \{x_i\}_{i=1}^{L}$ that belong to sets S and Q.

$$h = f_\theta(x) \tag{1}$$

Here, h $\in R^{L\times d}$ represents the output at the last layer of the encoder, where *d* denotes the embedding size. To represent a span T= (l, r) within a sentence x, where l and r represent the start and end positions of the span within the sentence, we derive its initial representation $T_{(l,r)}$ by concatenating two hidden states and applying a weight matrix W. Specifically, we concatenate the hidden state of the first token h_l with that of the last token h_r, then multiply the concatenated vector with the weight matrix W.

$$T_{(l,r)} = [h_l; h_r]W \tag{2}$$

We enumerate spans in the sentence with a maximum length of L, where L is a hyperparameter used

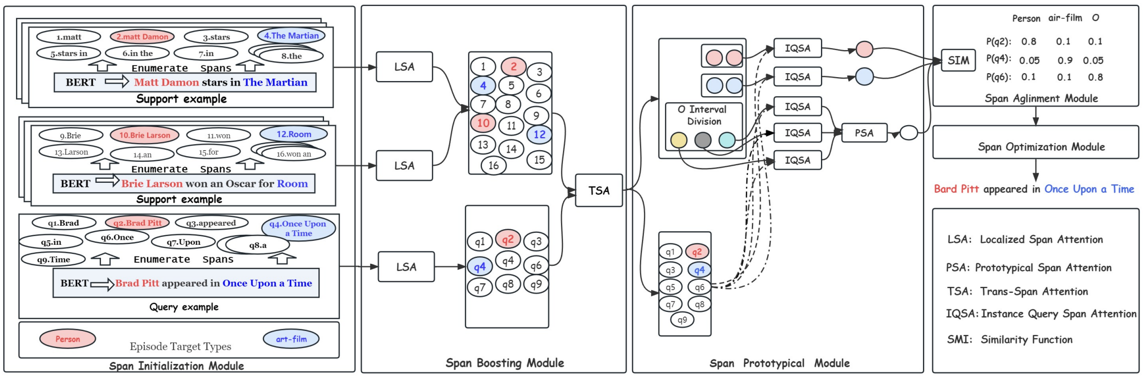

Figure 1: The model architecture of SSF. We only enumerate spans with lengths less than 2 for clarity.

to control the number of O-class spans, in order to reduce memory consumption and processing time.

3.2. Span Boosting Module

Through our research, we have found that incorporating information from other spans within the same sentence or across multiple sentences can enhance the accuracy of the model. In addition to examining the interactions between the query and the support set, we introduce the span boost module, which enhances the span representation by exploiting these contextual factors. The span boost module not only enhances model performance, but also provides a more sophisticated and nuanced representation for few-shot NER.

3.2.1. Localized Span Attention

Within a sentence, the semantic content of a particular span can often be deduced by examining its relationship to other spans. Given all the span representations of a sentence $S \in R^{B \times d}$, where B is the number of spans. We denote the i-th row of S as s_i, which represents the i-th span in the sentence.

$$H_i = s_i + PE_i \tag{3}$$

$$\hat{s_i} = MultiheadAttention(H_i, H_i, H_i) \tag{4}$$

To obtain the final LSA enhanced feature $\hat{out_i}$, we utilize a Feed Forward Neural network (FFN) (Vaswani et al., 2017) that incorporates Residual Connection (He et al., 2016) and Layer Normalization (Ba et al., 2016).

$$out_i = LayerNorm(s_i + \hat{s_i}) \tag{5}$$

$$\overline{out_i} = LayerNorm(FFN(out_i) + out_i) \tag{6}$$

$$\hat{out_i} = \overline{out_i} W_{lsa} + b \tag{7}$$

Here, PE_i denotes the i-th learnable position encoding. W_{lsa} represents the weight matrix for the linear transformation, and b denotes the bias term.

3.2.2. Trans-Span Attention

After implementing the LSA module, certain measures were taken to ensure that the spans of the query sentences and support sentences were aligned, which facilitated seamless localized span interactions. To enhance the query spans, we propose using Trans-Span Attention (TSA). The span representations of the query sentences, Q, as well as those of the support set, S, which have both been improved by the LSA module, are given by $Q \in R^{B_q \times d}$ and $S \in R^{B_s \times d}$. We use $\bar{q_i}$ to denote the i-th row of Q, and $\bar{s_j}$ to denote the j-th row of S. Finally, we obtain the final TSA-enhanced representation of $\bar{q_i}$ and $\bar{s_j}$ by following the steps below:

$$\hat{s_j} = MultiheadAttention(\bar{s_j}, \bar{q_i}, \bar{q_i}) \tag{8}$$

$$\overline{out_j} = \hat{s_j} W_s + b \tag{9}$$

$$Final_j = LayerNorm(\overline{out_j} + \hat{s_j}) \tag{10}$$

$$\hat{q_i} = MultiheadAttention(\bar{q_i}, \bar{s_j}, \bar{s_j}) \tag{11}$$

$$\overline{out_i} = \hat{q_i} W_q + b \tag{12}$$

$$Final_i = LayerNorm(\overline{out_i} + \hat{q_i}) \tag{13}$$

The final TSA-enhanced representation of $\bar{s_j}$ and $\bar{q_i}$ are represented by $Final_j$ and $Final_i$, respectively.

3.3. Span Prototypical Module

3.3.1. Instance Query Span Attention

Our research has shown that different support span intervals have varying impacts on a query span. Additionally, the exchange of information between the support spans reveals noticeable disparities. Regarding the i-th class that comprises m annotated spans with boosting representations $\bar{S_i} = [\bar{s}_i^1, ..., \bar{s}_i^m]$ within the support set, IQSA procures the corresponding prototypical representation $\bar{z}_i^m$ for a specified query span $\bar{q}_i^m$ as follow:

$$\begin{cases} a_m = Softmax(\bar{q}_i^m \bar{S}_i^T), \\ \bar{z}_i^m = \sum_{n=1}^{m} \alpha_i^n s_i^n \end{cases} \tag{14}$$

3.3.2. O-Type Division and Prototypical Span Attention

Representing O-Type spans using prototypical networks presents a significant challenge due to their diverse semantics and large quantities. A single prototypical vector may not be sufficient to capture all the subtleties of these spans. To overcome this issue, it is essential to consider the boundary information for each span. By dividing O-Type spans into three sub-classes based on their boundaries, we can better account for their diverse semantics. This approach provides a more comprehensive representation of the data and allows us to gain insights into the nuances of different sub-classes. Moreover, this framework can be extended to accommodate even more complex semantic structures of O-Type spans. Specifically, suppose we have a sentence with L annotated spans $[(l_1, r_1), ..., (l_n, r_n)]_{i=1}^{L}$, where l_n and r_n represent the left and right boundaries of the n-th annotated span. For each of the remaining spans (l_o, r_o), we assign it a sub-class O_{sub} as follows:

$$O_{sub} = \begin{cases} O_1, & \forall i, \text{ s.t. } r_o < l_i \vee l_o > r_i \\ O_2, & \exists i, \text{ s.t. } l_o \geq l_i \wedge r_o \leq r_i \\ O_3, & \text{Others} \end{cases} \quad (15)$$

where O_1 denotes the span that does not overlap with any entities, e.g. "won an" in support example of figure 1 and O_2 represents the span that is the sub-span of an entity, e.g. "matt" in support example of figure 1. After O partition, we get the prototypical representation of each O_{sub} , thus for a query span $\bar{q}_i^m$, we have 3 sub-class representations $\mathbf{Z_m^o} = [\mathbf{z_m^{o_1}}, \mathbf{z_m^{o_2}}, \mathbf{z_m^{o_3}}]$ for the class O. Then, we utilize Prototypical Span Attention (PSA) to achieve the final O representation as follow:

$$a_m = Softmax(\bar{q}_i^m Z_m^o) \quad (16)$$

$$\bar{z}_i^m = \sum_{n=1}^{3} \alpha_i^n Z_m^{o_n} \quad (17)$$

3.4. Span Alignment Module

The n-th query span, denoted as w_n, is processed by the previous span modules to obtain its enhanced representation $\bar{\mathbf{w}}_\mathbf{n}$ and corresponding prototypical vectors $\mathbf{A_n} = (a_n^o, a_n^1, ..., a_n^N)$. Subsequently, we predict the type of w_n in relation to the support set, represented by a_k, with a certain probability.

$$p(x_n = a_k | w_n) = \frac{exp(-L_2(\bar{\mathbf{w}}_\mathbf{n}, a_k^n))}{\sum_{k'} exp(-L_2(\bar{\mathbf{w}}_\mathbf{n}, a_{k'}^n))} \quad (18)$$

Here, L_2 refers to the Euclidean distance. Cross-entropy is used as the loss function:

$$\mathcal{L} = -\frac{1}{B_w} \sum_{n=1}^{B_w} log\ p(y_n^* | w_n) \quad (19)$$

where y_n^* is the gold label of w_n and B_w is the number of spans in the query.

3.5. Span Optimization Module

During the inference process, the span alignment module may output overlapping or conflicting spans. To address this issue, we propose an optimization module that incorporates Soft Non-Maximum Suppression (SoftNMS) (Bodla et al., 2017; Shen et al., 2021) into the beam search algorithm, which we call Adaptive Soft-Beam Non-Maximum Suppression (ASBNMS).

Algorithm 1 ASBNMS

Require: Predicted entity sets $entity\_sets$
Ensure: List of non-overlapping entities $final\_entities$

Initialize $final\_entities$ to an empty list
for $entity\_set$ in $entity\_sets$ **do**
  Sort $entity\_set$ by descending decay score
  Initialize $beam\_list$ with a new BeamNode from $entity\_set$
  Initialize all_beams to an empty set
  while there are updates to $beam\_list$ **do**
    Initialize $current\_beams$ to an empty list
    Set $updated$ to False
    for each $beam$ in $beam\_list$ **do**
      Get $valid\_tuples$ from $Expand(entity\_set)$
      if $valid\_tuples$ is empty **then**
        Add $beam$ to $current\_beams$
      else
        for each $valid$ in $valid\_tuples$ **do**
          Create a new BeamNode new_beam from $valid$
          if new_beam is not in all_beams **then**
            Add new_beam to $current\_beams$
            Insert new_beam into all_beams
          end if
        end for
      end if
    end for
    Remove duplicates from $current\_beams$
    Update $beam\_list$ with $current\_beams$ if different from the previous list
  end while
  Add the best result from $beam\_list$ to $final\_entities$
end for
return $final\_entities$

This algorithm provides a nuanced and fine-grained approach to conflict resolution, resulting in more accurate and reliable predictions. The ASBNMS algorithm expands all beam states at each step, followed by pruning of the newly generated states according to the prescribed beam capacity. For a beam state S containing non-overlapping spans $\{l_i, r_i, score_i, y_i\}_{i=1}^{L}$, we calculate the decayed score $score_i^{decay}$ for each non-overlapping span $s_i = (l_i, r_i, score_i, y_i)$.

$$\lambda_i = \sum_{t=1}^{T} w_i H(s_i, s_t) \tag{20}$$

$$H(s_i, s_t) = I(\theta_{i-1} < IoU(s_i, s_t) \leq \theta_i) \cdot f(s_i, s_t) \tag{21}$$

$$f(s_i, s_t) = \frac{\min(r_i - l_i, r_t - l_t)}{\max(r_i - l_i, r_t - l_t)} \tag{22}$$

$$score_i^{decay} = score_i * e^{-\lambda_i} \tag{23}$$

The indicator function is denoted by I. The $f(s_i, s_t)$ function is to calculate the ratio of lengths between spans. The overlap ratio of two spans, $\mathrm{IoU}(s_i, s_j)$, is calculated as the size of their intersection divided by the size of their union:

$$\mathrm{IoU}(s_i, s_j) = \frac{|\{l_i, ..., r_i\} \cap \{l_j, ..., r_j\}|}{|\{l_i, ..., r_i\} \cup \{l_j, ..., r_j\}|} \tag{24}$$

We partition the value of IoU into multiple intervals, with θ being the interval boundary value and a hyperparameter. We also assign weight w_i to each interval and finally get λ_i. Our goal in incorporating this module into the system's architecture is to enhance the model's performance and robustness.

4. Experiments

4.1. Settings

4.1.1. Datasets

We have selected two widely-used N-way K-shot benchmarks for assessing the performance of our SSF: FewNERD [1]Ding et al. (2021) and SNIPS Coucke et al. (2018). The FewNERD dataset is annotated with a hierarchy of eight coarse-grained entity types such as "Location", and 66 fine-grained entity types including "Location-GPE". The dataset consists of two tasks: Intra and Inter. In the Intra task, all entities in the train, development, and test sets belong to different coarse-grained types. On the other hand, in the Inter task, the train, development, and test sets may share coarse-grained types while maintaining mutually disjoint fine-grained entity types. The SNIPS dataset offers a diverse set of seven domains, each designed to facilitate slot-filling tasks. The sampling task of SNIPS employs a N-way K-shot approach, whereby all classes in the support set are endowed with K annotated examples. Each domain contained within SNIPS presents two distinctive few-shot slot-filling settings: the 1-shot and the 5-shot configurations.

[1] https://github.com/thunlp/Few-NERD

4.1.2. Evaluation

For evaluation on FewNERD, we employ episode evaluation as in Ding et al. (2021) and calculate micro F1 score over all test episodes. For evaluation on SNIPS, we calculate micro F1 score within each episode and then average over all episodes as in Hou et al. (2020). For all results, we report the mean and standard deviation based on 5 runs with different seeds.

4.1.3. Baselines

For FewNERD, we compare the proposed approach with ProtoBERT (Ding et al., 2021), NNShot (Yang and Katiyar, 2020), StructShot (Yang and Katiyar, 2020), CONTaiNER (Das et al., 2022), ESD (Wang et al., 2022b), DecomposedMeta (Ma et al., 2022), SpanProto (Wang et al., 2022a), MSDP (Dong et al., 2023), MeTNet (Han et al., 2023), and PromptNER (Zhang et al., 2023). For SNIPS, we compare the proposed approach with TransferBERT (Hou et al., 2020), MN+BERT (Hou et al., 2020), L-TapNet+CDT (Hou et al., 2020), Retriever (Yu et al., 2021), ConVEx (Henderson and Vulić, 2021), and Ma2021 (Ma et al., 2021a).

The baselines compared on the FewNERD dataset are all from the FewNERD leaderboard [2].

4.1.4. Implementation Details

We use `bert-base-uncased` from huggingface library as our base encoder following Ding et al. (2021). We use AdamW as our optimizer with a learning rate of 5e-4 at both the training and fine-tuning in testing time for all experiments. We set the dropout ratio to 0.1. The dimension of span representation d and the maximum span length L is set to 100 and 5. We set max_o_num to 100, which is the maximum number of O-type spans. We choose five random seeds from {6, 12, 3407, 42, 9999} and report the averaged results with standard deviations. We use grid search for hyperparameter setting, the search space is shown in Table 4. The total model has 110M parameters and trains in ≈240min on an A100 GPU.

4.2. Main Results

Table 2 and Table 3 present the main results of our proposed method compared to other baselines. Based on these results, we make the following observations: 1) SSF achieves the best performance, significantly outperforming the baselines. Compared to DecomposedMeta, the overall average results for FewNERD-INTRA and FewNERD-INTER

[2] https://paperswithcode.com/dataset/few-nerd

Models	Intra					Inter				
	$1\sim2$ shot		$5\sim10$ shot		Avg.	$1\sim2$ shot		$5\sim10$ shot		Avg.
	5 way	10 way	5 way	10 way		5 way	10 way	5 way	10 way	
ProtoBERT	20.76$_{\pm0.84}$	15.04$_{\pm0.44}$	42.54$_{\pm0.94}$	35.40$_{\pm0.13}$	28.44	38.83$_{\pm1.49}$	32.45$_{\pm0.79}$	58.79$_{\pm0.44}$	52.92$_{\pm0.37}$	45.75
NNShot	25.78$_{\pm0.91}$	18.27$_{\pm0.41}$	36.18$_{\pm0.79}$	27.38$_{\pm0.53}$	26.90	47.24$_{\pm1.00}$	38.87$_{\pm0.21}$	55.64$_{\pm0.63}$	49.57$_{\pm2.73}$	47.83
StructShot	30.21$_{\pm0.90}$	21.03$_{\pm1.13}$	38.00$_{\pm1.29}$	26.42$_{\pm0.60}$	28.92	51.88$_{\pm0.69}$	43.34$_{\pm0.10}$	57.32$_{\pm0.63}$	49.57$_{\pm3.08}$	50.53
CONTaiNER	40.40	33.82	53.71	47.51	43.86	56.1	48.36	61.90	57.13	55.87
ESD	36.08$_{\pm1.6}$	30.00$_{\pm0.70}$	52.14$_{\pm1.5}$	42.15$_{\pm2.6}$	40.09	59.29$_{\pm1.25}$	52.16$_{\pm0.79}$	69.06$_{\pm0.80}$	64.00$_{\pm0.43}$	61.13
DecomposedMeta	49.48$_{\pm0.85}$	42.84$_{\pm0.46}$	62.92$_{\pm0.57}$	57.31$_{\pm0.25}$	53.14	64.75$_{\pm0.35}$	58.65$_{\pm0.43}$	71.49$_{\pm0.47}$	68.11$_{\pm0.05}$	65.75
SpanProto	54.49$_{\pm0.39}$	45.39$_{\pm0.72}$	65.89$_{\pm0.82}$	59.37$_{\pm0.47}$	56.29	73.36$_{\pm0.18}$	66.26$_{\pm0.33}$	75.19$_{\pm0.77}$	70.39$_{\pm0.63}$	71.30
MSDP	56.35$_{\pm0.28}$	47.13$_{\pm0.69}$	66.80$_{\pm0.78}$	**64.69**$_{\pm\mathbf{0.51}}$	58.74	76.86$_{\pm0.22}$	69.78$_{\pm0.31}$	84.78$_{\pm0.69}$	81.50$_{\pm0.71}$	78.23
MeTNet	55.79$_{\pm0.23}$	47.18$_{\pm0.89}$	65.41$_{\pm0.35}$	60.71$_{\pm0.17}$	57.27	74.42$_{\pm0.61}$	67.91$_{\pm0.68}$	76.28$_{\pm0.32}$	71.96$_{\pm0.35}$	72.64
PromptNER	55.32$_{\pm1.03}$	50.29$_{\pm0.61}$	67.26$_{\pm1.02}$	60.42$_{\pm0.73}$	58.32	64.92$_{\pm0.71}$	62.28$_{\pm0.39}$	72.64$_{\pm0.16}$	70.13$_{\pm0.67}$	67.49
SSF (Ours)	**60.80**$_{\pm\mathbf{0.75}}$	**50.31**$_{\pm\mathbf{0.45}}$	**74.09**$_{\pm\mathbf{0.55}}$	61.69$_{\pm0.55}$	**61.72**	**83.21**$_{\pm\mathbf{0.80}}$	**75.87**$_{\pm\mathbf{0.50}}$	**91.24**$_{\pm\mathbf{0.30}}$	**85.95**$_{\pm\mathbf{0.50}}$	**84.06**

Table 2: F1 scores with standard deviations on FewNERD. The best results are in **boldface**.

	Models	We	Mu	Pl	Bo	Se	Re	Cr	Avg.
1-SHOT	TransferBERT	55.82$_{\pm2.75}$	38.01$_{\pm1.74}$	45.65$_{\pm2.02}$	31.63$_{\pm5.32}$	21.96$_{\pm3.98}$	41.79$_{\pm3.81}$	38.53$_{\pm7.42}$	39.06$_{\pm3.86}$
	MN+BERT	21.74$_{\pm4.60}$	10.68$_{\pm1.07}$	39.71$_{\pm1.81}$	58.15$_{\pm0.68}$	24.21$_{\pm1.20}$	32.88$_{\pm0.64}$	69.66$_{\pm1.68}$	36.72$_{\pm1.67}$
	ProtoBERT	46.72$_{\pm1.03}$	40.07$_{\pm0.48}$	50.78$_{\pm2.09}$	68.73$_{\pm1.87}$	60.81$_{\pm1.70}$	55.58$_{\pm3.56}$	67.67$_{\pm1.16}$	55.77$_{\pm1.70}$
	Ma2021	-	-	-	-	-	-	-	69.3(unk)
	L-TapNet+CDT	71.53$_{\pm4.04}$	60.56$_{\pm0.77}$	66.27$_{\pm2.71}$	**84.54**$_{\pm\mathbf{1.08}}$	76.27$_{\pm1.72}$	70.79$_{\pm1.60}$	62.89$_{\pm1.88}$	70.41$_{\pm1.97}$
	ESD	78.25$_{\pm1.50}$	54.74$_{\pm1.02}$	71.15$_{\pm1.55}$	71.45$_{\pm1.38}$	67.85$_{\pm0.75}$	71.52$_{\pm0.98}$	78.14$_{\pm1.46}$	70.44$_{\pm0.47}$
	SSF (Ours)	**85.59**$_{\pm\mathbf{1.50}}$	**69.25**$_{\pm\mathbf{0.75}}$	**83.48**$_{\pm\mathbf{0.65}}$	74.48$_{\pm2.29}$	**84.40**$_{\pm\mathbf{0.45}}$	**79.44**$_{\pm\mathbf{0.66}}$	**94.64**$_{\pm\mathbf{1.31}}$	**81.64**$_{\pm\mathbf{0.47}}$
5-SHOT	TransferBERT	59.41$_{\pm0.30}$	42.00$_{\pm2.83}$	46.07$_{\pm4.32}$	20.74$_{\pm3.36}$	28.20$_{\pm0.29}$	67.75$_{\pm1.28}$	58.61$_{\pm3.67}$	46.11$_{\pm2.29}$
	MN+BERT	36.67$_{\pm3.64}$	33.67$_{\pm6.12}$	52.60$_{\pm2.84}$	69.09$_{\pm2.36}$	38.42$_{\pm4.06}$	33.28$_{\pm2.99}$	72.10$_{\pm1.48}$	47.98$_{\pm3.36}$
	ProtoBERT	67.82$_{\pm4.11}$	55.99$_{\pm2.24}$	46.02$_{\pm3.19}$	72.17$_{\pm1.75}$	73.59$_{\pm1.60}$	60.18$_{\pm6.96}$	66.89$_{\pm2.88}$	63.24$_{\pm3.25}$
	Retriever	82.95(unk)	61.74(unk)	71.75(unk)	81.65(unk)	73.10(unk)	79.54(unk)	51.35(unk)	71.72(unk)
	ConVEx	71.5(unk)	77.6(unk)	79.0(unk)	84.5(unk)	84.0(unk)	73.8(unk)	67.4(unk)	76.8(unk)
	Ma2021	89.39(unk)	75.11(unk)	77.18(unk)	84.16(unk)	73.53(unk)	82.29(unk)	72.51(unk)	79.17(unk)
	L-TapNet+CDT	71.64$_{\pm3.62}$	67.16$_{\pm2.97}$	75.88$_{\pm1.51}$	84.38$_{\pm2.81}$	82.58$_{\pm2.12}$	70.05$_{\pm1.61}$	73.41$_{\pm2.61}$	75.01$_{\pm2.46}$
	ESD	84.50$_{\pm1.06}$	66.61$_{\pm2.00}$	79.69$_{\pm1.35}$	82.57$_{\pm1.37}$	82.22$_{\pm0.81}$	80.44$_{\pm0.80}$	81.13$_{\pm1.84}$	79.59$_{\pm0.39}$
	SSF (Ours)	**91.05**$_{\pm\mathbf{0.70}}$	**77.90**$_{\pm\mathbf{0.65}}$	**89.52**$_{\pm1.50}$	**94.87**$_{\pm\mathbf{0.57}}$	**95.13**$_{\pm\mathbf{0.20}}$	**87.99**$_{\pm\mathbf{0.48}}$	**96.54**$_{\pm\mathbf{0.30}}$	**90.35**$_{\pm\mathbf{0.39}}$

Table 3: F1 scores and standard deviations are presented for seven domains of the SNIPS dataset. The best results are highlighted in **boldface**. The term 'unk' denotes methods for which deviations have not been reported in the corresponding paper. A comparison is made between 1-shot and 5-shot settings for the baselines, as ConVEx and Retriever do not provide 1-shot results in their publications.

learning rate	[5e-5, 1e-4, 3e-4, 5e-4,1e-2]
seed	[6, 12, 3407, 42, 9999]
dropout	[0.1,0.2,0.3,0.4,0.5]
bert learning rate	[5e-6, 1e-5, 2e-5, 3e-5, 5e-5]
span dimension	[50, 100, 150, 200]
beam size	[1, 2, 3, 4, 5, 6, 7]
θ	[0,0.1,0.3,0.5,0.7,1]

Table 4: The searching scope of hyperparameters.

show an improvement of 8.51 and 18.31 in F1 score, respectively. 2) Compared to a similar span-based method, SpanProto, our approach demonstrates better results in both FewNERD-INTRA and FewNERD-INTER, leading by 5.06 and 12.76 in average F1 score, respectively. This highlights the superior performance of our method compared to similar span-based methods. 3) In the 5-way 5$\sim$10 shot setting on FewNERD-INTER, SSF achieves an F1 score of 91.24, significantly outperforming other methods. 4) For the SNIPS dataset, our method achieves an average F1 score of 81.64 in the 1-shot setting and 90.35 in the 5-shot setting, significantly outperforming previous state-of-the-art methods.

4.3. Ablation Study

In order to validate the contributions of different components in the proposed approach, we perform ablation studies by removing each component of the SSF individually: 1) *Ours w/o Localised Span Attention*, where we remove the localised span attention component. As a result, the span cannot be aware of other spans within the same sentence. 2) *Ours w/o Trans Span Attention*, where we remove the trans span attention component. As a result, the span cannot be aware of other spans within the other sentence. 3) *Ours w/o Instance Query Span Attention*, where we attain the prototypical representation for each class through averaging. 4) *Ours w/o O-type Division and Prototypical Span*

Ablation Models	F1
SSF	$91.24_{\pm 0.30}$
Ours *w/o* Localized Span Attention	$83.10_{\pm 0.4}$
Ours *w/o* Trans-Span Attention	$80.6_{\pm 1.5}$
Ours *w/o* Instance Query Span Attention	$84.2_{\pm 0.6}$
Ours *w/o* O-type Division	$81.7_{\pm 1.3}$
Ours *w/o* ASBNMS	$85.3_{\pm 1.4}$

Table 5: The effect of our proposed mechanisms on the validation set of FewNERD (inter, 5 way 5∼10 shot). We report the average result of 5 different runs with standard deviations. *w/o* denotes *without*.

Attention, where we directly attain the prototypical representation of the O-class, excluding the sub-classifications of the O-type spans. 5) *Ours w/o ASBNMS*, where we remove the ASBNMS algorithm.

As shown in Table 5, the results demonstrate that: 1) The superiority of the TSA module over its LSA counterpart is evident, due to its ability to augment span representation beyond the constraints of isolated sentences. Unlike LSA, which can only enhance span representation within the boundaries of a single sentence, TSA is able to enhance it across multiple sentences, thereby providing greater access to and integration of diverse information. 2) Upon removing IQSA to attain the prototypical representation of a class through averaging, the average F1 score experiences a 7-point decrease. Failure to account for the sub-classes of O-type spans leads to an additional 9.54 decrease in average F1 score. This exemplifies the necessity of such constituents, which are able to enhance performance. 3) The final outcome obtained in the absence of ASBNMS underscores the importance of our post-processing algorithm in this span-level few-shot NER framework.

4.4. Error Analysis

Following the error analysis methodology of Wang et al. (2022b), we undertook an error analysis on two distinct categories, namely false positive with incorrect span boundaries (FP-Span) and false positive with correct span boundaries but incorrect types (FP-Type). The former refers to extracted entities with incorrect span boundaries, while the latter describes entities that possess accurate span boundaries but are classified with erroneous entity types. As demonstrated in Table 6, our SSF model outperforms other strong baselines and has significantly fewer false positive predictions. Interestingly, the FP-Type errors remain notably lower than those of rival models, thereby substantiating the efficacy of our framework. Remarkably, the span-level prototype networks exhibit exceptional performance in classifying entity types.

Methods	Total	F1	FP-Span	FP-Type
ProtoBERT	30.4k	44.44	86.70%	13.30%
NNShot	21.7k	54.29	84.70%	15.30%
StructShot	14.5k	57.33	80.00%	20.00%
ESD	9.4k	66.46	72.80%	27.20%
Ours	**6.5k**	**83.21**	89.90%	10.10%

Table 6: Error analysis of 5-way 1∼2 shot on FewNERD-INTER. 'Total' denotes the total wrong prediction of two types.

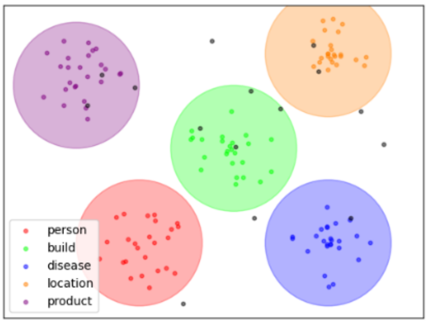

Figure 2: The t-SNE visualization displays span representations using 5-way, 5∼10 shot episode data from FewNERD-INTER. The points are color-coded to denote entity spans with different types, while the circle represents the prototype region. Instances of false positives are indicated by black markings.

4.5. Visualization

We visualize the representations learned by SSF in the 5-way 5∼10 shot setting on FewNERD-INTER. We created a visualization using the t-SNE algorithm (Van der Maaten and Hinton, 2008), which is shown in figure 2. Upon examination, it becomes apparent that the SSF method effectively clusters span representations of the same entity class while dispersing those belonging to different classes. Thus, compared to other baseline models, the proposed SSF method is better able to assign an appropriate entity classification to a query span. It does this by measuring the similarities between the span representation and the prototype of each entity class.

5. Related Work

Few-Shot Learning and Meta-Learning Recently, the field of Natural Language Processing has shown a growing interest in few-shot Learning

(Han et al., 2018; Geng et al., 2019; Chen et al., 2019; Brown et al., 2020; Schick and Schütze, 2021; Gao et al., 2021). Few-shot learning is a complicated problem that seeks to develop models that can quickly adapt to different tasks with minimal labelled data. The basic concept behind meta-learning is to facilitate the efficient acquisition of novel skills by the model. Some common meta-learning algorithms for few-shot learning include optimisation-based learning (Kulkarni et al., 2016), metric-based learning (Snell et al., 2017), and augmentation-based learning (Wei and Zou, 2019), among others.

Few-Shot NER Few-shot NER aims to identify and classify entity types based on low-resource data. Existing few-shot NER methods can be roughly categorized into two types: prompt-based and metric-based meta-learning, which approach either token-level (Fritzler et al., 2019; Hou et al., 2020; Yang and Katiyar, 2020; Tong et al., 2021) or span-level classification (Yu et al., 2021; Wang et al., 2022b). The first type mainly focuses on exploring the general pre-trained language model knowledge for NER via prompt learning (Cui et al., 2021; Ma et al., 2021b; Zhang et al., 2022; Chen et al., 2022; Cui et al., 2022). Cui et al. (2021) proposed template-based BART, which treated original sentences as the source sequence, and statement templates filled by candidate spans as the target sequence. By introducing templates, this method outperforms traditional sequence labeling in few-shot scenarios, but it would be time-consuming to enumerate and classify all candidate spans. LightNER (Chen et al., 2022) integrates continuous prompts into the self-attention matrix and develops a semantically informed answer space, replacing label-specific layers. The subsequent category aims to acquire a feature space with strong generalisability in the source domain before classifying test samples using nearest class prototypes (Snell et al., 2017; Fritzler et al., 2019; Ji et al., 2022) or neighbour samples (Das et al., 2022; Yang and Katiyar, 2020). It is noteworthy that currently, state-of-the-art few-shot named entity recognition methods rely on prototypical networks.

6. Conclusion

This study presents a streamlined span factorization approach for few-shot NER. The proposed technique, SSF, treats few-shot NER as a span-level alignment problem and decomposes it into four modules designed to improve the accuracy of the span alignment. The study achieves significant improvements over previous state-of-the-art results.

Limitations

Our SSF model can only be applied to few-shot NER tasks. In the future, we plan to extend it to other NER scenarios, such as few-shot cross-lingual NER.

Ethical Considerations

Our contribution to this work is purely methodological. Specifically, we have devised a span-based prototypical network to augment the performance of few-shot NER. Thus, our contribution does not entail any direct negative social repercussions.

Acknowledgements

This work was supported by the Hunan Provincial Science and Technology Department (NO.2023GK2057).

7. Bibliographical References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. 2017. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-*

IJCNLP), pages 4217–4226, Hong Kong, China. Association for Computational Linguistics.

Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. 2022. LightNER: A lightweight tuning paradigm for low-resource NER via pluggable prompting. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2374–2387, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

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

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. *arXiv preprint arXiv:2203.09770*.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2022. Container: Few-shot named entity recognition via contrastive learning. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

Guanting Dong, Zechen Wang, Jinxu Zhao, Gang Zhao, Daichi Guo, Dayuan Fu, Tingfeng Hui, Chen Zeng, Keqing He, Xuefeng Li, Liwen Wang, Xinyue Cui, and Weiran Xu. 2023. A multi-task semantic decomposition framework with task-specific pre-training for few-shot ner.

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.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Chengcheng Han, Renyu Zhu, Jun Kuang, FengJiao Chen, Xiang Li, Ming Gao, Xuezhi Cao, and Wei Wu. 2023. Meta-learning triplet network with adaptive margins for few-shot named entity recognition. *arXiv preprint arXiv:2302.07739*.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Matthew Henderson and Ivan Vulić. 2021. ConVEx: Data-efficient and few-shot slot labeling. In *Proceedings of the 2021 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3375–3389, Online. Association for Computational Linguistics.

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

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

Vivek Kulkarni, Yashar Mehdad, and Troy Chevalier. 2016. Domain adaptation for named entity recognition in online media with word embeddings. *CoRR*, abs/1612.00148.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*, pages 260–270.

Jianqiang Ma, Zeyu Yan, Chang Li, and Yang Zhang. 2021a. Frustratingly simple few-shot slot tagging. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1028–1033, Online. Association for Computational Linguistics.

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

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

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Meihan Tong, Shuai Wang, Bin Xu, Yixin Cao, Minghui Liu, Lei Hou, and Juanzi Li. 2021. Learning from miscellaneous other-class words for few-shot named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6236–6247, Online. Association for Computational Linguistics.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022a. Spanproto: A two-stage span-based prototypical network for few-shot named entity recognition. *CoRR*, abs/2210.09049.

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

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, pages 6381–6387.

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

Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. Few-shot intent classification and slot filling with retrieved examples. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 734–749, Online. Association for Computational Linguistics.

Mozhi Zhang, Hang Yan, Yaqian Zhou, and Xipeng Qiu. 2023. Promptner: A prompting method for few-shot named entity recognition via k nearest neighbor search. *arXiv preprint arXiv:2305.12217*.

Xinghua Zhang, Bowen Yu, Yubin Wang, Tingwen Liu, Taoyu Su, and Hongbo Xu. 2022. Exploring modular task decomposition in cross-domain named entity recognition. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 301–311.

Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. 2020. Example-based named entity recognition. *CoRR*, abs/2008.10570.

8. Language Resource References

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

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.