KCL: Few-shot Named Entity Recognition with Knowledge Graph and Contrastive Learning

Shan Zhang, Bin Cao*, Jing Fan*
Zhejiang University of Technology
Hangzhou, China
{zhangshan,bincao,fanjing}@zjut.edu.cn

Abstract

Named Entity Recognition(NER), as a crucial subtask in natural language processing(NLP), is limited to a few labeled samples(a.k.a. few-shot). Metric-based meta-learning methods aim to learn the semantic space and assign the entity to its nearest label based on the similarity of their representations. However, these methods have trouble with semantic space learning and result in suboptimal performance. Specifically, the label name or its description is widely used for label semantic representation learning, but the label information extracted from the existing label description is limited. In addition, these methods focus on reducing the distance between the entity and the corresponding label, which may also reduce the distance between the labels and thus cause misclassification. In this paper, we propose a few-shot NER method that harnesses the power of **K**nowledge Graph and **C**ontrastive **L**earning to improve the prototypical semantic space learning. First, KCL leverages knowledge graphs to provide rich and structured label information for label semantic representation learning. Then, KCL introduces the idea of contrastive learning to learn the label semantic representation. The label semantic representation is used to help distance the label clusters in the prototypical semantic space to reduce misclassification. Extensive experiments show that KCL achieves significant improvement over the state-of-the-art methods.

Keywords: named entity recognition, few-shot learning, knowledge graph, contrastive learning

1. Introduction

Named Entity Recognition (NER)(Mikheev et al., 1999) aims to extract entity mentions from the text and classify them into pre-defined entity categories such as person and location. As a key sub-task of natural language processing (NLP), NER extracts key information for downstream tasks like machine translation(Wang et al., 2017) and text categorization(Rigutini et al., 2005), and helps the models to understand semantic information deeply. Conventional NER methods require a mount of annotated data to ensure the prediction performance(Li et al., 2020b). Actually, NER always suffers from annotated data scarcity because labeling is time-consuming and labor-intensive. Therefore, performing NER based on very limited annotated data(a.k.a. few-shot ENR) has recently garnered a lot of attention(Huang et al., 2021; Chen et al., 2022b; Zhang et al., 2021).

Metric-based meta-learning methods such as prototypical networks are widely used for few-shot NER tasks(Vinyals et al., 2016; Snell et al., 2017; Fritzler et al., 2019). The basic idea of these methods is to learn a common semantic space from rich source domains and use the semantic space to perform NER tasks in target domains by a few samples. In the semantic space, the entities belonging to the same label cluster around a single prototype. As the label representation, the prototype is the average of the entity representations. Then, in the inference phase, the entity can be assigned to the closest label based on the similarity of the entity and label representations. Due to that the label representation learning also benefits from the label's semantic information itself(Zhou et al., 2018; Athiwaratkun et al., 2020; Wang et al., 2021), some studies(Ma et al., 2022a; Hou et al., 2020; Luo et al., 2021) use the label name as the extra information for label representation learning. Specifically, the label name is used to obtain the label semantic representation, and the weighted sum of the label semantic representation and prototype representation is used as the label representation. Nevertheless, these studies still face the following two challenges:

(1) Limited label semantic representation learning. The semantic information contained in the label name is limited, which further affects label semantic representation learning. To this end, some few-shot NER methods based on data augmentation(Zhou et al., 2018) and pre-trained(Athiwaratkun et al., 2020; Wang et al., 2021) use the label description instead of the label name, since the description contains more semantic information about the label. For example, the description of the label "Person" is "A person is a **being** who has certain capacities or attributes such as **reason**, **morality**...". The words in bold are semantic information about "Person" and are called label properties in this paper. However, only a few properties are contained in the description, which

*Corresponding author

LREC-COLING 2024, pages 9681–9692
20-25 May, 2024.

means that the label semantic information is still limited.

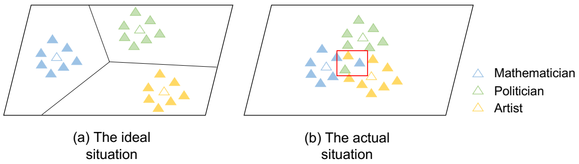


Figure 1: An example of label clusters instance in the semantic space.

(2) Misclassification caused by label-label similarity. Prototypical networks aim to make entities of the same label close together in the semantic space. This poses a potential challenge in which the distance between label clusters may be shortened, leading to misclassification. Specifically, Figure 1(a) shows the ideal situation of label clusters distributed in the semantic space, where the triangle represents the class 'Person'. Blue, green, and yellow triangles represent the Person's sub-classes 'Mathematician', 'Politician', and 'Artist' respectively. There should be a large distance between these sub-classes to avoid misclassification. Actually, as shown in Figure 1(b), because the entities (within the red box) of the three sub-classes have the common features(a.k.a. semantic overlap) about 'Person', these entities tend to get close which also pulls into the distance of the sub-class clusters. In this situation, the metric-based methods will result in misclassification.

To remedy this, Hou et al. (2020) proposes L-TapNet that uses the linear matrix to map different domains to different M spaces to avoid label interference between domains. Then L-TapNet uses reference vectors in the M space to further distance label prototypes. However, the simple linear mapping just scales and biases the distribution of the label clusters, which can not effectively change the distribution and overcome the complicated semantic overlap problem.

In this paper, to deal with these challenges, we propose KCL, a few-shot NER method that harnesses the power of **K**nowledge Graph(KG) and **C**ontrastive **L**earning. Firstly, we introduce knowledge graphs into the few-shot NER task to improve the label semantic representation learning. Knowledge graph records labels and their corresponding properties, which can provide more richer and structured semantic information than the label description. Specifically, we use the attention mechanism(Vaswani et al., 2017) to compute the label semantic representation. The goal is to make label semantic representation learning 'pay more attention' to the effective semantic information of the label name and properties. Then, the label semantic representation is obtained by fusing the name and property representations.

To effectively utilize the label semantic information to help reduce misclassification caused by label-label similarity, we introduce the idea of contrastive learning(Chen et al., 2020). Contrastive learning aims to bring positive sample pairs closer and negative sample pairs farther away in the semantic space. We use this idea to learn the label semantic representation. Then the label semantic representation is used to make the label clusters evenly distributed instead of occupying a narrow cone in the prototypical semantic space(Gao et al., 2021). Specifically, we first assign the learnable anchor-point vectors for each label semantic representation. Then, the anchor-point vector and the corresponding label semantic representations are regarded as the positive sample pair while the vector and the rest of the label semantic representations are negative sample pairs. After that, the similarity function is learned to make the label semantic representations close to its corresponding anchor-point vector and relatively far from other label semantic representations. Finally, we use the linear mapping proposed by L-TapNet to map the anchor-point vectors, label semantic representations, and samples of each domain to M space. In M space, the label cluster is close to the corresponding label semantic representation and the anchor-point vector, which achieves the purpose of extending the distance between label clusters.

Our contributions are summarized as follows:

1. We propose to apply knowledge graphs to provide the external semantic information for label semantic representation learning in few-shot NER tasks.
2. We introduce the idea of contrastive learning to learn the strong distribution of prototype clusters in the prototypical semantic space to reduce misclassification.
3. Extensive experimental evaluation demonstrates the outstanding performance of KCL for few-shot NER.

2. Task Formulation

In few-shot learning, given a sentence $x = (x_1, x_2, ..., x_L)$, NER models aim to output the label sequence $y = (y_1, y_2, ..., y_L)$ for x by just a few training samples. Each domain is a NER task and is a set of (x,y) pairs. In the training phase, NER models are trained on source domains T = {T_1, T_2, ..., T_m}, where these source domains have rich annotated sample pairs. In the inference phase, NER models are evaluated on unseen target domains T' = {T'_1, T'_2, ..., T'_n} by fine-tuning on the support set S. The support set follows the "N-way K-shot" principle that each label of N labels includes K

samples, where K is small such as 1 or 5.

Formally, the task of few-shot NER is defined as follows: given a query sentence x and a few-shot support set S, find the best label sequence y^* of x:

$$y^* = \arg\max_y P(y|x, S) \quad (1)$$

3. Methodology

In this section, we describe the proposed few-shot NER method KCL in detail. First, to improve the label semantic representation learning, KCL applies the knowledge graph to provide the extra label information. Second, in order to effectively utilize the label semantic information to reduce misclassification caused by label-label similarity, KCL introduces the idea of contrastive learning to learn the label semantic representations and help label clusters evenly distributed in the prototypical semantic space.

The overview framework of KCL is shown in Figure 2. Specifically, KCL first obtains the label name embedding and the corresponding property (provided by KG) embeddings. Then KCL performs the attention mechanism for the label name and property embeddings to get the label semantic embedding. In addition, KCL assigns the anchor-point vectors for each label. Contrastive learning is used to make the label semantic representation and the corresponding anchor-point vector close in the prototypical space. After that, KCL maps the label semantic embeddings, anchor-point vectors, and token embeddings of the same domain in M space. Then, the weighted sum of label semantic embedding, anchor-point vector, and prototype embedding is used as the label embedding. Finally, KCL assigns labels to each token by calculating the similarity between the token embeddings and the label embeddings. The specific details of knowledge graph application (3.1) and contrastive learning for prototypical space learning(3.2) are as follows.

3.1. Knowledge graph for label semantic representation learning

Knowledge graphs record the label properties and the property descriptions, which can be used to provide rich label information for few-shot NER. The detailed description of the KG application is composed of the following two steps:

(1) Knowledge extraction. We use Google Knowledge Graph[1] to obtain the label information. The knowledge graph contains millions of entries describing people, places, things, etc. in the real world. These entries make up the nodes of the graph. Therefore, we can view labels as nodes and query their related information in the graph. Google Knowledge Graph Search API provides the labels search link called Schema.org[2]. If we input the queried label, Schema.org will return its related items. As shown in Figure 3, each item mainly includes the following information: (a) **Property.** A property of the queried label. (b) **Expected Type.** The type of the property. (c) **Description.** The description of the property. In this paper, we use **Property** and **Description** as the extra label information other than the label name. Specifically, we first obtain the search results for labels in HTML format from Schema.org. Then we extract each item including the information of **Property** and **Description** from the HTML files. Finally, we concatenate the property and property description into a sequence as a piece of property information from one item.

It is worth noting that not all labels can be searched for the corresponding properties and descriptions. In the case where the label has no properties and property descriptions, we divide them into the following three situations: (a) The queried label has synonyms. In this case, we take the properties of the label's synonym as that of the label. Intuitively, the properties of similar words should also be similar. For example, the label 'Location' is similar to the label 'Place', the property 'address' of 'Place' can also be that of 'Location'. (b) The queried label has no synonyms but has its own descriptions. For example, the search result for the label 'Abstract' is its description, that is 'An abstract is a short description that summarizes a CreativeWork.'. Therefore, we use the label description as the label information. (c) The queried label has no synonyms and its own descriptions. In this case, we give up considering the property information.

(2) Knowledge application. In this phase, we use the property information obtained from the knowledge extraction phase for label semantic representation learning. It is worth noting that, for each label, due to limitations of storage and computation, we use downsampling to randomly select k property information from the label's all property information extracted from the knowledge graph. Therefore, we can obtain k pieces of property information P_i = $\{p_i^1$, p_i^2, ..., $p_i^k\}$ as the label's property information. Furthermore, for each label semantic representation learning, we use the property information of all labels instead of only considering its own property information. The motivation is that there can be common properties between labels, for example, 'address' is a property of the label 'Person' and 'Organization' at the same time. Therefore, using all property information can provide rich label

[1] The Google Knowledge Graph is from source: https://developers.google.com/knowledge-graph?hl=zh-cn

[2] The entities search link: https://schema.org/docs/gs.html

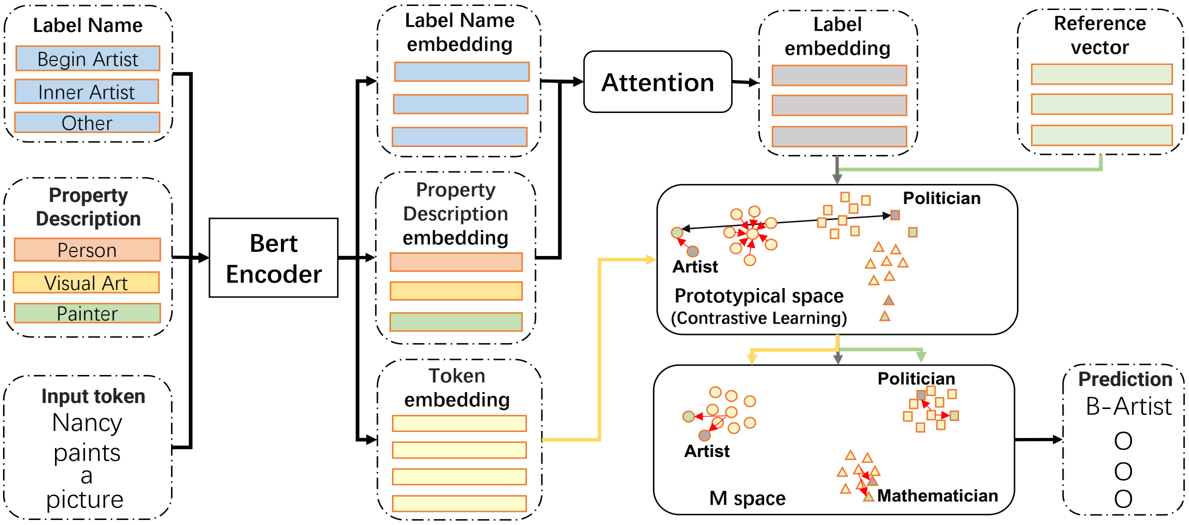
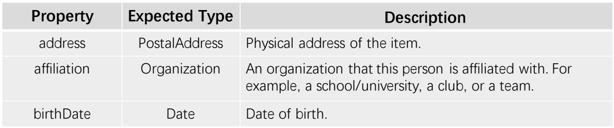


Figure 2: The overview framework of proposed KCL.



Property	Expected Type	Description
address	PostalAddress	Physical address of the item.
affiliation	Organization	An organization that this person is affiliated with. For example, a school/university, a club, or a team.
birthDate	Date	Date of birth.

Figure 3: The search result of label 'Person' from Sechma.org.

information. In addition, we also considered the label name n_i for the label semantic representation learning. In conclusion, the label semantic information $K_i = \{n_i, P_1, P_2, ..., P_N\}$ of label l_i consists of one label name n_i and the Nk property information, where $K_i \in R^{(Nk+1)\times d}$ (d is the embedding dimension) and N is the number of labels. Finally, the label semantic information K_i is used for the label semantic representation learning of label l_i.

In the label semantic representation learning process, we use the embedding of p_i as the property embedding. To make the label semantic representation learning more "focus" on the properties related to itself among all the properties, we use the attention mechanism. In the conventional NER scenario, the attention mechanism aims to learn the importance of different tokens in the sequence. The attention mechanism maps the tokens into three vectors(**query**, **key**, and **value**) and outputs the token embedding. The token embedding is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

In this paper, we use the attention mechanism to learn the importance of different properties and the label name for the label. Therefore, for each label, label name n_i is **query** and **key** is K_i. Specifically, we compute the weight value $w_i \in R^{Nk+1}$ of K_i for the label name n_i:

$$w_i = softmax(n_i^T K_i) \tag{2}$$

Then, the weight w_i is assigned to **value** K_i and the weighted sum of values of **value** is as the label semantic representation s_i:

$$s_i = Attention(w_i, K_i) \tag{3}$$

Finally, we obtain all label semantic representations $S = \{s_1, s_2, ..., s_N\}$ for each domain.

3.2. Contrastive learning for prototypical space learning

In the prototypical semantic space, the distance of the prototype clusters corresponding to labels is close caused by the semantic overlap problem, which results in misclassification. In addition, the learned embeddings in the space tend to occupy a narrow cone thus aggravating the shortening of distance. To this end, KCL uses contrastive learning to effectively learn the label semantic representation, which further helps to change the distribution of the label clusters in the space.

According to the optimization objective of the contrastive learning that the samples in the positive sample pair are close to each other, and that in the negative sample pair are far away, we construct positive and negative sample pairs for label semantic representations S. Due to that the label names and property information are extracted from the text, and the label semantic representation is learned based on these semantic information. The semantic overlap problem still exists and affects representation learning. To this end, KCL assigns the trainable anchor-point vectors $V = [v_1, v_2, ..., v_N]$ to each s_i, because the anchor-point vector is semantic-agnostic which can reduce the interference of the semantic overlap. Specifically, we first assign an initialization vector to each label semantic representation as an anchor-point vector. Then the anchor-point v_i, and the corresponding label semantic representation s_i form positive sample pair(v_i, s_i) while v_i and the other label semantic representations form negative sample pairs(v_i, s_j), respectively. Formally, given an anchor-point v_i, the label semantic representation s_i, and the remaining $N-1$ label semantic representations, the loss function for an anchor-point is as follows:

$$loss_{CL_i} = -log\frac{exp(score(v_i, s_i))}{\sum_j exp(score(v_i, s_j))} \tag{4}$$

where $score$ is the dot product. Then the $loss_{CL}$ is minimized on entire prototype clusters:

$$loss_{CL} = \sum_{i}^{N} loss_{CL_i} \tag{5}$$

Based on the semantic space learned by the contrastive learning, we use the projector M in Hou et al. (2020) to map the samples of different domains, the corresponding label semantic representations S, and anchor-point vectors V to different M spaces. The motivation is to avoid the interference of labels between domains. In M space, for each label, its label semantic representation s_i, label prototype c_i, and anchor-point vector v_i align closely while the distance between these of different labels is further separated. Then, KCL fuses s_i, c_i, and v_i to obtain the label representation l_i:

$$l_i = (1-\alpha)\cdot c_i + \alpha \cdot ((1-\beta)\cdot v_i + \beta \cdot s_i) \tag{6}$$

where the α and β are weight factors to evaluate the importance of s_i, c_i and v_i. Finally, KCL assigns labels to each token by calculating the similarity between the token representations and the label representations.

In addition, because the few-shot NER also benefits from considering the dependencies between labels, we use the CRF framework to learn the label dependencies. CRF framework includes two modules: (1)Emission module, which is used to consider the correlations of entity-label. and computes the similarity(called emission score F_{ems}) between the entity and the label. (2)Transition module, which takes into account the connection of label-label and constructs the label transition matrix to record the transition score F_{tra} that is the probability of the current label given the previous label. In this paper, we use CDT proposed by Hou et al. (2020) to obtain the transition matrix for the few-shot NER, and the loss function of CRF is as follows:

$$loss_{CRF} = -log\frac{exp(F_{ems}+F_{tra})}{\sum_{y'\in Y} exp(F_{ems}+F_{tra})} \tag{7}$$

Finally, the loss $loss_{KCL}$ is consisted of CRF and contrastive learning:

$$loss_{KCL} = (1-\lambda)\cdot loss_{CRF} + \lambda \cdot loss_{CL} \tag{8}$$

where λ is the adjustable parameter that balances weights of the two losses and $0 \leq \lambda \leq 1$.

4. Experiment

This section experimentally evaluates the generalization ability of KCL on multiple few-shot NER datasets. First, we introduce the experiment setting. Second, we compare KCL against existing methods on 1-shot/5-shot NER scenarios to demonstrate the effectiveness of KCL. In addition, we design ablation experiments to illustrate the necessity both of KG application and the use of contrastive learning. Finally, we give the parameter analysis experiments, including regulating the number k of label property information extracted from KG and the weight factor λ used to balance the weights of $loss_{CRF}$ and $loss_{CL}$.

4.1. Settings

Dataset We use 4 different datasets from different domains as the benchmark datasets: 1) CoNLL-2003 (Sang and De Meulder, 2003); 2) GUM (Zeldes, 2017); 3) WNUT-2017 (Derczynski et al., 2017); 4) and Ontonotes (Pradhan et al., 2013). To simulate the few-shot situation, we adapt the sample collection method proposed by Hou et al. (2020) to construct 1-shot/5-shot datasets from the above original datasets. It is worth noting that, due to device limitations and time costs, we randomly sample a part of each original dataset. Therefore, the number of the few-shot samples is smaller than that in Hou et al. (2020). Tabel 1 shows the detailed statistics of the original datasets used to construct few-shot experiment data.

Table 1: Statistic of Dataset

Dataset	Domain	# Labels	# Sent
CoNLL	News	5	4703
GUM	Wiki	12	5300
WNUT	Social	7	10433
OntoNotes	Mixed	19	3438

Evaluation We conduct main experiments on 1-shot and 5-shot scenarios following previous work (Hou et al., 2020; Wang et al., 2021). We use four-fold cross-validation to test the generalization ability of our method on the four benchmark datasets. For each fold, we use one dataset as the test set, one as the validation set, and the remaining two datasets as the training set. We randomly generate four-fold experimental data so that the same dataset will not be repeated as a test set or a validation set. F1-score is the evaluation metric and the results are averaged over 5 runs with different random seeds $\pm$ one standard deviation.

Comparison Methods We divide comparison methods into two categories, namely the few-shot NER method and the traditional NER method.

The few-shot NER methods consist of two types: (1) Methods without label dependency and considering label dependence. Specifically, Four methods without label dependency include ***Matching NetworkMNet***(Vinyals et al., 2016) classifies each entity according to its similarity with the samples of each class; ***WarmProtoZero***(***WPZ***)(Fritzler et al., 2019) adopts a similar strategy as MN, except replacing the matching network with the prototypical network; ***TapNet***(Yoon et al., 2019) constructs different mapping spaces for different tasks instead of using the common prototypical semantic space. In these mapping spaces, entities corresponding to the same label are close to each other, and the distances between labels are stretched; ***L-TapNet***(Hou et al., 2020) uses TapNet to learn the label representation and also considers label information including name and description. (2) Methods considering label dependence. To consider label dependency, we use the CRF framework where the above four methods (MNet, WPZ, TapNet, and L-TapNet) are emission modules, respectively. In the transition module, the state-of-the-art method ***CDT***(Hou et al., 2020) which constructs an abstract label transition score is used for label dependency transfer in the specific few-shot task. In addition, ***SpanNER***(Ma et al., 2022c), as a span-level NER method, which divides the NER task into two subtasks: span detection and entity typing, to bypass the token-wise label dependency.

The traditional NER method. We use ***LSTM+CRF*** method as the traditional NER method. The LSTM is used for effectively learning the contextual information in text and CRF is used for considering label dependence.

Hyperparameters We use the uncased BERT-Base (Kenton and Toutanova, 2019) to calculate contextual embeddings for all baseline models and our model KCL. We use Adam optimizer (Kingma and Ba, 2015) to train the models with batch size 4 and the learning rate is selected from {1e-4, 1e-5, 1e-6}. For CDT which is used for transition score, we set the learning rate η for the meta-parameter update, which is taken from {1e-2, 1e-3, 1e-4}. The hyper-parameter value selections of number k of label property and λ in loss function are described in section 4.4. We run all experiments on NVIDIA RTX 3090 GPU.

4.2. Main Results

Table 2 shows the 1-shot NER results of KCL and all baselines. Each row represents the F1-scores of test domains using the corresponding method.

Result of 1-shot setting For the results in the 1-shot scenario, we conclude four main observations as follows.

(1) KCL achieves better performance than methods considering label dependency that MNet, WPZ, TapNet and L-TapNet are emission modules and CDT is the transition module. Compared to these methods, KCL improves at least 5, 6, 6, and 3 F1-score points respectively on average. These results show the strong prediction ability of KCL. KCL also improves 7 F1-score points on average compared to SpanNER which considers the span-level label dependency. This illustrates the effectiveness of KCL in considering the token-level label dependency. In addition, the F1-score of LSTM-CRF on average is 1.78, which is at least 23 percentage points lower than that of KCL. This indicates that, for the traditional NER model, a few training samples will cause serious overfitting.

(2) Compared to the few-shot NER method that considers label information, KCL demonstrates better predictive performance. L-TapNet uses the label name and description as the label information for label representation learning. Compared to L-TapNet and L-TapNet+CDT, KCL improves by 10.4 and 3.86 F1-score points on average. This indicates that, in comparison with using label names and descriptions as label information, KCL utilizes the knowledge graph to provide richer label information and effectively to learn label representations using the contrastive learning idea.

(3) KCL maintains excellent generalization ability in all domains. Specifically, compared to all baselines on the four benchmark datasets, KCL improves by 1.26, 2.77, 3.34, and 0.41 F1-score points respectively on average. This demonstrates that KCL has superior improvement on few-shot NER.

(4) The methods(MNet, WPZ, L-TapNet, and TapNet) equipped with CDT have better generalization ability than those without label dependency. These results illustrate the significance of considering the label dependency for few-shot NER. In addition, our KCL performs significantly better than baseline methods without label dependency. These results also illustrate the effectiveness of our work.

Result of 5-shot setting As shown in Table 3, the result of the 5-shot NER shows that KCL also achieves the best performance. Compared to the baselines that ignore or weaken the label dependency, KCL improves at least 6 F1-score points on average. In the case of considering label dependency, KCL also achieves the best F1-score on average compared to all baselines. The results are

K-shot	Model	Wiki	SocialMedia	OntoNotes	News	Ave.
1-shot	LSTM+CRF	1.26±0.10	1.34±0.23	0.45±0.07	4.08±0.34	1.78±0.35
	SpanNER	5.45±0.14	19.57±1.05	9.33±1.59	39.18±1.36	18.38±0.82
	MNet	2.96±0.07	20.59±0.64	6.42±0.82	39.44±0.95	17.35±0.32
	MNet+CDT	3.45±0.38	24.55±2.09	9.38±2.29	42.87±1.29	20.06±1.36
	WPZ	2.91±0.24	20.50±0.82	6.23±0.43	34.06±1.05	15.93±0.26
	WPZ+CDT	4.11±0.64	24.01±1.47	8.78±1.75	42.11±1.90	19.76±1.34
	TapNet	3.30±0.47	19.00±0.79	8.32±0.54	28.66±3.46	14.82±1.06
	TapNet+CDT	3.40±0.44	22.50±2.15	11.51±1.66	41.60±1.11	19.75±1.02
	L-TapNet	3.27±0.54	19.63±0.71	8.30±0.69	30.22±1.47	15.36±0.27
	L-TapNet+CDT	7.17±0.94	27.66±1.03	17.58±1.36	35.20±2.04	21.90±0.45
	KG-only (ours)	7.47±0.50	30.14±2.44	19.65±1.33	38.90±4.64	24.04±1.39
	KCL (ours)	**8.43±0.37**	**30.43±2.12**	**20.92±0.93**	**43.28±3.09**	**25.76±0.89**

Table 2: F1-scores with standard deviations on the four benchmark datasets in 1-shot

consistent with the 1-shot setting, which demonstrates the generalization ability of KCL in more shots situations.

It is worth noting that, the F1-score of MNet+CDT is higher than that of KCL. The reason is as follows. When dataset News is the test set, datasets Wiki and OntoNotes are training sets. The above training sets have many samples corresponding to labels in News, so when performing tests on News, a large amount of common knowledge is transferred from the training sets to News. MNet maps all domains into the same semantic space, which can make good use of the transferred common knowledge. KCL uses the TapNet that maps different domains to different spaces, which may hinder the utilization of transferred common knowledge. In the 1-shot setting, F1-score of KCL is higher than that of MNet. The reason is that each label has only one sample, which means that little transferred common knowledge available for MNet to utilize. However, KCL uses knowledge graphs to provide additional semantic information and uses contrastive learning to effectively learn the semantic information, which improves the generalization performance of the model.

4.3. Ablation Study

In this section, we conduct the ablation analysis to indicate the necessity of KG application and the use of contrastive learning in our method (KCL).

Effectiveness of KG application To prove the effectiveness of applying knowledge graphs to few-shot NER, we perform the ablation experiments that only consider label name and label properties extracted from KG to obtain the label semantic representation(KG-only) and ignore the contrastive learning. Specifically, we directly map the label semantic representations to M space without using contrastive learning. In the M space, we use the idea of L-TapNet to assign the reference vectors to labels, and then the weighted sum of the label semantic representation, the reference vector, and the prototype representation is used as the label representation. Then we use CDT as the transition module.

Table 2 and Table 3 show the results of KG-only on all domains in 1-shot/5-shot settings. We can see KG-only is more advantageous than baselines. Specifically, in the 1-shot scenario, KG-only achieves the best performance on the above datasets. In the 5-shot setting, KG-only also has the best generalization ability on Wiki, SocialMedia, and OntoNotes. This shows that it's important to provide the extra semantic information in few-shot NER. In addition, the F1-scores of KG-only are lower than that of MNet in 1-shot/5-shot scenarios, the reason is concluded in **Result of 5-shot setting** of section 4.2.

Necessity of using contrastive learning To verify that using contrastive learning is essential for KCL to learn the semantic space, we compare KG-only to KCL which considers contrastive learning. As shown in Tables 2 and 3, we can see that KCL achieves higher F1-scores than KG-only on all datasets in 1-shot and 5-shot situations. This indicates the necessity of using the idea of contrastive learning to further learn the prototypical semantic space based on considering the label semantic representation.

4.4. Hyper-parameter value selection

In this section, we give the experiments of hyper-parameter value selections about parameters k and λ in the 1-shot setting. For all experiments, the F1-scores are averaged over 3 runs with different random seeds ± one standard deviation.

Value selection of parameter k In order to find the optimal k of properties to provide additional semantic information, we take k from {1, 2, 3, 4, 5}. Figure 4 shows the results of different values of k on the four benchmark datasets. For dataset News, the F1-score is highest when k is 5, which

K-shot	Model	Wiki	SocialMedia	OntoNotes	News	Ave.
5-shot	LSTM+CRF	4.35±0.09	4.79±0.22	4.81±1.07	11.06±0.31	6.25±0.25
	SpanNER	4.74±0.78	16.38±7.30	9.66±0.31	39.59±2.24	17.59±1.88
	MNet	3.80±0.82	18.45±2.78	9.58±1.23	40.70±4.86	18.13±1.61
	MNet+CDT	6.17±0.63	20.58±3.70	14.62±1.93	**48.87±4.19**	22.56±1.14
	WPZ	2.77±0.34	17.03±3.28	9.41±0.82	28.84±3.91	14.51±0.92
	WPZ+CDT	2.20±1.17	19.69±2.96	17.45±1.92	37.65±3.84	19.25±1.84
	TapNet	3.04±0.56	14.37±1.69	10.85±0.96	31.69±2.40	14.99±0.26
	TapNet+CDT	5.34±1.60	22.12±2.46	15.66±2.66	40.46±4.45	20.89±0.88
	L-TapNet	3.82±0.57	16.51±2.11	11.82±1.42	32.12±2.99	16.07±0.62
	L-TapNet+CDT	5.51±0.47	19.58±3.59	19.78±2.79	32.45±4.67	19.33±1.25
	KG-only (ours)	7.75±2.33	23.25±3.03	19.65±1.51	35.48±7.75	21.53±1.85
	KCL (ours)	**9.06±1.79**	**25.55±3.86**	**23.34±2.62**	39.26±3.70	**24.30±1.11**

Table 3: F1-scores with standard deviations on the four benchmark datasets in 5-shot

illustrates that KCL obtains the most effective information. About the dataset SocialMedia, as k increases, the F1-score first increases and then tends to be stable. This indicates that KCL obtains enough semantic information as k is 2. For datasets Wiki and OntoNotes, as k is 4, the F1-scores of the two datasets are highest compared to that of the other four values. Therefore, we choose 4 as the optimal value for the two datasets. It is worth noting that, as the number of label properties increases, F1-score does not increase all the time but floats up and down. The reason is that the properties are obtained by random sampling, as the number of properties increases, useless label properties for domains may be added, which affects the generalization performance of KCL.

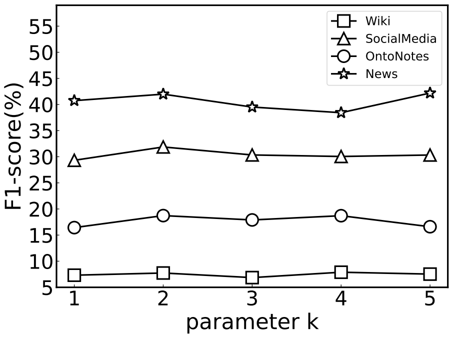


Figure 4: The hyper-parameter value selection of parameter k

Value selection of parameter λ In this section, we adjust parameter λ from {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} to balance the weight of $loss_{CRF}$ and $loss_{CL}$ in the loss function. The F1-scores of different values on four benchmark datasets are shown in Figure 5. For datasets News and OntoNotes, as λ increases, the F1-score is the highest. After that, there is an overall downward trend. About the dataset SocialMedia, the overall trend of F1-scores is downward as the value of λ increases. For the above three datasets, the best weight values of $loss_{CL}$ are all less than 0.5. The possible reason is that there are useless label properties extracted from KG which leads to inaccurate label semantic representation. Therefore, it is necessary to reduce the weight of $loss_{CL}$ to avoid misleading model training. On dataset Wiki, we use 0.8 as the optimal value corresponding to the highest F1-score. This also demonstrates that KCL obtains effective label properties and increases the weight of $loss_{CL}$ for semantic space learning.

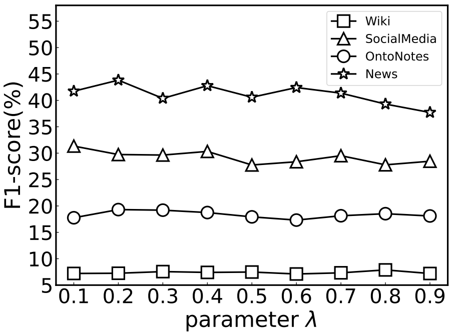


Figure 5: The hyper-parameter value selection of parameter λ

5. Related work

As a classification problem, the few-shot NER studies focus on modeling the correlation of entities-labels and have been widely explored with similarity-based methods(Mettes et al., 2019; Tong et al., 2021; Li et al., 2021). These methods aim to learn a prototype for each label and classify an entity by finding the nearest prototype in a mapping space. Specifically, there are two aspects to learning the prototype for the label:

Label semantics. The conventional prototype network(Tong et al., 2021; Zhu and Wang, 2019;

Lu et al., 2016; Chen et al., 2022a) uses the average of entity representations as their corresponding label prototype, which oversimplifies the learning of label semantic representations. To solve this problem, some studies(Ma et al., 2022a; Hou et al., 2020; Luo et al., 2021) proposed that label names should be leveraged, because label names are also words that appear in the text and thus semantically related to other words that appear in the text. Moreover, the label descriptions are also used to provide rich label information for the label semantic representations learning(Ma et al., 2022b; Wang et al., 2021). Different from the existing methods, our method proposes to apply knowledge graph to the label semantic representations learning. The reason is that the knowledge graph contains properties information about labels and thus can provide richer label semantic information compared with label descriptions.

Relative distance between labels. The similarity-based methods mainly focus on making the distance between entities of the same label close and ignore the distance between labels, which may cause the situation that the distance between sub-labels under the same parent label is too close to cause misclassification. To address this problem, (Hou et al., 2020) proposes to construct different mapping spaces for NER tasks in different domains, and learns reference vectors to distance prototype clusters. Different from this work which uses the simple linear mapping just scale and bias the distribution of the prototype clusters, we use contrastive learning to make the prototype clusters evenly distributed in the prototypical semantic space. Das et al. (2022) uses contrastive learning to improve the relative distance between entities, where the entities with similar semantics are close and those with different semantics should be separated. Compared to it which is label-independent, our method considers the label information as the extra semantic information to improve the semantic space learning.

6. Conclusion

In this paper, we propose a few-shot NER method based on knowledge graph and contrastive learning. To improve the label semantic representation learning, we use knowledge graphs that contain properties and property descriptions of labels and thus provide rich and structured label information for label semantic representation learning. In addition, to effectively utilize the label semantic information to help reduce misclassification caused by label-label similarity, we introduce the idea of contrastive learning to learn the label semantic representation and then help extend the distance between label clusters in the prototypical semantic space. Experiment results validate that both applying knowledge graphs and contrastive learning can improve the prototypical semantic space learning and further improve the few-shot NER accuracy.

Acknowledgment

This research was partially supported by National Natural Science Foundation of China (No. 62276233 and No. 62072405), Key Research Project of Zhejiang Province (2023C01048).

7. References

Ben Athiwaratkun, Cicero dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 375–385.

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. 2022a. 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.

Yuxuan Chen, Jonas Mikkelsen, Arne Binder, Christoph Alt, and Leonhard Hennig. 2022b. A comparative study of pre-trained encoders for low-resource named entity recognition. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 46–59.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2022. Container: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353.

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

John Domingue and Martin Dzbor. 2004. Magpie: supporting browsing and navigation on the semantic web. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 191–197.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. 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 55–65.

A Fritzler, V Logacheva, and M Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the ACM Symposium on Applied Computing*, pages 993–1000.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

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.

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

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

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, pages 4171–4186.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

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

Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. 2021. Adaptive prototype learning and allocation for few-shot segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8330–8339. IEEE.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020b. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Yanan Lu, Yue Zhang, and Donghong Ji. 2016. Multi-prototype chinese character embedding. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)*, pages 855–859.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2773–2782.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022a. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuan-Jing Huang. 2022b. Template-free prompt tuning for few-shot ner. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732.

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

Pascal Mettes, Elise van der Pol, and Cees Snoek. 2019. Hyperspherical prototype networks. *Advances in neural information processing systems*, 32.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Leonardo Rigutini, Marco Maggini, and Bing Liu. 2005. An em based training algorithm for cross-language text categorization. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 529–535. IEEE.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Felix Sasaki, Milan Dojchinovski, and Jan Nehring. 2017. Chainable and extendable knowledge integration web services. In *Knowledge Graphs and Language Technology: ISWC 2016 International Workshops: KEKI and NLP&DBpedia, Kobe, Japan, October 17-21, 2016, Revised Selected Papers 15*, pages 89–101. Springer.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

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

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with books*, 20th edition. The Phantom Editors Associates, Gotham City.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021. Learning from language description: Low-shot named entity recognition via decomposed framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1618–1630.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for wmt17. In *Proceedings of the Second Conference on Machine Translation*, pages 410–415.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pages 7115–7123. PMLR.

Amir Zeldes. 2017. The gum corpus: creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Tao Zhang, Congying Xia, Philip S Yu, Zhiwei Liu, and Shu Zhao. 2021. Pdaln: Progressive domain adaptation over a pre-trained model for low-resource cross-domain named entity recognition. In *EMNLP*.

Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. Zero-shot open entity typing as type-compatible grounding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2065–2076.

Yuying Zhu and Guoxin Wang. 2019. Can-ner: Convolutional attention network for chinese named entity recognition. In *Proceedings of NAACL-HLT*, pages 3384–3393.