A Domain-Transfer Meta Task Design Paradigm for Few-Shot Slot Tagging

Fengyi Yang[1,2,3], **Xi Zhou**[1,2,3*], **Yating Yang**[1,2,3], **Bo Ma**[1,2,3], **Rui Dong**[1,2,3], **Abibulla Atawulla**[1,2,3]

[1] Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China
yangfengyi17@mails.ucas.edu.cn, {zhouxi, yangyt, mabo, dongrui}@ms.xjb.ac.cn, aibibulaatawula19@mails.ucas.ac.cn

Abstract

Few-shot slot tagging is an important task in dialogue systems and attracts much attention of researchers. Most previous few-shot slot tagging methods utilize meta-learning procedure for training and strive to construct a large number of different meta tasks to simulate the testing situation of insufficient data. However, there is a widespread phenomenon of overlap slot between two domains in slot tagging. Traditional meta tasks ignore this special phenomenon and cannot simulate such realistic few-shot slot tagging scenarios. It violates the basic principle of meta-learning which the meta task is consistent with the real testing task, leading to historical information forgetting problem. In this paper, we introduce a novel domain-transfer meta task design paradigm to tackle this problem. We distribute a basic domain to each target domain based on the coincidence degree of slot labels between these two domains. Unlike classic meta tasks which only rely on small samples of target domain, our meta tasks aim to correctly infer the class of target domain query samples based on both abundant data in basic domain and scarce data in target domain. To accomplish our meta task, we propose a Task Adaptation Network to effectively transfer the historical information from the basic domain to the target domain. We carry out sufficient experiments on the benchmark slot tagging dataset SNIPS and the name entity recognition dataset NER. Results demonstrate that our proposed model outperforms previous methods and achieves the state-of-the-art performance.

Introduction

Slot tagging, also called slot filling, is a crucial task in human-machine dialogue systems. The purpose of slot tagging is to identify pre-defined semantic slots from utterances and then the extracted slots are involved in downstream tasks such as dialogue state tracking. Benefiting from the rapid development of deep learning, slot tagging has made great progress in recent years. Researchers have proposed a series of effective algorithms, which usually require a large amount of annotated data.

In reality, it is not feasible to obtain a large amount of annotated conversation data in brand-new dialogue systems. In this situation, the data-driven methods may lead to serious overfitting problems. Therefore, a robust model is required in this data scarcity scenario, which can effectively learn from limited samples. Inspired by human ability to learn new things quickly, researchers proposed few-shot learning (FSL) algorithms (Snell, Swersky, and Zemel 2017; Sung et al. 2018) to overcome the problem of lacking training samples. Generally, few-shot learning methods utilize meta-learning procedure for training and construct a series of meta tasks which are identical to testing tasks. A meta task is constructed by sampling a small training set (support set) and test set (query set) from rich data domains (source domains). The objective of each meta task is to correctly classify query set samples based on only a small amount of data. These meta tasks are selected from different source domains, resulting in strong generalization ability of the meta-learning model. However, the design of traditional meta tasks is not completely suitable for few-shot slot tagging tasks. In slot tagging, it is common for two domains to contain overlap labels, such as *time* and *number_of_people* can be universal for airline ticket booking and restaurant booking. Traditional meta-learning models aim to minimize the loss on multiple different meta tasks and they don't focus on the specific label in each meta task. In other words, these overlap labels in different domains are regarded as different labels in traditional meta task design strategy. It violates the basic principle of

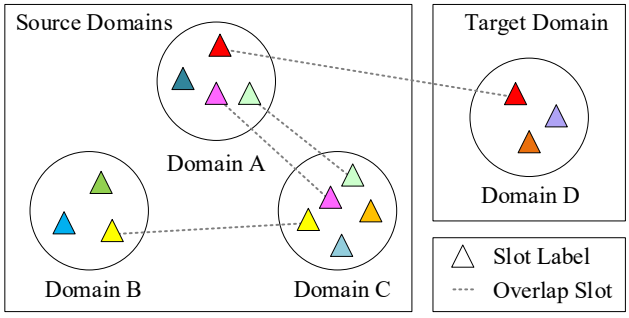

Figure 1: An illustration of slot label distribution in the realistic slot tagging task. There is a widespread phenomenon of overlap slot between two domains. This phenomenon exists not only between the two source domains, but also between the source domain and the target domain.

*Corresponding Author

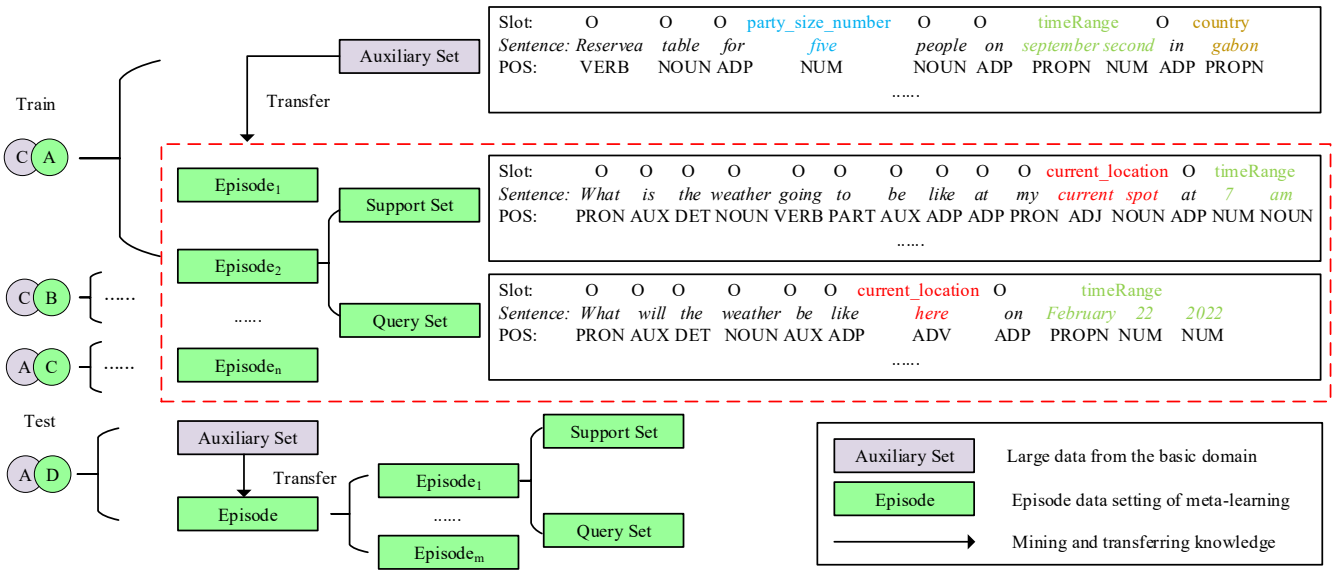

Figure 2: Meta tasks of our proposed method, which the objective is to correctly infer the class of query set samples depending on the support set and auxiliary set. The contents in the red dotted line box are the traditional meta task settings, simulating the situation of data scarcity.

meta-learning which the meta task is consistent with the real testing task, causing historical information forgetting problem.

In this paper, we introduce a novel meta task design paradigm to tackle this problem. We define the basic domain corresponding to each domain based on the coincidence degree of slot labels between these two domains. For example, in Figure 1, Domain A is the basic domain of Domain C. Similarly, Domain A is also the basic domain of Domain D. As shown in Figure 2, the auxiliary set is a collection of a large amount of data in the basic domain, which contains both useful information and a lot of noise. It should be noted that if there is no slot intersection between one domain and any other source domains, we regard a random source domain as the basic domain. We form the novel meta task which the objective is to correctly infer the class of query set samples depending on the support set and auxiliary set. To accomplish our meta task, we propose a Task Adaptation Network to effectively transfer the historical information from the basic domain to the target domain. Task Adaptation Network adopt a two-phase meta-learning framework. Specifically, a basic model is trained on the auxiliary set in a supervised learning manner to learn the historical information of overlap slots. Next, a meta model is designed to learn the meta knowledge of novel slots, which uses traditional meta-learning procedure. Each sentence is fed into basic model and meta model separately and a task adaptation feature fusion module is introduced to fuse the historical information with meta knowledge of sentences depending on different meta tasks. In addition, linguistic features as a universal cross-domain knowledge, playing a guiding role for few-shot slot tagging tasks. To effectively explore the knowledge of linguistic features, we propose a Linguistic Features Enhanced Task Adaptation Network (LFETAN). It explicitly introduces part-of-speech features into the model to assist in constructing the vector representation of words.

The primary contributions of this paper are as follows:

- We introduce a novel domain-transfer meta task design paradigm for few-shot slot tagging, which can simulate realistic few-shot slot tagging tasks and alleviate the problem of historical information forgetting.
- We propose a novel Linguistic Features Enhanced Task Adaptation Network to accomplish our meta task. It effectively fuses overlap labels historical information, novel labels meta knowledge and cross-domain linguistic features.
- We conduct abundant experiments on both slot tagging and named entity recognition datasets to test the performance of our method in few-shot scenarios. Sufficient experimental results show that our method is superior to previous methods and achieves new state-of-the-art results.

Related Work

Few-Shot Slot Tagging

Slot tagging aims to correctly infer the slot label of each word in sentences, which is generally considered as a sequence labeling problem. Most of previous few-shot slot tagging methods adopt metric-based meta-learning strategy for training and testing (Finn, Abbeel, and Levine 2017). (Hou et al. 2020) introduce the conditional random fields (CRFs) (Sutton and McCallum 2012) into the few-shot slot tagging task and modify two key functions (emission and transition scores) in CRF according to the characteristics of the few-shot scenario. Moreover, (Zhu et al. 2020) point out that

the dot product similarity function is not suitable for measuring the word-label similarity in such few-shot tasks and they propose a more appropriate similarity measure function with better generalization ability. (Wang et al. 2021) utilizes contrastive learning to model episode-level relationship and transfers prior knowledge from source domains. These methods follow the traditional meta task design strategy, ignoring the particularity of the slot tagging task. Different from these metric-based methods, (Hou et al. 2022) first apply prompting learning methods for few-shot slot tagging. However, prompting methods need to fine-tune on target domain and has high requirements on the quantity and quality of data.

Generalized Few-Shot Learning

Generalized few-shot learning (GFSL) is an extension of few-shot learning, aiming to correctly classify samples in a joint label space where both existing and novel classes are present. GFSL is a new investigation domain and current works mainly focus on computer vision. (Gidaris and Komodakis 2018) introduce an attention based few-shot classification weight generator and enhance the recognition ability of base and novel classes. (Kukleva, Kuehne, and Schiele 2021) propose a three-stage framework for GFSL, which strikes a balance between preventing base classes, learning novel classes and ultimately correcting. (Fan et al. 2021) find some neglected but useful features by analyzing transfer learning based few-shot object detection and utilize these features to simply and effectively combine the base detector with the novel detector. In the field of natural language processing, most of the current GFSL work focuses on the classification problem. (Yang et al. 2022) extract the diversity features in base categories and utilize these features to enhance features in novel categories for generalized few-shot intent detection. (Chen et al. 2022) extend GFSL on relation classification and explore a new problem called open generalized few-shot relation classification. Although few-shot slot tagging has overlap slot phenomenon, it is still different from the setting of GFSL because most of slots in source domains do not appear in the target domain.

Problem Formulation

In this section, we define the few-shot slot tagging task followed by (Hou et al. 2020). We denote the input sentence as $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ and the corresponding tag sequence as $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$, where n represents the number of words in this sentence. Furthermore, each domain is defined as $D = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{|D|}$, which contains a series of $(\boldsymbol{x}, \boldsymbol{y})$ pairs. The few-shot slot tagging task requires the model to be trained on the source domains D_S and directly tested on the target domain D_T without fine-tuning. Source domains contain a large number of labeled samples while the target domain only includes few labeled data. For the target domain, these few labeled samples form a support set $S = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{|S|}$, which contains K samples (K-shot) for each of N tags (N-way). Given such a support set S and a query $\boldsymbol{x}$, the objective of the few-shot slot tagging task is to find the best tag sequence $\boldsymbol{y}^*$ corresponding to this query $\boldsymbol{x}$. The mathematical formulation is shown in Equation 1.

$$\boldsymbol{y}^* = arg\max_{\boldsymbol{y}} p_\theta\left(\boldsymbol{y}|\boldsymbol{x}, S\right) \tag{1}$$

where θ refers to the parameters of the model, the $(\boldsymbol{x}, \boldsymbol{y})$ pair and the support set S belong to the target domain.

Method

Overall Architecture

In this section, we introduce an overview of the Linguistic Features Enhanced Task Adaptation Network, as shown in Figure 3.

We first distribute a basic domain to each domain based on the slot coincidence degree between these two domains. The slot coincidence degree is calculated by the Jaccard similarity coefficient between slot label spaces of two domains. Given the label set Y_a of domain a and the label set Y_b of domain b, the slot coincidence degree can be calculated as follow:

$$degree\left(Y_a, Y_b\right) = \frac{|Y_a \bigcap Y_b|}{|Y_a \bigcup Y_b|} \tag{2}$$

We regard the domain with the highest slot coincidence degree in sourse domains as basic domain. During training, we select one source domain to simulate the target domain in a meta task and choose its basic domain from the remaining source domains.

Our meta task formulation of few-shot slot tagging can be defined as follows. We regard the total samples in the basic domain as the auxiliary set, $A = D_b = \{(\boldsymbol{x}_b^{(i)}, \boldsymbol{y}_b^{(i)})\}_{i=1}^{|D_b|}$. In the target domain, few labeled samples constitute the support set, $S = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t^{(i)})\}_{i=1}^{|S|}$. while a subset of the remaining samples serve as the query set, $Q = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t^{(i)})\}_{i=1}^{|Q|}$.

In K shot slot tagging setting, support set contains K samples for each of N tags. Given an auxiliary set, a support set and a query set, the objective of our meta task is to correctly infer the class $\boldsymbol{y}^*$ of a query set sample $\boldsymbol{x}_t$ depending on both auxiliary set and support set,

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} p_\theta(\boldsymbol{y}|\boldsymbol{x}, (A, S)) \tag{3}$$

where θ refers to the parameters of the model.

There are two phases in our model. In the first phase, we train a Basic Slot Tagging Network on the auxiliary set for the basic domain slot tagging task. Basic Slot Tagging Network is trained in a supervised learning manner and adopt a CRF framework proposed by (Hou et al. 2020). In the second phase, we use the similar CRF framework to train a Task Adaptation Meta Network, which can determine the knowledge transfer degree from the basic domain according to tasks. Besides, a POS Encoder is designed to acquire POS features and POS features are used to assist in constructing word embeddings.

Basic Slot Tagging Network

Basic Slot Tagging Network (BSTN) aims to learn a Basic Encoder (BE) storing historical information, which is used to transfer historical information from the basic domain to

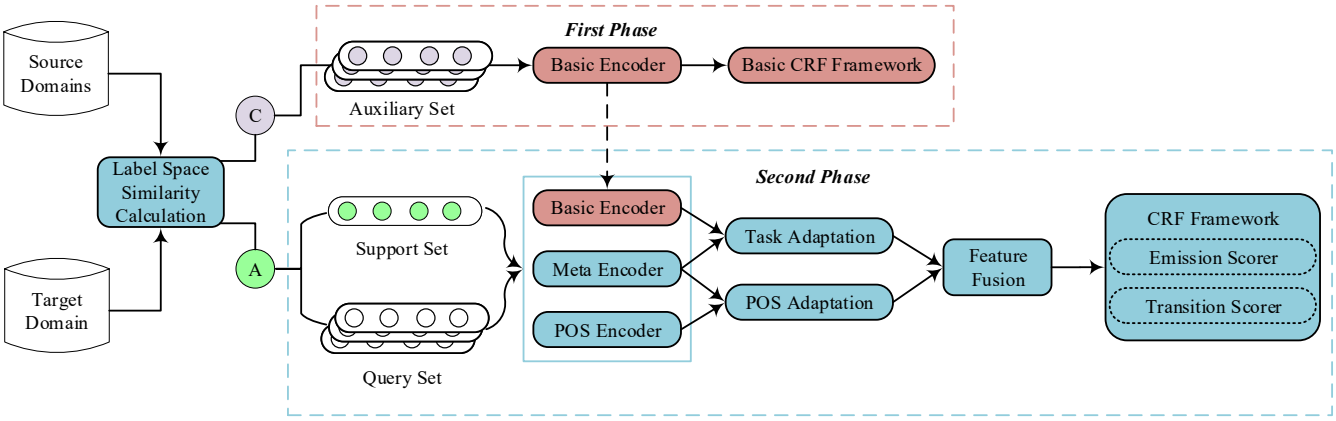

Figure 3: An overview of the Linguistic Features Enhanced Task Adaptation Network. There are two training phases in our method. In the first phase, we select a basic domain and train a corresponding slot tagging model on it. This basic model is based on a large number of annotated data and obtain historical information of overlap labels. In the second phase, we train a slot tagging model on the target domain based on a few labeled data. This meta model is trained on different domains and acquire meta knowledge of novel labels. We design a Task Adaptation module which fuses historical information and meta knowledge. In addition, we use POS features to assist in constructing word embeddings.

the target domain in the second phase. In order to adapt to the second stage tasks, we train the slot tagging task on the auxiliary set and employ a CRF-based few-shot slot tagging framework (Hou et al. 2020). Since the auxiliary set samples are from the same domain, this few-shot framework can still learn appropriate historical information. We construct the auxiliary set A into the form of few-shot episode data setting and each episode includes a support set S_A and a query set Q_A.

There are two essential components in the few-shot CRF framework: Transition Scorer and Emission Scorer. Transition Scorer aims to learn the dependencies between labels. Given a basic domain query set sentence $\boldsymbol{x}_b = (x_1, x_2, \cdots, x_n)$ and a K-shot basic domain support set S_A, the dependency between two labels is the transition probability:

$$f_T(y_{i-1}, y_i) = p(y_i|y_{i-1}) \tag{4}$$

This transition probability is calculated by Collapsed Dependency Transfer (CDT) mechanism (Hou et al. 2020) and the Transition Scorer output is claculated via Equation 5:

$$TRANS(\boldsymbol{y}_b) = \sum_{i=1}^{n} f_T(y_{i-1}, y_i) \tag{5}$$

Emission Scorer aims to learn the correspondence between each word and each label. The correspondence is expressed as

$$\begin{aligned} f_E(y_i, \boldsymbol{x}_b, S_A) &= p(y_i|\boldsymbol{x}_b, S_A) \\ &= Sim(BE(\boldsymbol{x}_b)_i, c_{y_i}) \end{aligned} \tag{6}$$

where BE is Basic Encoder, c_{y_i} is the label embedding of y_i which is calculated based on S_A. In this paper, we use a pre-trained language model BERT (Devlin et al. 2019) as the Basic Encoder and it encodes each word x_i in the sentence as word embedding, $e_i = BE(x_i)$, $e_i \in \mathbb{R}^d$. Prototypical network (Snell, Swersky, and Zemel 2017) is used to calculate the label embedding c_{y_i}, which is the average vector of the corresponding words in the support set S_A. We use Vector Projection Similarity (VP) (Zhu et al. 2020) as the similarity function,

$$Sim(BE(\boldsymbol{x}_b)_i, c_{y_i}) = BE(\boldsymbol{x}_b)_i^T \cdot \frac{c_{y_i}}{\|c_{y_i}\|} \tag{7}$$

The Emission Scorer output is

$$EMIT(\boldsymbol{y}_b, \boldsymbol{x}_b, S_A) = \sum_{i=0}^{n} f_E(y_i, \boldsymbol{x}_b, S_A) \tag{8}$$

The label probability of label $\boldsymbol{y}_b$ is as follows:

$$\begin{aligned} p(\boldsymbol{y}_b|\boldsymbol{x}_b, S_A) = \frac{1}{Z} exp(&TRANS(\boldsymbol{y}_b) + \\ &\lambda \cdot EMIT(\boldsymbol{y}_b, \boldsymbol{x}_b, S_A)) \end{aligned} \tag{9}$$

$$\begin{aligned} Z = \sum_{\boldsymbol{y}_b^{'} \in \boldsymbol{Y}_b} exp(&TRANS(\boldsymbol{y}_b^{'}) + \\ &\lambda \cdot EMIT(\boldsymbol{y}_b^{'}, \boldsymbol{x}_b, S_A)) \end{aligned} \tag{10}$$

λ is a scaling parameter used to balance weights of these two scores. The loss function of Basic CRF Slot Tagging Network is defined as:

$$L_b = -logp(\boldsymbol{y}_b|\boldsymbol{x}_b, S_A). \tag{11}$$

Task Adaptation Meta Network

Task Adaptation Meta Network (TAMN) aims to learn the meta knowledge from the target support set and fuse it with historical information according to different tasks. Historical information is stored in Basic Encoder and the parameters of

BE are fixed during TAMN training. We employ a Meta Encoder (ME) to capture the meta knowledge, which is trained across multiple episodes. A Task Adaptation is designed to fuse meta knowledge and historical information.

Given a query set sentence $\boldsymbol{x}_t = (x_1, x_2, \cdots, x_n)$ and a K-shot support set S. Each word x_i is encoded by Basic Encoder and Meta Encoder respectively,

$$e_i^b = BE(x_i), e_i^b \in \mathbb{R}^d \tag{12}$$

$$e_i^m = ME(x_i), e_i^m \in \mathbb{R}^d \tag{13}$$

Task Adaptation fuses these two feature vectors by,

$$e_i^f = w_i^b \odot e_i^b + (1 - w_i^b) \odot e_i^m \tag{14}$$

where w_i^b is the weight of the historical information, $\odot$ is the element-wise product. w_i^b is calculated by two fully-connected linear layer, the detailed process is as Equation 15 and 16:

$$h_i^b = Relu(W[e_i^b \oplus e_i^m] + d) \tag{15}$$

$$w_i^b = Sigmoid(W^{'} h_i^b + d^{'}) \tag{16}$$

where $\oplus$ is concatenation operation, W, d, $W^{'}$, $d^{'}$ all are the parameters of two linear layer.

The fused features are input into a CRF framework which is the same as the Basic Slot Tagging Network. We define the loss function of Task Adaptation Meta Network as

$$L_m = -logp(\boldsymbol{y}_t|\boldsymbol{x}_t, (A, S)) \tag{17}$$

POS Adaptation Network

As a basic task in natural language processing, part of speech tagging has been well solved. We can use the POS tagging model to mark the POS label z for each word in both the support set S and the query set Q automatically. The enhanced support set with POS label is $S_E = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t^{(i)}, \boldsymbol{z}_t^{(i)}\}_{i=1}^{|S|}$, the POS query set is $Q_P = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{z}_t^{(i)})\}_{i=1}^{|Q|}$. In this work, we employ SpaCy's pre-trained POS tagger[1] to mark POS labels.

In the second phase, we train a POS Encoder (PE) through the POS tagging task, which is used to acquire POS features. Each word x_i is encoded by POS Encoder, the POS feature vector

$$e_i^p = PE(x_i), e_i^p \in \mathbb{R}^d \tag{18}$$

We use the same CRF framework as the Basic Slot Tagging Network for POS tagging. It should be noted that since all POS tags are known, the POS label embedding c_{z_i} is the mean vector of the corresponding words in both enhanced support set S_E and POS query set Q_P. The loss function of POS tagging is:

$$L_{pos} = -logp(\boldsymbol{z}_t|\boldsymbol{x}_t, (S_E, Q_P)) \tag{19}$$

[1]https://spacy.io/api/annotation#pos-tagging

For each word x_i, we use the average vector of the POS feature vector e_i^p and the fusion word vector e_i^f as the final word embedding e_i^f. The final word embedding is used in few-shot CRF framework and the loss function of POS Adaptation Network is calculating in Equation 20:

$$L_{ada} = -logp(\boldsymbol{y}_t|\boldsymbol{x}_t, (A, S_E, Q_P)) \tag{20}$$

Finally, in the second phase, the LFETAN Loss is calculated as:

$$L_{all} = L_{meta} + L_{pos} + L_{ada} \tag{21}$$

Experiments

Datasets

We evaluate our proposed model on the benchmark slot tagging dataset SNIPS (Coucke et al. 2018) and the name entity recognition dataset NER, followed the same data split provided by (Hou et al. 2020). These two datasets are both in the form of few-shot episode data setting (Vinyals et al. 2016) , where each episode contains a support set and a query set. SNIPS dataset contains 7 domains with different label sets: Weather (We), Music (Mu), PlayList (Pl), Book (Bo), Search Screen (Se), Restaurant (Re) and Creative Work (Cr). NER dataset treats 4 dataset as different domains: CoNLL-2003 (News) (Sang and Meulder 2003) , GUM (Wiki) (Zeldes 2017), WNUT-2017 (Social) (Derczynski et al. 2017) and OntoNotes (Mixed) (Pradhan et al. 2013). For each dataset, we follow (Hou et al. 2020) to select one domain for testing, one domain for validation and the remaining domains for training. We need to distribute a basic domain to each domain. Note that the basic domain is only selected from the training set. We build different meta tasks for training. In each training meta task, we select one domain from training set as target domain. During testing, target domain is the test set domain and its basic domain is also selected from training set.

Baselines

SimBERT is a metric-based method that assigns labels based on cosine similarity of word embedding generated by non-fine-tuned BERT.

TransferBERT is a domain transfer-based method using parameter sharing of BERT, which is pre-trained on source domains and fine-tuned on the target domain support set.

WPZ+BERT (Fritzler, Logacheva, and Kretov 2019) is a few-shot slot tagging model based on prototypical network (Snell, Swersky, and Zemel 2017). It regards slot tagging as classification of each word, pre-trains on source domains and then performs word classification directly on the target domain.

L-TapNet+CDT+PWE (Hou et al. 2020) is a few-shot slot tagging method using CRF framework. It introduces the collapsed dependency transfer mechanism to transfer knowledge of label dependencies from source domains.

L-ProtoNet+CDT+VPB (Zhu et al. 2020) introduces different distance functions on the basis of L-TapNet+CDT+PWE and uses the distance function VPB to improve the performance of the model.

Model	We	Mu	PI	Bo	Se	Re	Cr	Avg.
SimBERT	36.10	37.08	35.11	68.09	41.61	42.82	23.91	40.67
TransferBERT	55.82	38.01	45.65	31.63	21.96	41.79	38.53	39.06
WPZ+BERT (2019)	46.72	40.07	50.78	68.73	60.81	55.58	67.67	55.77
L-TapNet+CDT+PWE (2020)	71.53	60.56	66.27	84.54	76.27	70.79	62.89	70.41
L-ProtoNet+CDT+VPB (2020)	73.12	57.86	69.01	82.49	75.11	73.34	70.46	71.63
MCML (2021)	72.30	58.33	69.64	82.90	77.23	72.79	**79.57**	73.25
InversePrompt* (2022)	63.37	53.04	63.33	75.21	56.58	65.73	59.10	62.34
Ours	**76.57**	**63.37**	**75.02**	**87.01**	**80.34**	**76.61**	71.02	**75.71**

Table 1: F1 scores on 1-shot slot tagging of SNIPS. $\star$ indicates the results reproduced by using the source codes.

Model	We	Mu	PI	Bo	Se	Re	Cr	Avg.
SimBERT	53.46	54.13	42.81	75.54	57.10	55.30	32.38	52.96
TransferBERT	59.41	42.00	46.07	20.74	28.20	67.75	58.61	46.11
WPZ+BERT (2019)	67.82	55.99	46.02	72.17	73.59	60.18	66.89	63.24
L-TapNet+CDT+PWE (2020)	71.64	67.16	75.88	84.38	82.58	70.05	73.41	75.01
L-ProtoNet+CDT+VPB (2020)	82.93	69.62	80.86	91.19	86.58	81.97	76.02	81.31
MCML (2021)	81.79	69.70	80.78	91.53	87.09	82.49	**81.07**	82.06
InversePrompt* (2022)	79.08	69.08	75.80	87.87	80.22	76.75	71.71	77.22
Ours	**85.14**	**70.26**	**83.63**	**93.36**	**90.17**	**83.77**	77.33	**83.38**

Table 2: F1 scores on 5-shot slot tagging of SNIPS. $\star$ indicates the results reproduced by using the source codes.

Model	1-shot					5-shot				
	News	Wiki	Social	Mixed	Avg.	News	Wiki	Social	Mixed	Avg.
SimBERT	19.22	6.91	5.18	13.99	11.32	32.01	10.63	8.20	21.14	18.00
TransferBERT	4.75	0.57	2.71	3.46	2.87	15.36	3.62	11.08	35.49	16.39
WPZ+BERT (2019)	32.49	3.89	10.68	6.67	13.43	50.06	9.54	17.26	13.59	22.61
L-TapNet+CDT+PWE (2020)	44.30	12.04	20.80	15.17	23.08	45.35	11.65	23.30	20.95	25.31
L-ProtoNet+CDT+VPB (2020)	43.47	10.95	28.43	33.14	29.00	56.30	18.57	35.42	44.71	38.75
MCML (2021)	-	-	-	-	-	-	-	-	-	-
InversePrompt* (2022)	22.36	10.71	15.18	21.90	17.54	25.23	21.57	19.74	30.01	24.14
Ours	**46.86**	**15.55**	**30.91**	**42.71**	**34.01**	**61.06**	**25.86**	**36.00**	**52.47**	**43.85**

Table 3: F1 scores on 1-shot and 5-shot slot tagging of NER. $\star$ indicates the results reproduced by using the source codes.

MCML (Wang et al. 2021) is a current state-of-the-art meta-learning method for the few-shot slot tagging task. It utilizes contrastive learning to model episode-level relationship and transfers prior knowledge from source domains.

InversePrompt (Hou et al. 2022) is a current state-of-the-art prompting method for few-shot slot tagging. It introduces a novel inverse paradigm for prompting methods and proposes an Iterative Prediction Strategy for learning.

Implementation Details

We take the uncased BERT-Base (Devlin et al. 2019) as Basic Encoder, Meta Encoder and POS Encoder to embed words into different semantic vector representations. We use ADAM (Kingma and Ba 2015) to train our model and set Basic Encoder, Meta Encoder and POS Encoder learning rate as 1e-5, other modules learning rate as 1e-3. We set the word vector dimension d is 768 and learn the scaling parameter λ during training. To prevent the impact of randomness, we use 10 different random seeds to carry out our experiments and report the final average results.

Comparisons with State-of-the-arts

Table 1 and Table 2 indicate an overall improvement of our method for 1-shot and 5-shot slot tagging on SNIPS dataset compared to previous baselines. Table 3 shows the results of these methods on NER dataset.

Overall, our model outperforms all baselines on the two benchmark datasets and reach state-of-the-art results. On the SNIPS dataset, the average F1 score of our model on 1-shot and 5-shot situation is 2.46 and 1.32 higher than that of MCML, respectively. On the NER dataset, our model outperforms InversePrompt by F1 scores of 16.47 and 19.71 on 1-shot and 5-shot situation in average respectively. It can be seemed that the performance improvement of our model is

Model	1-shot	5-shot
Ours	**75.71**	**83.38**
w/o BSTN	72.66	82.22
w/o TAMN	74.55	82.52
w/o POS	73.36	82.67

Table 4: Ablation study of F1 score on SNIPS.

more obvious in 1-shot situation. This is because the smaller number of support set samples, the more important it is to transfer knowledge from source domains.

In terms of specific domains, our model outperforms all baselines in all four domains on the NER dataset and performs worse than MCML in only one domain (Cr) on the SNIPS dataset. We find the Cr domain has a unique characteristic after carefully examining the label fields of all domains. It has only two slots and both of those slots appear in other domains. This unique characteristic is very beneficial to MCML because it has a module specifically designed for this feature. From the results, we can see that the F1 score of MCML on Cr domain increases slightly when increasing the number of support set samples from 1 to 5. This is due to such particular module relies more on learning overlapping knowledge from source domains than on the number of samples in the support set. Compared with MCML, our model is more stable for general slot tagging scenarios.

We also observe that the InversePrompt is not as good as such metric-based meta-learning model in this cross-domain few-shot setting. Generally, prompting methods need to train directly on the small amount of data in the target domain, rather than training in source domains. (Hou et al. 2022) tests the performance of InversePrompt in this cross-domain few-shot slot tagging situation, pretrains the model in source domains and fine-tunes it on the target domain support set. According to this strategy, we conduct complete experiments on our dataset using the code provided by (Hou et al. 2022). Results demonstrate that our proposed method outperforms InversePrompt under all testing circumstances.

Ablation Study

In this section, we conduct a large number of ablation studies to observe the effect of each individual component in our model. We test the performance of our full model and its ablations on SNIPS dataset. As shown in Table 4, the average performance of the model decreases to varying degrees after removing these three modules separately.

"w/o BSTN" means that we employ traditional meta task and use POS features to enhance word embeddings. In 1-shot and 5-shot settings, the performance decreases by 3.05 and 1.16, respectively. It can be seen that BSTN has the most impact on the performance improvement on both 1-shot and 5-shot scenarios, especially in the 1-shot setting. It proves the effectiveness of our novel meta task design paradigm, especially in the case of low resources. This is mainly because our meta task can effectively simulate realistic few-shot slot tagging scenarios and learn the ignored historical knowledge form source domains.

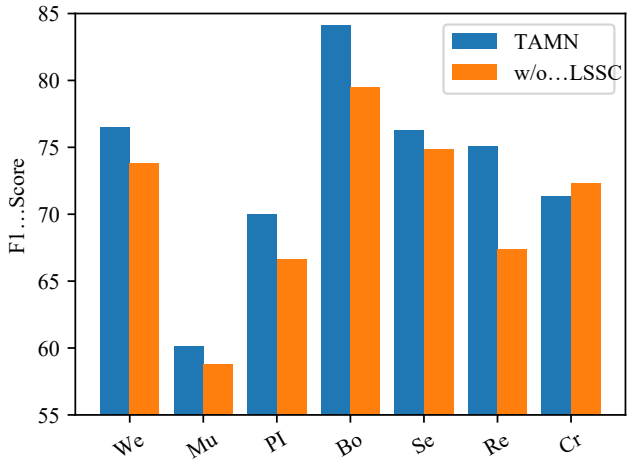

Figure 4: The impact of the basic domain selection.

"w/o TAMN" means that we no longer set a separate loss function for the traditional meta task and change the final loss function to $L_{all} = L_{pos} + L_{ada}$. It the 1-shot setting, changing the final loss function has the least impact on the model. It proves the ability of learning historical information is more important when resources are scarce. With the increase of support set samples, the ability of learning meta knowledge becomes more and more important.

"w/o POS" means that we only use Task Adaptation Meta Network to train the model, ignoring the influence of POS. The final loss function is modified to $L_{all} = L_{meta}$. In the 1-shot and 5-shot settings, the F1 scores are reduced by 2.35 and 0.71. We further explore the impact of the basic domain selection on the Task Adaptation Meta Network. In TAMN, we use Label Space Similarity Calculation (LSSC) to determine the basic domain. It can help to filter out a large amount of irrelevant information compared with directly using the entire source domains as the basic domain. As shown in Figure 4, we compare the results of these two basic domain selection strategies on 1-shot slot tagging of SNIPS. The performance of the TAMN without LSSC decreases in six domains, but improves in the Cr domain because all Cr domain slots appear in source domains. It proves directly using the entire source domains as the basic domain can bring both historical information and a large amount of noise.

Conclusion

In this paper, we introduce a novel domain-transfer meta task design paradigm to simulate realistic few-shot slot tagging tasks and alleviate the problem of historical information forgetting. To accomplish our meta task, we propose a novel Linguistic Features Enhanced Task Adaptation Network. It can effectively fuse overlap labels historical information, novel labels meta knowledge and cross-domain linguistic features. Experimental results show that LFETAN is superior to previous state-of-the-art methods on both slot tagging and named entity recognition datasets in few-shot scenarios. In the future, we will explore the effectiveness of our meta task design paradigm in other tasks.

Acknowledgments

This research is supported by the Natural Science Foundation for Distinguished Young Scholars of Xinjiang Uygur Autonomous Region (2022D01E04), the Xinjiang Science and Technology Major Project (No.2020A02001-1), the Youth Innovation Promotion Association of Chinese Academy of Sciences (Grant No.[2019]26), the Tianshan Innovative Research Team of Xinjiang (Grant No.2020D14045), the West Light Foundation of The Chinese Academy of Sciences (Grant No.2019-XBQNXZ-B-008) and the Youth Innovation Promotion Association of Chinese Academy of Sciences (Grant No.2021436).

References

Chen, X.; Wang, G.; Ren, H.; Cai, Y.; Leung, H.; and Wang, T. 2022. Task-Adaptive Feature Fusion for Generalized Few-Shot Relation Classification in an Open World Environment. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30: 1003–1015.

Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; Primet, M.; and Dureau, J. 2018. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Derczynski, L.; Nichols, E.; van Erp, M.; and Limsopatham, N. 2017. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In Derczynski, L.; Xu, W.; Ritter, A.; and Baldwin, T., eds., *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, 140–147. Association for Computational Linguistics.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Fan, Z.; Ma, Y.; Li, Z.; and Sun, J. 2021. Generalized Few-Shot Object Detection Without Forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 4527–4536. Computer Vision Foundation / IEEE.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 1126–1135. PMLR.

Fritzler, A.; Logacheva, V.; and Kretov, M. 2019. Few-shot classification in named entity recognition task. In Hung, C.; and Papadopoulos, G. A., eds., *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, 993–1000. ACM.

Gidaris, S.; and Komodakis, N. 2018. Dynamic Few-Shot Visual Learning Without Forgetting. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 4367–4375. Computer Vision Foundation / IEEE Computer Society.

Hou, Y.; Che, W.; Lai, Y.; Zhou, Z.; Liu, Y.; Liu, H.; and Liu, T. 2020. Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 1381–1393. Association for Computational Linguistics.

Hou, Y.; Chen, C.; Luo, X.; Li, B.; and Che, W. 2022. Inverse is Better! Fast and Accurate Prompt for Few-shot Slot Tagging. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, 637–647. Association for Computational Linguistics.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kukleva, A.; Kuehne, H.; and Schiele, B. 2021. Generalized and Incremental Few-Shot Learning by Explicit Learning and Calibration without Forgetting. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 9000–9009. IEEE.

Pradhan, S.; Moschitti, A.; Xue, N.; Ng, H. T.; Björkelund, A.; Uryupina, O.; Zhang, Y.; and Zhong, Z. 2013. Towards Robust Linguistic Analysis using OntoNotes. In Hockenmaier, J.; and Riedel, S., eds., *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, 143–152. ACL.

Sang, E. F. T. K.; and Meulder, F. D. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Daelemans, W.; and Osborne, M., eds., *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, 142–147. ACL.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical Networks for Few-shot Learning. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 4077–4087.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H. S.; and Hospedales, T. M. 2018. Learning to Compare: Relation

Network for Few-Shot Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 1199–1208. Computer Vision Foundation / IEEE Computer Society.

Sutton, C.; and McCallum, A. 2012. An Introduction to Conditional Random Fields. *Found. Trends Mach. Learn.*, 4(4): 267–373.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 3630–3638.

Wang, H.; Wang, Z.; Fung, G. P. C.; and Wong, K. 2021. MCML: A Novel Memory-based Contrastive Meta-Learning Method for Few Shot Slot Tagging. *CoRR*, abs/2108.11635.

Yang, F.; Zhou, X.; Wang, Y.; Atawulla, A.; and Bi, R. 2022. Diversity Features Enhanced Prototypical Network for Few-shot Intent Detection. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 4447–4453. ijcai.org.

Zeldes, A. 2017. The GUM corpus: creating multilayer resources in the classroom. *Lang. Resour. Evaluation*, 51(3): 581–612.

Zhu, S.; Cao, R.; Chen, L.; and Yu, K. 2020. Vector Projection Network for Few-shot Slot Tagging in Natural Language Understanding. *CoRR*, abs/2009.09568.