Prior Omission of Dissimilar Source Domain(s) for Cost-Effective Few-Shot Learning

Zezhong Wang[1,2,*], **Hongru Wang**[1,2,*], **Wai-Chung Kwan**[1,2], **Kam-Fai Wong**[1,2]
[1]The Chinese University of Hong Kong, Hong Kong, China
[2]MoE Key Laboratory of High Confidence Software Technologies, China
[1,2]{zzwang, hrwang, wckwan, kfwong}@se.cuhk.edu.hk

Abstract

Few-shot slot tagging is an emerging research topic in Natural Language Understanding (NLU). Conventional few-shot approaches use all the data from the source domains without considering inter-domain relations and implicitly assume each sample in the domain contributes equally. However, our experiments show that transferring knowledge from dissimilar domains will introduce extra noises that decrease the performance of models. We propose an effective similarity-based method to select data from the source domains to tackle this problem. In addition, we propose a Shared-Private Network (SP-Net) for the few-shot slot tagging task. The words from the same class would have some shared features. We extract those shared features from the limited annotated data on the target domain and merge them as the label embedding to help us predict other unlabelled data on the target domain. The experiment shows that our method outperforms the state-of-the-art approaches with fewer source data. The result also proves that some training data from dissimilar sources are redundant and even negative for the adaptation.

1 Introduction

Slot tagging (Tur and De Mori, 2011), one of the crucial problems in Natural Language Understanding (NLU), aims to recognize pre-defined semantic slots from sentences and usually is regarded as a sequence labeling problem (Sarikaya et al., 2016). For example, given the sentence "Book a ticket to London", the word "London" should be recognized as the slot "CITY" by the NLU model.

Currently, most of the methods for the slot tagging task have a notorious limitation in that they require a lot of annotated data. However, there are almost infinite long tail domains in the real scenarios (Zhu et al., 2014) so it is nearly impossible to annotate sufficient data for each domain. Therefore, few-shot learning methods (Ravi and Larochelle, 2016) have received attention as they can transfer the knowledge learned from the existing domains to new domains quickly with limited data.

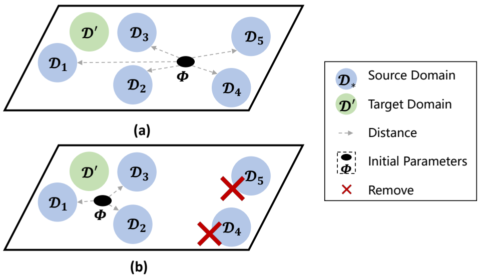

Figure 1: The difference between training with (a) all data and (b) data selection. The dashed line represents the distance among different domains in the parameter space with the centroid (Φ). With data selection, we remove the dissimilar domains $\mathcal{D}_4$ and $\mathcal{D}_5$ from training and the centroid will be closer to the target domain $\mathcal{D}'$.

Current works (Yoon et al., 2019; Liu et al., 2020; Wang et al., 2021) proposed various methods to improve the performance of slot tagging few-shot learning, but most of them focus on "how" to transfer rather than "what" should be transferred. The knowledge from the not-relevant source domain is hard to help the model identify the slots in the new domain. Further, such kind of knowledge is redundant and sometimes could be regarded as noises that even deteriorate the performance (Wang et al., 2019; Meftah et al., 2021). We observe this phenomenon and prove the existence of the negative transfer in the experiment. To this end, we propose a similarity-based method to evaluate the inter-domain relation and indicate which domains should be selected for training. Specifically, we calculate three different similarities, including target vocabulary covered (TVC), TF-IDF similarity (TIS), and label overlap (LO) between domains, and combine them with different weights. The combined similarity function selects data from both corpus level and label level, which is more com-

[*] These authors contributed equally to this work.

prehensive. In this way, the dissimilar sources will be rejected, and the initial parameters of the model will be naturally more closed to the local optimum of the target domain. A high-level intuition of the difference between training with all data and training with data selection is shown in Figure 1.

After selecting pertinent data, we also propose a solution about “how" to transfer knowledge for a few-shot slot tagging task. Specifically, we build a Shared-Private Network to capture stable label representations under the few-shot setting. Many works (Hou et al., 2020; Zhu et al., 2020; Liu et al., 2020) try to enhance the accuracy of slot identification from the label representation engineering. They assign each label with a semantic vector (Snell et al., 2017; Hou et al., 2020; Zhu et al., 2020; Yoon et al., 2019) rather than a simple one-hot encoding. However, the quality of the label representations highly depends on the volume of the training samples and suffers from the unstable problem under the few-shot setting due to the extremely biased data distribution. Hence, we propose the Shared-Private Network separate the shared and private features from the limited samples. The words with the same label share common information. They are extracted and saved as shared features. Other parts are regarded as detailed information related to the words and will be saved as private features. After filtering the detailed information out, the label representation generated according to the shared features will be more robust against the annotation shortage problems in the few-shot setting.

The contributions of this work are as follows:

- We propose a similarity-based method to measure the relation among domains to guide data selection and to avoid negative knowledge transfer in few-shot learning.
- We propose the Shared-Private Network to extract more stable label representation with limited annotations.
- We prove the existence of negative transfer via experiments and give explanations about this phenomenon via visualization.

2 Related Work

Convention studies in slot tagging mainly focus on proposing and utilizing deep neural networks to recognize the semantic slots in given contexts (Shi et al., 2016; Kim et al., 2017). However, most of these models need a large amount of annotated data which is scarce in the real world, especially for those minority domains. Recent works (Bapna et al., 2017; Shah et al., 2019; Rastogi et al., 2019; Liu et al., 2020) propose several few-shot learning methods for slot tagging and developed domain-specific model with limited annotated data. It is worth noting that, due to the lack of annotation on the target domain, both approaches paid attention to label representation engineering rather than using conventional one-hot encoding directly. But building label representation with limited annotations is still a challenge. To stabilize the effectiveness of label representation, we proposed a Shared-Private network to learn representation from shared information of words.

Besides that, negative transfer that transferring knowledge from the source can have a negative impact on the target has been founded in many tasks (Wang et al., 2019; Chen et al., 2019; Gui et al., 2018). Because of this phenomenon, recent methods for relation analysis between source and target domains have been proposed. Gururangan et al. (2020) use vocabulary overlap as the similarity between two datasets and emphasize the significant impact of domain-adaptive for pre-training. Dai et al. (2019) study different similarity methods, including target vocabulary covered (TVC), language model perplexity (PPL), and word vector variance (WVV) to select data for pre-training tasks. However, a single similarity function does not work well in the few-shot setting. Different similarity methods always give diverse data selection strategies and are hardly consistent. To this end, we propose a comprehensive indicator that combines three similarity functions to guide the data selection in the few-shot setting.

3 Problem Definition

We follow the same task definition as Hou et al. (2020). Given a sentence $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ as a sequence of words, slot tagging task aims to assign the corresponding label series $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ to indicate which classes the words should belong to. A domain $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N_\mathcal{D}}$ is a set of $(\mathbf{x}, \mathbf{y})$ pairs that from same scenario and $N_\mathcal{D}$ is the number of sentences in domain $\mathcal{D}$.

In few-shot setting, models are trained from source domain $\{\mathcal{D}_1, \mathcal{D}_2, \cdots\}$ and are applied to the target domain $\{\mathcal{D}'_1, \mathcal{D}'_2, \cdots\}$ which are new to the models. It is worth note that there are only few labeled samples, which make up the support set

$\mathcal{S} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N_S}$, in each target domain $\mathcal{D}'_j$. For each unique N labels (N-way) in support set $\mathcal{S}$, there are K annotated samples (K-shot). Besides that, the samples in the target domain $\mathcal{D}'_j$ are unlabeled.

Thus, few-shot slot tagging task is defined as follows: given a K-shot support set $\mathcal{S}$ and a query sentence $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, determine the corresponding labels sequence $\mathbf{y}^*$:

$$\mathbf{y}^* = (y_1^*, y_2^*, \cdots, y_n^*) = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x},\mathcal{S}) \quad (1)$$

4 Data Selection

In this section, we first show the existence of negative knowledge transfer among domains. The phenomenon demonstrates the necessity of data selection. Then introduce our similarity-based data selection strategy that can be used to avoid negative knowledge transfer to improve performance in few-shot slot tagging.

4.1 Negative Knowledge Transfer

Due to negative knowledge transfer, some knowledge the model learned before is useless and may affect the judgment of the model on the new domains, which will degrade the performance. In the preliminary study, we train the model with all different combinations of source domains and record their performance. The relation between the number of source domains and their corresponding performance is shown in Figure 2. Overall, with more training domains, the performance would be better. However, comparing the maximum values, it is evident that training with three source domains outperforms training with 4. This phenomenon indicates that more source domains may even decrease the performance and proves the existence of negative knowledge transfer. It also inspires us that the model will achieve a better result with proper data selection.

4.2 Selection Strategy

An indicator is needed to select data or source domains before training to avoid negative knowledge transfer. Given a group of data from source domain and the data of target domain, the indicator should output a score that can reflect how fit are these source data for transferring knowledge to the target. Ideally, the indicator score behaves linearly with the performance so that a higher indicator score can lead to better performance. In this way, the group of source data with the highest indicator score can be selected as the best choice for training.

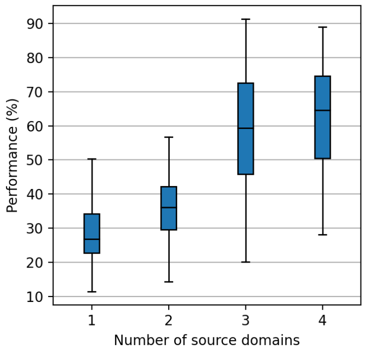

Figure 2: The relationship between performance (y-axis), specifically the F1 score, and the number of source domains (x-axis).

The data that can be leveraged includes the source domains $\{\mathcal{D}_1, \cdots, \mathcal{D}_M\}$ with sufficient labels, the support set $\mathcal{S}_j$ with labels in the target domain $\mathcal{D}'_j$, and the query set $\mathcal{Q}_j$ without labels. Notice that the data in the support set $\mathcal{S}_j$ is much less than the query set $\mathcal{Q}_j$. Considering the attributes mentioned above and the data we can use, we investigate three similarity functions as indicators for data selection.

Target Vocabulary Covered (TVC) is a significant corpus level feature that represents the overlap of vocabulary between source domain(s) and a target domain and is defined as:

$$\mathbf{TVC}(\mathcal{D}_i, \mathcal{D}'_j) = \frac{\left|V_{\mathcal{D}_i} \cap V_{\mathcal{D}'_j}\right|}{\left|V_{\mathcal{D}'_j}\right|} \quad (2)$$

where $V_{\mathcal{D}_i}$ and $V_{\mathcal{D}'_j}$ are the vocabularies (sets of unique tokens) of the source domain $\mathcal{D}_i$ and the target domain $\mathcal{D}'_j$ respectively and $|\cdot|$ is the norm operation that indicates the size of the set. Intuitively, if most of words in the target domain have already appeared in the sources, the word embeddings should have been well trained so that improves the performance.

TF-IDF Similarity (TIS) is another corpus level feature (Bao et al., 2020). We treat each domain as a document and calculate their `tf-idf` features (Salton and Buckley, 1988; Wu et al., 2008). Cosine similarity is used to evaluate the correlation between the sources and the target. Compared with TVC, TIS assigns each word a weight according to the term frequency and inverse document frequency, which takes fine-grained corpus feature into account. The details are shown below:

$$\mathrm{tf}_{i,j} = \frac{n_{ij}}{\sum_k n_{k,j}} \tag{3}$$

where n_{ij} is the times of word t_i appeared in domain $\mathcal{D}_j$.

$$\mathrm{idf}_i = \lg\left(\frac{M}{|\{j : t_i \in \mathcal{D}_j\}_{j=1}^{M}|}\right) \tag{4}$$

where M is the total number of domains. And the `tf-idf` feature is the product of `tf` and `idf`:

$$\text{tf-idf}_j = \mathrm{tf}_{i,j} \cdot \mathrm{idf}_i \tag{5}$$

tf-idf$_j$ can be regarded as the word distribution feature of the domain j, and cosine similarity is used to evaluate the correlation between the two domains:

$$\mathrm{TIS}(\mathcal{D}_i, \mathcal{D}_j) = \frac{\mathrm{tfidf}_{\mathcal{D}_i} \cdot \mathrm{tfidf}_{\mathcal{D}_j}}{||\mathrm{tfidf}_{\mathcal{D}_i}||_2 \cdot ||\mathrm{tfidf}_{\mathcal{D}_j}||_2} \tag{6}$$

where $|| \cdot ||_2$ is the Euclidean norm.

Label Overlap (LO) is a label level feature that represents the overlap of labels between source domains and the target domain. Although labels are scarce in the target domain under the few-shot setting, the types of labels are not. Every label on the target domain at least appeared K times (K-shot) in the support set $\mathcal{S}$ and therefore, the types of the labels are complete. Hence, label overlap is also a good choice as a data selection indicator:

$$\mathrm{LO}(Y_i, Y_j) = \frac{|Y_i \cap Y_j|}{|Y_j|} \tag{7}$$

where Y_i and Y_j stand for the unique label set of the source domain $\mathcal{D}_i$ and the target domain $\mathcal{D}'_j$, respectively.

Each similarity function only focus on a single aspect, i.e. the corpus level information or the label level. Therefore, it is inevitable to introduce bias when we select data with them. Naturally, we come up with a strategy that combines all three similarity scores as the indicator to give a more stable guidance for data selection. Assume that one of the combinations, i.e. $C_{\theta_1,\theta_2,\theta_3}(\mathrm{TVC}_i, \mathrm{TIS}_i, \mathrm{LO}_i) = \theta_1\mathrm{TVC}_i + \theta_2\mathrm{TIS}_i + \theta_3\mathrm{LO}_i$, is linear with the performance, our goal is to find the best value of θ_1, θ_2, and θ_3. For a better reading experience, $C_{\theta_1,\theta_2,\theta_3}(\mathrm{TVC}_i, \mathrm{TIS}_i, \mathrm{LO}_i)$ is abbreviated to C_i.

Algorithm 1 Training with combination of source domains

```
Require: Set of source domains $\{\mathcal{D}_1, \cdots, \mathcal{D}_M\}$; Target domain $\mathcal{D}'$; Model $\mathcal{F}$;
for $1 \le i \le M$ do
    all_combination = $combination(\{\mathcal{D}_1, \cdots, \mathcal{D}_M\}, i)$
    // Select $i$ domain(s) from $M$ for training.
    for $1 \le j \le$ |all_combination| $- 1$ do
        combination = all_combination[$j$]
        // e.g. combination = $[\mathcal{D}_1, \mathcal{D}_3]$
        $\mathcal{D}_{\text{training}} \leftarrow Merge(\text{combination})$
        TVC = $TVC(\mathcal{D}_{\text{training}}, \mathcal{D}')$
        TIS = $TIS(\mathcal{D}_{\text{training}}, \mathcal{D}')$
        LO = $LO(\mathcal{D}_{\text{training}}, \mathcal{D}')$
        train $\left(\mathcal{F}(\mathcal{D}_{\text{training}})\right)$ until Loss converge
        $\hat{p}_i$ = eval(($\mathcal{F}(\mathcal{D}')$))
    end for
end for
```

Following the least squares method (Merriman, 1877), we design the objective function as follows:

$$\begin{aligned} &\underset{\theta_1,\theta_2,\theta_3,w,b}{\arg\min}\ \frac{1}{N_E}\sum_{i=1}^{N_E} \left|\left|[wC_i + b] - \hat{p}_i\right|\right|^2 \\ &\text{s.t.}\quad w > 0, b \ge 0 \end{aligned} \tag{8}$$

where w and b are the weight and bias of the linear function to simulate the linear relation between the indicator score and the performance. N_E is the number of the experiments, and $\hat{p}_i$ is the actual performance of the experiment i. TVC$_i$, TIS$_i$, and LO$_i$ are the TVC score, TIS score, and LO score between the source domains and the target domain in the experiment i.

To solve the problem in equation (8), we design a scheme to generate samples with the combination of source domains. We pre-define the number of source domains and enumerate all combinations. The three similarity scores between the combination of source domains and target domains will be calculated and recorded. Then we train the model with the combination and record the final performance on the target domain. In this way, we get sufficient tuples $(\texttt{TVC}, \texttt{TIS}, \texttt{LO}, p)$ to figure out the optimum θ_1, θ_2, and θ_3 (see Algorithm 1).

With sufficient samples, we fit them with the linear function in equation (8) and optimize w, b, θ_1, θ_2, and θ_3 via `SGD` (Curry, 1944). Due to the data distribution bias of different domains, we finally assign different w_j and b_j for each target domain $\mathcal{D}'_j$ to acquire a better linear relation. For the combination weights θ_1, θ_2, and θ_3, we keep same for different target domains. Further, we still have the following points to declare:

- The parameters w and b are learnable but unnecessary for data selection. They are not a part of the indicator and are only used to observe the linear relation between the combination similarity scores and the corresponding performance.
- Due to the cross-validation setting in the real dataset (e.g., SNIPS), to avoid data leakage of the target domain, we obtain θ_1, θ_2, and θ_3 according to the validation domain for each target. The combination from the validation domain still works well on the target and can prove the generality of this strategy.
- Although training with a combination of source domains is time-consuming, it can be adapted to different domains once the optimum combination weights have been found.

After that, we can select domains according to the optimum w^*, b^*, θ_1^*, θ_2^*, and θ_3^*. The domains which can achieve a higher combined similarity score may lead to better performance, and this can be formulated as:

$$\arg\max_i \quad w^* \left(\theta_1^* \text{TVC}_i + \theta_2^* \text{TIS}_i + \theta_3^* \text{LO}_i\right) + b^* \tag{9}$$

And due to $w > 0$, equation (9) is equivalent to:

$$\arg\max_i \quad \theta_1^* \text{TVC}_i + \theta_2^* \text{TIS}_i + \theta_3^* \text{LO}_i \tag{10}$$

In this way, the domain specific w and b are eliminated.

5 Shared-Private Network

Based on the Prototypical Network (Snell et al., 2017), we propose the Shared-Private Network (SP-Net) to gain more representative label embeddings. The workflow is divided into two stages: SP-Net extracts label embeddings for each class from the support set in the first stage. SP-Net predicts each query sentence in the second stage according to the label embeddings extracted from stage one. Figure 3 illustrates this process.

(a) Encode Firstly, sentences are encoded into word embeddings via BERT (Devlin et al., 2019). Given a sentence $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ as a sequence of words, BERT will generate their corresponding contextual word embeddings $\mathbf{E} = (E_1, E_2, \cdots, E_n)$, where $E_i \in \mathbb{R}^h$. h is the hidden size of the word embedings.

(b) Extract shared features Although words are different, there is common information among words from the same class. Intuitively, the same class words always appear in a similar context with similar syntax. And in some cases, they can even be replaced with each other without any grammatical mistakes. For example, even though we replace the phrase “Hong Kong" with “New York" in Figure 3, the sentence still makes sense. Common information can help us generate scalable label embeddings that can represent most of the words in a class. The shared layer in the framework is designed for this. In this work, we implement the shared layer with a residual linear function, and the shared feature of a word is calculated as follows:

$$E_i^s = E_i + \text{RELU}(E_i W_s + b_s) \tag{11}$$

where $W_s \in \mathbb{R}^{h \times h}$ and $b_s \in \mathbb{R}^h$ are the weight and bias of the shared layer, respectively. RELU is the rectified linear unit function (Maas et al., 2013).

(c) Extract private features Besides the shared information, each word still has its specific information. Recalling the phrase replacing case mentioned in Figure 3, although the sentence is without any grammatical mistakes after phrase replacing, the meaning has been changed. This is due to the private information carried by the word. The private information is ineffective and can be harmful to label embeddings as they lack generality. Less private information can lead to a better quality of label embeddings, and therefore, the private layer is designed to extract private information from the word embeddings. The private layer is also implemented with a residual linear function, and the private feature of a word is calculated as follows:

$$E_i^p = E_i + \text{RELU}(E_i W_p + b_p) \tag{12}$$

where $W_p \in \mathbb{R}^{h \times h}$ and $b_p \in \mathbb{R}^h$ is the weight and bias of the private layer, respectively. So far, the shared layer and private layer are symmetrical and share the same design.

(d) orthogonality constrain To ensure the shared features and private features are separated completely, we introduce the following constraints:

- The shared features of the words in the same class should be close to each other.
- The private features of words should be diverse even though they belong to the same class.
- The shared and private features of a word should not overlap.

For the first requirement, Chen et al. (2020) proposed using contrastive loss that can make the same

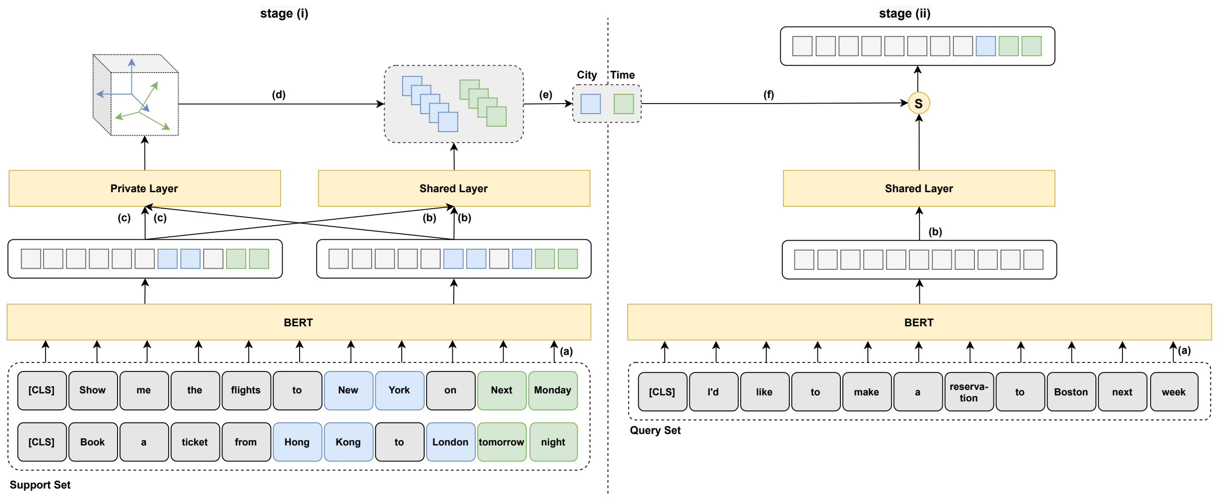

Figure 3: This is the workflow of SP-Net. In this case, the support set contains 2 sentences, and the query set contains 1. The details of processes (a) encode, (b) extract shared features, (c) extract private features, (d) orthogonality constrain, (e) extract label embeddings, and (f) predict are introduced in the main body.

samples close and different samples far apart. The similarity between samples are defined as follows:

$$\text{sim}(E_i^s, E_j^s) = \frac{{E_i^s}^\top E_j^s}{\|E_i^s\|\|E_j^s\|} \tag{13}$$

The loss in the first requirement is defined as:

$$\mathcal{L}_1 = \mathop{\mathbb{E}}_{c}\left[-\log \frac{\sum\limits_{\{i;y_i=c\}}\sum\limits_{\{j;y_j=c\}} e^{\text{sim}(E_i^s,E_j^s)/\tau}}{\sum\limits_{\{i;i\in\mathcal{S}\}}\sum\limits_{\{j;j\in\mathcal{S}\}} e^{\text{sim}(E_i^s,E_j^s)/\tau}}\right] \tag{14}$$

where τ is the temperature parameter and c is the class. The numerator is the sum of the similarity scores whose class is c. The denominator is the sum of all the similarity scores. Specifically, embeddings in the same class present a high similarity score and the numerator is large, and the loss decreases.

For the second requirement, according to the covariance of two variables, we define the divergence between two embeddings as:

$$D(E_i^p, E_j^p) = (E_i^p - \mathbb{E}^p)^T(E_j^p - \mathbb{E}^p) \tag{15}$$

where $\mathbb{E}^p$ is the mean vector of all private embeddings in the set. The loss in the second requirement is:

$$\mathcal{L}_2 = -\frac{1}{|\mathcal{S}|^2}\sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}}\log D(E_i^p, E_j^p) \tag{16}$$

where $|\mathcal{S}|$ is the size of the support set, i.e., the number of words. Higher divergence among the private embeddings will lead to lower loss. We also implement `L2-norm` to restrain the increase of the parameters.

The third requirement refines the shared features further. We introduce the orthogonality constraints (Liu et al., 2017) to force the shared embedding independent of the private embedding:

$$\mathcal{L}_3 = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left\|{E_i^s}^\top E_i^p\right\|_2 \tag{17}$$

where $\|\cdot\|_2$ is the Euclidean norm.

(e) Extract label embeddings Label embeddings are extracted from shared embeddings for each class. We take the mean vector of the shared embeddings which belong to class c as the label embedding:

$$E^c = \frac{1}{|\{y_i = c\}|}\sum_{\{y_i=c\}} E_i^s \tag{18}$$

where E^c is the label embedding of the class c.

(f) Predict We calculate the similarity between shared embeddings of the query sentence with the label embeddings. We provide various options, and here we take cosine similarity as an example:

$$p_i^c = \frac{{E_i^s}^\top E^c}{\|E^s\|\|E^c\|} \tag{19}$$

where p_i^c is the similarity between word i with class c and can also be regarded as the confidence that the word belongs to this class. The class with the highest similarity will be considered as the prediction for the word. We take the binary cross-

	Model	We	Mu	Pl	Bo	Se	Re	Cr	Avg.
1-shot	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
	L-TapNet+CDT+PWE (Hou et al., 2020)	71.53	60.56	66.27	**84.54**	76.27	70.79	62.89	70.41
	L-ProtoNet+CDT+VPB (Zhu et al., 2020)	73.12	57.86	69.01	82.49	75.11	73.34	70.46	71.63
	BERT-ProtoNet	60.01	43.33	52.42	44.37	47.86	50.91	39.04	45.65
	SP-Net	70.67	59.27	69.58	82.80	76.92	72.49	74.63	72.34
	SP-Net + Domain Selection	**76.07**	**64.29**	**71.10**	84.19	**81.63**	**73.66**	**76.41**	**75.34** (+3.71)
5-shot	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
	L-TapNet+CDT+PWE(Hou et al., 2020)	71.64	67.16	75.88	84.38	82.58	70.05	73.41	75.01
	L-ProtoNet+CDT+VPB(Zhu et al., 2020)	82.93	69.62	80.86	91.19	86.58	**81.97**	76.02	81.31
	BERT-ProtoNet	68.98	59.31	62.42	81.35	78.91	67.57	71.69	70.03
	SP-Net	83.92	69.37	79.47	89.43	87.95	77.75	80.31	81.17
	SP-Net + Domain Selection	**84.03**	**71.09**	**82.01**	**92.13**	**89.44**	80.71	**80.88**	**82.90** (+1.59)

Table 1: F1 scores of few-shot slot tagging on SNIPS dataset.

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
L-TapNet+CDT+PWE	**44.30**	12.04	20.80	15.17	23.08	45.35	11.65	23.30	20.95	25.31
L-ProtoNet+CDT+VPB	43.47	10.95	**28.43**	33.14	29.00	56.30	18.57	35.42	44.71	38.75
SP-Net	43.95	**13.02**	27.77	**34.05**	**29.70**	**57.70**	18.62	**36.41**	**44.97**	39.42
SP-Net Domain Selection	43.95	13.02	27.77	34.05	29.70 (+0.70)	57.70	**21.11**	36.41	44.97	**40.05** (+1.30)

Table 2: F1 scores of few-shot slot tagging on NER dataset. The performance improvements of SP-Net Domain Selection compared to all baselines are significant (validated by Student's t-test with p-value < 0.01).

entropy loss to measure the error in each class:

$$\mathcal{L}_4 = \frac{1}{|Q|}\sum_{i}^{|Q|}\sum_{c}^{C} y_i \log p_i^c + (1-y_i)\log(1-p_i^c) \tag{20}$$

where C is the number of unique labels in the query set and $|Q|$ is the number of words in the query set.

Finally, we combine the $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, and $\mathcal{L}_4$ with different weights as the cost function:

$$\mathcal{L} = \alpha\mathcal{L}_1 + \beta\mathcal{L}_2 + \gamma\mathcal{L}_3 + \delta\mathcal{L}_4 \tag{21}$$

where α, β, γ, and δ are hyperparameters determined by the experiments.

6 Experiments

6.1 Dataset

We evaluate the proposed method following the same experiment setting provided by Hou et al. (2020) on SNIPS (Coucke et al., 2018) and NER dataset (Zhu et al., 2020). SNIPS contains seven domains, including Weather (We), Music (Mu), PlayList (Pl), Book (Bo), Search Screen (Se), Restaurant (Re), and Creative Work (Cr), and the sentences in SNIPS are annotated with token-level BIO labels for slot tagging. Each domain will be tested in turn following the cross-validation strategy. Five domains are used for training and one for evaluation in each turn. In each domain, the data are split into 100 episodes (Ren et al., 2018). For the sake of fair peer comparison, the selection of evaluation domain and episodes construct are kept the same with Hou et al. (2020). The NER dataset contains four domains: News, Wiki, Social, and Mixed. In addition, because the number of domains in the NER dataset is too short, we randomly split domains into pieces and select those pieces via the combined similarity function. More training details can be found in the appendix.

6.2 Baselines

SimBERT assigns a label to the word according to cosine similarity of word embedding of a fixed BERT.

TransferBERT directly transfers the knowledge from the source domain to the target domain by parameter sharing.

L-TapNet+CDT+PWE (Hou et al., 2020) combines with the label name representation and a special CRF framework.

L-ProtoNet+CDT+VPB (Zhu et al., 2020) utilizes the powerful distance function VPB to boost the performance of the model.

BERT-ProtoNet is the model proposed in this work which is without the Shared-Private layer. This model is used for ablation study.

SP-Net is the Shared-Private Network proposed in this work.

SP-Net + Domain Selection is also SP-Net, but it is trained with the selected data according to the data selection strategy we proposed.

6.3 Main Results

Table 1 shows the results of 1-shot and 5-shot on the SNIPS dataset. Generally speaking, the SP-Net achieves the best performance on the 1-shot setting and comparable performance on the 5-shot setting (0.14% adrift of SOTA). The data selection strategy dramatically enhances the performance on both the 1-shot and 5-shot settings. With the data selection, the performance of SP-Net is far beyond other baselines.

The result on the NER dataset also proves the effectiveness of our method (See Table 2). It is noticed that, due to the short of the data, combined similarity select all data on most domains except Wiki of 5-shot task. Therefore the result of SP-Origin and SP-Domain Selection are nearly the same.

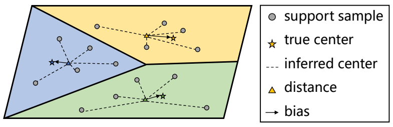

Figure 4: This is diagram shows the automatic correction of distribution bias when the number of supports increased. The circles are samples in the support set and triangles are the inferred center, as well as label embedding, according to the supports. Stars are the true center of classes.

The effect of the Shared-Private Network is more remarkable if the number of support samples is less. The SP-Net outperforms all baseline in the 1-shot setting, but in 5-shot, it achieves comparable performance. The shared-private Network essentially corrects the bias between the label embedding and the center of the class. The bias will be more severe if the support is less. With the increase in the number of supports, bias could be suppressed to some extent (see Figure 4). Some other methods, like label description (Hou et al., 2020), can also correct such kind of bias if enough supports are given. But when the supports are highly scarce, Shared-Private Network performs the best.

6.4 Analysis

We further visualize the relation between the performance with the similarity function and compare combined similarity with TVC in Figure 5. We firstly sample some combinations of source domains and train the model. Then we calculate their similarity with the target domain and record performance. From the left part of Figure 5, the performance generally positively correlates with TVC. However, its precision is poor, so that cannot be used as an indicator. Points around the green line have similar TVC scores, but the performance is quite diverse, i.e., the green points' are from 20% to 70%. A similar conclusion can be drawn from the horizontal direction: blue points around the blue line have identical performance, but their TVC scores are 36% to 87%. Therefore, data selection with TVC suffers severe performance fluctuation. By comparison, there is an apparent positive linear correlation between combined similarity and performance in the target domain (See the right part of Figure 5).

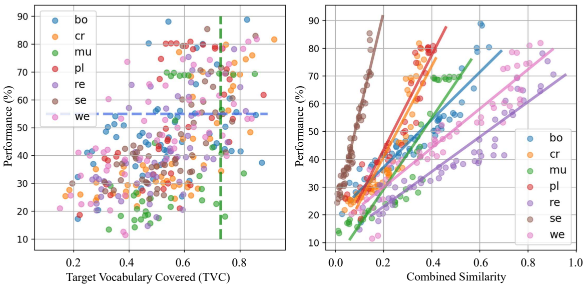

Figure 5: The relation between performance (y-axis) and the similarity function (x-axis). Different target domains are in different colors.

More analysis of (1) the comparison between the combination similarity function with its component TVC, TIS, and LO, and (2) inter-domain relations are shown in the appendix.

7 Conclusions and Future Work

In this paper, we prove the existence of negative knowledge transfer in few-shot learning and propose a similarity-based method to select pertinent data before training. We propose a Shared-Private Network (SP-Net) for the few-shot slot tagging task. We prove the effectiveness and advantages of both the data selection method and SP-Net with experiments. In the future, we will investigate the relations among domains and improve our data selection method to select episodes or samples rather than domains.

Acknowledgements

This work is supported by Innovation & Technology Commission HKSAR, under ITF Project No. PRP/054/21FX.

References

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

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling.

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

Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. 2019. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Haskell B Curry. 1944. The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics*, 2(3):258–261.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2019. Using similarity measures to select pretraining data for NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1460–1470, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Lin Gui, Ruifeng Xu, Qin Lu, Jiachen Du, and Yu Zhou. 2018. Negative transfer detection in transductive transfer learning. *International Journal of Machine Learning and Cybernetics*, 9(2):185–197.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

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.

Young-Bum Kim, Sungjin Lee, and Ruhi Sarikaya. 2017. Speaker-sensitive dual memory networks for multi-turn slot tagging. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 541–546. IEEE.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer.

Sara Meftah, Nasredine Semmar, Youssef Tamaazousti, Hassane Essafi, and Fatiha Sadat. 2021. On the hidden negative transfer in sequential transfer learning for domain adaptation from news to tweets. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 140–145, Kyiv, Ukraine. Association for Computational Linguistics.

Mansfield Merriman. 1877. *A List of Writings Relating to the Method of Least Squares: With Historical and Critical Notes*, volume 4. Academy.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2018. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Ruhi Sarikaya, Paul A Crook, Alex Marin, Minwoo Jeong, Jean-Philippe Robichaud, Asli Celikyilmaz, Young-Bum Kim, Alexandre Rochette, Omar Zia Khan, Xiaohu Liu, et al. 2016. An overview of end-to-end language understanding and dialog management for personal digital assistants. In *2016 ieee spoken language technology workshop (slt)*, pages 391–397. IEEE.

Darsh J Shah, Raghav Gupta, Amir A Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values.

Yangyang Shi, Kaisheng Yao, Hu Chen, Dong Yu, Yi-Cheng Pan, and Mei-Yuh Hwang. 2016. Recurrent support vector machines for slot tagging in spoken language understanding. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 393–399.

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

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Hongru Wang, Zezhong Wang, Gabriel Pui Cheong Fung, and Kam-Fai Wong. 2021. Mcml: A novel memory-based contrastive meta-learning method for few shot slot tagging.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019. Characterizing and avoiding negative transfer.

Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–37.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7115–7123. PMLR.

Su Zhu, Ruisheng Cao, Lu Chen, and Kai Yu. 2020. Vector projection network for few-shot slot tagging in natural language understanding.

X. Zhu, D. Anguelov, and D. Ramanan. 2014. Capturing long-tail distributions of object subcategories. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922.