Slot Transferability for Cross-domain Slot Filling

[1]Hengtong Lu, [1]Zhuoxin Han, [1]Caixia Yuan, [1]Xiaojie Wang,
[2]Shuyu Lei, [2]Huixing Jiang, [2]Wei wu
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan-Dianping Group, Beijing, China
{luhengtong, hanzhuoxin, yuancx, xjwang}@bupt.edu.cn
{leishuyu, jianghuixing, wuwei30}@meituan.com

Abstract

Cross-domain slot filling focuses on using labeled data from source domains to train a slot filling model for target domains. It is of great significance for transferring a dialogue system into new domains. Most of the existing work focused on building a cross-domain transfer model. From the perspective of slots themselves, this paper proposes a model-agnostic Slot Transferability Measure (STM) for evaluating the transferability from a source slot to a target slot, specifically, the degree that labeled data of the source slot is helpful to train the slot filling model for the target slot. We also give a STM-based method for a model to select helpful source slots and their labeled data for a given target slot. Experimental results on multiple existing models and datasets show that our method significantly outperforms state-of-the-art baselines in cross-domain slot filling. The code is available at https://github.com/luhengtong/STM-for-cdsf.git.

1 Introduction

As an important task in task-oriented dialog systems, slot filling aims to identify task-related slot information in user utterances. When a task (or domain) has a large amount of labeled data, most existing slot filling models can achieve desired performance. However, there is usually little or even no labeled data for a new task. How to train the slot filling model in the new task (target task) with the labeled data of one or more existing tasks (source tasks) is of great significance for the rapid expansion of the application of task-oriented dialog systems.

Existing work can be mainly classified into two categories. The first is to establish implicit semantic alignment between slot representations of the source task and the target task, the model trained with the source task data is directly used for the target task (Bapna et al., 2017; Lee and Jha, 2019; Shah et al., 2019). The second is to use a two-stage strategy (Liu et al., 2020), which treats all slot values as entities. First, it trains a generic entity recognition model using source task labeled data to identify all candidate slot values in the target task. Then, the candidate slot value is classified into the target task slot by comparing the similarity between its representation and the target task slot information.

Most of the existing work has focused on building cross-task transferable models that leverage the association information between source tasks and target tasks, and the model is always trained using the labeled data of all the source tasks without distinction. However, not all the source task data will have transferable value to the target task, and the value of different source tasks data to a particular target task may be different. For example, flight-ticket-reservation task and train-ticket-reservation task have high similarity so that the labeled data of the former will be helpful to the latter. While the flight-ticket-reservation task and the weather-inquiry task have great difference so that the labeled data of the former has no or only little value to the latter, and even has a negative effect on the target model. Furthermore, even though the source task is similar to the target task, not every source slot will be useful for all the slots of the target task. For example, the labeled data for leaving-time slot in flight-ticket-reservation task may be helpful for the slot filling of leaving-time in train-ticket-reservation task, but not useful for the train-type slot. Therefore, finding valuable source slots that can provide transferable information for slot filling in target slot and then training a model based on the labeled data of these slots can make better use of the data in source tasks. This is the starting point of this paper which is different from the existing work.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4970–4979
August 1–6, 2021. ©2021 Association for Computational Linguistics

In achieving this goal, we firstly propose slot transferability measure (STM) and give a method to calculate the STM. By comparing the STM between the target slot and each source slot, we can select different set of source slots for different target slot. Only the labeled data of these source slots are used to train the slot filling model for the target slot. To be more specific, we fuses distribution similarity of the slot value representations and of the slot value context representations between target slot and source slot as STM between two slots. All source slots are sorted according to their STMs with the target slot. Labeled data of the source slot with the highest STM are used to train the model, and then the labeled data of the source slot with the second highest STM is added to train the model. The process continues until the model gains no improvement on validation set of target slots. Those source slots and their labeled data are used to build the final slot filling model for the target slots.

Our main contributions are three-fold as follows.

1. We propose a metric called STM to measure the transferability between two slots. To our best knowledge, it is the first study on the transferability between two slots. The STM is model-agnostic.

2. We also propose a STM-based method to select source slots and their labeled data for training slot filling model for target slots.

3. Experimental results on several existing models and datasets show that this method brings consistent performance improvement for cross-domain slot filling.

2 Related work

As a key component of dialog system, the slot filling task has been studied extensively. Traditional supervised learning methods have made great achievements with a large amount of labeled data (Liu and Lane; Mesnil et al., 2015; Hakkani-Tür et al., 2016; Kurata et al., 2016; Liu and Lane, 2016; Goo et al., 2018; E et al., 2019). However, there is little or even no labeled data for a new task, the cross-domain slot filling task which uses labeled data in source tasks to training model for target task is gaining increasing attention (Yazdani and Henderson, 2015; Bapna et al., 2017; Zhu and Yu, 2018; Lee and Jha, 2019; Shah et al., 2019; Liu et al., 2020; Zhu et al., 2020). There are mainly two streams of methods in previous work.

The first is to establish implicit semantic alignment of the slot representations between the source task and the target task (Bapna et al., 2017; Lee and Jha, 2019; Shah et al., 2019; Liu et al., 2020). Bapna et al. (2017) proposed the Concept Tagging model (CT), which unified the slot filling model on the source tasks and the target task by combining the slot representations modeled by slot description information, and then conducting BIO 3-way classification. Based on CT, Lee and Jha (2019) proposed the Zero-Shot Adaptive Transfer model (ZAT), which introduced an attention layer in building representations of slot description; Meanwhile, Shah et al. (2019) proposed the Robust Zero-shot Tagger (RZT) model, which used a small number of sample slot values of the target slot to constrain the slot filler to avoid the negative transfer caused by the misalignment of slot names.

The second is a coarse-to-fine approach, which first identifies all candidates of slots and then classifies them into corresponding slots. Liu et al. (2020) proposed a Coarse-to-fine approach (Coach). They first predicted whether the tokens are slot value candidates, and then identified their specific slot types based on the similarity between the tokens and the representation of each slot description. In addition, Coach utilized a template regularization method which clusters the representations of semantically similar utterance into a similar vector space. It greatly improves the robustness of the model.

Most of these efforts focus on building a cross-task transferable model by exploiting the correlation information between source and target tasks. All source data is used to train the transfer model no matter if the data is helpful for target slot filling. On the contrast, this paper proposes a new method to select parts of source slots and their labeled data for model training.

3 Methodology

This section describes the cross-domain slot filling method proposed in this paper. First, we propose the concept of slot transferability and its measurement STM in Section 3.1. Then we describe the method of finding source slots for target slot based on the STM in Section 3.2. Finally, we introduce how this method can be deployed and implemented on existing models in Section 3.3. The STM is model-agnostic and will be validated on multiple existing models.

3.1 Slot Transferability Measure

Given slots s_a and s_b, the transferability from s_a to s_b refers to the degree of that the slot filling information of s_a can be used for slot filling of s_b, denoted as $\mathrm{STM}(s_a, s_b)$.

Let $p_v(s_i)$ be the distribution of slot value representation of slot $s_i(i = \{a, b\})$, $p_c(s_i)$ be the distribution of slot value context representation of slot $s_i(i = \{a, b\})$. We define the transferability from slot s_a to slot s_b as Equation (1):

$$\mathrm{STM}_\beta(s_a, s_b) = 1 - \tanh\left(\frac{(1+\beta^2) sim(p_v(s_a), p_v(s_b)) sim(p_c(s_a), p_c(s_b))}{\beta^2 sim(p_v(s_a), p_v(s_b)) + sim(p_c(s_a), p_c(s_b))}\right) \tag{1}$$

where $sim(p, q)$ denotes the similarity between distribution p and q. The β parameter determines the weight of similarity between distributions of slot value context representations. $\beta > 1$ favors similarity between distributions of slot value context representations, $\beta < 1$ lends more weight to similarity between distributions of slot value representations. The larger the $\mathrm{STM}_\beta(s_a, s_b)$, the higher the transferability from slot s_a to slot s_b.

Maximum Mean Discrepancy (MMD) is employed to calculate $sim(p, q)$. MMD is usually used as a loss function in transfer learning (Tzeng et al., 2014; Zhang et al., 2015; Long et al., 2015, 2016, 2017; Yan et al., 2017). It minimizes the difference between different domains to obtain the domain-invariant features. It serves as test statistics to determine if two distributions are the same, as well as measure the similarity between two distributions. The smaller the MMD is, the higher the similarity between distributions is. Let $\mathcal{F}$ be a class of functions $\mathcal{F} : X \to R$. Let $X = \{x_1, ..., x_m\}$ and $Y = \{y_1, ..., y_n\}$ be samples composed of independent and identically distributed observations drawn from distribution p and q, respectively. MMD is defined as Equation (2), and the square of the MMD can be empirically estimated by Equation (3) (Borgwardt et al., 2006):

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} (E_p[f(x)] - E_p[f(y)]) \tag{2}$$

$$\mathrm{MMD}^2[\mathcal{F}, X, Y] = \frac{1}{m(m-1)} \sum_{i \neq j}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i,j=1}^{m,n} k(x_i, y_j) \tag{3}$$

where k is the kernel function. Gaussian kernel functions are usually used as. Therefore, the similarity between the distributions of slot value representations of slots s_a and s_b, and the similarity between distributions of slot value context representation of slots s_a and s_b are Equations (4) and (5) respectively:

$$sim(p_v(s_a), p_v(s_b)) = \mathrm{MMD}^2[\mathcal{F}, \Omega_{va}, \Omega_{vb}] \tag{4}$$

$$sim(p_c(s_a), p_c(s_b)) = \mathrm{MMD}^2[\mathcal{F}, \Omega_{ca}, \Omega_{cb}] \tag{5}$$

where Ω_{vi} and Ω_{ci} is the sample set of the slot values representation distribution and the sample set of the slot value context representation distribution corresponding to slots $s_i(i = \{a, b\})$.

Given labeled data $D_{sa} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_a}$ for slot s_a , where $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \cdots, x_l^{(i)})$ is a sequence of words, $y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \cdots, y_l^{(i)})$ is the corresponding label sequence. Since $x^{(i)}$ contains the slot value of slot s_a, there is either "B-s_a" (the slot value is a word) or "B-s_a" and "I-s_a" (the slot value includes several words) in the label sequence. We first extract all slot value words from labeled dataset D_{sa}, and have the sample set Ω_{va} of the slot value representation of slot s_a, as shown in Equation (6):

$$\Omega_{va} = \left\{ E(x_j^{(i)}) \mid if\ I_{va}(x_j^{(i)}, s_a) \right\}_{i=1,j=1}^{i=N_a,j=l} \tag{6}$$

where E is a word embedding mapping, and I_{va} indicates whether $x_j^{(i)}$ is the slot value of slot s_a defined as Equation (7):

$$I_{va}(x_j^{(i)}, s_a) = \begin{cases} 1, & if\ y_j^{(i)} = B\text{-}s_a \\ & or\ y_j^{(i)} = I\text{-}s_a \\ 0, & otherwise \end{cases} \tag{7}$$

Then we extract the slot values context. N words before and after the slot values are extracted to form the sample set Ω_{ca} as shown in Equation (8):

$$\Omega_{ca} = \left\{ E(x_k^{(i)}) \mid if\ I_{ca}\ (x_k^{(i)}, s_a) \right\}_{i=1,k=1}^{i=N_a,k=l} \tag{8}$$

where E is a word embedding mapping, and I_{ca} indicates whether $x_k^{(i)}$ is the slot value context for slot s_a defined as Equation (9):

$$I_{ca}(x_k^{(i)}, s_a) = \begin{cases} 1,\ if\ x_k^{(i)} \neq B\text{-}s_a\ and\ x_k^{(i)} \neq I\text{-}s_a \\ \quad and\ (y_{k+N}^{(i)} = B\text{-}s_a\ or\ y_{k-N}^{(i)} = I\text{-}s_a \\ \quad or\ y_{k-N}^{(i)} = B\text{-}s_a) \\ 0,\ otherwise \end{cases} \tag{9}$$

Similarly, we can obtain Ω_{vb} and Ω_{cb}. Based on Ω_{va}, Ω_{ca}, Ω_{vb} and Ω_{cb}, we can calculate

$sim(p_v(s_a), p_v(s_b))$ and $sim(p_c(s_a), p_c(s_b))$ based on Equations (4) and (5) and then calculate $\text{STM}_\beta(s_a, s_b)$ based on Equation (1).

Slot transferability has the following two properties:

Symmetry STM is symmetric. Let the transferability from s_b to s_a be $\text{STM}_\beta(s_b, s_a)$, we have:

$$\text{STM}_\beta(s_a, s_b) = \text{STM}_\beta(s_b, s_a) \qquad (10)$$

Relativity When comparing the STM between two slot pairs, it is meaningful only their source slot or target slot is the same. When $\text{STM}_\beta(s_a, s_b) < \text{STM}_\beta(s_c, s_b)$, the transferability from s_a to s_b is higher than that from s_c to s_b, or when $\text{STM}_\beta(s_a, s_b) < \text{STM}_\beta(s_a, s_c)$, the transferability from s_a to s_b is higher than that from s_a to s_c. The comparison between $\text{STM}_\beta(s_a, s_b)$ and $\text{STM}_\beta(s_c, s_d)$ is meaningless.

3.2 Selection of source slots based on STM

Given a slot set $S = \{s_1, \cdots, s_{ns}\}$ from source tasks and the target slot s_t. Each source slot s_a has a labeled dataset $D_{sa}^T = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_{sa}^T}$, and the target slot s_t is given a labeled dataset $D_{st}^V = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N_{st}^V}$ for validation. We select a slot set S_t for training the target slot filling model from S basing on the following steps.

1. For each slot s_i in the source slot set, we calculate the transferability $\text{STM}_\beta(s_i, s_t)$.

2. After sorting $[s_1, \cdots, s_{ns}]$ from big to small basing on $\text{STM}_\beta(s_i, s_t)$, we get $[s_{(1)}, \cdots, s_{(n)}]$, the slots sequence according to the order of transferability from highest to lowest.

3. First, we select a slot filling model M and define the union of training data corresponding to the first h slots (h is initialized to 1) in the sorted list $[s_{(1)}, \cdots, s_{(ns)}]$ as $D_h^{ACC} = D_{(1)}^T \cup D_{(2)}^T \cup \cdots \cup D_{(h)}^T$. After replacing the B-$*$ and I-$*$ tags on D_h^{ACC} and D_d^V with the labels B and I, we train the model M with D_h^{ACC} and then test the trained model on D_{st}^V to get the corresponding F1 value, which is denoted as $f_h = M(D_h^{ACC}, D_{st}^V)$. Then $h = h + 1$, till F1 gets to its maximum, then $S_t = [s_{(1)}, \cdots, s_{(h)}]$.

3.3 Model training

Given a set of source tasks $T = \{T_1, \cdots, T_n\}$, a target task T_{tgt} and the slot set $S_i = \{s_1, \cdots, s_{N_i}\}$ corresponding to task T_i. We define the set of all source task slots as $S_{union} = S_1 \cup \cdots \cup S_n$, and the set of target task slots as S_{tgt}. For an existing cross-domain slot filling model M_{base}, we deploy our approach on the model by following these steps.

Firstly, the training set and validation set corresponding to the target task and the source task are divided into the training set and validation set corresponding to each slot according to whether the slots contained in each sample. Then select the corresponding source slot set S_{ti} for each slot s_{ti} in the target task slot set from all the source tasks set S_{union}. We combine the source slot set corresponding to all target task slots to get the source slot set for the target task $S_{tgt} = S_{t1} \cup \cdots \cup S_{tN_{tgt}}$ and then replace the labels corresponding to all slots in the source task training set that are not in S_{tgt} with labels O. Finally, the source slot set S_{tgt} and the training data after replacement are used to train the model M_{base}.

4 Experiments

In this section we describe the dataset used for evaluation, the baseline models used for comparison, and more details of the experimental settings.

4.1 Datasets

To evaluate the effectiveness of our approach, we conduct experiments on SNIPS (Coucke et al., 2018). In order to further evaluate the generalization ability of our approach, we also construct a cross-task slot filling dataset called MultiWoz-Slot (MWS) based on the multi-domain task-oriented dialog dataset MultiWoz (Budzianowski et al., 2018; Eric et al., 2020). Table 1 displays some statistics about the two datasets. Details about the two datasets and how the MWS dataset is constructed are described as follows.

SNIPS SNIPS is a public SLU dataset that contains 7 tasks (intents) and 39 slots, and each task contains approximately 2000 training samples. As shown in Table 1, the data contains a total of 14,484 samples, the vocabulary size is 12,134, the average length of the sample utterance is 9, and the average number of slots in each sample is 2.6.

Multiwoz-Slot MultiWoz is a public multidomain task-oriented dialogue dataset that contains 7

MWS	vocab size	utters num	avg len	avg slot num	slot num	slots
attraction[1]	1132	4616	15.3	1.2	3	$area^{123}$, $arrive^{45}$, day^{235}, $depart^{45}$, $dest^{45}$, $food^{3}$, $leave^{45}$, $name^{123}$, $people^{235}$, $price^{23}$, $stars^{2}$, $stay^{2}$, $time^{3}$, $type^{12}$
hotel[2]	1481	11258	16	1.7	8	
restaurant[3]	1734	11669	14.4	1.8	7	
taxi[4]	961	1758	15.7	1.3	4	
train[5]	1391	10538	13.9	1.7	6	
total	3314	39839	14.9	1.6	14	

SNIPS	vocab size	utters num	avg len	avg slot num	slot num	slots
AddToPlaylist[1]	3261	2042	9.2	2.7	5	$album^{4}$, $artist^{14}$, $best\_rating^{5}$, $city^{23}$, $country^{23}$, $condition\_description^{3}$, $condition\_temperature^{3}$, $cuisine^{2}$, $current\_location^{3}$, $entity\_name^{1}$, $year^{4}$, $geographic\_poi^{3}$, $location\_name^{7}$, $playlist\_owner^{1}$, $object\_type^{567}$, $party\_size\_description^{2}$, $rating\_unit^{5}$, $movie\_type^{7}$, $served\_dish^{2}$, $service^{4}$, $sort^{24}$, $genre^{4}$, $music\_item^{14}$, $object\_location\_type^{7}$, $object\_name^{56}$, $object\_part\_of\_series\_type^{5}$, $object\_select^{5}$, $facility^{2}$, $party\_size\_number^{2}$, $playlist^{14}$, $movie\_name^{7}$, poi^{2}, $rating\_value^{5}$, $restaurant\_name^{2}$, $restaurant\_type^{2}$, $state^{23}$, $timeRange^{237}$, $track^{4}$, $spatial\_relation^{237}$
BookRestaurant[2]	2639	2073	12	3.2	14	
GetWeather[3]	2260	2100	9.5	2.3	9	
PlayMusic[4]	2961	2100	7.1	2.2	9	
RateBook[5]	1906	2056	8.8	3.8	7	
SearchCreativeWork[6]	3222	2054	7.8	1.7	2	
SearchScreeningEvent[7]	1718	2059	8.7	2.2	7	
total	12134	14484	9	2.6	39	

Table 1: Some statistics about SNIPS and MWS. The upper script on task indicates the task id. The upper script on slot indicates the task it belongs.

tasks and 24 slots. Since the (hospital, police) tasks have little conversation data and only appear in the training data, we use user-side utterance for just five tasks (attractions, hotel, restaurant, taxi, train) to construct the MWS dataset, which contains 14 slots. When constructing the training, validation and test data of a task in MWS, we extract the user-side utterance containing the task separately from the conversations in the training set, validation set and test set of Multiwoz. Since the training set of the target task is generally used as the final test set in the cross-domain slot filling task, we combine the validation set and test set as validation set for each task. Table 1 shows the number of slots and the number of sample of training set and validation set included in each task in MWS. As shown in Table 1, the data contains a total of 39,839 samples, and the vocabulary size is 3,314, the average length of the sample utterance is 15, and the average number of slots in per sample is 1.7.

Compared to SNIPS, MWS has smaller vocabulary size and the number of slots in each task. However, MWS has more samples in each task, so when it is used as a cross-domain slot filling dataset, its source tasks have more sufficient training samples, and the correlation between these tasks is stronger.

4.2 Models

We conduct our experiments on the following models.

Concept Tagger (CT) A cross-domain slot filling model proposed by Bapna et al. (2017), which uses the information of the slot descriptions to establish implicit alignment between target slots and source slots.

Robust Zero-shot Tagger (RZT) A model proposed by Shah et al. (2019), which uses the slot value sample of slots to improve the robustness of the model on the target task based on the CT model.

Coarse-to-fine Approach (Coach) A two-stage cross-domain slot filling method proposed by Liu et al. (2020), which splits the cross-domain slot filling task into two stages: coarse-grained BIO 3-way classification and fine-grained slot type classification, and uses slot descriptions in the second stage to help recognize unseen slots.

Coach+TR A variant of Coach proposed by Liu et al. (2020), which further uses template regularization on the basis of Coach to improve the performance of the model on similar or the same slots, is the state-of-the-art model.

4.3 Implementation Details

We deploy the proposed method on above slot filling models CT, RZT, Coach, and Coach+TR.

β is set to 1. A two-layer BiLSTM (Schmidhuber and Hochreiter, 1997) model is used for selecting source slots for all models. 300 dimensions Glove (Pennington et al., 2014) vector is used for word embedding. The hidden layer dimension is set to 300, the learning rate is 0.001. We train the model 30 epochs and select the model with the best performance on the validation set as the final model.

Model		CT		RZT		Coach		Coach+TR	
Data/Domain ↓ Training Setting →		ALL	STM_1	ALL	STM_1	ALL	STM_1	ALL	STM_1
MWS	attraction	74.52	**84.99**	74.96	**84.22**	67.26	**73.35**	65.06	**76.16**
	hotel	58.81	47.73	50.63	43.82	59.03	**61.12**	59.00	**60.74**
	restaurant	69.93	63.47	66.29	61.01	78.65	65.36	79.00	71.53
	taxi	51.61	**69.32**	52.82	**66.25**	63.88	**81.17**	70.04	**79.34**
	train	80.78	79.66	80.02	**81.37**	77.68	**82.85**	77.91	**85.23**
	Average F1	67.13	**69.03**	64.94	**67.33**	69.30	**72.77**	70.20	**74.60**
SNIPS	AddToPlaylist	38.82	**41.95**	42.77	**42.92**	45.23	**53.36**	50.90	50.54
	BookRestaurant	27.54	**31.17**	30.68	30.21	33.45	32.60	34.01	32.89
	GetWeather	46.45	**53.03**	50.28	**62.32**	47.93	**60.91**	50.47	**62.38**
	PlayMusic	32.86	23.09	33.12	22.33	28.89	**35.60**	32.01	**34.45**
	RateBook	14.54	**15.39**	16.43	**25.37**	25.67	16.37	22.06	**25.39**
	SearchCreativeWork	39.79	38.72	44.45	42.63	43.91	**49.88**	46.65	**52.21**
	SearchScreeningEvent	13.83	**14.13**	12.25	**15.15**	25.64	23.75	25.63	**26.05**
	Average F1	30.55	**31.07**	32.85	**34.42**	35.82	**38.92**	37.39	**40.56**

Table 2: The main result of the four models (CT, RZT, Coach, Coach +TR) trained using original data (All data) and data selected by our method (STM_1). Scores in each row are F1 of target task.

In order to make a fair comparison, we use the same settings with Liu et al. (2020) to construct the cross-domain slot filling model. We concatenate the 100-dimensional character-level representation and the 300-dimensional word-level representation as word representation. We set the hidden layer dimension of all the BiLSTM encoders to 300 and set the dropout rate to 0.3. We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0005. The samples of each task in SNIPS are divided into two parts: 500 samples as validation set, and the remaining samples as training set. When a task is set as a target task, its training set is used as a test set. We evaluate on two datasets respectively. For each test, we choose one task as the target task and set the other tasks as the source tasks.

5 Result and Discussion

In this section, we describe and analyze the experimental results. Firstly, the main results of the experiment are described in Section 5.1. Then, we analyze the impact of some factors on STM in Section 5.2.

5.1 Main Results

Quantitative Analysis Table 2 shows the main result of the four models. For each model, the first column is F1 of the model trained by all labeled data available, i.e., the original way for using the model. The second column is F1 of the model trained by labeled data selected by method proposed in the paper. As can be seen from the Table 2:

1. Our method improves the average F1 (average on all target tasks) of all four models on two datasets consistently, e.g. our method improves the average F1 of coach+TR by 4.4 points on MWS, and by 3.17 points on SNIPS.

2. Our method improves the performance of all four models on most target tasks, even improves several of them by more than 10 points.

Qualitative analysis We perform a qualitative analysis of the STM on the MWS dataset. Figure 1 shows the thermal diagram of slot transferability between any two slots in MWS. Each cell in the figure represents the value of STM_1 between the slot labeled in the horizontal axis and the vertical axis. The higher the brightness, the higher the transferability. The figure is symmetric because the STM_1 is symmetric. It can be found that the slot with high transferability to each other are roughly divided into 7 categories, as shown in Table 3. After observing the data, we find that there are mainly three kinds of slots in the same category. The first type is the slot with high coincidence degree of slot value set. For example, ”attraction-name” and ”taxi-dest” have some common values, such as ”adc theatre”, ”all saints church”, ”county folk museum” and so on. The second kind of slots is that the slot values appear in similar context. For example, ”attraction-name” and ”hotel-name” have some common context words, such as ”about”, ”for”, ”at” and so on. The third kind of slots are the slots with higher coincidence degree, as well as similar context of slot values, such as ”attraction-area” and ”hotel-area”. These phenomena are consistent with the definition of STM.

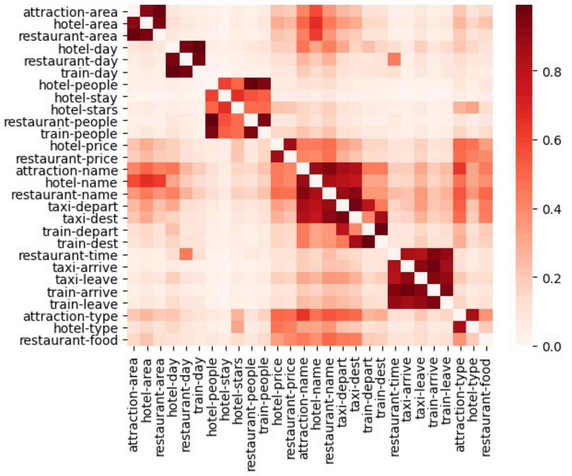

Figure 1: the thermal diagram of slot transferability between any two slots in MWS.

Category	Slots
1	*attraction-area*, *hotel-area*, *restaurant-area*
2	*hotel-day*, *restaurant-day*, *train-day*
3	*hotel-people*, *hotel-stars*, *hotel-stay*, *restaurant-people*, *train-people*
4	*hotel-price*, *restaurant-price*
5	*attraction-name*, *hotel-name*, *restaurant-name*, *taxi-depart*, *taxi-dest*, *train-depart*, *train-dest*
6	*restaurant-time*, *taxi-arrive*, *taxi-leave*, *train-arrive*, *train-leave*
7	*attraction-type*, *hotel-type*, *restaurant-food*

Table 3: The MWS slots categories. The slots in the same categories have the high transferability.

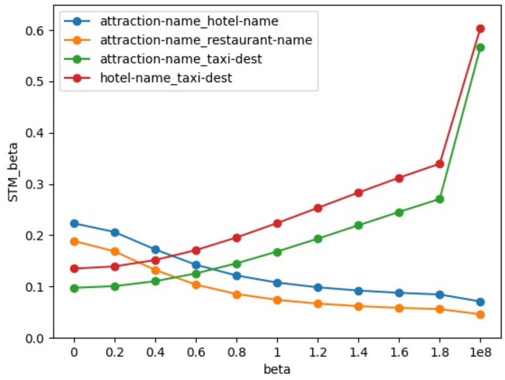

Figure 2: The impact of β on STM.

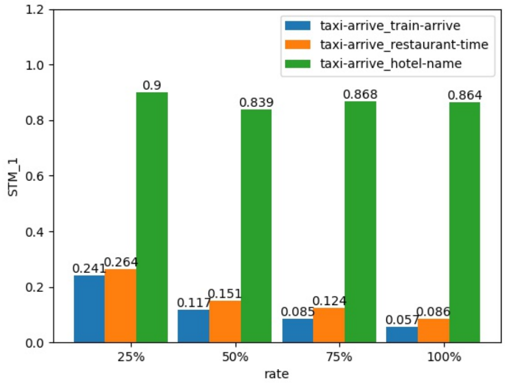

Figure 3: The impact of sample number on STM.

5.2 The impact of some factors on STM

There are three main factors in the calculation of STM_{beta}. The following is an experimental analysis of the impact of the three factors on STM.

The impact of β on STM β parameter determines the weight of similarity between distributions of slot value context representations . We randomly select four slot pairs which are (attraction-name, hotel-name), (attraction-name, restaurant-name), (attraction-name, taxi-dest), and (hotel-name, taxi-dest). In the first two groups, the slot values appear in similar context, but the sets of slot values almost have no intersection. In the last two groups, the sets of slot values set have high consistency, but the contexts of slot values are not similar. β is range from 0 and $+\infty$. When $\beta = 0$, STM_β only measures the similarity of distributions between slot values representations of the two slots, and when $\beta = +\infty$, STM_β only measures the similarity of distributions between the slot value context representation of the two slots. As shown in Figure 2, the STM_β of the first two groups increased with the increase of β, while the STM_β of the last two groups decreased with the increase of β. Therefore, when β increases, the effect of slot value similarity on STM_β becomes greater, and the effect of slot value context similarity on STM_β becomes smaller.

The impact of sample number on STM In order to measure the impact of sample number on STM_β, we randomly selected three slot pairs which are (taxi-arrive, train-arrive), (taxi-arrive, restaurant-time) and (taxi-arrive, hotel-name) for comparative experiment. We select 25%, 50%, 75% and 100% samples from the validation set used to calculate STM_β on the three groups of slots. The experimental results are shown in Figure 3. According to the figure, although the absolute values of STM_β of the three slot pairs changed, their relative relations didn't change. That is, sam-

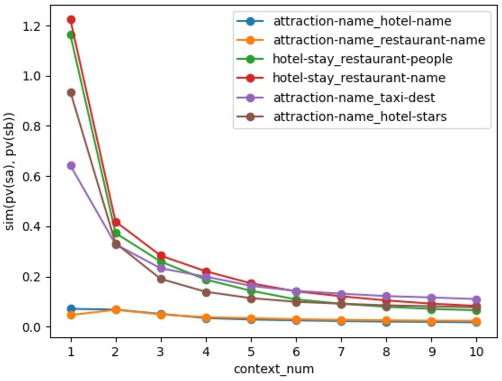

Figure 4: The impact of N on $sim(p_c(s_a), p_c(s_b))$.

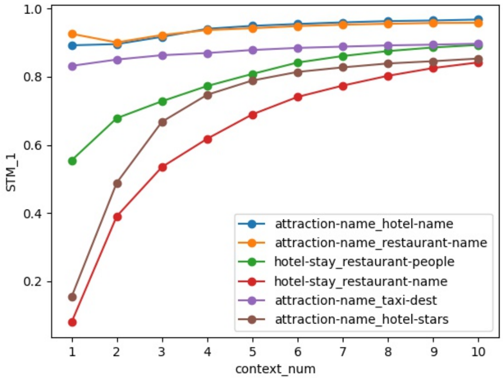

Figure 5: The impact of N on STM.

ple size will affect the value of STM_β. However, for the same source slot, the relationship between STM_β to different target slots does not change.

The impact of N on STM When calculating slot transferability, we fuse distribution similarity of the slot value representations and of the slot value context representations. We select one word before and after slot value as slot value context. In order to measure the impact of slot context window size N on slot transferability, (attraction-name, hotel-name), (attraction-name, restaurant-name), (attraction-name, taxi-dest), (attraction-name, hotel-stars), (hotel-stay, restaurant-people) and (hotel-stay, restaurant-name) are randomly selected for comparing. The first two groups have similar context, the middle two have similar slot values sets, and the last two have low similarity in both slot values and context. N is range from 1 and 10. We observe the change of context representation distribution similarity $sim(p_c(s_a), p_c(s_b))$ and STM_1 among 6 groups of slots. As shown in Figure 4 and Figure 5, the similarity among context representation distributions increases with the increase of context window size N, and the context similarity among 6 groups of slots tends to be the same. In addition, STM increases with the increase of window size N, and the distinction of STM between different types of slot pairs decreases. We conjecture this is due to the fact that the context we extracted contains too much slot-independent context when the window size N becomes large.

5.3 Running time analysis

The method proposed in Sec 3.2 does increase the running time. However, there are two sides of running time. Selecting slots by a Bi-LSTM cost some times, while training the model with selected (less) data saves times. We don't calculate the save minutes in training, however, we find the increase of time consumption in slots selection is small, it is acceptable considering the performance improvements it brings. Since the process in Sec 3.2 is offline and once for a new domain, and the model used in Sec 3.2 is a simple Bi-LSTM model, it increases only a little time. To be more detailed, we conducted experiments on one Titan V GPU, the average running time of the method in Sec 3.2 is 80 minutes for a new domain.

6 Conclusions and Future Work

In this paper, we propose a metric STM to measure the slot transferability of the slots across task, and the calculation of this metric is model-agnostic. Based on this metric, we also propose a cross-domain slot filling method to improve the performance of the existing models by selecting the source slots with high transferability for the target slots. The results on several existing models and datasets show that our method can bring consistent performance improvement to the slot filling models of the target tasks, which show the effectiveness of the STM. We also further explore the impact of some factors on STM. In the future, we hope to use STM to further guide the improvements of models.

Acknowledgments

We would like to thank anonymous reviewers for their suggestions and comments. The work was supported by the National Natural Science Foundation of China (NSFC62076032).

References

Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. *Proc. Interspeech 2017*, pages 2476–2480.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

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. *CoRR*, abs/1805.10190.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471, Florence, Italy. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 422–428.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun Nung Chen, and Ye Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *The 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling.

Sungjin Lee and Rahul Jha. 2019. Zero-shot adaptive transfer for conversational language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6642–6649.

Bing Liu and Ian Lane. Recurrent neural network structured output prediction for spoken language understanding.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 19–25, Online. Association for Computational Linguistics.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2016. Unsupervised domain adaptation with residual transfer networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 136–144.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR.

Gregoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, and Dong and Yu. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio Speech Language Processing*, 23(3):530–539.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jürgen Schmidhuber and Sepp Hochreiter. 1997. Long short-term memory. *Neural Comput*, 9(8):1735–1780.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5484–5490, Florence, Italy. Association for Computational Linguistics.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474.

Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. 2017. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2272–2281.

Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 244–249, Lisbon, Portugal. Association for Computational Linguistics.

Xu Zhang, Felix Xinnan Yu, Shih-Fu Chang, and Shengjin Wang. 2015. Deep transfer network: Unsupervised domain adaptation. *arXiv preprint arXiv:1503.00591*.

Su Zhu and Kai Yu. 2018. Concept transfer learning for adaptive language understanding. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 391–399.

Su Zhu, Zijian Zhao, Rao Ma, and Kai Yu. 2020. Prior knowledge driven label embedding for slot filling in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1440–1451.