# Cross-domain Slot Filling with Distinct Slot Entity and Type Prediction

Shudong Liu, Peijie Huang(✉), Zhanbiao Zhu, Hualin Zhang, and Jianying Tan

College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China
{sudan,zhuzhanbiao,vol_chris,jytan}@stu.scau.edu.cn, pjhuang@scau.edu.cn

**Abstract.** Supervised learning approaches have been proven effective in slot filling, but they need massive labeled training data which is expensive and time-consuming in a given domain. Recent models for cross-domain slot filling adopt transfer learning framework to cope with the data scarcity problem. However, these cross-domain slot filling models rely on the same encoder representation in different stages for slot entity task and slot type task, which decrease the performance of both tasks. Besides, these models treat different source domains equally and ignore the shared slot-related information in different domains, which may damage the performance of cross-domain learning. In this paper, we present a pipeline approach for cross-domain slot filling (**PCD**) by learning distinct contextual representations for slot entity identification and slot type alignment, and fusing slot entity information at the input layer of the slot type alignment model for incorporating global context. Moreover, we also present a simple yet effective instance weighting scheme (**Iw**) to our approach for better capturing the slot entities in the cross-domain setting. Experiments on multiple domains show that our approach achieves state-of-the-art performance in cross-domain slot filling. Ablation analysis and further experiments also prove the effectiveness of each part of our model, especially in the identification of slot entities.

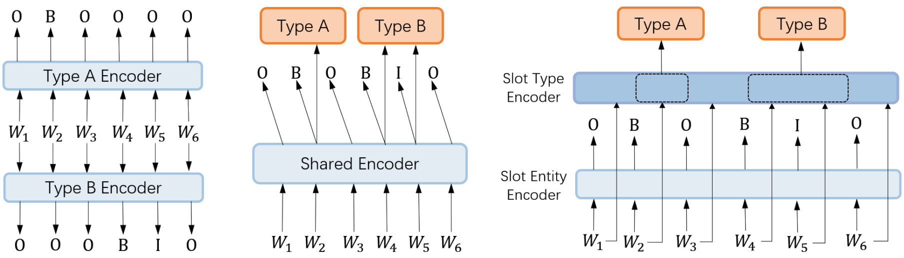
**Keywords:** Spoken language understanding · Slot filling · Cross-domain learning · Instance weighting scheme · Zero-shot learning

## 1 Introduction

Spoken language understanding (SLU) is the core component of intelligent personal digital assistants (IPDAs) such as Microsoft Cortana, Google Assistant, Amazon Alexa, and Apple Siri [1]. It typically consists of intent detection and slot filling. Slot filling models capture useful semantic information which has been shown helpful for related NLP tasks.

Recently, supervised joint learning approaches have shown their effectiveness in slot filling [2–5]. Such joint models for intent detection and slot tagging have taken the state of the art of slot filling to a new level. However, such approaches are expensive and time-consuming due to the difficulties in


L. Wang et al. (Eds.): NLPCC 2021, LNAI 13028, pp. 517–528, 2021.
https://doi.org/10.1007/978-3-030-88480-2_41



(a) CT model [11] (b) Coach model [14] (c) Our pipeline framework

**Fig. 1.** Cross-domain slot filling frameworks.

collecting high-quality labeled training data with different domains. This limitation has motivated us to explore cross-domain slot filling for fast adaptation to new domains. Cross-domain adaptation copes with the data scarcity problem in low-resource target domains [6–10]. The key challenge of slot filling in a new domain is identifying unseen slot types without any supervision signals. Common approaches for cross-domain slot filling are focusing on employing slot description (e.g., the description of slot label *restaurant_type* is "restaurant type") to predict unseen slots [11–15].

Existing cross-domain slot filling models can be classified into two main categories. As shown in Fig. 1 (a), the first part of work, such as the CT model [11], conducts slot filling individually for each slot type [12]. They generate word-level representations, then interact with the representation of each slot type description in semantic space. The final predictions are independent for each slot type based on the fused features. Besides, slot examples were also used to increase the robustness of domain adaptation [13]. However, such models exist a multiple prediction problem. Unlike the above models, as shown in Fig. 1 (b), Liu et al. [14] proposed a two-stage slot filling framework to avoid the multiple prediction problem and learn the general pattern of slot entities. They use the shared representation to identify whether the tokens are slot entities or not by a BIO (Begin/In/Out) 3-way classifier, and then predict their specific slot types based on slot descriptions. For example, given a movie-related utterance "*find andreas hofer at elevenses*", the model will first capture the slot entity "*andreas hofer*" and then classify its label as "*movie_name*". He et al. [15] further leveraged contrastive learning and adversarial attack to improve model robustness.

Though achieving the promising performance, these two-stage models still suffer from two issues: (1) They use the same encoder for identifying slot entities (by using BIO structure) in stage one and predicting the specific slot types for each entity in stage two. However, the information captured in cross-domain learning in slot entity identification and slot type alignment is different. The slot entity identification is to detect the entity boundary while the slot type alignment is to predict slot labels by contexts. The performance of these two-stage models drops in both tasks since affecting each other. (2) Such approaches treat each source domain corpus equally. However, in cross-domain learning, different source domains have different contributions to the target domain, and some of them may even cause negative transfer problems [16]. For example, given target domain "GetWeather", the model can get more improvements from "BookRestaurant" domain because of the location-related shared slots, but fewer improvements from the "PlayMusic" domain with no related shared slots at all.

To meet the above challenges, in this work, we propose a pipeline approach for cross-domain slot filling by learning distinct contextual representations for slot entities and slot types. The overall principle can be seen in Fig. 1 (c). It learns two independent encoders for the slot entity model and slot type model. To capture the entity information from the entity model in slot types prediction, we add boundary markers into the second encoder. In addition, we introduce an instance weighting scheme to control the contribution of different source domains to the target domain. The core idea is to compute the similarities between domains, which are used to adjust learning rates for the utterances of different domains. By doing so, the model tends to learn more shared-information in more similar domains, rather than in less similar domains.

Our main contributions are summarized as follows: (1) We propose a pipeline approach for cross-domain slot filling with distinct contextual representations for slot entities and slot types. (2) We introduce a simple yet effective instance weighting scheme for better capturing slot entities and alleviating the negative transfer problem. (3) Experiments in the zero-shot/few-shot settings on SNIPS and SMP-ECDT datasets show that our approach outperforms the state-of-the-art models. Ablation study and quantitative analysis also prove the effectiveness of the proposed model.

## 2 Our Approach

Figure 2 illustrates our pipeline model architecture by a sample user utterance "*find andreas hofer at elevenses*" and its corresponding slots. The pipeline model consists of a slot entity and a slot type model. The slot entity model predicts whether tokens are slot entities or not (BIO labels) and learns the slot entity pattern with the instance weighting scheme. The slot type model classifies the slot entities into related types with slot descriptions [11] and boundary markers.

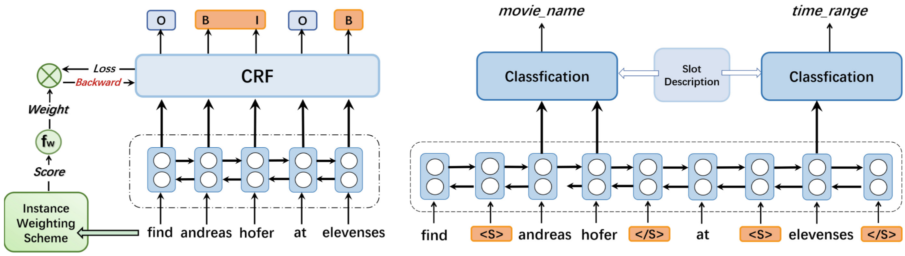
### 2.1 Slot Entity Model

Following prior work, we utilize the BiLSTM-CRF structure [17] to encode the hidden states of tokens and predict the BIO labels. The input of the model is an utterance consisting of $n$ tokens denoted as $W = [w_1, ..., w_n]$. Let $E$ be the embedding layer for utterances. We formulate the whole process as follows:

$$[\boldsymbol{h_1}, \boldsymbol{h_2}, ..., \boldsymbol{h_n}] = BiLSTM(E(W)) \tag{1}$$

$$[\boldsymbol{p_1}, \boldsymbol{p_2}, ..., \boldsymbol{p_n}] = CRF([\boldsymbol{h_1}, \boldsymbol{h_2}, ..., \boldsymbol{h_n}]) \tag{2}$$

where $[\boldsymbol{h_1}, \boldsymbol{h_2}, ..., \boldsymbol{h_n}]$ is the hidden layer and $[\boldsymbol{p_1}, \boldsymbol{p_2}, ..., \boldsymbol{p_n}]$ is the logits for the 3-way classification.




(a) Slot entity model (b) Slot type model

**Fig. 2.** The core architecture of our proposed pipeline model (PCD-Iw). Fig (a) displays the slot entity model with instance weighting scheme to identify whether the tokens are slot entities or not. Fig (b) shows the slot type model with boundary markers to match the specific slot types based on slot type descriptions. The boundary markers can be generated from the prediction results of the slot entity model.

## 2.2 Slot Type Model

The slot type model aims to classify the type of the slot entities predicted by the slot entity model. Prior work [14,15] used the same encoder since it has captured the information about which parts are the entities to focus on in predicting slot types. However, due to the different granularity of the information to be captured by the two tasks in cross-domain setting, using the shared representation directly will damage the performance of the model. Hence, we build a new model for classifying slot types.

To capture the entity boundaries and highlight the slot entities, inspired by Zhong and Chen [18], we insert boundary markers at the input layer in this model. Specifically, given an input utterance $W$ and a corresponding predicted slot entity, we define text markers as $\langle S\rangle$ and $\langle /S\rangle$, and insert them into the input utterance before and after the slot entities (Fig. 2 (b)). Let $\widehat{W}$ denote this modified sequence with text markers inserted:

$$\widehat{W} = ... \langle S\rangle, w_{START(i)}, \cdots, w_{END(i)}, \langle /S\rangle ... \tag{3}$$

By doing so, the position information and boundary of the slot entity can be explicitly used for the slot type prediction, which realizes the effect of the original shared encoder. We then apply another BiLSTM encoder on $\widehat{W}$ to generate the context-aware representations:

$$[\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_{n+2m}] = BiLSTM(E(\widehat{W})) \tag{4}$$

where $m$ denotes the number of the predicted slot entities in this utterance. We take its hidden states between the start and end markers($\langle S\rangle$ and $\langle /S\rangle$) to denote the slot representation. The representation $\boldsymbol{r_i}$ of $i^{th}$ slot entity can be denoted as:

$$\boldsymbol{r_i} = BiLSTM([\boldsymbol{h}_{START(i)}, ..., \boldsymbol{h}_{END(i)}]) \tag{5}$$

Following Shah et al. [13] and Liu et al. [14], we sum the embedding of the slot description tokens as the description representation. Then we can obtain a slot description matrix $\boldsymbol{M}_{desc} \in \mathbb{R}^{n_s \times d_s}$ where $n_s$ is the number of all the possible slot types and $d_s$ is the dimension of slot description representation. Finally, we calculate the dot product as classification logits $\boldsymbol{s}_i = \boldsymbol{r}_i \cdot \boldsymbol{M}_{desc}$ and get the cross-entropy loss.

### 2.3 Instance Weighting Scheme

Negative transfer often occurs in cross-domain learning because of the wide differences in the distribution of different domains. Especially in slot entities predicting, for the similar sentence structure, the slot entities that need to be captured in different domains are usually different. For example, slot *object_name* usually appears after the phrase "What is" in domain "SearchCreativeWork". However, it will have a bad effect in the domain without slot *object_name*, leading to the prediction of redundant entities in the slot entity model.

Since we can get all the possible slots in the target domain (Table 1 and 2), we introduce a simple yet effective instance weighting scheme by using the ratio of shared slots. We quantify the similarity between the data of different source domains and the specified target domain. For a target domain ($td$), the scoring function for calculating the similarity of different source domains ($sd$) is as follows:

$$score(sd, td) = \frac{|Slot_{shared}|}{|Slot_{sd}|} \cdot \frac{|Slot_{shared}|}{|Slot_{td}|} \tag{6}$$

where $|Slot_{sd}|$ and $|Slot_{td}|$ are the numbers of slot types for the source domain and the target domain respectively, and $|Slot_{shared}|$ is the number of shared slots of the source domain and the target domain. For example, in Table 1, *timeRange* and *spatial_relation* are the shared slots of "GetWeather" and "FindScreeningEvent" domains. Then we define a function $f_w(\cdot)$ to transform the scores into weights as follows:

$$weight(sd, td) = f_w(score) = \alpha + \beta \cdot score(sd, td) \tag{7}$$

where $\alpha$ and $\beta$ are the hyper-parameters and are used to tune the magnitude of similarity. For the utterances of different source domains , the learning rate is controlled by the similarity weight, which is computed as:

$$LR(sd, td) = \epsilon \cdot weight(sd, td) \tag{8}$$

where $\epsilon$ represents the initial learning rate for the source domain. Finally, $LR(sd, td)$ will be used to update the model parameters with the loss of the CRF layer in slot entity model.

## 3 Experiment

### 3.1 Dataset

To evaluate the efficiency of our proposed model, we conduct experiments on two benchmark datasets.

**Table 1.** Detailed statistics of SNIPS dataset.

| Domain | Slots | |
|---|---|---|
| | Cross-domain shared | Domain-specific |
| AddToPlaylist | artist, playlist, music_item | playlist_owner, entity_name |
| BookRestaurant | country, state, timeRange, sort, spatial_relation, city | party_size_number, poi, restaurant_type, facility, party_size_description, served_dish, cuisine, restaurant_name |
| GetWeather | country, state, timeRange, city, spatial_relation | spacurrent_location, condition_description condition_temperature, geographic_poi |
| PlayMusic | sort, artist, playlist, music_item | year, album, genre, track, service |
| RateBook | object_type, object_name | object_part_of_series_type, rating_value, object_select, best_rating, rating_unit |
| SearchCreativeWork | object_type, object_name | - |
| FindScreeningEvent | timeRange, object_type, spatial_relation | object_location_type, movie_type, movie_name, location_name |

**Table 2.** Detailed statistics of SMP-ECDT dataset.

| Domain | Slots | |
|---|---|---|
| | Cross-domain shared | Domain-specific |
| cookbook | keyword | dishName, utensil, ingredient |
| epg | datetime_time, datetime_date, category, name, code, area | tvchannel |
| map | startLoc_poi, endLoc_poi, startLoc_city, endLoc_city, endLoc_province, endLoc_area, location_city, location_province, type, startLoc_area | location_area, location_poi |
| message | name, content, category, teleOperator | receiver, headNum |
| poetry | keyword, author, name | queryField, dynasty |
| train | startDate_date, category, startLoc_city, endLoc_city, startLoc_area, endLoc_area, startLoc_province, endLoc_province, startLoc_poi, startDate_time | – |
| video | name, category, timeDescr, area, popularity, artist | tag, scoreDescr |

- **SNIPS**. We execute the experiments on the crowd-sourced benchmark corpus SNIPS [19] that widely used for slot filling. It is a public spoken language understanding dataset that contains 39 slot types across 7 domains (intents). The scheme corresponding to each domain is described in Table 1. To test our model, for each time, we choose one domain as the target domain and the other six domains as the source domains.
- **SMP-ECDT**. SMP-ECDT corpus[1,2] consists of 29 domains and 124 slot labels. Due to the large number of domains and the small amount of data in each domain, we only selected the top 7 domains with the largest amount of data as target domains for the experiment. The statistics of these domains are listed in Table 2. For each time, we choose one domain as the target domain and the other 28 domains as the source domains.

[1] http://conference.cipsc.org.cn/smp2019/evaluation.html.
[2] https://github.com/OnionWang/SMP2019-ECDT-NLU.

### 3.2 Baselines

We compare our model with the existing baselines:

- **Concept Tagger (CT)**. Bapna et al. [11] utilized slot descriptions to fill slots for each slot type individually and cope with the unseen slot types.
- **Coarse-to-fine Approach (Coach)**. Liu et al. [14] proposed a coarse-to-fine procedure with BIO 3-way classification and slot type prediction. It also further introduced a template regularization (TR) to improve the performance of similar or the same slot types. We use their best model Coach+TR to compare with but we call it Coach simply.
- **Contrastive Zero-Shot Learning with Adversarial Attack (CZSL-Adv)**. He et al. [15] used contrastive loss to leverage auxiliary slot description information and introduced an adversarial attack (Adv) training strategy to improve model robustness. Since the paper does not provide the code of adversarial attack part, we only use the CZSL model to compare with in the experiments on SMP-ECDT dataset.

### 3.3 Implementation Details

For a fair comparison under cross-domain settings, we follow most of the setups in Liu et al. [14] and He et al. [15]. For all BiLSTM encoders, We set the hidden size to 200 and a dropout [20] rate to 0.3. Following Liu et al. [14], for every word in SNIPS, we concatenate the word-level [21] and character-level [22] embeddings. For SMP-ECDT datasets, we use the public Chinese pre-training character-level embeddings [23]. We combine the samples from all source domains for training, split 500 data samples in the target domain as the validation set for choosing the best model and the remainder are used for the test set.

Following Liu et al. [14], we used tokenized slot names as the slots descriptions of SNIPS (e.g., the description of slot label *restaurant_type* is "restaurant type"). For SMP-ECDT dataset, we also define a simple Chinese slot description for each slot type. For example, the slot description of "*TV_channel*" is "电视频道" (The Chinese word embedding of "TV channel").

In instance weighting scheme, we set the hyper-parameters $\alpha$ and $\beta$ to 0.2 and 15. We take the instance weighting scheme on the loss of the CRF layer. We use Adam optimizer [24] to optimize all parameters with a learning rate of 0.0005. We set the batch size to 32 and use the early stop of patience 5. All data shown in the following results are the average of several independent experiments.

### 3.4 Overall Results

We use F1 score to evaluate the performances on each domain. Table 3 and Table 4 show the experiment results of the proposed model on SNIPS and SMP-ECDT datasets respectively. PCD-Iw denotes our proposed model and PCD represents our model without instance weight scheme. Scores in each row represent the performance of the leftmost target domain.

**Table 3.** Slot F1-scores on SNIPS for different target domains under zero/few-shot learning settings. * indicates the significant improvement over all baselines ($p < 0.05$)

| Training setting | Zero-shot | | | | | | Few-shot on 50 samples | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Domain↓ Model→ | CT | Coach | CZSL | CZSL-Adv | PCD | PCD-Iw | CT | Coach | CZSL | CZSL-Adv | PCD | PCD-Iw |
| AddToPlaylist | 38.82 | 50.90 | 53.29 | 53.89 | 52.84 | **55.83***| 68.69 | 74.68 | 77.71 | 76.18 | 80.24 | **80.37*** |
| BookRestaurant | 27.54 | 34.01 | 37.97 | 34.06 | 36.84 | **38.41*** | 54.22 | 74.82 | 77.35 | 76.28 | **77.41*** | 76.59 |
| GetWeather | 46.45 | 50.47 | 48.70 | 52.24 | 56.04 | **59.80*** | 63.23 | 79.64 | 81.85 | 83.28 | 84.23 | **85.09*** |
| PlayMusic | 32.86 | 32.01 | 29.14 | 34.59 | 31.81 | **36.31*** | 54.32 | 66.38 | 65.59 | 68.17 | 66.44 | **69.76*** |
| RateBook | 14.54 | 22.06 | 29.55 | **31.53** | 30.26 | 28.25 | 76.45 | 84.62 | 84.31 | 87.22 | 89.16 | **89.49*** |
| SearchCreativeWork | 39.79 | 46.65 | 49.32 | 50.61 | 49.78 | **51.81*** | 66.38 | 64.56 | 66.41 | 66.49 | **70.00*** | 69.65 |
| FindScreeningEvent | 13.83 | 25.63 | 25.95 | **30.05** | 27.75 | 26.95 | 70.67 | 83.85 | 81.14 | 83.26 | 84.10 | **86.43*** |
| Average F1 | 30.55 | 37.39 | 39.13 | 40.99 | 40.76 | **42.50*** | 64.85 | 75.51 | 76.34 | 77.27 | 78.80 | **79.63*** |

**Table 4.** Slot F1-scores on SMP-ECDT for different target domains under zero/few-shot learning settings. * indicates the significant improvement over all baselines ($p < 0.05$)

| Training Setting | Zero-shot | | | | | Few-shot on 5 samples | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Domain↓ Model→ | CT | Coach | CZSL | PCD | PCD-Iw | CT | Coach | CZSL | PCD | PCD-Iw |
| cookbook | 1.35 | 16.95 | 15.00 | 16.54 | **22.27*** | 3.47 | 38.07 | 43.18 | **48.62*** | 42.31 |
| epg | 9.50 | 18.84 | 20.54 | **25.41*** | 24.59 | 13.95 | 31.37 | 29.54 | 39.92 | **39.93*** |
| map | 16.75 | 22.15 | 23.42 | 22.95 | **26.66*** | 18.39 | **35.71** | 32.02 | 28.33 | 28.40 |
| message | 11.19 | 29.87 | 25.23 | 26.59 | **29.89*** | 30.86 | 33.87 | 34.86 | 31.79 | **36.63*** |
| poetry | 19.03 | 43.19 | 43.66 | 43.41 | **43.81** | 21.96 | 50.48 | 53.67 | 45.74 | **65.52*** |
| train | 84.58 | **85.71** | 85.09 | 83.96 | 84.05 | 84.95 | 85.16 | 85.14 | **86.65*** | 86.31 |
| video | 19.41 | 26.39 | 32.13 | **36.68*** | 32.53 | 22.14 | 30.56 | 30.82 | 34.42 | **35.07*** |
| Average F1 | 23.21 | 34.73 | 35.01 | 36.50 | **37.69*** | 26.94 | 43.60 | 43.92 | 45.07 | **47.73*** |

From Table 3 and 4, we can observe that our model significantly outperforms all the baselines and achieves the state-of-the-art performance in the zero/few-shot settings. In the zero-shot setting, compared with the best prior work, PCD achieves 1.51% and 2.68% improvement on SNIPS dataset and SMP-ECDT dataset respectively. Moreover, since we did not use adversarial attack training to improve the robustness of the model, the PCD-Iw actually reaches 3.37% improvement on SNIPS dataset compared to the CZSL model (F1 score of 39.13). In the few-shot setting, PCD achieves 2.36% and 3.81% improvement in two datasets. In addition, without instance weighting scheme the PCD framework has also improved in every experiment setting. These results indicate the effectiveness of our proposed framework.

### 3.5 Analysis on Slot Entity Identification

Since our PCD-Iw approach especially the instance weighting scheme has a promotion effect on the identification of slot entities, we analyze this effect separately. The results are shown in Tables 5 and 6. The scores are calculated from our slot entity model and the first step in two-stage models [14,15].

**Table 5.** BIO F1-scores on SNIPS for different target domains under zero/few-shot learning settings.

| Training setting | Zero-shot | | | | Few-shot on 50 samples | | | |
|---|---|---|---|---|---|---|---|---|
| Domain↓ Model→ | Coach | CZSL | PCD | PCD-Iw | Coach | CZSL | PCD | PCD-Iw |
| AddToPlaylist | 57.06 | 57.43 | 61.77 | **65.43** | 79.08 | 80.98 | 84.93 | **85.17** |
| BookRestaurant | 59.49 | 59.51 | 60.39 | **65.29** | 82.56 | 83.75 | 84.63 | **87.06** |
| GetWeather | 57.14 | 59.76 | 66.24 | **71.15** | 79.95 | 84.96 | 89.58 | **90.28** |
| PlayMusic | 48.48 | 49.53 | 52.22 | **59.46** | 70.24 | 74.72 | 77.68 | **82.04** |
| RateBook | 32.23 | **38.13** | 34.39 | 35.87 | 86.69 | 89.56 | 89.10 | **90.08** |
| SearchCreativeWork | 48.88 | 48.27 | 49.97 | **54.66** | 66.69 | 67.90 | 70.41 | **71.47** |
| FindScreeningEvent | 37.73 | 40.71 | 44.24 | **46.43** | 84.02 | 85.19 | 84.93 | **88.22** |
| Average F1 | 48.72 | 50.51 | 52.74 | **56.90** | 78.45 | 81.01 | 83.04 | **84.90** |

**Table 6.** BIO F1-scores on SMP-ECDT for different target domains under zero/few-shot learning settings.

| Training setting | Zero-shot | | | | Few-shot on 5 samples | | | |
|---|---|---|---|---|---|---|---|---|
| Domain↓ Model→ | Coach | CZSL | PCD | PCD-Iw | Coach | CZSL | PCD | PCD-Iw |
| cookbook | 65.84 | 70.50 | 73.24 | **74.35** | 71.74 | 71.11 | **74.71** | 74.31 |
| epg | 29.72 | 34.89 | 41.46 | **42.70** | 38.61 | 35.28 | 49.19 | **49.28** |
| map | 53.15 | 57.00 | 55.95 | **57.32** | 56.21 | 52.63 | 54.50 | **56.25** |
| message | 38.35 | 33.50 | 36.17 | **44.58** | 39.16 | 40.73 | 42.06 | **46.47** |
| poetry | 51.15 | **53.08** | 51.41 | 52.05 | 53.50 | 54.68 | 52.69 | **74.86** |
| train | **92.13** | 89.70 | 91.04 | 89.20 | 91.26 | 89.65 | 93.00 | **93.57** |
| video | 32.39 | 41.58 | 42.23 | **42.51** | 35.12 | 36.79 | 40.91 | **44.15** |
| Average F1 | 51.82 | 54.32 | 55.93 | **57.52** | 55.09 | 54.41 | 58.15 | **62.70** |

As can be seen from Table 5 and 6, our model achieves the state-of-the-art performance in almost all domains under zero/few-shot settings. In the zero-shot setting, PCD achieves 6.39% and 3.40% improvement on SNIPS dataset and SMP-ECDT dataset respectively. In the few-shot setting, PCD achieves 3.89% and 8.29% improvement in two datasets. The result also verifies our assumptions that instance weighting scheme can be used for alleviating the problem of negative transfer and improving the performance of capturing slot entities.

### 3.6 Ablation Study

From Table 3 and 4, we can see that compared with PCD-Iw model, the PCD model has a performance decline of 1.0%-2.5%, which indicates that both our PCD model and the instance weighting scheme have an improvement effect. As can be seen from Table 5 and 6, compared with the PCD-Iw model, the performance of the PCD model in identifying slot entities decreases by 1.5% to 5.0%. The relatively high gap indicates that the instance weighting scheme has a more significant improvement in the identification of slot entities.

### 3.7 Analysis on Seen and Unseen Slots

Following the baselines setting, we also split the test set into "unseen" and "seen" parts. Table 7 shows the results on seen and unseen slots in two datasets. We can observe that our approach consistently outperforms the baselines both on the unseen and seen slots in the two settings and two datasets. Our pipeline model is to promote all the slot types and the instance weighting scheme also alleviates the problem of negative transfer. Therefore, our approaches generally improve on both unseen and seen slot types.

**Table 7.** Average F1-scores on SNIPS and SMP-ECDT for seen and unseen slots across all target domains.

| Dataset | SNIPS | | | | SMP-ECDT | | | |
|---|---|---|---|---|---|---|---|---|
| Setting | 0 sample | | 50 samples | | 0 sample | | 5 samples | |
| | unseen | seen | unseen | seen | unseen | seen | unseen | seen |
| CT | 27.10 | 44.18 | 62.05 | 69.64 | 11.85 | 30.95 | 18.29 | 34.64 |
| Coach | 34.09 | 51.93 | 76.49 | 80.16 | 18.98 | 44.15 | 31.45 | 44.78 |
| CZSL | 34.57 | 52.69 | 77.15 | 80.09 | 17.05 | 46.74 | 32.74 | 43.41 |
| CZSL-Adv | 36.35 | 55.43 | 78.48 | 79.36 | – | – | – | – |
| PCD | 35.79 | 55.63 | 78.84 | 80.75 | 20.73 | 48.70 | 29.84 | 46.29 |
| PCD-Iw | **36.98** | **56.96** | **80.61** | **81.66** | **21.12** | **49.08** | **39.76** | **49.17** |

## 4 Conclusions

In this paper, we propose a new pipeline approach with distinct slot entity and type prediction for cross-domain slot filling. Our approach consists of a slot entity identification model and slot type alignment model, which uses distinct contextual representations for learning and boundary markers for connecting two sub models. Moreover, we introduce an effective instance weighting scheme to control the contribution of different source domains by adjusting learning rates. Experiments show that our approach significantly outperforms existing cross-domain slot filing models, especially in the accuracy of slot entity identification.

**Acknowledgements..** This work was supported by Natural Science Foundation of Guangdong Province (No. 2021A1515011864), Guangzhou Key Laboratory of Intelligent Agriculture (No. 201902010081), National Natural Science Foundation of China (No. 71472068), Educational Commission of Guangdong Province of China (No. 2020KTSCX01) and Innovation Training Project for College Students of Guangdong Province (No. S202110564051, No. S202010564169).

## References

1. Sarikaya, R.: The technology behind personal digital assistants: an overview of the system architecture and key components. IEEE Signal Process. Mag. **34**(1), 67–81 (2017)
2. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016), San Francisco, CA, USA, 8–12 September, pp. 685–689 (2016)
3. Goo, C., et al.: 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 (NAACL-HLT 2018), New Orleans, Louisiana, USA, 1–6 June, Volume 2 (Short Papers), pp. 753–757 (2018)
4. Haihong, E., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August, Volume 1: Long Papers, pp. 5467–5471 (2019)
5. Qin, L., Che, W., Li, Y., Wen, H., Liu, T.: A stack-propagation framework with token-level intent detection for spoken language understanding. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), Hong Kong, China, 3–7 November, pp. 2078–2087 (2019)
6. Jaech, A., Heck, L.P., Ostendorf, M.: Domain adaptation of recurrent neural networks for natural language understanding. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016), San Francisco, CA, USA, 8–12 September, pp. 690–694 (2016)
7. Guo, J., Shah, D.J., Barzilay, R.: Multi-source domain adaptation with mixture of experts. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), Brussels, Belgium, 31 October–4 November, pp. 4694–4703 (2018)
8. Lin, B.Y., Lu, W.: Neural adaptation layers for cross-domain named entity recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), Brussels, Belgium, 31 October–4 November, pp. 2012–2022 (2018)
9. Liu, Z., et al.: Zero-shot cross-lingual dialogue systems with transferable latent variables. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), Hong Kong, China, 3–7 November, pp. 1297–1303 (2019)
10. Liu, Z., Winata, G.I., Fung, P.: Zero-resource cross-domain named entity recognition. In: Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP@ACL 2020), Online, 9 July, pp. 1–6 (2020)
11. Bapna, A., Tür, G., Hakkani-Tür, D., Heck, L.P.: Towards zero-shot frame semantic parsing for domain scaling. In: Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017), Stockholm, Sweden, 20–24 August, pp. 2476–2480 (2017)
12. Lee, S., Jha, R.: Zero-shot adaptive transfer for conversational language understanding. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019) Honolulu, Hawaii, USA, 27 January–1 February, pp. 6642–6649 (2019)

13. Shah, D.J., Gupta, R., Fayazi, A.A., Hakkani-Tür, D.: Robust zero-shot cross-domain slot filling with example values. In: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August, Volume 1: Long Papers, pp. 5484–5490 (2019)
14. Liu, Z., Winata, G.I., Xu, P., Fung, P.: Coach: A coarse-to-fine approach for cross-domain slot filling. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), Online, 5–10 July, pp. 19–25 (2020)
15. He, K., Zhang, J., Yan, Y., Xu, W., Niu, C., Zhou, J.: Contrastive zero-shot learning for cross-domain slot filling with adversarial attack. In: Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020), Barcelona, Spain (Online), 8–13 December, pp. 1461–1467 (2020)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
17. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016), San Diego California, USA, 12–17 June, pp. 260–270 (2016)
18. Zhong, Z., Chen, D.: A frustratingly easy approach for entity and relation extraction. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), Online, 6–11 June, pp. 50–61 (2021)
19. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. CoRR abs/1805.10190 (2018)
20. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
21. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguistics **5**, 135–146 (2017)
22. Hashimoto, K., Xiong, C., Tsuruoka, Y., Socher, R.: A joint many-task model: growing a neural network for multiple NLP tasks. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), Copenhagen, Denmark, 9–11 September, pp. 1923–1933 (2017)
23. Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., Du, X.: Analogical reasoning on Chinese morphological and semantic relations. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, (ACL 2018), Melbourne, Australia, 15–20 July, Volume 2: Short Papers, pp. 138–143 (2018)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015