

Bridge to Target Domain by Prototypical Contrastive Learning and Label Confusion: Re-explore Zero-Shot Learning for Slot Filling

Liwen Wang^{1*}, Xuefeng Li^{1*}, Jiachi Liu¹, Keqing He², Yuanmeng Yan¹, Weiran Xu¹

¹Beijing University of Posts and Telecommunications, Beijing, China

²Meituan Group, Beijing, China

{w_liwen, lixuefeng, ljcl1997, yanyuanmeng, xuweiran}@bupt.edu.cn
{hekeqing}@meituan.com

Abstract

Zero-shot cross-domain slot filling alleviates the data dependence in the case of data scarcity in the target domain, which has aroused extensive research. However, as most of the existing methods do not achieve effective knowledge transfer to the target domain, they just fit the distribution of the seen slot and show poor performance on unseen slot in the target domain. To solve this, we propose a novel approach based on prototypical contrastive learning for zero-shot slot filling. The prototypical contrastive learning aims to reconstruct the semantic constraints of labels, and we introduce the label confusion strategy to establish the label dependence between the source domains and the target domain on-the-fly. Experimental results show that our model achieves significant improvement on the unseen slots, while also set new state-of-the-arts on slot filling task.¹

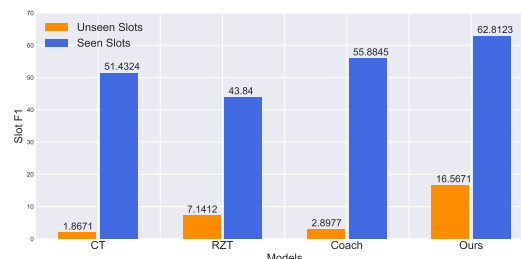
1 Introduction

Slot filling, as an important part of the task-oriented dialogue system, is mainly used to extract specific information in user utterances. Traditional supervised methods have shown remarkable performance in slot filling tasks (Liu and Lane, 2016; Goo et al., 2018; E et al., 2019; He et al., 2020b; Wu et al., 2020; He et al., 2020a; Oguz and Vu, 2020; Qin et al., 2020), but they require a large amount of domain-specific labeled data.

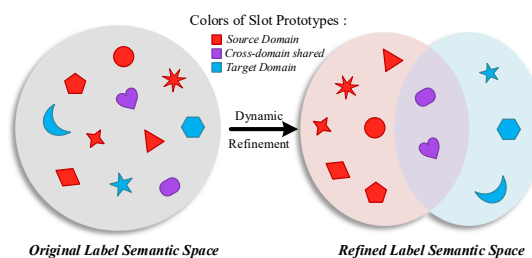
Recently, there has been increasing interest in the zero-shot cross-domain slot filling task, whose goal is to identify unseen slots in the target domain without any labeled data. Previous methods can be classified into two types: one-stage and two-stage. In the one-stage framework, (Bapna et al., 2017; Shah

*The first two authors contribute equally. Weiran Xu is the corresponding author.

¹Our source code is available at: <https://github.com/W-lw/PCLC>



(a) Performance of some zero-shot cross-domain slot filling models on seen and unseen slots in *GetWeather* domain in SNIPS.



(b) The left part is the original distribution, where the slot prototypes of the source domain and the target domain are chaotic and lack constraint relations. The right part is the distribution after refinement, where slot prototypes of the source domain and target domain are separated.

Figure 1: (a) Performance of the models in *GetWeather* domain; (b) a demonstration of slot prototypes refinement process;

et al., 2019b; Lee and Jha, 2019) perform slot filling task for each slot type respectively, and the slot type description is then integrated into the prediction process to achieve zero-shot adaptation. The main drawback is that the model possibly predicts multiple slot types for one entity span. To avoid the above problem, (Liu et al., 2020b,a; He et al., 2020c) decompose the slot filling task into two stages. First, all slot entities in the utterances are identified by the coarse-grained binary sequence labeling model. Then slot types are classified by mapping the entity value to the representation of the corresponding slot label in the semantic space.

However, we find that these methods have poor performance on unseen slot in the target domain, as

shown in Fig 1(a). In the cross-domain slot filling task, there are always seen slots and unseen slots in the target domain. The former exists in both the source domain and the target domain, while the latter only exists in the target domain. Although previous methods have achieved good overall performance on the cross-domain slot filling task, we find that their high performance mostly comes from the seen slots, while the performance on the unseen slots remains very low. We argue that these methods don't achieve domain adaptation well. Actually, these methods lack explicit modeling of the association between the source and target domain. They directly utilize the slot name embedding as its slot prototype, whose distribution is often chaotic in semantic space due to lack of modeling for constraint relationships of slot prototypes in different domains. And due to the lack of data in the target domain, the model can't learn the mapping relationship between the slot value in the target domain and the slot prototype. Therefore, when making predictions, the model can only correctly predict the seen slot type, and the prediction for unseen slots is almost random. An intuitive way to solve this problem is to refine the label semantic space and establish the constraint relationship between source and target domains, as shown in Fig 1(b).

In this paper, we propose a novel method based on **Prototypical Contrastive learning** and **Label Confusion** strategies (**PCLC**) to dynamically refine the constraint relationship between slot prototypes in the semantic space. First, we introduce prototypical contrastive learning (Medina et al., 2020; Cao et al., 2020; Li et al., 2020; Yue et al., 2021; Li et al., 2021; Zhao et al., 2021), which makes the mapped slot value embeddings close to its corresponding slot prototype and away from other slot prototypes, to enhance the accuracy of mapping between feature space and semantic space. Second, we introduce a label obfuscation strategy to establish the dependency between the slots of the source domain and the target domain. In the training process of the source domain, we confuse the original one-hot label into the probability distribution of the source domain and the target domain by calculating the similarity between the slot prototypes in the source domain and the target domain.

Our contributions are three-fold. (1) We evaluate the performance of existing methods on cross-domain slot filling. It turns out these methods do not achieve domain adaptation effectively as the

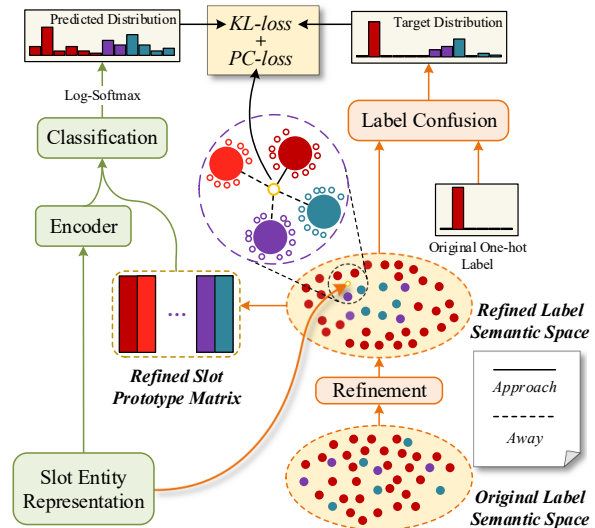


Figure 2: Illustration of proposed method **PCLC**

performance varies widely between unseen slots and seen slots. (2) We propose a novel method based on prototypical contrastive learning and label confusion to refine semantic space and enhance the domain adaptation. (3) Experiments demonstrate that the performance of our proposed method has improved significantly on unseen slots, and the overall performance outperforms the state-of-the-art models on both zero-shot and few-shot settings.

2 Method

2.1 Overall Architecture

Consistent with Coach (Liu et al., 2020b), we adapt the two-stage framework as the backbone of our model. In the first stage, we utilize a BiLSTM-CRF structure (Lample et al., 2016) (as a BIO-label sequence tagger) to encode the input utterance and identify the slot entities.

In the second stage, our model encodes the slot entity and predicts the label for it by calculating the similarity with the slot prototypes in the label semantic space. Unlike the previous works (Liu et al., 2020b; He et al., 2020c) which directly utilize the slot name embedding as slot prototypes, we introduce Prototypical Contrastive learning and Label Confusion strategies (PCLC) strategies to dynamically refine the constraint relationship between slot prototypes in the semantic space, as shown in Fig 2. In the training procedure, we use an MLP layer to encode the original slot name embedding. So we can obtain a dynamically updated slot prototype matrix. We will introduce these in the following subsections.

2.2 Prototypical Contrastive Learning

There is a problem of utilizing slot name embedding as the slot prototypes that the distribution of slot name embedding is very chaotic and somewhat dense in semantic space. Therefore, when the slot values are mapped to the semantic space, they can hardly establish a correct relationship with the corresponding slot prototype. So, we introduce prototypical contrastive learning (Li et al., 2021) to enhance the precision of the mapping function from feature space to semantic space and reduce the density of slot prototype distribution in the label semantic space.

As shown in Fig 2, given a slot value r_k and the corresponding label y_k , we will map r_k to the refined semantic space, and obtain the prototypical contrastive loss:

$$\mathcal{L}_{pc}(r_k, y_k) = -\log \frac{\exp(r_k \cdot z^{y_k} / \tau)}{\sum_{c=1}^C \exp(r_k \cdot z^c / \tau)} \quad (1)$$

, where z^c is the representation of slot prototype with label c , τ is temperature factor and C is the number of slot categories. By optimizing the above objective function \mathcal{L}_{pc} , slot values can be close to corresponding slot prototype in semantic space and be away from other slot prototypes.

2.3 Label Confusion

Now that prototypical contrastive learning has separated the slot prototypes and established the relation between slot values and slot prototypes, we need to establish the dependency between the slots of the source domain and the target domain by label confusion.

In the training process of the source domain, we confuse the original one-hot label into the probability distribution of the source domain and the target domain by calculating the similarity between the slot prototypes in the source domain and the target domain. Given a slot value r and its corresponding label y , we first calculate the cosine similarities between z^y and slot prototypes z^j of all slot label $j \in \mathcal{T}$ in the target domain:

$$s^j = \text{cosine_similarity}(z^y, z^j) \quad (2)$$

$$\mathcal{D}_{\text{tgt}} = \text{norm}(\{s^j\}_{j \in \mathcal{T}}) \quad (3)$$

, where the $\text{norm}(\cdot)$ indicates the L_1 norm. We regard the \mathcal{D}_{tgt} as the soft distribution over the slot labels in the target domain and then concatenate them with the one-hot distribution on the source domain with label confusion factor λ :

$$\mathcal{D}_{\text{src}} = \text{one_hot}(y) \quad (4)$$

$$\mathcal{D}_{\text{smooth}} = \text{concat}(\lambda \cdot \mathcal{D}_{\text{src}}, (1 - \lambda) \cdot \mathcal{D}_{\text{tgt}}) \quad (5)$$

, where $\text{one_hot}(y)$ indicates the one-hot distribution of the source slot label y and $\text{concat}(\cdot)$ indicates the concatenation operation. Then we can obtain the KL-Divergence loss:

$$\mathcal{D}_{\text{pre}} = \log_softmax(\mathcal{M}_{\text{proto}} \cdot r_k) \quad (6)$$

$$\mathcal{L}_{kl} = \text{KL-Divergence}(\mathcal{D}_{\text{pre}}, \mathcal{D}_{\text{smooth}}) \quad (7)$$

, where $\mathcal{M}_{\text{proto}}$ donate the representation matrix of the slot prototypes. The final loss function is $\mathcal{L} = \mathcal{L}_{pc} + \alpha \mathcal{L}_{kl}$, where α is a hyperparameter.

3 Experiments

3.1 Setup

Dataset. We evaluate our method on SNIPS (Coucke et al., 2018), a public spoken language understanding dataset which contains 7 domains and 39 slots. To simulate the cross-domain scenarios, we follow the setup of (Liu et al., 2020b), choosing one domain as the target domain and the left six domains as the source domains.

Baselines. We compare our approach with the following baselines:

- **Concept Tagger (CT)** A method proposed by (Bapna et al., 2017), which utilizes the slot description to boost the performance on the unseen slots in the target domain.
- **Robust Zero-shot Tagger (RZT)** Based on CT, a method proposed by (Shah et al., 2019a), which adds additional example values of slots to improve the robustness of the model.
- **Coarse-to-fine Approach (Coach)** A two-stage framework proposed by (Liu et al., 2020b): the first stage performs coarse-grained BIO labeling task, and the second stage utilizes slots descriptions to perform fine-grained slot type classification task. And it has a stronger variant², which applies template regularization to improve performance.
- **Contrastive Zero-Shot Learning with Adversarial Attack (CZSL-Adv)** A method proposed by (He et al., 2020c) based on Coach, which utilizes contrastive learning and adversarial attacks to improve the performance and robustness of the framework.

²For brevity, all the *Coach* we mentioned in subsequent experiments refer to this stronger variant.

Training Setting Domain ↓ ~ Model →	Zero-shot							Few-shot on 50 samples						
	CT	RZT	Coach	CZSL-Adv	LC	PCL	PCLC	CT	RZT	Coach	CZSL-Adv	LC	PCL	PCLC
AddToPlaylist	38.82	42.77	50.90	53.89	53.53	55.60	59.24	68.69	74.89	74.68	76.18	81.55	79.46	81.59
BookRestaurant	27.54	30.68	34.01	34.06	34.89	35.28	41.36	54.22	54.49	74.82	76.28	76.48	76.87	77.54
GetWeather	46.45	50.28	50.47	52.24	49.16	48.61	54.21	63.23	58.87	79.64	83.28	83.83	79.61	83.85
PlayMusic	32.86	33.12	32.01	34.59	29.83	33.02	34.95	54.32	59.20	66.38	68.17	66.25	67.48	66.86
RateBook	14.54	16.43	22.06	31.53	22.20	23.25	29.31	76.45	76.87	84.62	87.22	87.14	89.06	87.88
SearchCreativeWork	39.79	44.45	46.65	50.61	48.73	53.73	53.51	66.38	67.81	64.56	66.49	69.63	70.88	71.06
FindScreeningEvent	13.83	12.25	25.63	30.05	20.99	23.99	27.17	70.67	74.58	83.85	83.26	82.32	83.46	81.56
Average F1	30.55	32.85	37.39	40.99	37.05	39.07	42.82	64.85	66.67	75.51	77.27	78.17	78.12	78.62

Table 1: Slot F1-scores on SNIPS for different target domains under zero-shot and few-shot learning settings. LC denotes adding label confusion strategy and PCL denotes adding prototypical contrastive loss.

Implementation Details We follow the setup of (Liu et al., 2020b), selecting one domain as the target domain at a time, and use 500 samples in this domain as a validation set, the rest as a test set. Samples from the remaining six domains are used for training. We use a two-layer BiLSTM with a hidden layer size of 200 as encoder, and concatenate the word-level and character-level embedding as input. The dropout rate of BiLSTM is set to 0.3 and the learning rate of the Adam optimizer is set to 0.0005. We set the batch size to 64 and use the early stop of patience 15 to ensure the stability of the model. For all the experiments, we train and test our model on the 2080Ti GPU. It takes an average of 2.5 hours to run with 30 epochs under the zero-shot setting and 5 hours to run with 60 epochs under the few-shot setting.

3.2 Main Results

As illustrated in Table 1, our method PCLC outperforms the SOTA model by 1.83% on the average F1-score under zero-shot setting, and 1.35% under few-shot setting. Besides, compared to Coach, the model we directly modify, our method achieves superior performance by 4.7% under zero-shot setting and 3.1% under few-shot setting. The significant performance improvement proves that the combined use of the two strategies we proposed can help establish a better mapping relation between slots values and slot prototypes in label semantic space.

3.3 Results on Seen slots and Unseen Slots

In this section, we re-explore all these methods’ transferability by measuring their performance directly on unseen slots and seen slots respectively³. The experimental results are shown in Table 2. We can observe that the previous methods perform very

³Liu et al. (2020b) split SNIPS into *seen* and *unseen* part according to whether an utterance includes the unseen slots, which may introduce some bias to the actual performance when testing model on unseen slots.

settings	0 samples		50 samples	
	unseen	seen	unseen	seen
CT	3.38	37.23	52.65	65.66
RZT	2.19	40.99	50.28	61.63
Coach	9.31	46.22	68.59	74.55
LC	9.15	48.65	75.09	79.60
PCL	10.71	49.08	74.78	79.44
PCLC	17.38	51.68	76.64	78.99

Table 2: Average F1-scores for seen and unseen slots across all target domains³

limited on the unseen slot under the zero-shot setting, while PCLC has achieved a very significant improvement on the unseen slot under both settings. For huge improvement on unseen slots, we hypothesize that with our proposed two strategies our model does achieve a more effective knowledge transferring to the target domain instead of overfitting the seen slot. It is necessary to mention that there still exists huge gap of model performance on unseen and seen slot, which is worth further study. More details about all domains can be found in Table 3.

3.4 Ablation Analysis

We compare the effect for the main performance of PCL and LC strategy in Table 1. We find that the LC strategy would cause a slight decrease in performance under the zero-shot setting while PCL can provide some performance boost, comparing to coach. Interestingly, both LC and PCL can achieve a significant improvement close to PCLC under the few-shot setting. Besides, we conduct the ablation experiment on unseen slots, as shown in Table 2. We find that LC and PCL don’t work very well individually on unseen slots under zero-shot setting. The experimental results show that both PCL and LC would bring significant improvement under the few-shot setting, but they need to be combined together for better performance under the zero-shot setting.

3.5 Visualization Analysis

To explore the effectiveness on the refinement of the slot prototype embedding in semantic space, we

samples	Zero-shot								Few-shot on 50 samples							
	CT		RZT		Coach		PCLC		CT		RZT		Coach		PCLC	
	unseen	seen	unseen	seen	unseen	seen	unseen	seen	unseen	seen	unseen	seen	unseen	seen	unseen	seen
AddToPlaylist	5.18	47.15	3.48	57.50	7.94	64.65	2.57	73.32	60.12	73.87	54.71	68.72	66.11	76.26	75.16	84.60
BookRestaurant	1.87	51.43	7.14	43.84	2.89	55.88	16.56	62.81	46.25	56.83	47.40	52.35	64.94	79.15	73.30	83.16
GetWeather	2.11	39.54	2.34	62.84	2.20	63.97	14.20	65.84	31.91	71.61	29.46	68.83	59.71	84.15	75.12	88.02
PlayMusic	0.13	46.48	0	47.45	8.78	31.69	17.53	45.17	35.36	62.27	35.21	61.80	54.24	68.02	62.79	69.56
RateBook	0.13	25.10	0.13	25.41	18.81	40.42	25.70	34.70	81.39	58.69	80.65	55.64	89.11	70.38	93.15	72.98
SearchCreativeWork	-	39.59	-	39.27	-	44.35	-	53.51	-	64.69	-	39.27	-	58.49	-	71.05
SearchScreeningEvent	10.84	11.32	0	10.61	15.21	27.27	22.71	29.66	60.82	71.67	54.21	56.43	77.41	85.36	80.29	83.52
Avg	3.38	37.23	2.19	40.99	9.31	46.22	17.38	51.68	52.65	65.66	50.28	61.63	68.59	74.55	76.64	78.99

Table 3: Detailed F1-scores for seen and unseen slots across all target domains. The bold part is the highest F1-score obtained by our model and baselines on the unseen and seen slots, under the zero-shot and the few-shot settings respectively.

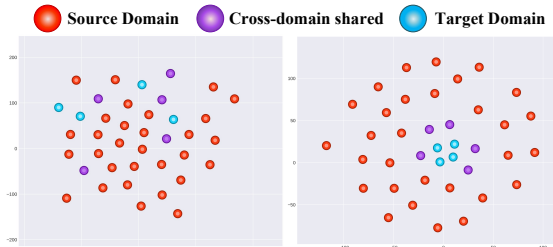


Figure 3: t-SNE projection of slot prototypes. The left represents the original distribution and the right represents refined distribution by our method. The target domain is *Getweather* while others are source domains.

do the t-SNE visualization for the slot prototype representation after refinement, as shown in Fig 3. It is observed that after refinement, the distribution of slot prototype in semantic space changes from a chaotic distribution to a separated distribution between the source domain and the target domain. This observation indicates our thoughts: prototypical contrastive learning stimulates different slots to be separated from each other, while label confusion is able to establish the constraint relations between the source domain and the target domain. The separation between the source domain and the target domain helps to build the mapping between slot value and slot prototype, and thus improve the accuracy on unseen slots.

3.6 Hyperparameter Analysis

The hyperparameter λ is used to control the degree of label confusion, which is set too small will make the prediction too random, while set too large the original effect will not be achieved. As shown in Figure 4, the model has the best performance when we set λ to 0.6

4 Conclusion & Discussion

In this paper, we propose a novel method based on Prototypical Contrastive learning and Label Confusion strategy (PCLC) for cross-domain slot filling. Our main contribution was to improve the domain adaptability of the model. The proposed method

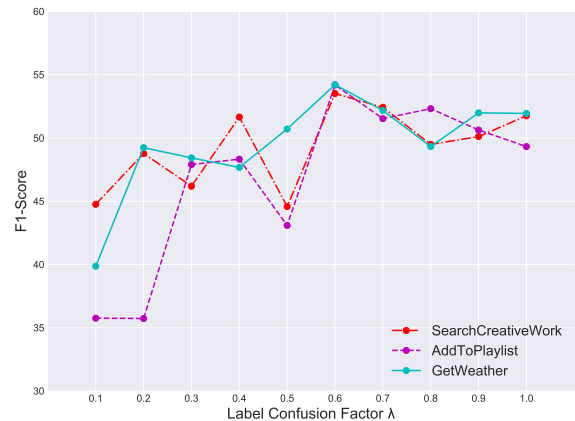


Figure 4: F1-scores of PCLC with different label confusion factor λ .

conducts a refinement process for label semantic space to re-establish the constraint relationship between different slots. Experiments show the effectiveness of our method, especially for recognizing the unseen slots.

As there is still a huge gap in model performance between unseen and seen slots, in future work, we will focus on improving the performance on unseen slots while maintaining the performance on seen slots. Representation disentanglement (Wang et al., 2021) can be used to disentangle domain-specific and domain-shared knowledge in the source domain, then we can preserve domain-shared knowledge and focus on establishing the relation of domain-specific knowledge between the source and the target domain.

5 Acknowledgements

We thank Jinzheng Zhao, Yuanyuan Qi and all anonymous reviewers for their helpful comments and suggestions. This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artificial Intelligence" Project No. MCM20190701.

References

- Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. [Towards zero-shot frame semantic parsing for domain scaling](#).
- Zeyu Cao, Xiaorun Li, and Liaoying Zhao. 2020. [Un-supervised feature learning by autoencoder and prototypical contrastive learning for hyperspectral classification](#).
- 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](#).
- 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.
- 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.
- Keqing He, Yuanmeng Yan, Si hong Liu, Z. Liu, and Weiran Xu. 2020a. [Learning label-relational output structure for adaptive sequence labeling](#). *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Keqing He, Yuanmeng Yan, and Weiran Xu. 2020b. [Learning to tag OOV tokens by integrating contextual representation and background knowledge](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 619–624, Online. Association for Computational Linguistics.
- Keqing He, Jinchao Zhang, Yuanmeng Yan, Weiran Xu, Cheng Niu, and Jie Zhou. 2020c. [Contrastive zero-shot learning for cross-domain slot filling with adversarial attack](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1461–1467, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [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*, pages 260–270, San Diego, California. Association for Computational Linguistics.
- Sungjin Lee and Rahul Jha. 2019. [Zero-shot adaptive transfer for conversational language understanding](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6642–6649.
- Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. 2021. [Prototypical contrastive learning of un-supervised representations](#).
- Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. 2020. [Contrastive clustering](#).
- Bing Liu and Ian Lane. 2016. [Attention-based recurrent neural network models for joint intent detection and slot filling](#).
- Zihan Liu, Genta Indra Winata, and Pascale Fung. 2020a. [Zero-resource cross-domain named entity recognition](#).
- Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020b. [Coach: A coarse-to-fine approach for cross-domain slot filling](#).
- Carlos Medina, Arnout Devos, and Matthias Grossglauser. 2020. [Self-supervised prototypical transfer learning for few-shot classification](#).
- Cennet Oguz and Ngoc Thang Vu. 2020. [A two-stage model for slot filling in low-resource settings: Domain-agnostic non-slot reduction and pretrained contextual embeddings](#). In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 73–82, Online. Association for Computational Linguistics.
- Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. [Agif: An adaptive graph-interactive framework for joint multiple intent detection and slot filling](#).
- Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019a. [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.
- Darsh J Shah, Raghav Gupta, Amir A Fayazi, and Dilek Hakkani-Tur. 2019b. [Robust zero-shot cross-domain slot filling with example values](#).
- Liwen Wang, Yuanmeng Yan, Keqing He, Yanan Wu, and Weiran Xu. 2021. [Dynamically disentangling social bias from task-oriented representations with adversarial attack](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3740–3750, Online. Association for Computational Linguistics.

Di Wu, Liang Ding, Fan Lu, and Jian Xie. 2020. [Slotrefine: A fast non-autoregressive model for joint intent detection and slot filling.](#)

Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. 2021. [Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation.](#)

Xiangyun Zhao, Raviteja Vemulapalli, Philip Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. 2021. [Contrastive learning for label-efficient semantic segmentation.](#)