Zero-shot Domain Adaptation with Inference Relation Paths for Spoken Language Understanding

Sixia Li* and Jianwu Dang*†
* Japan Advanced Institute of Science and Technology, Ishikawa, Japan
† Tianjin Key Laboratory of Cognitive Computing and Application, College of Intelligence and Computing, Tianjin University, Tianjin, China
E-mail: lisixia@jaist.ac.jp, jdang@jaist.ac.jp

***Abstract*—Zero-shot slot filling methods are proposed to tackle the problems of adapting to new domains with unseen slots. Due to lacking share information across domains based on semantic slot descriptions, the challenge in zero-shot slot filling is handling the unseen slots that are less similar to the training slots semantically. Since people utilizes not only explicit semantic information as estimation cues but also implicit semantic relations, this study attempts to find implicit sematic relation cues between slots and values to tackle unseen slots in zero-shot slot filling. We speculate that the inference paths between slots and values may be one of the implicit semantic relation cues and then investigated an amount of inference paths in the SnipsNLU dataset via the knowledge graph. The inference relation paths (IRPs) are found to implicitly build up semantic relations between slots and their values. Accordingly, we proposed a method to utilize the IRPs for zero-shot slot filling. Experimental results showed that the proposed method outperformed a strong baseline by 3.61 in terms of F1 score and achieved an improvement rate of 40% for semantically dissimilar unseen slots. These results demonstrate that the proposed method can provide effective share information for handling unseen slots.**

I. Introduction

Building task-oriented dialogue systems has become a hot topic in research and industry [1]. Spoken language understanding (SLU) is a key and intermediate component that connects input utterances to subsequent modules in dialogue systems. Therefore, the performance and generalization capacity of SLU affect the performance of the whole dialogue system.

Slot filling is an indispensable procedure in SLU that aims to extract semantic components from user's utterances and fill them into corresponding slots. Slot filling is usually treated as a sequence tagging task in practice, in which the inside-outside-beginning (IOB) tags are used to indicate the slot filling parts in the utterance. Figure 1 shows an example of slot filling for a given utterance 'Will it be windy at 4 pm in NY'. As shown in Figure 1, the slot filling procedure predicts IOB-slot tags for each token in the utterance, these tags indicate that the 'windy', '4 pm' and 'NY' are filled into the slots condition description, time range and state, respectively.

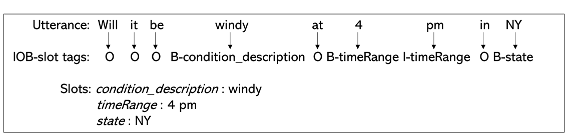

Fig. 1. Example of slot filling with IOB tags

Slot filling models have been remarkably improved due to deep learning [2, 3, 4, 5, 6]. However, the existing systems usually require to add some new domains for extending topics in practice. This demands a large amount of labeled data for adapting a slot filling model to new domains with typical supervised approaches, which is not always efficient and feasible. To handle unseen slots in new domains and boost the domain adapting, zero-shot approaches have been proposed [7, 8, 9, 10]. Zero-shot slot filling aims to train a model on source domains and adapt the model to handle target domains directly. Early works [10] used similarity representations to avoid retraining models for domain adaption. Recently, the concept tagger model (CT) [11] was proposed as a framework to predict slot spans within utterances for each slot with semantic slot descriptions. The zero-shot adaptive transfer network (ZAT) [13] further added an attention mechanism to improve the model generalization. Shah et al. [14] proposed a method to learn a model with slot descriptions and example slot values, which showed the usefulness of the example values.

In zero-shot slot filling, the prediction for each slot relies on two types of information: the context information from utterances and the slot information based on slot descriptions. The context information is not controllable since it depends on users or given data. Therefore, previous works mainly rely on semantic descriptions of slots for providing share information across domains, which enables the zero-shot slot filling model to deal with unseen slots in target domains. This kind of model can handle a slot in target domains if the slot is semantically similar to some others in the source domains. As pointed out by Bapna et al. [11], however, a method relying on conventional slot descriptions alone cannot deal well with the unseen slots that are less semantically similar to any other slot in the source domains due to lacking share information.

To find share information to tackle the problem mentioned above, we move our sights to how people estimate objects with less semantic similarity. In practice, people often use many implicit semantic relations as estimation cues, such as we can infer the situation of a golf ball from the kind of the golf clubs that professional golfers are using. The club here is an implicit sematic relation carrier, providing the information for estimating the ball's situation. Inspired by this, we attempt to find out a semantic relation carrier that can provide the semantic relation information between slots and their values and try to utilize such a semantic relation carrier to estimate unseen slot values in zero-shot slot filling. We speculate the inference path between a slot and its values might be an implicit semantic relation carrier such as the functions of different golf clubs in a golf game. In this study, our investigation proved the speculation. Accordingly, we propose a method to utilize the inference relation paths (IRPs) between a slot and its values to describe the slot information and then to tackle unseen slots in zero-shot slot filling.

The contributions of this study are as follows: (1) We found out that the IRPs implicitly carry semantic relations between slots and their values and function as additional semantic features for slots. (2) We proposed a method to utilize the IRPs as slot descriptions into zero-shot slot filling task. This enables a model to estimate unseen slot values considering the implicit semantic relations between slots and values. (3) The proposed method improves the zero-shot slot filling performance more than the leading baseline, especially on the unseen slots with less semantic similarity to the training slots.

Section 2 describes the IRPs in the knowledge graph. Section 3 investigates the sematic function of the IRPs between slots and their values in a widely used dataset. Section 4 describes the model construction and the experiment setting. Section 5 gives the results, and Section 6 concludes the paper.

II. Inference Relation Paths

In this study, we try to find out implicit semantic relation cues between slots and values to tackle unseen slots in zero-shot slot filling. Since the inferences are often utilized for obtaining semantic relations between the intent and target, analogically, the inference paths between slots and values may have implicit semantic relations. The necessary components to describe the inference between slots and values are the knowledge of the entity and its relations. Therefore, we focus on the descriptions of the entity knowledge and the relations of entities within the knowledge graph.

In the knowledge graph, entities are described as nodes, the edges connecting entity nodes represent the relations between entities. Generally, these relations are pre-defined rules of inference mappings in the knowledge graph, such as 'instance of' and 'subclass of'. Entity nodes are usually used as explicit semantic information in previous works [20, 21] but the semantic information carried by relations are not paid much attention to. To preliminarily verify our speculation that the inference paths between slots and their values could carry implicit semantic relations, we first extract a number of inference paths from a multi-domain dataset SnipsNLU [12] via large-scale knowledge graph Wikidata. Fig. 1 shows two examples of the inference paths from values to corresponding slots through entity nodes and relations. One is from the value 'Pelham Bay Park' to the slot 'POI'; the other one is from the value 'Guernsey' to the slot 'country'.

From the inference paths in Fig. 1, we found that although the entity nodes on the inference path are different, the IRPs carried the same analogic reasoning. The figure demonstrates that the same IRPs can correlate slots to the values by similar semantic relations. For instance, in Fig. 1, the IRP [instance of, subclass of, subclass of] correlates slots to the values 'Pelham Bay Park' and 'Guernsey', which represent place names. Therefore, a certain IRP may implicitly correspond to certain semantic information, correlating slots to certain values. A statistical analysis is conducted in the following section.

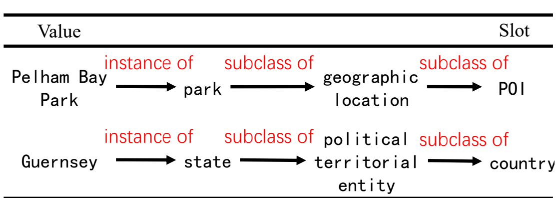

Fig. 2. Examples of inference paths from values to corresponding slots in the SnipsNLU dataset

III. Analysis for the Inference Relation Paths

To testify to the generality of an IRP carrying certain semantic information implicitly, we analyze an amount of IRPs between the slots and their values in a widely-used corpus, the SnipsNLU dataset. SnipsNLU contains seven domains with various slots, some of the slots describes entities in the real world, such as 'city' and 'album'. Some of the slots describes concepts, such as 'genre' and 'time range'. To analyze IRPs in the SnipsNLU, we first build an ontology containing available slots and their values for the SnipsNLU dataset. We extract slots and corresponding values from the annotation data of the SnipsNLU dataset and randomly select up to 30 values for each slot to balance the values of each slot. s. As a result, the ontology of the SnipsNLU contains 53 available slots out from the dataset, each of which has up to 30 values.

Then we try to identify the entities of the 53 available slots and their values in the ontology via Wikidata to obtain the IRPs between slots and their values. Some slots cannot be identified as entities directly, we explore the entities in the knowledge graph with similar semantic meanings to the slots, and then use the new-found entities as the slot entities for the slots to extract IRPs in subsequent processing. For instance, the slot 'served dish' cannot be identified as an entity directly, but we find that the entity 'dish' has similar semantic meaning to the slot 'served dish', so we use the entity 'dish' to extract the IRPs for slot 'served dish'. Then, depending on whether the slot value is an entity or not, we separate the IRPs into two groups. (1) If at least one value of a slot can be identified as entities in Wikidata, we manually extract the IRP with the shortest inference steps from the slot entity to a corresponding value entity. (2) For some slots, no corresponding values can be identified in Wikidata. For instance, the values of 'playlist owner' are all possessive pronouns that could not be identified to be entities. We set the IRPs as [instance of] between such slots and their values. After this processing, the slots in Group (1) have multiple kinds of IRPs corresponding to their values, and the slots in Group (2) have only one kind of IRPs. Among 53 available slots in the ontology built for SnipsNLU, 46 slots belong to Group (1) and 7 slots belong to Group (2).

For analyzing the extracted IRPs, we convert the values corresponding to each IRP into vector representations by word embeddings. Then we clarify the distributions of the IRP-corresponded values in the semantic space. To explain the distributions intuitively, we reduce their dimensions using the t-distributed Stochastic Neighbor Embedding (t-SNE) algo-

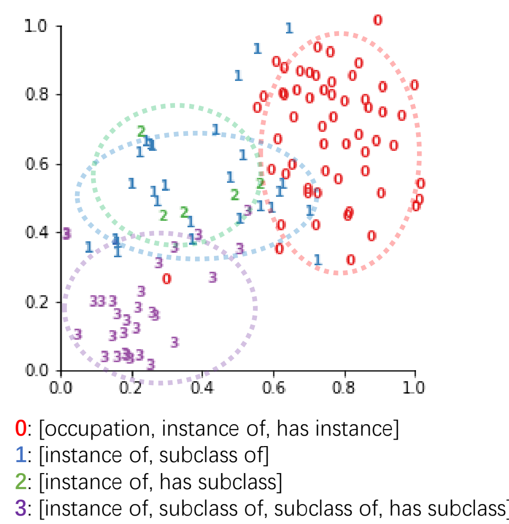

Fig. 3. Visualization example of value distributions in semantic space corresponding to four IRPs

rithm and show the 2-D distributions in a semantic space. We select the four IRPs, referred to as paths 0 3. Specifically, path 0 is [occupation, instance of, has instance], path 1 is [instance of, subclass of], path 2 is [instance of, has subclass], and path 3 is [instance of, subclass of, subclass of, has subclass]. Accordingly, the paths can be treated as three groups: path 0, path 3, and path 1&2. Fig. 2 shows an example of the value distribution of the four IRPs. In the Fig. 2, the values corresponding to path 0, path 3, and path 1&2 are distributed separately as three different clusters in the semantic space, where the values corresponding to paths 1 and 2 are overlapped heavily since paths 1 and 2 can be regarded as the same path since they have only one different step while other steps are identical.. This confirms that the IRPs carries implicit semantic information correlating the slot and its corresponding values. Therefore, the IRPs can be regarded as semantic cues that implicitly carry semantic relation information of slots and their values. Similar to estimating the ball's situation on the basis of the golf club the professional golfer is using, we can estimate the slot values on the basis of the IRPs. Therefore, the IRP can be used to provide share information for estimating unseen slot values.

Further analysis shows that the IRPs of each slot can be grouped into several patterns for efficient utilization in practice. For instance, the slot 'artist' in the 'AddToPlaylist' domain has 30 corresponding values. Among them, 28 values have the same IRP of [occupation, instance of, has instance] to 'artist', while the other 2 values have different IRPs to 'artist', which is [instance of, subclass of, subclass of, has part]. Accordingly, the IRPs of 'artist' can be grouped into two IRP patterns for efficient utilization. To investigate whether or not the same IRP patterns exist in different domains could provide share information across domains, for each domain, we count the number of shared slot IRP patterns between that domain and other domains and show them in Table I. Note that these numbers do not reflect the amount of share information provided by the IRP patterns since the effectiveness of share information is task-dependent. In Table I, unseen and seen show the number for unseen and seen slots in a domain, respectively. One can see that for unseen and seen slots, numerous identical IRP patterns exist in different domains. The cross-domain IRPs provide share information in practice for tackling unseen slots

TABLE I
THE SAME IRP PATTERNS AMONG SLOTS IN ALL DOMAINS IN SNIPSNLU DATASET

Domains	Unseen	Seen
AddToPlaylist	5	7
BookRestaurant	28	17
GetWeather	8	10
PlayMusic	19	11
RateBook	4	7
SearchCreativeWork	0	14
SearchScreeningEvent	5	7

IV. EXPERIMENT

To utilize the IRP for handling unseen slots, we propose a model to compliment IRP in zero-shot slot filling and conduct experiments on a widely used dataset SnipsNLU.

A. Model construction

So far, predicting inside-outside-beginning (IOB) tags on given utterances for each possible slot has been shown to be suitable for zero-shot slot filling [11, 13]. The IOB prediction indicates the spans within the utterance for the given slot. Our model employs the same paradigm to predict IOB tags for

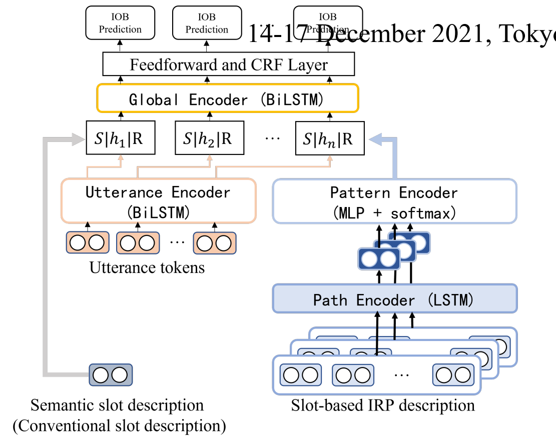

Fig. 4. Model construction utilizing slot-based IRP description for zero-shot slot filling

each slot. Fig. 3 shows the structure of the proposed model. The semantic slot description (conventional slot description) denotes the mean of the slot tokens' semantic representations of a given slot. The utterance tokens mean the semantic representations of the tokens in a given utterance. The slot-based IRP description contains the IRP patterns of a given slot. To obtain the representation for the specific relations in the IRP patterns, we apply pre-trained knowledge graph embeddings [15, 16, 17] to convert the specific relations in the knowledge graph into vector representations. In this study, we apply a TransE model [15, 18], which is pre-trained on Wikidata, to obtain embedding vector representations.

In the model, we encode a given utterance using a bidirectional long shot-term memory (LSTM) network. The output of the utterance encoder at the i-th time step represents the context information of the i-th token. For a given slot, each IRP pattern in the slot-based IRP description provides implicit semantic relation information for estimating how context information is related to the given slot. To utilize the implicit semantic relation information into the model, we use an LSTM to encode the path of each IRP pattern separately. The last hidden state of the path encoder is used as the path encoding for the corresponding IRP pattern. Then the pattern encoder takes the path encodings as inputs and uses a self-attention mechanism to encode the path encodings into one vector, such a vector represents the combining information of all IRP patterns of the given slot. The self-attention mechanism is computed as:

$$P = softmax(MLP(V))^T \cdot V \qquad (1)$$

where V is the path encodings of IRP patterns, $MLP(\cdot)$ is a fully connected layer, $softmax$ is the softmax function, T indicate the transpose of $softmax(MLP(V))$ and P is the output of the pattern encoder. We concatenate the utterance context information, the pattern encoder output P and the semantic slot description S as a general variable at each time step i and input it into the global encoder. In accordance with the finding that the CRF layer is useful for sequence-related tasks [12, 22], we use a conditional random field (CRF) layer following a feedforward layer to take the output of the global encoder as input and gives IOB tag predictions for the given slot. The model final output for slot filling is the merge of the spans indicated by IOB tags for all possible slots.

B. Experiment setting

In this study, we conduct the experiment on the SnipsNLU [12] dataset. The SnipsNLU is a widely used dataset for SLU containing 7 domains, each domain has 2000 utterances collected from real applications. We train a model by setting one domain as the target domain for zero-shot test and the other six domains as source domains for training. In total, we train seven models independently by setting each domain as the target domain and evaluate the model performance one by one. In the training process, data from source domains is merged for training and validation, while data from the target domain is used for the test alone. Models are trained with utterances and possible given slots. The training instances are divided into positive instances if the given slot occurs in the utterance, or into negative instances if the given slot does not occur. To make a fair comparison, we randomly sample positive and negative instances in a ratio of 1:3, which is the same as previous works [12, 13].

For comparison, CT [11] and ZAT [12] are used as baselines for evaluation. For embedding representations, we use nnlm-en-dim128 word embeddings for word tokens and use dim-100 TransE pre-trained embeddings for relation representations. We set 200 hidden units for the utterance encoder and the global encoder and set 128 hidden units for the path encoder in our model. The cross entropy loss function is used to compute the loss among IOB predictions. The Adam optimizer is used for optimizing with learning rate of 0.0005. We conduct the experiments three times and give an average performance. The conlleval script [23] is used to compute the slot F1 score for evaluation metrics.

V. Result and discussion

A. Experiment result

Table II shows the results of different model performances on each domain. The average is the average result over all domains. The underlined numbers indicate the best result for each domain. From Table II, one can see that our model achieved better performance on most domains than all baselines, with an average F1 score 3.61 higher than that of the strong baseline ZAT model. These results demonstrate that the slot-based IRP descriptions are effective for providing share information across domains in zero-shot slot filling.

TABLE II
Results of different model performance for each domain of SnipNLU dataset

Domains	CT	ZAT	Ours
AddToPlaylist	28.94	37.66	44.62
BookRestaurant	24.54	34.05	34.62
GetWeather	42.73	55.82	60.05
PlayMusic	27.18	31.54	39.42
RateBook	20.56	19.47	20.28
SearchCreativeWork	65.95	73.46	74.29
SearchScreeningEvent	24.57	33.05	37.04
Average	33.50	40.72	44.33

B. Discussion

To clarify the effectiveness of slot-based IRP descriptions on unseen and seen slots, we compare the F1 score on each slot between the proposed method and ZAT and show the comparison in the appendix. To measure how an unseen slot is semantically similar to the slots in training domains, we define the max cosine similarity between the semantic representation of an unseen slot and the semantic representation of the slot in training domains as e max semantic similarity (MSS). The level of MSS reflects that the training domains can provide share information for estimating the unseen slot values in the target domain on the basis of semantic representations in some extent. On the basis of the value of the MSS, we divide unseen slots into two categories: the low-MSS unseen slot (MSS<0.5) and the high-MSS unseen slot (MSS≥0.5). We compare the F1 scores on slots in each category between the proposed method and ZAT.

Fig. 4 shows sector diagrams of the comparison between the proposed method and ZAT. The blue parts indicate the ratio of the slots that the proposed method performed better (R_{ours}), while the orange ones show that ZAT performed better (R_{ZAT}). To evaluate the performance of both methods, we introduced an improvement rate R_{imp} as following equation,

$$R_{imp} = R_{ours} - R_{ZAT} \tag{2}$$

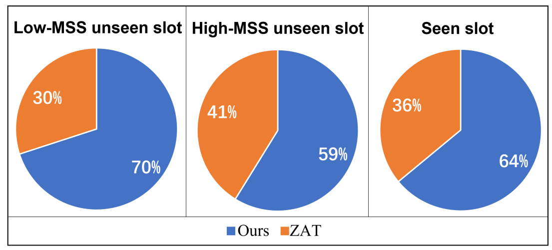

Fig. 5. Comparison of different methods on different slot categories

From the sector diagrams, one can see that the improvement rates are 40% for low-MSS unseen slots, 28% for seen slots, and 18% for high-MSS unseen slots. These results confirm that the slot-based IRP descriptions provided effective share information across domains for handling both unseen and seen slots, especially for the unseen slots with less semantic similarity to the slots in source domains. For the high-MSS unseen slots, the proposed method has a relatively smaller improvement over ZAT. The reason probably is that we simply utilized the mean of slot tokens' semantic representations as semantic slot descriptions in our model, which is not as exquisite as the semantic slot descriptions used in ZAT. To further improve the generalization of a zero-shot slot filling model for estimating unseen slots, our future work is to incorporate the proposed slot-based IRP descriptions with elaborated semantic slot descriptions.

To clarify the effectiveness of slot-based IRP descriptions on different types of slots, we further compare the F1 score on different slot types between the proposed method and ZAT. We divide slots into entity-type slots and abstract-type slots based on semantic meanings the slots describe. The entity-type slot describes entities or existence in the real world, such as 'album' and 'city'. The abstract-type slot describes concept semantic meaning, such as 'genre' and 'time range'.

Fig. 5 shows sector diagrams for a comparison. From the sector diagrams, one can see that the improvement rates are 60% and 50% for unseen and seen entity type slots, respectively, and 6% and 20% for unseen and seen abstract-type slots, respectively. These results demonstrate that slot-based IRP descriptions provide share information effectively on handling entity-type slots. Unlike that the entity relations obtained from the commonsense knowledge graph, abstract-type slots are not so effective to have the inference path. To deal with abstract-type slots in zero-shot slot filling more effectively, our future work is to utilize concept knowledge graph to describe abstract-type slots.

TABLE III
RESULTS OF ABLATION MODEL PERFORMANCE FOR EACH DOMAIN OF SNIPNLU DATASET, IRP ALONE SHOWS THE MODEL PERFORMANCE USING IRP DESCRIPTIONS ALONE; CONVENTIONAL METHOD SHOWS THE MODEL PERFORMANCE USING CONVENTIONAL SLOT DESCRIPTIONS ALONE

Domains	Our model (IRP+Conventional)	IRP alone	Conventional method
AddToPlaylist	44.62	43.72	37.93
BookRestaurant	34.62	12.42	33.56
GetWeather	60.05	35.72	59.32
PlayMusic	39.42	22.72	30.16
RateBook	20.28	6.92	18.80
SearchCreativeWork	74.29	33.11	68.08
SearchScreeningEvent	37.04	12.57	33.36
AVER	44.33	23.88 (-20.45)	40.17 (-4.16)

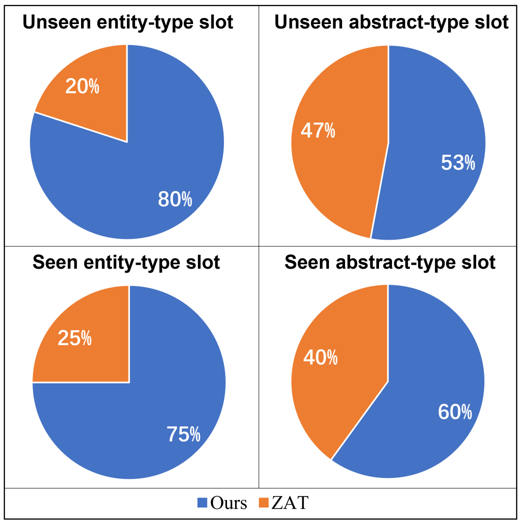

Fig. 6. Comparison of different methods on different types of slots

C. Ablation study

We conduct ablation experiments to clarify the effectiveness of the IRP descriptions and the conventional slot descriptions in our model. To do so, we train ablation models in the same way as described in the experiment setting and compare the model performance. The ablation models are trained using IRP descriptions alone and conventional slot descriptions alone. Table III shows the results of the proposed method and the ablation models. From Table III, one can see that without the explicit slot information, using IRP descriptions alone can achieve 23.88 in F1 score on average, which is over the half of the slot F1 score of the conventional model using explicit slot descriptions. This result demonstrates that the IRPs carry implicit relations between slots and their values and can provide share information across domains for zero-shot slot filling. The results also show that adding IRPs to the conventional model significantly improves the model performance by 4.16 in F1 score. These results demonstrate that the IRP slot descriptions and conventional slot descriptions have complementary effect on zero-shot slot filling.

VI. CONCLUSION

In this study, we analyzed the inference relation paths (IRPs) between slots and their values and found that the IRPs implicitly carry certain semantic relations, which function as additional sematic features for a slot. We implemented the IRP in a zero-shot slot filling model to tackle unseen slots. Experimental results demonstrated that the slot with IRP descriptions can provide effective share semantic information across domains for dealing with the unseen slots. The proposed method outperformed the strong baseline ZAT model in terms of both unseen and seen slot filling. The future work is to utilize the proposed slot-based IRP descriptions with more elaborated semantic slot descriptions and utilize concept-based knowledge graph to improve the model generalization on handling various unseen slots in zero-shot slot filling.

VII. ACKNOWLEDGEMENTS

This study is supported by JSPS KAKENHI Grant (20K11883).

REFERENCES

[1] Zhang Z, Takanobu R, Zhu Q, et al. Recent advances and challenges in task-oriented dialog systems[J]. Science China Technological Sciences, 2020: 1-17.
[2] Zhao L, Feng Z. Improving slot filling in spoken language understanding with joint pointer and attention[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2018: 426-431.
[3] Zhang X, Wang H. A joint model of intent determination and slot filling for spoken language understanding[C]//Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. 2016: 2993-2999.
[4] Yang X, Chen Y N, Hakkani-Tür D, et al. End-to-end joint learning of natural language understanding and dialogue manager[C]//2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 5690-5694.
[5] Liu B, Lane I. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling[J]. Interspeech 2016, 2016: 685-689.
[6] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//NAACL-HLT (1). 2019.
[7] Upadhyay S, Faruqui M, Tür G, et al. (Almost) zero-shot cross-lingual spoken language understanding[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 6034-6038.
[8] Xia C, Zhang C, Yan X, et al. Zero-shot User Intent Detection via Capsule Neural Networks[C]//Conference on Empirical Methods in Natural Language Processing. 2019.
[9] Kumar A, Muddireddy P R, Dreyer M, et al. Zero-Shot Learning Across Heterogeneous Overlapping Domains[C]//INTERSPEECH. 2017: 2914-2918.
[10] Ferreira E, Jabaian B, Lefevre F. Zero-shot semantic parser for spoken language understanding[C]//Sixteenth Annual Conference of the International Speech Communication Association. 2015.
[11] Bapna A, Tür G, Hakkani-Tür D, et al. Towards Zero-Shot Frame Semantic Parsing for Domain Scaling[J]. Proc. Interspeech 2017, 2017: 2476-2480.
[12] Lee S, Jha R. Zero-shot adaptive transfer for conversational language understanding[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33: 6642-6649.
[13] Coucke A, Saade A, Ball A, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces[J]. arXiv preprint arXiv:1805.10190, 2018.
[14] Shah D, Gupta R, Fayazi A, et al. Robust Zero-Shot Cross-Domain Slot Filling with Example Values[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 5484-5490.
[15] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[C]//Advances in neural information processing systems. 2013: 2787-2795.
[16] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]//Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. 2014: 1112-1119.

[17] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C]//Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). 2015: 687-696.

[18] Han X, Cao S, Lv X, et al. Openke: An open toolkit for knowledge embedding[C]//Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations. 2018: 139-144.

[19] Das R, Neelakantan A, Belanger D, et al. Chains of Reasoning over Entities, Relations, and Text using Recurrent Neural Networks[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. 2017: 132-141.

[20] Zhong P, Wang D, Miao C. Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations[C]//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: 165-176.

[21] Zhou H, Young T, Huang M, et al. Commonsense knowledge aware conversation generation with graph attention[C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2018: 4623-4629.

[22] Kumar, H., Agarwal, A., Dasgupta, R., & Joshi, S. (2018). Dialogue act sequence labeling using hierarchical encoder with CRF. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 3440–3447.

[23] Sang E T K, Buchholz S. Introduction to the CoNLL-2000 Shared Task Chunking[C]//Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop. 2000.

APPENDIX

Table A shows the comparison results of ZAT and the proposed method on each slot in different domains. The slots with '*' are unseen slots. The MSS of unseen slots indicate the max slot similarity that described in the section 5.2. According to the definition of the MSS, only unseen slots have MSS. The type shows the slot type, 'E' means the slot is an entity-type slot, 'A' means the slot is an abstract-type slot. The **Domain**/Slots indicates each domain and slots in the domain. The slot F1 score shows the comparison result of the ZAT model and the proposed method.

TABLE A
COMPARISON OF DIFFERENT METHODS ON EACH SLOT

MSS of unseen slots	Type	**Domain/Slots**	Slot F1 score	
		AddToPlaylist	ZAT	Ours
-	E	artist	47.14	55.47
0.73	E	entity name*	8.75	9.26
-	A	music item	84.06	86.75
-	E	playlist	31.75	47.14
0.72	A	playlist owner*	0.43	0.00
		BookRestaurant		
-	E	city	66.64	68.96
-	E	country	69.81	83.92
0.35	A	cuisine*	0.00	0.26
0.45	E	facility*	2.32	0.68
0.65	A	party size description*	2.46	0.00
0.49	A	party size number*	0.06	0.00
0.68	E	poi*	4.09	3.93
0.8	E	restaurant name*	16.30	18.45
0.69	A	restaurant type*	1.92	0.08
0.32	E	served dish*	2.01	7.50
-	A	sort	30.16	14.73
-	A	spatial relation	68.09	60.68
-	E	state	84.13	88.47
-	A	timeRange	45.33	53.95
		GetWeather		
-	E	city	63.38	70.74
0.65	A	condition description*	0.28	0.00
0.42	A	condition temperature*	0.00	4.39
-	E	country	52.52	71.12
0.62	A	current location*	0.00	0.19
0.68	E	geographic poi*	7.61	9.19
-	A	spatial relation	64.83	77.08
-	E	state	79.95	73.48
-	A	timeRange	83.78	82.76
		PlayMusic		
0.62	E	album*	0.62	8.15
-	E	artist	57.49	67.56
0.52	A	genre*	3.55	2.82
-	A	music item	61.56	69.76
-	E	playlist	6.60	9.37
0.44	E	service*	3.74	15.10
-	A	sort	41.27	46.14
0.35	A	track*	1.14	0.64
0.39	A	year*	0.08	8.37
		RateBook		
0.23	A	best rating*	0.00	0.00
-	E	object name	45.25	35.31
0.69	A	object part of series type*	0.77	1.40
0.65	A	object select*	0.00	18.84
-	A	object type	61.01	43.95
0.45	A	rating unit*	0.42	3.34
0.4	A	rating value*	1.53	10.28
		SearchCreativeWork		
-	E	object name	89.30	87.08
-	A	object type	40.06	51.46
		SearchScreeningEvent		
0.8	E	location name*	37.76	43.80
0.72	E	movie name*	40.41	40.92
0.65	A	movie type*	2.63	15.61
0.9	A	object location type*	21.25	14.75
-	A	object type	0.08	0.00
-	A	spatial relation	65.81	74.35
-	A	timeRange	84.24	83.76