Dual Class Knowledge Propagation Network for Multi-label Few-shot Intent Detection

Feng Zhang[1] **Wei Chen**[1,2,3*] **Fei Ding**[4,5] **Tengjiao Wang**[1,2,3]
[1]Key Lab of High Confidence Software Technologies (MOE), School of Computer Science, Peking University, Beijing, China
[2]Research Center for Computational Social Science, Peking University
[3]Institute of Computational Social Science, Peking University (Qingdao)
[4]School of Intelligence Science and Technology, Peking University
[5]Institute for Artificial Intelligence, Peking University
{zhangfeng,dingfei}@stu.pku.edu.cn, {pekingchenwei,tjwang}@pku.edu.cn

Abstract

Multi-label intent detection aims to assign multiple labels to utterances and attracts increasing attention as a practical task in task-oriented dialogue systems. As dialogue domains change rapidly and new intents emerge fast, the lack of annotated data motivates multi-label few-shot intent detection. However, previous studies are confused by the identical representation of the utterance with multiple labels and overlook the intrinsic intra-class and inter-class interactions. To address these two limitations, we propose a novel dual class knowledge propagation network in this paper. In order to learn well-separated representations for utterances with multiple intents, we first introduce a label-semantic augmentation module incorporating class name information. For better consideration of the inherent intra-class and inter-class relations, an instance-level and a class-level graph neural network are constructed, which not only propagate label information but also propagate feature structure. And we use a simple yet effective method to predict the intent count of each utterance. Extensive experimental results on two multi-label intent datasets have demonstrated that our proposed method outperforms strong baselines by a large margin.

1 Introduction

Multi-label intent detection aims to assign multiple labels to a single utterance rather than only one label since the user utterance often carries multiple different intents. As a practical task in real-world scenarios, multi-label intent detection attracts more attention with the rapid development of conversational AI (Louvan and Magnini, 2020). Previous works have achieved promising performance in multi-label intent detection, but they rely on large amounts of labeled training samples (Xu and Sarikaya, 2013; Qin et al., 2020a,b, 2021). However, due to fast-emerging intents and rapidly changing domains, the lack of adequate labeled data for each intent is inevitable, thus motivating the multi-label few-shot intent detection (Hou et al., 2021). Few-shot learning, a promising way to tackle data scarcity challenges, can recognize new classes from only a few labeled samples by exploiting the prior knowledge from source domain.

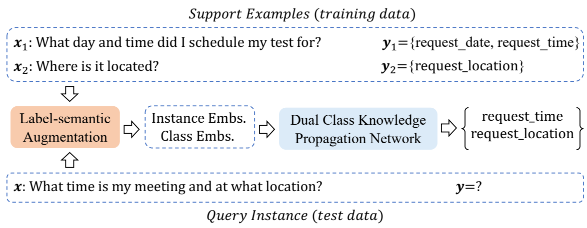


Figure 1: An example of 3-way 1-shot multi-label intent detection.

Recent works for multi-label intent detection mainly focus on threshold-based strategy which consists of relevance scoring and threshold estimation (Gangadharaiah and Narayanaswamy, 2019; Hou et al., 2021) to improve performance. Nevertheless, they suffer from the following two limitations. First, for multi-label utterances, most of them neglect the semantic connections between labels and input utterances and therefore learn the identical utterance representations for different labels, which confuses the similarity scoring. As Figure 1 shows, the embeddings for *request_date* and *request_time* are the same, i.e., the embedding of utterance x_1. Furthermore, in few-shot scenarios, it's critical to obtain proper intra-class and inter-class relations from only a few samples, while previous works overlook the high-order interaction of samples and just usually use a specific metric to compute the relevance score directly, e.g., cosine similarity.

* Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8605–8618

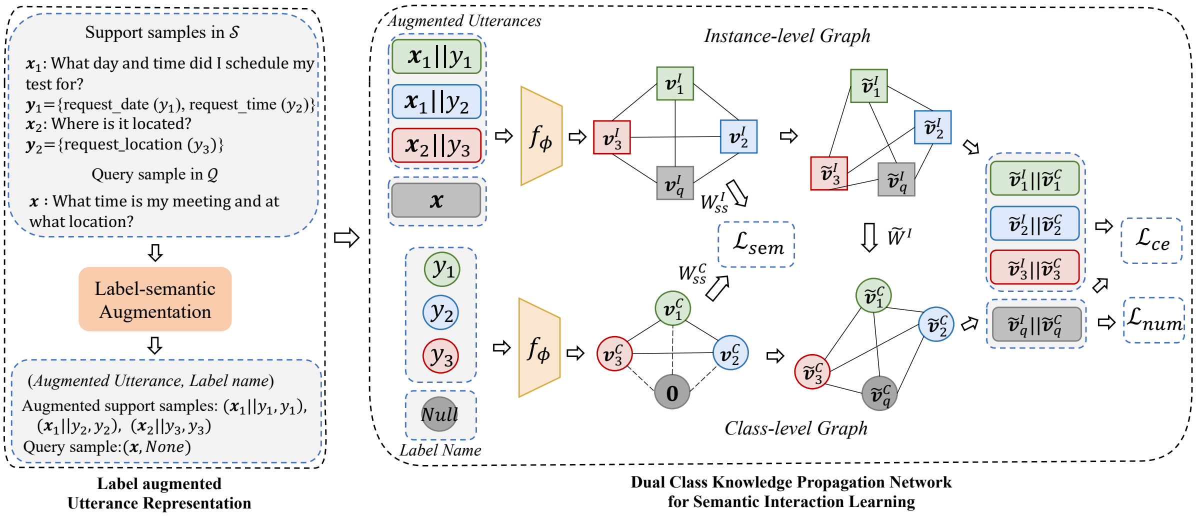


Figure 2: The architecture of the proposed framework DCKPN.

In this paper, we propose a novel framework for multi-label few-shot intent detection, as illustrated in Figure 2. It obtains well-separated class representations by fully exploiting the label semantic information and propagates class knowledge and feature structure by graph neural network to reduce intra-class variance and model proper inter-class relations. To deal with the confused embeddings for multi-label utterances, we introduce label-semantic augmentation (LA) to obtain well-separated representations without any complicated model architecture. Inspired by the effectiveness of label name information (Luo et al., 2021), LA appends each class label name after utterances to construct multiple label-specific utterances, where each label-specific utterance is corresponding to only one label. In this way, LA not only eliminates ambiguity but also generates more labeled training samples.

To better capture more complete and more adaptive interactions among samples, we propose a novel dual class knowledge propagation network, containing an instance-level graph and a class-level graph. It not only propagates label information from labeled support samples to unlabeled queries but also propagates feature structure in their latent space. The instance-level graph smoothing the representations among highly related utterances reduces the intrinsic gap between augmented support samples and queries. The class-level graph provides label correlation and generates completed pseudo class-level representations for queries with the guidance of the instance-level graph. Also, we design a calibration strategy to transfer label dependency into instance representations. Then the two representations are fused to classify queries. Finally, to select accurately multiple intents for an utterance, we employ an adaptive and powerful method to predict the number of intents.

The contributions of this paper are as follows: (1) We propose a novel dual class knowledge propagation network (DCKPN). DCKPN incorporates label semantic connections to guide feature propagation and label information propagation. (2) We introduce a novel label-semantic augmentation method for multi-label samples to obtain more discriminative representations. And we employ a simple inference module to predict the number of intents in utterances adaptively. (3) To verify the effectiveness of our proposed model, we conduct a series of experiments on two datasets. The empirical study shows that our model can achieve state-of-the-art performance in comparison with other strong baselines.

2 Related Work

Multi-label Classification The existing multi-label classification (MLC) methods mainly focus on learning enhanced representations (Liu et al., 2017) and modeling label dependency (Yang et al., 2018; Tsai and Lee, 2020). Recently, some works (You et al., 2019; Xiao et al., 2019) have been proposed to learn a label-specific representation by employing attention mechanism or utilize the label statistical co-occurrences to explore the semantic interactions (Ma et al., 2021), which shows the importance of exploring semantic interactions for MLC. However, these methods require large amounts of labeled samples for training, thus it's hard to be applied to fast-emerging new domains in few-shot scenarios.

Few-shot Learning Few-shot learning aims to recognize novel classes from a handful of annotated samples by leveraging the previous knowledge from source domains (Fei-Fei et al., 2006). Recently, meta-learning has become a successful paradigm for solving the few-shot problem, which can be divided into three categories: model-based methods (Munkhdalai and Yu, 2017), optimization-based methods (Finn et al., 2017; Yoon et al., 2018) and metric-based methods (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). However, these methods focus on single-label classification and only a few works investigate multi-label few-shot classification. Previous studies focus on image domain (Alfassy et al., 2019), audio domain (Cheng et al., 2019) and sentiment analysis (Hu et al., 2021). CTLR (Hou et al., 2021) is the first work to address multi-label few-shot intent detection tasks which proposes a meta-calibrated threshold mechanism and learns anchored label representation. However, it relies on a specific metric to estimate the relevance score, losing the inherent relations among samples and classes.

Propagation in Graph Model Graph neural networks are effective in modeling the relationship among different nodes and have shown powerful performance in many real applications, e.g., nodes classification (Kim et al., 2019) and link prediction (Zhang and Chen, 2018). In few-shot scenarios, several approaches have been explored. Specifically, DPGN (Yang et al., 2020) constructs both the distribution-level and instance-level graph. Liu et al. (2019) propose a transductive propagation network (TPN) by utilizing label propagation (Zhou et al., 2003), a classical and effective method to transfer knowledge from neighbors of each node. Rodríguez et al. (2020) yield a smoother embedding space by employing embedding propagation. Inspired by this, our method aims to propagate information both in intra-class and inter-class, called dual class knowledge propagation network (DCKPN).

3 Proposed Method

3.1 Problem Definition

Few-shot learning aims to train a model that can perform well in cases where only a few samples are given. We follow the episodic paradigm, an

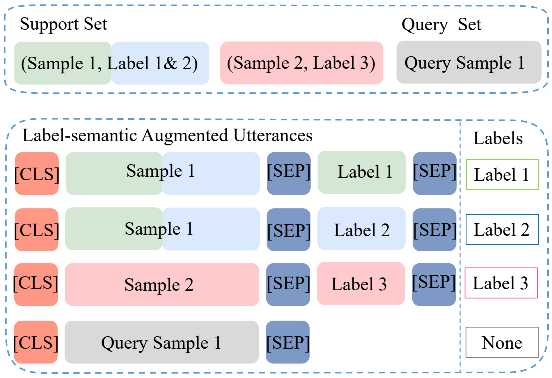


Figure 3: Illustration of a simple 3-way 1-shot task.

effective solution for few-shot learning, which is commonly employed in various studies (Snell et al., 2017; Yang et al., 2021). In general, the model is first trained on a series of tasks from source domains with numerous samples ($\mathcal{D}_{train}$) episode by episode until convergence, then directly adopted to another set of unseen novel tasks ($\mathcal{D}_{test}$), where a few labeled samples are available. There is no overlap between the training classes and testing classes. Specifically, each episode (task) has a support set $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N\times K}$, which usually includes N classes with K labeled samples for each class, i.e., the N-way K-shot setting, and a query set $\mathcal{Q} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{q}$, where q is the number of query samples.

Multi-label few-shot intent detection enables each utterance to be assigned with a set of intent labels simultaneously. To be more specific, for each example $(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{x}$ is the utterance and the label vector is denoted as $\boldsymbol{y} = \{y^1, y^2, ..., y^{|\mathcal{C}|}\} \in \mathbb{R}^{|\mathcal{C}|}$, where $\mathcal{C}$ is the set of intent labels in the episode and $y^i \in \{0, 1\}$. In this paper, we focus on multi-label few-shot intent detection, i.e., selecting multiple associated intent labels for each utterance in few-shot scenarios.

3.2 Label-semantic Augmentation

For multi-label few-shot intent detection tasks, one of the key points is to learn more discriminative class representations which could capture accurate class-relevant information while eliminating ambiguity via a few examples. To achieve this goal, we propose to use label name to augment the support set. Different from Luo et al. (2021), which have explored integrating label information into sentences, we propose to append each class label name after each sentence with multiple possible labels. As Figure 3 shows, for each utterance in support set, we append its labels one by one after it, i.e., "[CLS] utterance [SEP] label name [SEP]", thus generating different label-specific utterances corresponding to single different class labels. For each sample in query set, due to unknown labels, we take the common format "[CLS] utterance [SEP]" as input text.

There are two benefits of using the label name to augment utterances. First, we make use of different appended label name to distinguish the same utterance with multiple labels, thus obtaining more discriminative representations for each class. As illustrated in Figure 3, we can obtain different prototypes for class 1 and class 2, eliminating the problem that the multi-label utterances share the same representation essentially without any complex architecture or parameters. Second, this enables us to utilize the label name information as well as increase the number of support samples to eliminate the ambiguity caused by the scarcity of utterances. In Section 4.4, we analyze the experimental results to verify the improvements of augmentation.

3.3 Class Knowledge Propagation Network

Although one can directly calculate class prototypes via label-semantic augmented support set to classify the queries, doing so will result in the loss of intra-class and inter-class information, which is critical for multi-label classification tasks (Zhang et al., 2018). To mitigate these issues, we propose a dual class knowledge propagation network (DCKPN) to propagate class knowledge from support samples to queries. As illustrated in Figure 2, we first utilize the support and query samples to build the instance-level graph $\mathcal{G}^I = (V^I, W^I)$ and utilize label name information to build the class-level graph $\mathcal{G}^C = (V^C, W^C)$. Notably, in the following section, the support samples are label-semantic augmented samples described in Section 3.2, which means that each augmented support sample in instance-level graph has its corresponding single label in the class-level graph. Then, we modify the instance-level embeddings based on pairwise node relations. Guided by the class-level graph, we also calibrate relations of support-support nodes to integrate label semantic relevance explicitly. Next, we update class-level representations based on instance relations to generate pseudo query class-level representations. Finally, the two representations are merged to guide the inference of the queries.

3.3.1 Graph Construction

For each sample $(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{a}_i) \in \mathcal{S}$ in augmented support set, we employ a pre-trained language model as the backbone to extract its instance-level feature $\boldsymbol{v}_i^I = f_\phi(\boldsymbol{x}_i) \in \mathbb{R}^d$ and its class-level feature $\boldsymbol{v}_i^C = f_\phi(\boldsymbol{a}_i) \in \mathbb{R}^d$, where ϕ is the parameters of the backbone model and $\boldsymbol{a}_i$ is the label name of utterance $\boldsymbol{x}_i$. Due to unknown query labels, for each query sample $\boldsymbol{x}_j \in \mathcal{Q}$ in query set, we regard $\boldsymbol{v}_j^I = f_\phi(\boldsymbol{x}_j) \in \mathbb{R}^d$ as its instance-level feature and $\mathbf{0} \in \mathbb{R}^d$ as its class-level representation. Then, the adjacency matrix is defined as follows:

$$W_{ij} = \exp(-\frac{d(\boldsymbol{v}_i, \boldsymbol{v}_j)}{\sigma^2}), W_{ii} = 0, \quad (1)$$

where $\sigma^2 = Var(d(\boldsymbol{v}_i, \boldsymbol{v}_j))$ and $d(\boldsymbol{v}_i, \boldsymbol{v}_j) = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$. Thus, we obtain the instance-level graph $\mathcal{G}^I = (V^I, W^I)$, where $V^I \in \mathbb{R}^{(|\mathcal{S}|+|\mathcal{Q}|)\times d}$, and $W^I \in \mathbb{R}^{(|\mathcal{S}|+|\mathcal{Q}|)\times(|\mathcal{S}|+|\mathcal{Q}|)}$. While for class-level graph $\mathcal{G}^C = (V^C, W^C)$, $V^C \in \mathbb{R}^{(|\mathcal{S}|+|\mathcal{Q}|)\times d}$ consists of the feature embeddings of the label names in the support set $V_s^C \in \mathbb{R}^{|\mathcal{S}|\times d}$ as well as a matrix of zeros $\mathbf{0} \in \mathbb{R}^{|\mathcal{Q}|\times d}$ for queries, and $W^C \in \mathbb{R}^{(|\mathcal{S}|+|\mathcal{Q}|)\times(|\mathcal{S}|+|\mathcal{Q}|)}$ is as follows:

$$W^C = \begin{pmatrix} W_{ss}^C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \quad (2)$$

3.3.2 Instance-level Propagation

We now perform message passing on the instance-level graph to conduct higher-order interactions between augmented support embeddings and query embeddings. Specifically, we first normalize the weight matrix W^I as,

$$L^I = D^{-1/2} W^I D^{-1/2}, D_{ii} = \sum_j W_{ij}^I. \quad (3)$$

Then, using the propagation solution described in (Zhou et al., 2003), we calculate the propagation matrix and rectify the instance-level embeddings as,

$$\widetilde{V}^I = (I - \alpha L^I)^{-1} V^I, \quad (4)$$

where I is the identity matrix and $\alpha \in \mathbb{R}$ is a smoothing factor.

To further capture the inter-class relationships and reduce the intra-class variance, we utilize the guidance of relations of support-support pairs in class-level graph to calibrate the support-support pairs in instance-level graph, thus further encouraging the relation learning of instances with semantically related classes. For example, *request_temperature* share more semantic information with *request_low_temperature* than with *appreciate*. We implement the semantic loss by calibrating the pair relation in support set:

$$\mathcal{L}_{sem} = \frac{1}{|W_{ss}^C|} \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} |w_{ij}^C - w_{ij}^I|. \tag{5}$$

3.3.3 Class-level Propagation

As illustrated in Figure 2, the instance information flows back into the class-level graph to generate the completed semantic knowledge representations for both support set and query set. Concretely, we calculate the Laplacian of the updated instance embeddings $\widetilde{L}^I = \widetilde{D}^{-1/2}\widetilde{W}^I\widetilde{D}^{-1/2}$, where $\widetilde{D}_{ii} = \sum_j \widetilde{W}_{ij}^I$, $\widetilde{W}_{ij}^I = \exp(-d(\widetilde{\boldsymbol{v}}_i^I, \widetilde{\boldsymbol{v}}_j^I)/\widetilde{\sigma}^2)$, $\widetilde{\sigma}^2 = Var(d(\widetilde{\boldsymbol{v}}_i^I, \widetilde{\boldsymbol{v}}_j^I))$, and $\widetilde{\boldsymbol{v}}_i^I$ is obtained from Eq. (4). Next, the complete class-level embeddings are obtained as:

$$\widetilde{V}^C = (I - \alpha\widetilde{L}^I)^{-1}V^C, \tag{6}$$

where V^C is the original label name embedding matrix in class-level graph. Through the guidance of connectivity strength of instance pairs, we finally generate pseudo class representations for queries without labels. Thus, by explicitly establishing the bidirectional interaction between instance-level graph and class-level graph, we conduct feature propagation and derive completed representations to model intra-class and inter-class relationships.

3.4 Objective

Prototypical Classifier. Due to the two-level features being complementary, we concatenate the embeddings to obtain better feature representations. For each sample $\boldsymbol{x}_i$, the embedding is defined as follows:

$$\widetilde{\boldsymbol{v}}_i = \widetilde{\boldsymbol{v}}_i^I || \widetilde{\boldsymbol{v}}_i^C, \tag{7}$$

where $||$ denotes the concatenation and $\widetilde{\boldsymbol{v}}_i^I, \widetilde{\boldsymbol{v}}_i^C$ are sample $\boldsymbol{x}_i$'s instance-level and class-level embeddings respectively. The prototype $\widetilde{\boldsymbol{p}}_c$ of intent class $c \in \mathcal{C}$ can be calculated as $\widetilde{\boldsymbol{p}}_c = \frac{1}{|\mathcal{S}_c|}\sum_{\boldsymbol{x}_i \in \mathcal{S}_c} \widetilde{\boldsymbol{v}}_i$, where $\mathcal{S}_c$ denotes the set of support samples with intent class c and $\mathcal{C}$ is the class set. Given a query data $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{Q}$, we compute the conditional probability $p(y = c|\boldsymbol{x}_i, \mathcal{S})$ based on negative squared Euclidean distance:

$$p(y = c|\boldsymbol{x}_i, \mathcal{S}) = \frac{\exp(-d(\widetilde{\boldsymbol{v}}_i, \widetilde{\boldsymbol{p}}_c))}{\sum_{c' \in \mathcal{C}} \exp(-d(\widetilde{\boldsymbol{v}}_i, \widetilde{\boldsymbol{p}}_{c'}))}. \tag{8}$$

Finally, we perform cross-entropy loss on all query samples. Note that in the multi-label setting, as an utterance may have multiple labels, we need to consider $|\mathcal{C}|$ labels for each query sentence. The classification loss is calculated as:

$$\mathcal{L}_{ce} = -\frac{1}{|\mathcal{Q}|} \sum_{\boldsymbol{x}_i \in \mathcal{Q}} \sum_{j=1}^{|\mathcal{C}|} y_i^j \log p(y = j|\boldsymbol{x}_i, \mathcal{S}), \tag{9}$$

where $\boldsymbol{y}_i = \{y_i^1, ..., y_i^{|\mathcal{C}|}\}$ is the label and $y_i^j \in \{0, 1\}$.

Label Number Prediction. For multi-label few-shot intent detection, one of the challenges is to determine the number of intents in the utterance. We leverage a simple and effective method to decide the number of labels. Given an utterance $\boldsymbol{x}$, we can obtain its fused feature $\widetilde{\boldsymbol{v}}$, and then we use a multi-layer perceptron to predict the number of intents in $\boldsymbol{x}$:

$$\boldsymbol{n} = \mathrm{softmax}(\mathrm{MLP}(\widetilde{\boldsymbol{v}})), \tag{10}$$

where $\boldsymbol{n} \in \mathbb{R}^N$ is the indicator for the number of intents and N is the maximum count of possible intents. In meta-training stage, we calculate the cross entropy loss of intent count as follows:

$$\mathcal{L}_{num} = -\frac{1}{|\mathcal{Q}|} \sum_{\boldsymbol{x}_i \in \mathcal{Q}} \sum_{j=1}^{N} t_i^j \log(n_i^j), \tag{11}$$

where $\boldsymbol{t}_i = \{t_i^1, ..., t_i^N\}$ is the ground-truth intent count vector of $\boldsymbol{x}_i$ and $t_i^j \in \{0, 1\}$.

By combining Eqs. (5), (9) and (11), the overall loss of our proposed framework is:

$$\mathcal{L} = \mathcal{L}_{ce} + \beta\mathcal{L}_{sem} + \gamma\mathcal{L}_{num}, \tag{12}$$

where β and γ are adjustable weight parameters.

4 Experiments

4.1 Datasets and Setups

4.1.1 Datasets

We follow (Hou et al., 2021) to conduct experiments on two multi-label intent detection datasets: TourSG and StanfordLU. TourSG is a dataset of touristic information in Singapore, which includes 25751 samples annotated with multiple labels from 6 different domains: It (Itinerary), Ac (Accommodation), At (Attraction), Fo (Food), Tr (Transportation), and Sh (Shopping). StanfordLU is a Stanford dialogue dataset (Eric et al., 2017) and contains

	Model	It	Ac	At	Fo	Tr	Sh	Ave.
+E	TransferM	$14.34_{\pm 0.82}$	$14.75_{\pm 0.91}$	$16.13_{\pm 1.35}$	$11.79_{\pm 1.54}$	$13.64_{\pm 0.33}$	$14.32_{\pm 1.17}$	$14.16_{\pm 1.02}$
	MMN	$9.98_{\pm 1.80}$	$7.81_{\pm 0.70}$	$8.37_{\pm 0.86}$	$7.81_{\pm 0.35}$	$10.65_{\pm 1.62}$	$11.56_{\pm 0.79}$	$9.36_{\pm 1.02}$
	MPN	$12.24_{\pm 0.92}$	$10.38_{\pm 1.21}$	$10.00_{\pm 0.54}$	$10.47_{\pm 0.42}$	$13.61_{\pm 0.92}$	$11.41_{\pm 0.31}$	$11.35_{\pm 0.72}$
	CTLR	$39.98_{\pm 0.56}$	$51.55_{\pm 1.53}$	$55.16_{\pm 2.43}$	$52.16_{\pm 0.98}$	$55.36_{\pm 0.96}$	$52.20_{\pm 1.03}$	$51.07_{\pm 1.24}$
	DCKPN	$\mathbf{44.21}_{\pm 0.64}$	$\mathbf{55.91}_{\pm 0.35}$	$\mathbf{59.74}_{\pm 0.89}$	$\mathbf{56.55}_{\pm 0.83}$	$\mathbf{57.48}_{\pm 1.04}$	$\mathbf{55.71}_{\pm 0.82}$	$\mathbf{54.93}_{\pm 0.76}$
+B	TransferM	$16.78_{\pm 0.05}$	$18.62_{\pm 0.59}$	$14.92_{\pm 2.22}$	$16.40_{\pm 2.58}$	$15.68_{\pm 0.32}$	$14.50_{\pm 2.18}$	$16.15_{\pm 1.32}$
	MMN	$10.89_{\pm 3.35}$	$7.72_{\pm 1.44}$	$8.92_{\pm 1.45}$	$9.32_{\pm 1.40}$	$13.75_{\pm 0.70}$	$10.87_{\pm 4.31}$	$10.24_{\pm 2.11}$
	MPN	$13.77_{\pm 0.38}$	$12.38_{\pm 0.32}$	$13.46_{\pm 0.14}$	$10.23_{\pm 0.30}$	$16.19_{\pm 0.19}$	$15.79_{\pm 0.38}$	$13.64_{\pm 0.28}$
	CTLR	$44.58_{\pm 0.71}$	$57.11_{\pm 1.22}$	$60.34_{\pm 0.92}$	$56.49_{\pm 0.67}$	$60.18_{\pm 0.85}$	$55.60_{\pm 0.66}$	$55.72_{\pm 1.03}$
	DCKPN	$\mathbf{48.19}_{\pm 1.01}$	$\mathbf{58.32}_{\pm 0.68}$	$\mathbf{60.93}_{\pm 0.42}$	$\mathbf{58.22}_{\pm 1.36}$	$\mathbf{61.05}_{\pm 0.62}$	$\mathbf{57.62}_{\pm 0.64}$	$\mathbf{57.38}_{\pm 0.79}$

Table 1: The 1-shot average F1 scores of multi-label intent detection on TourSG dataset.

	Model	It	Ac	At	Fo	Tr	Sh	Ave.
+E	TransferM	$14.72_{\pm 0.53}$	$19.20_{\pm 1.59}$	$16.18_{\pm 1.03}$	$18.86_{\pm 1.04}$	$17.17_{\pm 1.19}$	$17.51_{\pm 1.63}$	$17.27_{\pm 1.17}$
	MMN	$14.11_{\pm 0.83}$	$10.58_{\pm 1.35}$	$17.80_{\pm 1.12}$	$12.74_{\pm 0.87}$	$18.01_{\pm 0.90}$	$16.76_{\pm 0.92}$	$15.00_{\pm 1.00}$
	MPN	$15.18_{\pm 0.63}$	$15.56_{\pm 0.54}$	$17.60_{\pm 1.15}$	$15.01_{\pm 0.19}$	$17.99_{\pm 0.36}$	$17.17_{\pm 1.09}$	$16.42_{\pm 0.66}$
	CTLR	$44.21_{\pm 0.71}$	$51.37_{\pm 1.22}$	$55.76_{\pm 0.92}$	$54.50_{\pm 0.58}$	$55.37_{\pm 0.95}$	$54.55_{\pm 0.86}$	$52.63_{\pm 0.87}$
	DCKPN	$\mathbf{47.76}_{\pm 0.73}$	$\mathbf{55.83}_{\pm 1.08}$	$\mathbf{59.48}_{\pm 0.32}$	$\mathbf{60.06}_{\pm 1.23}$	$\mathbf{60.23}_{\pm 0.63}$	$\mathbf{58.66}_{\pm 0.52}$	$\mathbf{57.00}_{\pm 0.75}$
+B	TransferM	$17.98_{\pm 1.80}$	$16.51_{\pm 1.95}$	$19.88_{\pm 4.17}$	$17.22_{\pm 3.01}$	$13.84_{\pm 1.40}$	$15.41_{\pm 2.81}$	$16.81_{\pm 2.52}$
	MMN	$15.65_{\pm 1.24}$	$16.42_{\pm 0.71}$	$19.90_{\pm 0.51}$	$12.23_{\pm 0.33}$	$16.81_{\pm 4.64}$	$17.13_{\pm 0.20}$	$16.36_{\pm 1.27}$
	MPN	$20.71_{\pm 0.98}$	$22.39_{\pm 1.95}$	$26.51_{\pm 0.72}$	$21.94_{\pm 1.59}$	$23.41_{\pm 1.31}$	$24.52_{\pm 3.31}$	$23.24_{\pm 1.64}$
	CTLR	$46.80_{\pm 0.83}$	$54.79_{\pm 0.80}$	$59.95_{\pm 0.46}$	$59.11_{\pm 0.39}$	$60.13_{\pm 0.44}$	$58.56_{\pm 0.30}$	$56.56_{\pm 0.54}$
	DCKPN	$\mathbf{49.58}_{\pm 0.72}$	$\mathbf{56.93}_{\pm 0.76}$	$\mathbf{60.65}_{\pm 0.48}$	$\mathbf{61.26}_{\pm 0.42}$	$\mathbf{60.89}_{\pm 0.42}$	$\mathbf{59.65}_{\pm 0.50}$	$\mathbf{58.16}_{\pm 0.55}$

Table 2: The 5-shot average F1 scores of multi-label intent detection on TourSG dataset.

Dataset	**TourSG**						**StanfordLU**		
Domain	It	Ac	At	Fo	Tr	Sh	Sc	Na	We
N_s	16	17	18	18	17	16	14	10	8
Prop. ML	23%	18%	16%	17%	18%	16%	21%	25%	4%

Table 3: The statistics of TourSG and StanfordLU datasets. N_s denotes the number of classes in each domain and *Prop. ML* denotes the proportion of multi-label utterances.

8038 utterances re-annotated by Hou et al. (2021) from 3 domains: Sc (Schedule), Na (Navigate) and We (Weather). Different from the standard N-way K-shot setting, an utterance may have multiple possible intents in multi-label few-shot intent detection and we refer you to (Hou et al., 2021) for more episode construction details. Table 3 shows the detailed dataset statistics.

4.1.2 Experiment Setups

Evaluation Metrics We follow (Hou et al., 2021) to use micro F1 scores to evaluate the performance. All reported results are the average results of 5 different runs.

Parameter Settings We conduct experiments on 1-shot and 5-shot settings. For each episode from different domains, the support set has N_s classes (N_s-way), where N_s is the label numbers in domain s. And the query set size is 16 for TourSG and 32 for StanfordLU. In terms of feature extraction, we use the pre-trained Electra-small (Clark et al., 2020) and Bert-base-uncased (Devlin et al., 2019) model. For the loss function, we set β as 0.1 and γ as 0.01. The smooth factor α is 0.1. We use AdamW (Loshchilov and Hutter, 2019) optimizer with the initial learning rate $2e-4$ for Electra, $1e-4$ for Bert, and the dropout rate is 0.1. All the hyperparameters are selected based on the performance of the development domain.

4.2 Baselines

We compare our proposed method with the following strong baselines.

TransferM is a standard fine-tune based transfer learning method, which is composed of a pre-trained language model as encoder and a multi-layer perceptron as classifier. It is trained on source

	Model	1-shot Sc	1-shot Na	1-shot We	Ave.	5-shot Sc	5-shot Na	5-shot We	Ave.
+E	TransferM	$16.96_{\pm 0.73}$	$22.99_{\pm 0.51}$	$21.01_{\pm 0.57}$	$20.32_{\pm 0.60}$	$16.99_{\pm 0.94}$	$23.79_{\pm 0.27}$	$23.92_{\pm 1.78}$	$21.57_{\pm 1.00}$
	MMN	$31.22_{\pm 4.96}$	$24.41_{\pm 3.28}$	$48.01_{\pm 1.10}$	$34.55_{\pm 3.11}$	$41.91_{\pm 4.49}$	$37.94_{\pm 1.38}$	$60.67_{\pm 1.23}$	$46.84_{\pm 2.37}$
	MPN	$32.44_{\pm 3.75}$	$17.83_{\pm 3.83}$	$38.86_{\pm 4.18}$	$29.71_{\pm 3.92}$	$35.92_{\pm 2.79}$	$27.65_{\pm 4.58}$	$58.07_{\pm 1.88}$	$40.55_{\pm 3.08}$
	CTLR	$40.61_{\pm 1.05}$	$40.76_{\pm 0.89}$	$46.16_{\pm 0.96}$	$42.51_{\pm 0.97}$	$51.83_{\pm 1.31}$	$46.44_{\pm 1.60}$	$54.17_{\pm 1.70}$	$50.82_{\pm 1.54}$
	DCKPN	$\mathbf{52.08}_{\pm 1.36}$	$\mathbf{51.37}_{\pm 0.82}$	$\mathbf{66.29}_{\pm 0.72}$	$\mathbf{56.58}_{\pm 0.97}$	$\mathbf{55.04}_{\pm 1.21}$	$\mathbf{55.64}_{\pm 0.91}$	$\mathbf{75.32}_{\pm 1.24}$	$\mathbf{62.00}_{\pm 1.12}$
+B	TransferM	$18.00_{\pm 0.62}$	$24.65_{\pm 0.79}$	$22.26_{\pm 0.64}$	$21.64_{\pm 0.68}$	$16.62_{\pm 0.18}$	$23.69_{\pm 0.46}$	$26.64_{\pm 2.04}$	$22.31_{\pm 0.89}$
	MMN	$39.18_{\pm 0.52}$	$35.35_{\pm 1.72}$	$45.87_{\pm 2.81}$	$40.13_{\pm 1.68}$	$43.65_{\pm 6.24}$	$51.94_{\pm 1.03}$	$46.65_{\pm 0.48}$	$47.41_{\pm 2.58}$
	MPN	$39.34_{\pm 1.38}$	$36.09_{\pm 0.77}$	$45.86_{\pm 2.50}$	$40.43_{\pm 1.55}$	$41.45_{\pm 2.83}$	$50.51_{\pm 2.94}$	$54.96_{\pm 9.76}$	$48.97_{\pm 5.18}$
	CTLR	$42.55_{\pm 0.40}$	$56.95_{\pm 0.77}$	$53.14_{\pm 1.89}$	$50.88_{\pm 1.02}$	$52.17_{\pm 1.29}$	$60.36_{\pm 1.55}$	$59.63_{\pm 2.23}$	$57.39_{\pm 1.69}$
	DCKPN	$\mathbf{53.81}_{\pm 0.72}$	$\mathbf{58.48}_{\pm 0.31}$	$\mathbf{74.02}_{\pm 0.74}$	$\mathbf{62.10}_{\pm 0.59}$	$\mathbf{57.81}_{\pm 0.62}$	$\mathbf{63.71}_{\pm 0.35}$	$\mathbf{93.83}_{\pm 0.36}$	$\mathbf{71.78}_{\pm 0.44}$

Table 4: The 1-shot and 5-shot average F1 scores of multi-label intent detection on StanfordLU dataset.

domains and fine-tuned by using support set from the target domain.

Multi-label Prototypical Network (MPN) is a variant of the vanilla prototypical network (Snell et al., 2017). It utilizes the negative Euclidean distance between queries and prototypes and selects a fixed threshold tuned on development domain to determine multiple labels.

Multi-label Matching Network (MMN) intends to train the model as MPN while leveraging Matching Network (Vinyals et al., 2016) to obtain the classification results by calculating the cosine similarity as distance measure.

CTLR (Hou et al., 2021) leverages meta calibrated threshold and anchored label representation to obtain dynamic threshold and separated label representations.

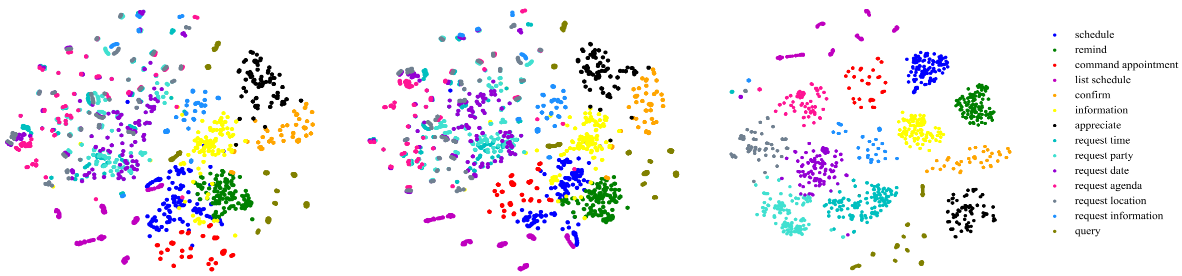
4.3 Main Results

Tables 1, 2, and 4 report the experimental results for 1-shot and 5-shot multi-label intent detection tasks on TourSG and StanfordLU. We use both Electra-small (+E) and Bert-base (+B) as feature encoders. The baseline results are taken from (Hou et al., 2021) and the top 1 results are highlighted in bold.

Average Improvements From the results, we can observe that DCKPN performs much better than other baselines, which demonstrates the superiority of our method. Specifically, in terms of average F1 scores of the TourSG dataset from Tables 1 and 2, DCKPN improves upon the most competitive baseline CTLR by 3.86% and 4.37% on 1/5-shot settings respectively when using Electra and by 1.66% and 1.60% when using Bert. In terms of average F1 scores of StanfordLU from Table 4, DCKPN improves upon CTLR by 14.07% and 11.18% on 1/5-shot settings for Electra and by 11.22% and 14.39% for Bert. The reason is that DCKPN accurately integrates label name as auxiliary information to learn more discriminative class prototypes and constructs the dual graph to obtain better feature embeddings and propagate class knowledge.

Parameter Efficiency Bert-base (110M params) may be too computationally intensive in real industry deployments, thus we additionally test model performance using Electra-small (14M params) that is nearly 10 times lighter. Interestingly, from Tables 1, 2 and 4, we can obtain that in terms of average F1 scores, our proposed method using Electra-small outperforms the strongest baseline CTLR using Bert-base by 0.44%, 5.70% and 4.61% in the 5-shot of TourSG and 1/5-shot of StanfordLU settings respectively and achieves competitive results in 1-shot of TourSG setting. As a result, larger models achieve better overall performance and our model reduces computational load while preserving competitive performance, which reflects the efficiency of our method.

Label Relationship When comparing to different baselines, our improvement on StanfordLU is much higher than those on TourSG. This large margin may come from the discrepancy in the label set of the two datasets. We find that the labels in StanfordLU are similar and fine-grained, where the label correlation is more informative while labels in TourSG are a little separated and the label correlation is less informative. To verify this, we show the relevance scores among labels using Eq. (1). We choose **Sc** domain and **Sh** domain from two



(a) Pic.1: CTRL. (b) Pic.2: DCKPN w/o DPN. (c) Pic.3: DCKPN.

Figure 4: Visualization of rectified support embeddings of Schedule in StanfordLU obtained from CTRL, DCKPN w/o DPN, and DCKPN respectively. Data points with the same color contain the same intent.

Setting	TourSG		StanfordLU	
	1-shot	5-shot	1-shot	5-shot
DCKPN	54.93	57.00	56.58	62.00
- LA	40.82	51.96	47.11	60.74
- CPN	53.61	54.86	51.77	59.62
- DPN	51.59	53.64	48.47	57.12

Table 5: Ablations on TourSG and StanfordLU. The average F1 scores of all domains are reported.

datasets respectively and visualize the relevance scores of the classes in Figure 5 in Appendix A. As can be seen, the classes in StanfordLU are more related than those in TourSG.

4.4 Ablation Studies

We conduct several ablation studies using Electra embeddings to examine the relative contributions of different components of our model in Table 5.

Label-semantic Augmentation For our model without label-semantic augmentation (LA), we directly use the original utterances as input of the encoder and concatenate their multiple labels as the corresponding class-level knowledge. From Table 5, massive F1 score decreases have been seen, particularly in 1-shot setting. On one hand, the model without LA is often confused by co-occurring intents in 1-shot setting, e.g., label 1 and 2 share an identical prototype in Figure 3, which can be separated by LA. On the other hand, utterances are ambiguous as described in Section 3.2, while LA is able to eliminate ambiguity and obtain more separated prototypes.

Impact of Dual Graph We independently remove class-level propagation network (CPN) and dual propagation network (DPN) including CPN and IPN. As presented in Table 5, DCKPN w/o CPN performs worse than DCKPN, especially for StanfordLU, which demonstrates the effectiveness of CPN for modeling label correlations in multi-label few-shot intent detection tasks. Moreover, when we remove DPN, the F1 scores drop hugely. The reason is that there exists an intrinsic gap between augmented support samples and original query samples while DPN reduces the feature gap and propagates label information.

4.5 Visualization

To better observe how the embeddings change with label-semantic augmentation and dual propagation network, we sample 50 episodes from Schedule domain in StanfordLU and use t-SNE (Van der Maaten and Hinton, 2008) to visualize the support embeddings obtained from CTRL (Hou et al., 2021), DCKPN w/o DPN and DCKPN. Note that each support embedding in CTRL is simply the average of utterance and label name embedding, that in DCKPN w/o DPN is the embedding of label-semantic augmented utterance without propagation, and that in DCKPN is obtained from our method. From Figure 4, it's easy to find that the distribution generated by CTRL has a lot of overlaps. The label enhancement in DCKPN w/o DPN can help to separate the embeddings to some extent. The dual propagation network can further generate more compact clusters and useful correlations.

5 Conclusion

In this paper, we propose a novel framework DCKPN for multi-label few-shot intent detection to capture the semantic interactions. We first exploit label-semantic augmentation to extract discriminative representations for user utterances with multiple intents. Then, we introduce an instance-level and a class-level propagation network to accomplish feature propagation and derive complete class representations. After that, we use the fused representations and an adaptive module to predict the possible multiple intents. Extensive experiments verify the effectiveness of our method and identify the cause of performance improvements our method brings. For future work, we will explore joint learning integrated with slot filling in dialogue systems.

Acknowledgements

This work is supported by the Science and Technology Project of State Grid Corporation of China (No. 5700-202341285A-1-1-ZN).

Limitations

In this paper, we explore label-semantic augmentation (LA) for multi-label few-shot intent detection via appending label name after utterances, which is similar to instruction learning or prompt learning. However, we don't further study the relationship between LA and instruction learning due to space limitations. We believe that instruction learning integrated with labels will inspire further investigation for multi-label few-shot intent detection.

References

Amit Alfassy, Leonid Karlinsky, Amit Aides, Joseph Shtok, Sivan Harary, Rogério Schmidt Feris, Raja Giryes, and Alexander M. Bronstein. 2019. Laso: Label-set operations networks for multi-label few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6548–6557.

Kai-Hsiang Cheng, Szu-Yu Chou, and Yi-Hsuan Yang. 2019. Multi-label few-shot learning for sound event recognition. In *21st IEEE International Workshop on Multimedia Signal Processing, MMSP*, pages 1–5.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Mihail Eric, Lakshmi Krishnan, François Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49.

Li Fei-Fei, Robert Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 594–611.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, pages 1126–1135.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 564–569.

Yutai Hou, Yongkui Lai, Yushan Wu, Wanxiang Che, and Ting Liu. 2021. Few-shot learning for multi-label intent detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 13036–13044.

Mengting Hu, Shiwan Zhao, Honglei Guo, Chao Xue, Hang Gao, Tiegang Gao, Renhong Cheng, and Zhong Su. 2021. Multi-label few-shot learning for aspect category detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 6330–6340.

Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. 2019. Edge-labeling graph neural network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 11–20.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. 2019. Learning to propagate labels: Transductive propagation network for few-shot learning. In *7th International Conference on Learning Representations, ICLR*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR*.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In

Proceedings of the 28th International Conference on Computational Linguistics, COLING*, pages 480–496.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, pages 2773–2782.

Qianwen Ma, Chunyuan Yuan, Wei Zhou, and Songlin Hu. 2021. Label-specific dual graph neural network for multi-label text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 3855–3864.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, pages 2554–2563.

Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021. GL-GIN: fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 178–188.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020a. TD-GIN: token-level dynamic graph-interactive network for joint multiple intent detection and slot filling. *CoRR*.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020b. Towards fine-grained transfer: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1807–1816.

Pau Rodríguez, Issam H. Laradji, Alexandre Drouin, and Alexandre Lacoste. 2020. Embedding propagation: Smoother manifold for few-shot classification. In *ECCV*, pages 121–138.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 4077–4087.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1199–1208.

Che-Ping Tsai and Hung-yi Lee. 2020. Order-free learning alleviating exposure bias in multi-label classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 6038–6045.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11).

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 3630–3638.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-specific document representation for multi-label text classification. 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*, pages 466–475.

Puyang Xu and Ruhi Sarikaya. 2013. Exploiting shared information for multi-intent natural language sentence classification. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association*, pages 3785–3789.

Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. 2020. DPGN: distribution propagation graph network for few-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 13387–13396.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING*, pages 3915–3926.

Shuo Yang, Lu Liu, and Min Xu. 2021. Free lunch for few-shot learning: Distribution calibration. In *9th International Conference on Learning Representations, ICLR*.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian model-agnostic meta-learning. In *Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 7343–7353.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 5812–5822.

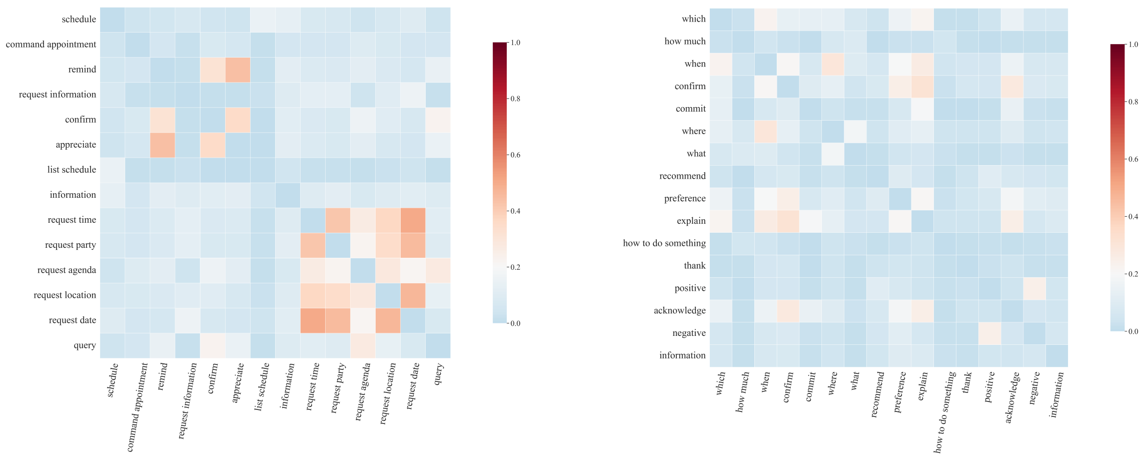
Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 5171–5181.

Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. 2018. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, ICMR*, pages 100–107.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328.

A Label Similarity

Figure 5(a) and 5(b) describes the label similarity of *schedule* (**Sc**) in StnafordLU and *shopping* (**Sh**) in TourSG using Bert-base embedding, which is calculated by Eq. (1). We can observe that (1) the classes in *schedule* are more similar and the classes in *shopping* are more distant. (2) the similarity of label description reflects co-occurrence between labels to some extent from the pragmatics view, e.g., in **Sc**, the similarity of *request date* and *request time* is high, and in real world scenarios, human usually asks the two details together, as well as *request date* and *request location*, which is crucial in multi-label tasks with low resources.



(a) Schedule

(b) Shopping

Figure 5: The label similarity of schedule and shopping domains. Darker colors denote higher similarities.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

4.1.2

☑ B1. Did you cite the creators of artifacts you used?
4.1.2

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1.1 and Table 3.

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.1.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.1 and 4.3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4.1.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.