Few-shot Learning for Multi-label Intent Detection

Yutai Hou, Yongkui Lai, Yushan Wu, Wanxiang Che* , Ting Liu
School of Computer Science and Technology, Harbin Institute of Technology, China
{ythou, yklai, car, tliu}@ir.hit.edu.cn, wuyushan@hit.edu.cn

Abstract

In this paper, we study the few-shot multi-label classification for user intent detection. For multi-label intent detection, state-of-the-art work estimates *label-instance relevance scores* and uses a *threshold* to select multiple associated intent labels. To determine appropriate *thresholds* with only a few examples, we first learn universal thresholding experience on data-rich domains, and then adapt the thresholds to certain few-shot domains with a calibration based on non-parametric learning. For better calculation of *label-instance relevance score*, we introduce label name embedding as anchor points in representation space, which refines representations of different classes to be well-separated from each other. Experiments on two datasets show that the proposed model significantly outperforms strong baselines in both one-shot and five-shot settings. Data and code are available at https://github.com/AtmaHou/FewShotMultiLabel.

Introduction

Intent detection, a fundamental component of task-oriented dialogue system (Young et al. 2013), is increasingly raising attention as a Multi-Label Classification (MLC) problem (Xu and Sarikaya 2013; Qin et al. 2020), since a single utterance often carries multiple user intents (See examples in Fig 1). In real-world scenarios, intent detection often suffers from lack of training data, because dialogue tasks/domains change rapidly and new domains usually contain only a few data examples. Recent success of Few-Shot Learning (FSL) presents a promising solution for such data scarcity challenges. It provides a more human-like learning paradigm that generalizes from only a few learning examples (usually one or two per class) by exploiting prior experience.

State-of-the-art works for multi-label intent detection focus on threshold-based strategy, where a common practice is estimating *label-instance relevance scores* and picking the intent labels with score higher than a *threshold* value (Xu et al. 2017; Gangadharaiah and Narayanaswamy 2019; Qin et al. 2020). Usually, the coordination and respective quality of the two modules, i.e. thresholding and relevance scoring, are crucial to the performance of MLC models. However, in few-shot scenarios, such multi-label setting poses unique challenges for both threshold estimation and label-instance relevance scoring.

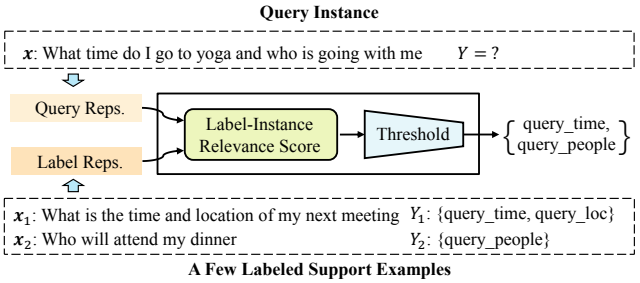

Figure 1: Example for one-shot multi-label intent detection.

For thresholding, previous works explore to tune a fixed threshold (Gangadharaiah and Narayanaswamy 2019; Qin et al. 2020) or to learn thresholds from data (Xu et al. 2017). But, these thresholds work well only when learning examples are sufficient. In few-shot scenarios, it is pretty hard to determine appropriate thresholds with only a few examples. Besides, it is also difficult to directly transfer the pre-learned thresholds due to the domain differences, such as differences in label number per instance, score density and scale.

Estimation of the label-instance relevance scores is also challenging. Few-shot learning has achieved impressive progress with similarity-based methods (Vinyals et al. 2016; Bao et al. 2020), where the relevance scores can be modeled as label-instance similarities. And the label representations can be obtained from corresponding support examples. Unfortunately, despite huge success in previous single-label tasks, these similarity-based methods become impractical for multi-label problems. When instances have multiple labels, representations of different labels may be obtained from the same support examples and become confused with each other. For the example in Fig 1, intents of *query_time* and *query_loc* share the same support example $\boldsymbol{x}_1$ and thus have the same label representation, which makes it impossible to predict correct labels with similarity scores.

In this paper, we study the few-shot learning problem of multi-label intent detection and propose a novel framework to tackle the challenges from both thresholding and label-instance relevance scoring.

To solve the thresholding difficulties of prior-knowledge

* Corresponding author.

transferring and domain adaption with limited examples, we propose a *Meta Calibrated Threshold* (MCT) mechanism that first learns universal thresholding experience on data-rich domains, then adapts the thresholds to certain few-shot domains with a Kernel Regression based calibration. Such combination of universal training and domain-specific calibration allows to estimate threshold using both prior domain experience and new domain knowledge.

To tackle the challenge of confused label representation in relevance scoring, we propose the *Anchored Label Representation* (ALR) to obtain well-separated label representations. Inspired by the idea of embedding label name as anchor points to refine representation space (Wang et al. 2018), ALR uses the embeddings of label names as additional anchors and represents each label with both support examples and corresponding anchors. Different from the previous single-label intent detection that uses label embedding as additional features (Chen, Hakkani-Tür, and He 2016), our label embeddings here have unique effects of separating different labels in metric space.

Finally, to encourage better coordination between thresholding and label-instance relevance scoring, we introduce the Logit-adapting mechanism to MCT that automatically adapts thresholds to different score densities.

Experiments on two datasets show that our methods significantly outperform strong baselines. Our contributions are summarized as follows: (1) We explore the few-shot multi-label problem in intent detection of task-oriented dialogue, which is also an early attempt for the few-shot multi-label classification. (2) We propose a Meta Calibrated Threshold mechanism with Kernel Regression and Logits Adapting that estimates threshold using both prior domain experience and new domain knowledge. (3) We introduce the Anchored Label Representation to obtain well-separated label representation for better label-instance relevance scoring.

Notions and Preliminaries

To ease understanding, we briefly introduce the task of multi-label classification and few-shot learning here.

Multi-label Classification

Multi-label task studies the classification problem where each single instance is sociated with a set of labels simultaneously. Suppose $\mathcal{X}$ denotes instance space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ denotes label space with N possible labels. Multi-label task learns a function $h(\cdot) : \mathcal{X} \to 2^{\mathcal{Y}}$ from multi-label training data $T = \{(\boldsymbol{x}_i, Y_i)\}_i^{N_T}$, where N_T is the size of datasets. For each learning example $(\boldsymbol{x}_i, Y_i)$, $\boldsymbol{x}_i \in \mathcal{X}$ is l-dimensional input and $Y_i \subseteq \mathcal{Y}$ is the corresponding label set. Then for an unseen instance $\boldsymbol{x}$, the classifier predicts $Y = h(\boldsymbol{x}) \subseteq \mathcal{Y}$ as the associated label set.

In most cases (Zhang and Zhou 2013), multi-label model learns a real-value function $f : \mathcal{X} \to \mathcal{Y}$. $f(\boldsymbol{x}, y)$ evaluates the *label-instance relevance score*, which reflects the confidence of label $y \in \mathcal{Y}$ being the proper label of x. Then multi-label classifier is derived as $h(\boldsymbol{x}) = \{y \mid f(\boldsymbol{x}, y) > t, y \in \mathcal{Y}\}$, where t is the *threshold* value.

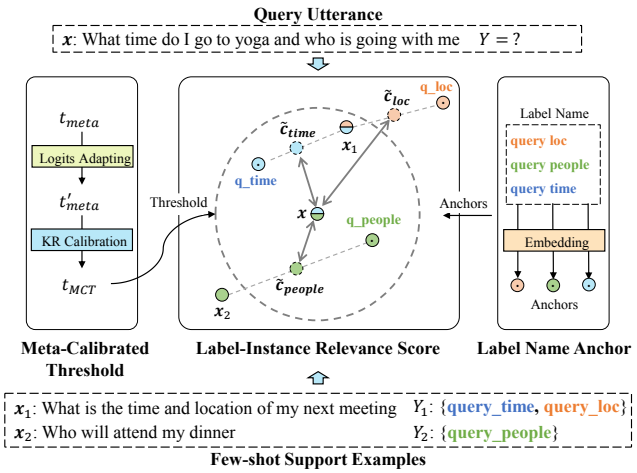

Figure 2: Proposed framework for few-shot multi-label intent detection. For each query $\boldsymbol{x}$, we compute label-instance relevance scores according to its similarity to each Anchored Label Representation $\tilde{\boldsymbol{c}}$. Then we pick the labels that have score higher than a threshold value t derived from the proposed Meta-Calibrated Threshold mechanism.

Few-shot Learning

Few-shot learning extracts prior experience that allows quick adaption on new tasks (Finn 2018). The learned prior experience often includes meta knowledge general to different domains and tasks, such as similarity metric and model architecture. On the new task, few-shot model uses these prior knowledge and a few labeled examples (*support set*) to predict the class of an unseen item (*query*).

Few-shot learning is often achieved with similarity based methods, where models are usually first trained on a set of source domains (tasks) $\{\mathcal{D}_1, \mathcal{D}_2, \ldots\}$, then directly work on another set of unseen target domains (tasks) $\{\mathcal{D}'_1, \mathcal{D}'_2, \ldots\}$ without fine-tuning. On each target domain, given a query $\boldsymbol{x}$, model predicts the corresponding label y by observing a labeled support set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_S}$. S usually includes k examples (K-shot) for each of N labels (N-way).

For few-shot multi-label intent detection, we define each query instance as user utterance with a sequence of words $\boldsymbol{x} = (x_1, x_2, \ldots, x_l)$. And instead of predicting single label, model predicts a set of intent labels $Y = \{y_1, y_2, \ldots, y_m\}$ with multi-label support set $S = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{N_S}$.

Method

In this section, we first present the overview of our framework, and then introduce the proposed Meta Calibrated Threshold and Anchored Label Representation. In the last, we introduced how the entire framework is optimized.

The workflow of our framework is shown in Fig 2: given an unseen query utterance and a labeled support set, we first gather representation of each label from support set, and calculate label-instance relevance scores with sentence-label similarity. Then we use a threshold to dichotomize the label space into relevant and irrelevant label sets. In the

framework, the MCT helps to estimate reasonable thresholds under few-shot scenarios, and the ALR provides well-separated label representation.

Proposed Framework for Few-shot Multi-label Intent Detection

We define the few-shot multi-label classification (MLC) similar to the aforementioned normal MLC. Given a query sentence $\boldsymbol{x}$ and a support set S, our framework predicts the associated label set Y as:

$$Y = h(\boldsymbol{x}, S) = \{y \mid f(\boldsymbol{x}, y, S) > t, y \in \mathcal{Y}\},$$

where f calculates label-instance relevance scores, and t is threshold value.

To achieve few-shot multi-label classification, we adopt the prevailing similarity-based method to calculate the label-instance relevance scores.

Firstly, we derive the representation of each label from the support set S. Supposing $\boldsymbol{c}_i$ is the representation vector of label y_i, we compute relevance score between query sentence $\boldsymbol{x}$ and label y_i as follow:

$$f(\boldsymbol{x}, y_i, S) = \textsc{Sim}(E(\boldsymbol{x}), \boldsymbol{c}_i),$$

where $E(\cdot)$ is an embedder function. SIM is a similarity function, and we use the dot-product similarity. We adopt BERT (Devlin et al. 2019) as the embedder, and the sentence embedding $E(\boldsymbol{x})$ is calculated as the averaged embedding of its tokens. To get well-separated label representations, we adopt the **Anchored Label Representation** to obtain $\boldsymbol{c}_i$.

Then, we estimate a threshold value t that integrates both prior knowledge from source domains and observation of examples from target domains. To achieve this, we propose the **Meta Calibrated Threshold** to estimate threshold t.

Meta Calibrated Threshold

In this section, we introduce a thresholding method for few-shot learning setting. In few-shot learning setting, models are trained and tested on different domains, which often have different preferences for threshold selection. Further, it is necessary to label each instance with different thresholds, because instances vary in label number and density of label-instance relevance scores.

To achieve this, we first learn a domain-general meta threshold, and then calibrate it to adapt to both target domain and specific queries.

Meta Threshold with Logits-Adaptiveness To achieve domain-general thresholding, we present a Meta Threshold t_{meta} that has automatic adaptability and is jointly optimized on various domains.

For automatic adaptability, we propose the Logit-Adaptive mechanism for meta threshold, which automatically adapts the threshold to both specific queries and domains. Specifically, considering that instances vary in scale/density of relevance scores and the value of threshold is always between the maximum and the minimum score, we propose to set the threshold as an interpolation of maximum and minimum scores:

$$t_{\text{meta}} = T(\boldsymbol{x}, S) = r \max_{\forall y \in \mathcal{Y}} f(\boldsymbol{x}, y, S) + (1 - r) \min_{\forall y \in \mathcal{Y}} f(\boldsymbol{x}, y, S),$$

where $T(\cdot)$ is the thresholding function and r is the interpolation rate learned in source domains.[1] As shown in Fig 3, such interpolation-based thresholds vary for different queries by adapting to different densities of label-instance relevance scores, and are more general than fixed thresholds.

Besides, Logit-Adaptive mechanism also promotes better coordination between thresholding and relevance scoring.

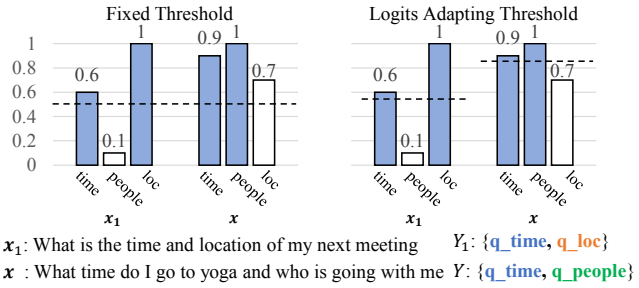

Figure 3: Example of fixed threshold and logit-adapting threshold. Colored score-bars correspond to correct labels. It is impossible to find a fixed threshold fitting both $\boldsymbol{x}_1$ and $\boldsymbol{x}$, but logit-adapting threshold can adapt to both cases with $r = 0.5$.

Threshold Calibration with Kernel Regression Till now, we learn a Meta Threshold t_{meta}, which is general to various domains but lacks domain-specific knowledge. To remedy this, we estimate a domain/query-specific threshold t_{est} by observing the support set, and use t_{est} to calibrate the Meta Threshold. However, owing to the absence of golden thresholds, it is hard to directly learn a model to estimate thresholds. Therefore, we turn to estimate the number of labels and indirectly deduce the threshold.

To estimate label numbers with domain-specific knowledge, we adopt *Kernel Regression* (KR) (Nadaraya 1964) and estimate the label number of a query according to its similarity to support examples. As a non-parametric method, KR can work on unseen domains without tuning. Compared to other non-parametric regression methods, such as KNN regression, KR allows to make use of all support examples and consider the distance influences.

Formally, given a support set S, we estimate the label number n of query $\boldsymbol{x}$ as the weighted average label number of support examples, where the weights are computed as kernel similarity between query and support examples:

$$n = \frac{1}{Z} \sum_{\forall (\boldsymbol{x'}, \boldsymbol{y'}) \in S} \text{Kernel}(\tilde{E}(\boldsymbol{x}), \tilde{E}(\boldsymbol{x'}); \lambda) \cdot |\boldsymbol{y'}|.$$

Here, Z is the normalizing factor, and we use Gaussian Kernel: $\text{Kernel}(\boldsymbol{a}, \boldsymbol{b}; \lambda) = \exp(-(\boldsymbol{a} - \boldsymbol{b})^2 / \lambda)$, where λ is bandwidth factor. $\tilde{E}(\boldsymbol{x})$ is a feature extractor that returns a feature

[1] Extreme cases of outputting all labels can be covered by imposing a small negative perturbation on t_{meta}.

vector related to the label number of a sentence $\boldsymbol{x}$. For the intent detection task of this paper, we consider the linguistics features related to the intent number of a sentence, including *sentence length*, *# of conjunctions*, *# of predicates*, *# of punctuations*, *# of interrogative pronouns* and encode these features with an MLP projection layer.[2]

Then, we derive a domain/query-specific threshold t_{est} from the estimated label number n. Specifically, we find a threshold value t_{est} that filters out top-n label-instance relevance scores of $\boldsymbol{x}$. One intuitive idea is to directly use the $(n+1)_{\text{th}}$ largest score as threshold. But, such threshold is derived from only one label-instance relevance score. So we further improve it to make use of all relevance scores by directly estimating threshold with the learned kernel weights:

$$t_{\text{est}} = \frac{1}{Z} \sum_{\forall(\boldsymbol{x'},\boldsymbol{y'})\in S} \text{Kernel}(\tilde{E}(\boldsymbol{x}), \tilde{E}(\boldsymbol{x'}); \lambda) \cdot T'(|\boldsymbol{y'}|; \boldsymbol{x}, S, f),$$

where $T'(n; \boldsymbol{x}, S, f)$ is a function that returns the $(n+1)_{\text{th}}$ largest label-instance relevance scores of query $\boldsymbol{x}$.

Finally, we use query-specific threshold t_{est} to calibrate the domain-general meta threshold t_{meta}. The final threshold for query $\boldsymbol{x}$ is computed as:

$$t = \alpha \times t_{\text{meta}} + (1-\alpha) \times t_{\text{est}}, \tag{1}$$

where α is hyper-parameter that measures the importance of the prior thresholding experience.

Anchored Label Representation

Label representation is essential in the label-instance relevance score. To get high-quality label-instance similarity modeling, label representations should be (1) well-separated from each other and (2) able to fully express the semantic information of the corresponding category.

Label Representation for Few-shot Learning For few-shot learning, the label representations are mainly obtained from support set examples. One of the most classic ideas is to get a prototypical representation of each category as label representation (Snell, Swersky, and Zemel 2017). And the prototypical representation of label y_i is calculated as the averaged embedding of support examples: $\boldsymbol{c}_i = \frac{1}{M_i}\sum_j^{M_i} E(\boldsymbol{x}_j)$, where each $\boldsymbol{x_j} \in \{\boldsymbol{x} \mid (\boldsymbol{x}, Y) \in S \wedge y_i \in Y\}$ is the support instance labeled with y_i, and M_i is the total number of such instances in support set S.

Such label representations have no constraint on the separability between labels. Further, in multi-label setting, different labels may share the same support examples. This can lead to mix-up and ambiguity between label representations.

Represent Label with Anchor Because obtaining the label representation with only support examples leads to ambiguity, we propose to additionally represent the labels with label-specific anchors, which emphasizes the difference between different categories. Label names are often naturally well separated from each other and contain good expressions of category-specific semantics (Wang et al. 2018). Therefore, intuitively, we use semantic embedding of label names as the anchors and represent each label with both anchor and support examples. For label y_i, we compute the anchored label representation $\tilde{\boldsymbol{c}}_i$ with an interpolation factor β:

$$\tilde{\boldsymbol{c}}_i = \beta \times E(y_i) + (1-\beta) \times \boldsymbol{c}_i, \tag{2}$$

where $\boldsymbol{c}_i$ is the prototypical representation obtained with support examples. Here, label name embedding $E(y_i)$ acts as a deflection of the prototypical representation vector. This allows label representations to be separated from each other and better describe the category semantics.

Optimization

Following Vinyals et al. (2016), we train the MLC framework with a series of few-shot learning episodes, where each episode contains a few-shot support-set and a query-set. Simulating few-shot situation on data rich domains ensures consistence between training and few-shot testing. Besides, the framework is optimized on different domains alternatively, which encourages both meta threshold t_{meta} and label-instance relevance scoring function f to be domain-general. We take the Sigmoid Cross Entropy loss (Cui et al. 2019) for MLC training:

$$\text{Loss} = \tfrac{1}{N}\textstyle\sum_i^N \Big\{\mathbb{I}(y_i \notin Y^*) \cdot \sigma(f_{y_i}) - \mathbb{I}(y_i \in Y^*) \cdot \sigma(f_{y_i})\Big\},$$

where N is the number of possible labels and $f_{y_i} = f(\boldsymbol{x}, y_i, S)$. Y^* is the golden label set. $\mathbb{I}(\cdot)$ is an indicator function.[3] σ is the Sigmoid function. Due to the undifferentiable process of picking thresholds with label numbers, we pre-train the kernel parameters before the whole framework learning process, i.e. bandwidth and MLP projection layer, on source domains with the loss of MLE($n_{\text{est}}, n_{\text{gold}}$).

This is a meta-learning process that learns meta parameters (of Meta Threshold, Kernel Regression, similarity metric computing) to improve non-parametric learning of unseen few-shot tasks, a.k.a learning to learn.

Experiment

We evaluate our method on the multi-label intent detection task of 1-shot/5-shot setting, which transfers knowledge from source domains (training) to an unseen target domain (testing) containing only a 1-shot/5-shot support set.

Dataset We conduct experiments on public dataset TourSG (Williams et al. 2012) and introduce a new multi-intent dataset StanfordLU. These two datasets contain multiple domains and thus allow to simulate the few-shot situation on unseen domains. TourSG (DSTC-4) contains 25,751 utterances annotated with multiple dialogue acts and 6 separated domains about touristic information for Singapore: Itinerary (It), Accommodation (Ac), Attraction (At), Food (Fo), Transportation (Tr), Shopping (Sh). StanfordLU is an re-annotated version of Stanford dialogue dataset (Eric et al. 2017) containing 8,038 user utterances from 3 domains:

[2] We extract these linguistics features with StanfordTagger-Base (Toutanova et al. 2003).

[3] $\mathbb{I}(True) = 1$ and $\mathbb{I}(False) = 0$

Domain	1-shot $\lvert S\rvert$	5-shot$\lvert S\rvert$	P. ML	$\lvert\mathcal{Y}\rvert$
It	12.56	48.44	22.7%	16
Ac	13.93	59.95	18.2%	17
At	14.40	65.71	16.0%	18
Fo	14.92	63.77	17.4%	18
Tr	13.97	59.77	18.1%	17
Sh	13.12	55.53	16.4%	16
Sc	11.07	52.88	21.3%	14
Na	7.34	34.29	24.6%	10
We	7.45	36.40	3.8%	8

Table 1: Overview of few-shot multi-intent detection data from TourSG (above midline) and StanfordLU (below midline). $|S|$ is the average support set size. P. ML denotes the proportion multi-label sentences. $|\mathcal{Y}|$ is the label numbers.

Schedule (Sc), Navigate (Na), Weather (We). We re-annotate each utterance with intent labels, which are not included in the original dataset.

Few-shot Data Construction To simulate the few-shot situation, we reconstruct the dataset into few-shot learning form, where each sample is the combination of a query instance $(\boldsymbol{x^q}, \boldsymbol{y^q})$ and corresponding K-shot support set $\mathcal{S}$. Table 1 shows the overview of the experiment data.

Different from the single-label classification problem, multi-label instance is associated with multiple labels. There, we cannot guarantee that each label appears K times while sampling the support sentences. To cope with this, we approximately construct K-shot support set $\mathcal{S}$ with the Minimum-including Algorithm (Hou et al. 2020). It constructs support set generally following two criteria: (1) All labels within the domain appear at least K times in $\mathcal{S}$. (2) At least one label will appear less than K times in $\mathcal{S}$ if any $(\boldsymbol{x}, \boldsymbol{y})$ pair is removed from it.[4]

For each domain, we sample N_s different K-shot support sets. Then, for each support set, we sample N_q unincluded utterances as queries (query set). Each support-query-set pair forms one **few-shot episode**. Eventually, we get N_s episodes and $N_s \times N_q$ samples for each domain.

For TourSG, we construct 100 few-shot episodes for each source domain and 50 few-shot episodes for each target domain. And the query set size is 16. Because StanfordLU has fewer domains, we construct 200 few-shot episodes for each source domain and 50 few-shot episodes for each target domain. And query set size is 32.

Evaluation To conduct robust evaluation under few-shot setting, we cross-validate the models on different domains. Each time, we pick one target domain for testing, one domain for development, and use the rest domains of the same dataset as source domains for training.[5]

When testing model on a target domain, we evaluate micro F1 scores within each few-shot episode. Then we average F1 scores from all episodes as the final result to counter the randomness from support-sets. To control the nondeterminacy of neural network training (Reimers and Gurevych 2017), we report the average score of 5 random seeds.

Implements For sentence embedding and label name, we average the token embedding provided by pretrained language model and we use *Electra-small* (Clark et al. 2020) and uncased *BERT-Base* (Devlin et al. 2019) here. Besides, we adopt embedding tricks of Pairs-Wise Embedding (Hou et al. 2020) and Gradual Unfreezing (Howard and Ruder 2018). We use ADAM (Kingma and Ba 2015) to train the models with batch size 4. Learning rate is set as 1e-5 for both our model and baseline models. We set α (Eq. 1) as 0.3 and vary β (Eq. 2) in $\{0.1, 0.5, 0.9\}$ considering label name's anchoring power with different datasets and support-set sizes. For the MLP of kernel regression, we employ ReLU as activation function and vary the layers in $\{1, 2, 3\}$ and hidden dimension in $\{5, 10, 20\}$. The best hyperparameter are determined on the development domains.

Baselines

We compare our model with two kinds of strong baseline: fine-tune based transfer learning methods (TransferM) and similarity-based FSL methods (MPN and MMN).

TransferM is a domain transfer model with large pretrained language model and a multi-label classification layer. Following popular MLC settings, we use a fixed threshold tuned on dev set. We pretrain it on source domains and select the best model on the dev set. We deal with mismatch of label set by re-training classifying layers for different domains. On target domain, model is fine-tuned on support set.

Multi-label Prototypical Network (MPN) is a similarity based few-shot learning model that calculates sentence-label relevance score with a prototypical network (Snell, Swersky, and Zemel 2017) and uses a fixed threshold tuned on dev set. It is pre-trained on source domains and directly works on target domains without fine-tuning.

Multi-label Matching Network (MMN) is all the same as MPN but employs the Matching Network (Vinyals et al. 2016) for label-instance relevance score calculation.

Main Results

Here we evaluate the proposed method on both 1-shot and 5-shot multi-label intent detection.

Result of 1-shot setting. Table 2 and Table 4 (Left) show the results of 1-shot multi-label intent detection. Each column respectively shows the F1 scores of taking a certain domain as target domain (test) and use others as source domain (train & dev). When using BERT embedding, our model outperforms the strongest baseline by average F1 scores of

[4]Removing steps have a preference for instances with more labels. So we randomly skip removing by the chance of 20%.

[5]For example of the TourSG dataset, each round, model is trained on $4\times100\times16 = 6400$ samples, validated on $1\times50\times16 = 1600$ samples, and tested on $1 \times 50 \times 16 = 800$ samples. Then, with all the 6 cross-evaluation rounds, each model is totally tested on $800 \times 6 = 4800$ samples.

	Model	It	Ac	At	Fo	Tr	Sh	Ave.
+E	TransferM	14.34±0.82	14.75±0.91	16.13±1.35	11.79±1.54	13.64±0.33	14.32±1.17	14.16±1.02
	MMN	9.98±1.80	7.81±0.70	8.37±0.86	7.81±0.35	10.65±1.62	11.56±0.79	9.36±1.02
	MPN	12.24±0.92	10.38±1.21	10.00±0.54	10.47±0.42	13.61±0.92	11.41±0.31	11.35±0.72
	MPN+ALR	28.74±2.18	34.94±1.91	35.06±3.83	34.62±2.69	35.53±1.97	31.87±2.31	33.46±2.48
	Ours	**39.98**±0.56	**51.55**±1.53	**55.16**±2.43	**52.16**±0.98	**55.36**±0.96	**52.20**±1.03	**51.07**±1.24
+B	TransferM	16.78±0.05	18.62±0.59	14.92±2.22	16.40±2.58	15.68±0.32	14.50±2.18	16.15±1.32
	MMN	10.89±3.35	7.72±1.44	8.92±1.45	9.32±1.40	13.75±0.70	10.87±4.31	10.24±2.11
	MPN	13.77±0.38	12.38±0.32	13.46±0.14	10.23±0.30	16.19±0.19	15.79±0.38	13.64±0.28
	MPN+ALR	40.99±1.54	51.57±1.04	54.91±0.31	51.90±1.98	54.87±0.82	50.76±1.30	50.83±1.17
	Ours	**44.58**±0.71	**57.11**±1.22	**60.34**±0.92	**56.49**±0.67	**60.18**±0.85	**55.60**±0.66	**55.72**±1.03

Table 2: F1 scores on 1-shot multi-label intent detection on TourSG dataset. +E and +B denote use Electra-small (14M params) and BERT-base (110M params) as embedder respectively. Ave. shows the averaged scores.

	Model	It	Ac	At	Fo	Tr	Sh	Ave.
+E	TransferM	14.72±0.53	19.20±1.59	16.18±1.03	18.86±1.04	17.17±1.19	17.51±1.63	17.27±1.17
	MMN	14.11±0.83	10.58±1.35	17.80±1.12	12.74±0.87	18.01±0.90	16.76±0.92	15.00±1.00
	MPN	15.18±0.63	15.56±0.54	17.60±1.15	15.01±0.19	17.99±0.36	17.17±1.09	16.42±0.66
	MPN+ALR	29.74±3.18	30.91±2.51	34.28±3.06	33.61±2.70	35.90±2.85	33.44±3.58	32.98±2.98
	Ours	**44.21**±0.71	**51.37**±1.22	**55.76**±0.92	**54.50**±0.58	**55.37**±0.95	**54.55**±0.86	**52.63**±0.87
+B	TransferM	17.98±1.80	16.51±1.95	19.88±4.17	17.22±3.01	13.84±1.40	15.41±2.81	16.81±2.52
	MMN	15.65±1.24	16.42±0.71	19.90±0.51	12.23±0.33	16.81±4.64	17.13±0.20	16.36±1.27
	MPN	20.71±0.98	22.39±1.95	26.51±0.72	21.94±1.59	23.41±1.31	24.52±3.31	23.24±1.64
	MPN+ALR	45.51±0.51	53.71±0.95	58.16±0.53	56.91±0.51	57.62±0.70	54.86±0.59	54.46±0.63
	Ours	**46.80**±0.83	**54.79**±0.80	**59.95**±0.46	**59.11**±0.39	**60.13**±0.44	**58.56**±0.30	**56.56**±0.54

Table 3: F1 scores on 5-shot multi-label intent detection on TourSG dataset. Ave. shows the averaged scores.

39.57 on TourSG and 10.45 on Stanford dataset. Our improvement on TourSG is much higher than those on StanfordLU. We find the gap mainly comes from the difference in label set characteristics of two datasets and will analyze this latter.

BERT is sometimes too computational costly in real-world applications. So we also evaluate model performance with lighter embedding of Electra-small. It only has 14M parameters, which is much smaller than the 110M of BERT. Again, our model achieves the best performance with Electra. Interestingly, our Electra-based model is even better than all baselines using BERT, which is especially valuable in scenarios with limited computing resources.

When comparing to traditional transferring-based method (TransferM), all non-finetune-based methods (Ours, MPN and MMN) gain huge improvements on StanfordLU dataset. This mainly comes from the superiority of the non-finetune-based in overfitting resistance. For TourSG, domain gaps are lower and labels of different domains are similar, which makes it easier to transfer source domain parameters. As a result, TransferM performs slightly better than MPN on TourSG. In contrast, our models have stable performance on both types of datasets, which reflects the model versatility.

Our model can be regarded as *MPN+ALR+MCT*. Thus, the stepped growth between *MPN*, *MPN+ALR* and *Ours* demonstrates the effectiveness of Both ALR and MCT.

Result of 5-shot setting. Table 3 and Table 4 (Right) show the 5-shots results. The results are consistent with 1-shot setting in general trending. Our methods achieve the best performance. By comparing 1-shot and 5-shots results, we find that our 1-shot model is able to outperform most 5-shot baselines. This indicates that our model can better exploit prior experience and rely less on support examples.

Analysis

Ablation Test To understand the contribution of each framework component, we conduct 1-shot/5-shots ablation study with Electra embedding in Table 5. We independently remove two main components: *Anchored Label Representation* (ALR) and *Meta Calibrated Threshold* (MCT).

When ALR is removed, we represent each label with only prototypical embeddings constructed from support examples. Huge F1 score drops are witnessed especially on TourSG. On one hand, TourSG has similar labels across different domains, which greatly benefits the label embedding learning of ALR. On the other hand, model without ALR are often confused by co-occurring intents, such as "thank" and "confirm", which can be easily separated by ALR.

For our model without MCT, we use a vanilla threshold tuned on source domains. We find MCT has more impacts on 5-shot settings. This is because logits adapting and KR calibration of MCT exploit the relation between specific query and support set, which promotes model to benefits from more support examples.

	Model	1-shot				5-shot			
		Sc	Na	We	Ave.	Sc	Na	We	Ave.
+E	TransferM	16.96±0.73	22.99±0.51	21.01±0.57	20.32±0.60	16.99±0.94	23.79±0.27	23.92±1.78	21.57±1.00
	MMN	31.22±4.96	24.41±3.28	**48.01**±1.10	34.55±3.11	41.91±4.49	37.94±1.38	**60.67**±1.23	46.84±2.37
	MPN	32.44±3.75	17.83±3.83	38.86±4.18	29.71±3.92	35.92±2.79	27.65±4.58	58.07±1.88	40.55±3.08
	MPN+ALR	33.35±1.00	28.88±1.57	45.58±1.98	35.93±1.52	44.52±6.21	42.39±2.32	54.42±4.78	47.11±4.44
	Ours	**40.61**±1.05	**40.76**±0.89	46.16±0.96	**42.51**±0.97	**51.83**±1.31	**46.44**±1.60	54.17±1.70	**50.82**±1.54
+B	TransferM	18.00±0.62	24.65±0.79	22.26±0.64	21.64±0.68	16.62±0.18	23.69±0.46	26.64±2.04	22.31±0.89
	MMN	39.18±0.52	35.35±1.72	45.87±2.81	40.13±1.68	43.65±6.24	51.94±1.03	46.65±0.48	47.41±2.58
	MPN	39.34±1.38	36.09±0.77	45.86±2.50	40.43±1.55	41.45±2.83	50.51±2.94	54.96±9.76	48.97±5.18
	MPN+ALR	38.81±1.13	41.08±1.76	**54.16**±2.12	44.68±1.67	51.30±1.69	47.80±3.73	**60.08**±2.64	53.06±2.69
	Ours	**42.55**±0.40	**56.95**±0.77	53.14±1.89	**50.88**±1.02	**52.17**±1.29	**60.36**±1.55	59.63±2.23	**57.39**±1.69

Table 4: F1 scores of multi-label intent detection on StanfordLU dataset. +E and +B denote use Electra-small (14M params) and BERT-base (110M params) as embedder respectively. Ave. shows the averaged scores.

Setting	TourSG		StanfordLU	
	1-shot	5-shots	1-shot	5-shots
Ours	51.07	52.63	42.51	50.82
- ALR	-38.53	-31.33	-11.33	-10.31
- MCT	-12.52	-16.70	-10.12	-17.45

Table 5: Ablation study over two main components of proposed framework: Anchored Label Representation and Meta Calibrated Threshold. Result is the averaged accuracy of all domains.

Setting	TourSG		StanfordLU	
	1-shot	5-shots	1-shot	5-shots
MCT (Ours)	51.07	52.63	42.51	50.82
- Logits Adapting	-6.13	-7.36	-2.45	-7.78
- Meta Learning	-2.73	-3.36	-3.59	-9.35
- KR Calibration	-0.86	-2.28	-2.08	-8.96

Table 6: Detailed ablation test over Meta Calibrated Threshold.

By comparing the opposite influence of shot number on ALR and MCT, we found our framework reaches a balance, since its two components are respectively good at transferring prior knowledge and exploiting domain knowledge.

Analysis over components of Meta Calibrate Threshold We further disassemble the MCT and to see the contributions of the three sub-components.

When we remove *Logits-Adapting*, we use a single learnable value as meta-threshold. The performance drops indicate that Logits-Adaptiving threhold provides better domain generalization than traditional threshold values.

If we remove *Meta Learning* of threshold, we replace the meta threshold with a fixed Logits-Adapting threshold, and calibrate it without learning of meta parameters in Kernel Regression. We address the performance loss to the fact that meta learning process provides prior experience which effectively aids the non-parametric learning of target domains.

Setting	TourSG		StanfordLU	
	1-shot	5-shots	1-shot	5-shots
ALR	68.16	67.28	15.67	21.91
ALR + MT	77.85	78.18	51.24	51.38
ALR + MT + KR	**82.26**	**82.05**	**80.92**	**84.70**

Table 7: Analysis of label number accuracy. ALR denotes model with Anchored Label Representation. MT is Meta Threshold. KR is Calibration with Kernel-Regression.

For our model without *KR Calibration*, we directly predict labels with meta thresholds. The score drops show calibration helps by adapting thresholds to different domains. The drops on TourSG is limited, because the domain gap of TourSG is small, and meta threshold learned on source domains are often good enough even without calibration.

Label Number Accuracy Analysis To understand the impact of thresholding module, we conduct accuracy analysis of whether model can predict correct number of labels. As Table 7 presents, when adding Meta Threshold and KR Calibration, we can observe a continuous increase in the label number accuracy. This shows that both Meta Threshold and KR Calibration can greatly help model to decide proper label numbers.

Related Work

Usually, multi-label classification (MLC) methods rely on thresholds to predict multiple labels. For MLC problem in NLP, Wu, Xiong, and Wang (2019) leverage meta learning to estimate thresholds for data-rich setting. For thresholding of intent detection, Gangadharaiah and Narayanaswamy (2019) leverage a fixed threshold over intent possibility. Xu et al. (2017) learn threshold with linear regression.

Without threshold, one solution to MLC is Label Powerset (LP) (Tsoumakas, Katakis, and Vlahavas 2010; Tsoumakas and Vlahavas 2007), which regards combination of multiple labels as a single label. Xu and Sarikaya (2013) explore idea of LP in multi-label intent detection. However, LP often

suffers from data sparseness from label combination even in data-rich settings. In addition to LP, Kim, Ryu, and Lee (2017) propose to learn multi-intent detection from single intent data. They first detect sub-sentence, and then predict single intents on sub-sentences. But, their method is limited by the explicit conjunctions in data, and it is hard to learn to detect sub-sentence in few-shot setting.

Few-shot learning in NLP has been widely explored for single-label classification tasks, including text classification (Sun et al. 2019; Geng et al. 2019; Yan, Zheng, and Cao 2018; Yu et al. 2018; Bao et al. 2020; Vlasov, Drissner-Schmid, and Nichol 2018), relation classification (Lv et al. 2019; Gao et al. 2019; Ye and Ling 2019), sequence labeling (Hou et al. 2020). However, few-shot multi-label problem is less investigated. Previous works focus on computer vision (Xiang et al. 2019; Alfassy et al. 2019) and signal processing (Cheng, Chou, and Yang 2019). Rios and Kavuluru (2018) investigate few-shot MLC for medical texts. But, their method requires descriptions and EMR structure of labels, which are often hard to obtain and not available in our task. For the use of label name semantics, it has been proven to be effective for data scarcity problem of both slot filling (Bapna et al. 2017; Lee and Jha 2019; Shah et al. 2019; Hou et al. 2020) and intent detection (Xia et al. 2020; Krone, Zhang, and Diab 2020; Chen, Hakkani-Tür, and He 2016). Our method shares the similar idea but introduces it to tackle the special challenges of multi-label setting.

Conclusion

In this paper, we explore the few-shot learning problem of multi-label intent detection. To estimate a reasonable threshold with only a few support examples, we propose the Meta Calibrated Threshold that adaptively combines prior experience and domain-specific knowledge. To obtain label-instance relevance score under few-shot setting, we introduce a metric learning based method with Anchored Label Representation. It provides well-separated label representations for label-instance similarity calculation. Experiment results validate that both the Meta Calibrated Threshold and Anchored Label Representation can improve the few-shot multi-label intent detection.

Acknowledgments

We are grateful for the helpful comments and suggestions from the anonymous reviewers. This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 61772153.

References

Alfassy, A.; Karlinsky, L.; Aides, A.; Shtok, J.; Harary, S.; Feris, R.; Giryes, R.; and Bronstein, A. M. 2019. LaSO: Label-Set Operations networks for multi-label few-shot learning. In *Proc. of the CVPR*, 6548–6557.

Bao, Y.; Wu, M.; Chang, S.; and Barzilay, R. 2020. Few-shot Text Classification with Distributional Signatures. In *Proc. of the ICLR*.

Bapna, A.; Tur, G.; Hakkani-Tur, D.; and Heck, L. 2017. Towards zero-shot frame semantic parsing for domain scaling. *arXiv preprint arXiv:1707.02363* .

Chen, Y.-N.; Hakkani-Tür, D.; and He, X. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6045–6049. IEEE.

Cheng, K.-H.; Chou, S.-Y.; and Yang, Y.-H. 2019. Multi-label Few-shot Learning for Sound Event Recognition. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 1–5.

Clark, K.; Luong, M.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proc. of the ICLR*.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proc. of the CVPR*, 9268–9277.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the NAACL-HLT, Volume 1 (Long and Short Papers)*, 4171–4186.

Eric, M.; Krishnan, L.; Charette, F.; and Manning, C. D. 2017. Key-Value Retrieval Networks for Task-Oriented Dialogue. In Jokinen, K.; Stede, M.; DeVault, D.; and Louis, A., eds., *Proc. of the SIGdial*, 37–49.

Finn, C. B. 2018. *Learning to Learn with Gradients*. University of California, Berkeley.

Gangadharaiah, R.; and Narayanaswamy, B. 2019. Joint Multiple Intent Detection and Slot Labeling for Goal-Oriented Dialog. In *Proc. of the ACL*, 564–569.

Gao, T.; Han, X.; Xie, R.; Liu, Z.; Lin, F.; Lin, L.; and Sun, M. 2019. Neural Snowball for Few-Shot Relation Learning. *arXiv preprint arXiv:1908.11007* .

Geng, R.; Li, B.; Li, Y.; Zhu, X.; Jian, P.; and Sun, J. 2019. Induction networks for few-shot text classification. In *Proc. of the EMNLP-IJCNLP*, 3895–3904.

Hou, Y.; Che, W.; Lai, Y.; Zhou, Z.; Liu, Y.; Liu, H.; and Liu, T. 2020. Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network. In *Proc. of the ACL*.

Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proc. of the ACL, Volume 1: Long Papers*, 328–339.

Kim, B.; Ryu, S.; and Lee, G. G. 2017. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications* 76(9): 11377–11390.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of the ICLR*. URL http://arxiv.org/abs/1412.6980.

Krone, J.; Zhang, Y.; and Diab, M. 2020. Learning to Classify Intents and Slot Labels Given a Handful of Examples. *arXiv preprint arXiv:2004.10793* .

Lee, S.; and Jha, R. 2019. Zero-shot adaptive transfer for conversational language understanding. In *Proc. of the AAAI*, volume 33, 6642–6649.

Lv, X.; Gu, Y.; Han, X.; Hou, L.; Li, J.; and Liu, Z. 2019. Adapting Meta Knowledge Graph Information for Multi-Hop Reasoning over Few-Shot Relations. *arXiv preprint arXiv:1908.11513* .

Nadaraya, E. A. 1964. On estimating regression. *Theory of Probability & Its Applications* 9(1): 141–142.

Qin, L.; Xu, X.; Che, W.; and Liu, T. 2020. TD-GIN: Token-level Dynamic Graph-Interactive Network for Joint Multiple Intent Detection and Slot Filling. *arXiv preprint arXiv:2004.10087* .

Reimers, N.; and Gurevych, I. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proc. of the EMNLP*.

Rios, A.; and Kavuluru, R. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proc. of the EMNLP*, volume 2018, 3132.

Shah, D. J.; Gupta, R.; Fayazi, A. A.; and Hakkani-Tür, D. 2019. Robust Zero-Shot Cross-Domain Slot Filling with Example Values. In *Proc. of the ACL, Volume 1: Long Papers*, 5484–5490.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Proc. of NeurIPS*, 4077–4087.

Sun, S.; Sun, Q.; Zhou, K.; and Lv, T. 2019. Hierarchical Attention Prototypical Networks for Few-Shot Text Classification. In *Proc. of the EMNLP-IJCNLP*, 476–485.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of the NAACL*, 173–180.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2010. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering* 23(7): 1079–1089.

Tsoumakas, G.; and Vlahavas, I. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *European conference on machine learning*, 406–417. Springer.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching networks for one shot learning. In *Proc. of NeurIPS*, 3630–3638.

Vlasov, V.; Drissner-Schmid, A.; and Nichol, A. 2018. Few-Shot Generalization Across Dialogue Tasks. *arXiv preprint arXiv:1811.11707* .

Wang, G.; Li, C.; Wang, W.; Zhang, Y.; Shen, D.; Zhang, X.; Henao, R.; and Carin, L. 2018. Joint embedding of words and labels for text classification. In *Proc. of the ACL*, 2321–2331.

Williams, J. D.; Raux, A.; Ramachandran, D.; and Black, A. 2012. Dialog state tracking challenge handbook. Technical report, Technical report, Microsoft Research.

Wu, J.; Xiong, W.; and Wang, W. Y. 2019. Learning to Learn and Predict: A Meta-Learning Approach for Multi-Label Classification. In *Proc. of the EMNLP*, 4353–4363.

Xia, C.; Zhang, C.; Nguyen, H.; Zhang, J.; and Yu, P. 2020. CG-BERT: Conditional Text Generation with BERT for Generalized Few-shot Intent Detection. *arXiv preprint arXiv:2004.01881* .

Xiang, L.; Jin, X.; Ding, G.; Han, J.; and Li, L. 2019. Incremental Few-Shot Learning for Pedestrian Attribute Recognition. In Kraus, S., ed., *Proc. of the IJCAI*, 3912–3918.

Xu, G.; Lee, H.; Koo, M.-W.; and Seo, J. 2017. Convolutional neural network using a threshold predictor for multi-label speech act classification. In *IEEE international conference on big data and smart computing (BigComp)*, 126–130.

Xu, P.; and Sarikaya, R. 2013. Exploiting shared information for multi-intent natural language sentence classification. In *Proc. of Interspeech*, 3785–3789.

Yan, L.; Zheng, Y.; and Cao, J. 2018. Few-shot learning for short text classification. *Multimedia Tools and Applications* 1–12.

Ye, Z.; and Ling, Z. 2019. Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification. In *Proc. of the ACL, Volume 1: Long Papers*, 2872–2881.

Young, S.; Gašić, M.; Thomson, B.; and Williams, J. D. 2013. Pomdp-based statistical spoken dialog systems: A review. In *Proc. of the IEEE*, volume 101, 1160–1179. IEEE.

Yu, M.; Guo, X.; Yi, J.; Chang, S.; Potdar, S.; Cheng, Y.; Tesauro, G.; Wang, H.; and Zhou, B. 2018. Diverse Few-Shot Text Classification with Multiple Metrics. In *Proc. of the NAACL-HLT, Volume 1 (Long Papers)*, 1206–1215.

Zhang, M.-L.; and Zhou, Z.-H. 2013. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* 26(8): 1819–1837.