Diversity Features Enhanced Prototypical Network for Few-Shot Intent Detection

Fengyi Yang[1,2,3], **Xi Zhou**[1,2,3*], **Yi Wang**[1,2,3], **Abibulla Atawulla**[1,2,3] and **Ran Bi**[1,2,3]
[1]Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China
yangfengyi17@mails.ucas.edu.cn, {zhouxi, wangyi}@ms.xjb.ac.cn, {aibibulaatawula19, biran19}@mails.ucas.ac.cn

Abstract

Few-shot Intent Detection (FSID) is a challenging task in dialogue systems due to the scarcity of available annotated utterances. Although existing few-shot learning approaches have made remarkable progress, they fall short in adapting to the Generalized Few-shot Intent Detection (GFSID) task where both seen and unseen classes are present. A core problem of the simultaneous existence of these two tasks is that limited training samples fail to cover the diversity of user expressions. In this paper, we propose an effective Diversity Features Enhanced Prototypical Network (DFEPN) to enhance diversity features for novel intents by fully exploiting the diversity of known intent samples. Specially, DFEPN generates diversity features of samples in the hidden space via a diversity feature generator module and then fuses these features with original support vectors to get a more suitable prototype vector of each class. To evaluate the effectiveness of our model on both FSID and GFSID tasks, we carry out sufficient experiments on two benchmark intent detection datasets. Results demonstrate that our proposed model outperforms existing state-of-the-art methods and keeps stable performance on both two tasks.

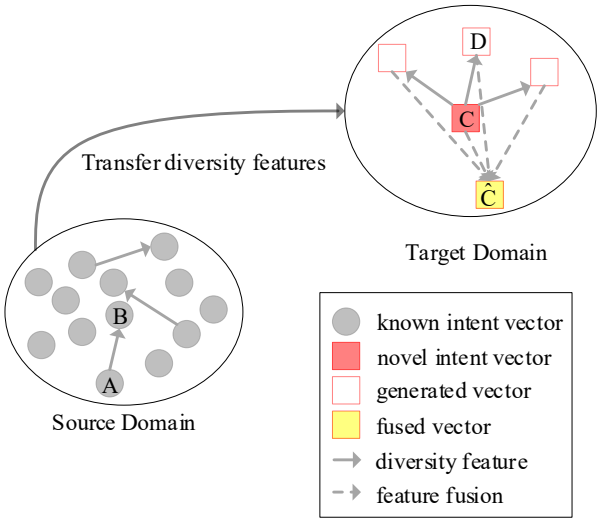

Figure 1: An illustration of transferring diversity features from the source domain to the target domain. Firstly, we randomly select sample pairs from a source domain (three sample pairs in this figure) and regard internal differences of sample pairs as diversity features (e.g., "B" – "A" is a diversity feature). Next, these diversity features are added to a novel intent vector in the target domain to obtain generated vectors (e.g., "D" = "C" + "B" – "A", "D" is a generate vector). Finally, fusing generated vectors with the original novel intent vector and then getting a fused vector "Ĉ".

1 Introduction

Intent Detection (ID), also called Intent Classification, is a crucial task in human-machine dialogue systems. ID is to identify user's real intents behind given utterances and then extracted intents are involved in downstream tasks. Recent advance in deep learning and availability of massive data have prompted a series of effective detection algorithms in ID domain.

However, the amount of conversation data is typically limited in brand-new dialogue systems. Therefore, the system should be able to learn the ID model from limited samples. In this Few-shot Intent Detection (FSID) task, data-driven methods could easily cause serious overfitting problems. Researchers propose few-shot learning (FSL) algorithms [Vinyals *et al.*, 2016; Sung *et al.*, 2018] to overcome the problem of lacking training samples. The driving force is to use the knowledge outside the target domain and few labeled data to construct a model with a good discrimination ability of recognizing novel classes. Generally, few-shot learning algorithms use the meta-learning procedure for training and testing [Finn *et al.*, 2017]. To simulate the situation of insufficient data, meta-learning procedure constructs a larger number of different meta-tasks in the training stage. A meta-task is constructed by sampling a small training set (support set) and test set (query set) from rich data domains. In each meta task, the objective is to correctly infer the class of query set samples by using support set samples. In the testing stage, the model can immediately identify the novel category according to limited samples instead of training a new model.

*Corresponding Author

While the dialogue system keeps operating, new intents are continuously emerging, therefore the ID model should be able to distinguish the existing intents and newly identified intents appropriately. Existing few-shot learning approaches fall short in adapting to this Generalized Few-shot Intent Detection (GFSID) task [Xia *et al.*, 2020] where both seen and unseen classes are present. A main challenge of both FSID and GFSID tasks is that limited training samples fail to cover the diversity of user expressions.

In this paper, motivated by the phenomenon of word analogies in word2vec [Mikolov *et al.*, 2013], we propose a novel method named Diversity Features Enhanced Prototypical Network (DFEPN) to tackle the above challenge. In word2vec, a surprising property of word vectors is that word analogies can often be solved with vector arithmetic, e.g., "King" = "Queen" + "man" – "woman". Among them, "King" and "Queen" belong to same category while "man" and "woman" belong to another category. Similar to word vectors, the difference between two sentence vectors from the same category can be regarded as a diversity feature of user expression. We treat an intent category containing abundant labeled data as a source domain and a novel intent category as a target domain. Therefore, we propose to transfer diversity features from source domains to target domains, and Figure 1 shows a simple example. Firstly, we randomly select sample pairs from a source domain and calculate diversity features by the internal differences of sample pairs. After that, diversity features are added to a novel intent vector to obtain generated vectors. Fusing generated vectors with the original novel intent vector and getting a fused vector. In this way, we enhance features for novel intents in the high dimensional space. Besides, our method uses meta-learning procedure and construct abundant meta-tasks for training. To learn the interactive knowledge between support and query set vectors, we reuse the process of transferring diversity features to enhance query set vectors at the same time.

The primary contributions of this paper are as follows:

- We propose a novel method, DFEPN, to solve the FSID and GFSID problem. DFEPN uses diversity features of source domains to enhance the diversity of samples in target domains.
- We improve the performance of the method by enhancing support and query set vectors at the same time. It extracts the interactive knowledge between support and query set vectors.
- We conduct experiments on two benchmark intent detection datasets to test the performance of our method in both FSID and GFSID scenarios. Experimental results show that our method is superior to previous state-of-the-art methods.

2 Related Work

2.1 Few-shot Learning

Few-shot learning aims to learn a model, which is trained on known category samples and can classify unknown category samples well. FSL methods can be divided into three types: optimization-based methods, metric-based methods and generation-based methods. Optimization-based methods aim to learn general initialization model parameters to ensure that parameters can be optimized well in few steps [Finn *et al.*, 2017; Rusu *et al.*, 2018]. Metric-based methods, including Matching Network (MN) [Vinyals *et al.*, 2016], Prototypical Network (PN) [Snell *et al.*, 2017] and Relation Network (RN) [Sung *et al.*, 2018], aim to learn an appropriate classifier by modeling the distance distribution of samples. Generation-based methods aim to synthesize new samples for target classes based on few samples [Chen *et al.*, 2020; Sun *et al.*, 2021].

2.2 Generalized Few-shot Learning

Generalized few-shot learning is an extension of FSL, requiring the model to correctly classify samples in a joint label space that consists of existing and emerging categories. GFSL is a new investigation domain and current works mainly focus on the computer vision. Li et al. [Li *et al.*, 2019] learn global class representations in the joint label space utilizing both existing and emerging category training samples. Shi et al. [Shi *et al.*, 2019] use graph-convolution to model inter-class relationships and learn representative prototypes for joint classes. In the field of intent detection, Xia et al. [Xia *et al.*, 2020] explore this problem first and simulate the distribution of unlabeled samples by utilizing labeled samples. Nguyen et al. [Nguyen *et al.*, 2020] consider that coarser-grained semantics can provide additional knowledge and propose multiple semantic components via multi-head self-attention.

3 Problem Formulation

Few-shot intent detection (FSID) and generalized few-shot intent detection (GFSID) are defined formally in this section.

The intent with a large number of labeled data is defined as seen intent and the intent with few labeled data as novel intent. We denote seen intent label space as Y_s, novel intent label space as Y_n, and $Y_s \cap Y_n = \varnothing$. We define the seen intent set, $D_s = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{N_s}, y_{N_s})\}, y_i \in Y_s$ where x denotes a user utterance and N_s denotes the number of labeled samples. Similarly, the novel intent set $D_n = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{N_n}, y_{N_n})\}, y_i \in Y_n$.

As summarized in Equation (1), the objective of FSID is to determine the most likely category of a new unlabeled sample x whose label belongs to Y_n.

$$\hat{y} = \arg\max_{y \in Y_n} p(y \mid x, D_n) \tag{1}$$

Unlike FSID, GFSID aims at classifying an unlabeled sample not only as the novel intents but also as the seen intents. We denote a joint intent set as Y_j, $Y_j = Y_s \cup Y_n$, a joint intent set as $D_j = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{N_j}, y_{N_j})\}, y_i \in Y_j$.

As summarized in Equation (2), given an unlabeled sample x whose label belongs to Y_j, the objective function of GFSID is to infer the most likely category of x.

$$\hat{y} = \arg\max_{y \in Y_j} p(y \mid x, D_j) \tag{2}$$

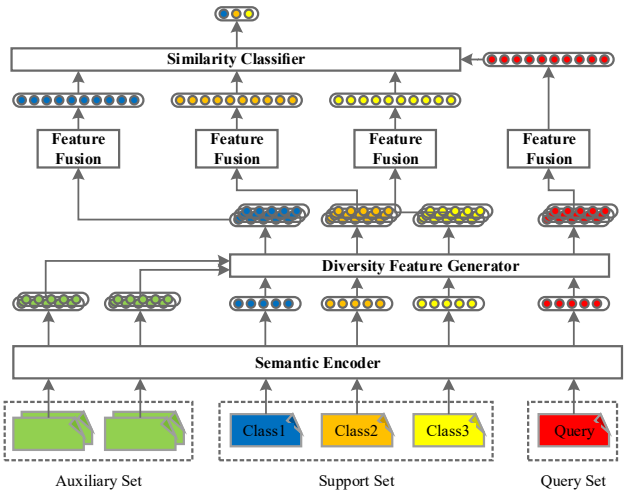

Figure 2: An overview of the Diversity Features Enhanced Prototypical Network.

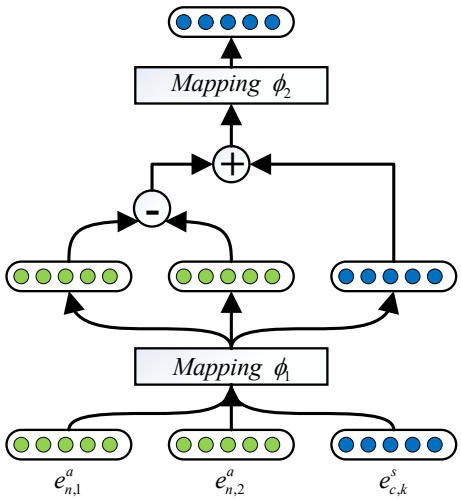

Figure 3: Structure of the Diversity Feature Generator.

4 Method

4.1 Overall Architecture

An overview of the Diversity Features Enhanced Prototypical Network (DFEPN) is shown in Figure 2. Our model framework includes four components: Semantic Encoder, Diversity Feature Generator, Feature Fusion and Similarity Classifier.

Our method uses a meta-learning framework and construct multiple training episode, simulating the situation of insufficient training samples. Generally, a training episode consists of support set and query set. Our model needs an additional auxiliary set to generate the diversity features of samples. We randomly select M intents (called auxiliary intents) from the training set and all examples belonging to these intents form the auxiliary training set D_{aux}. All samples of the remaining intents (called base intents) form the base training set D_{base}. For a C-way K-shot problem, we randomly select C intents from the base training set and then choose K samples within each selected intent. These samples constitute the support set, $S = \bigcup_{c=1}^{C}\{x^s_{c,k}, y^s_{c,k}\}_{k=1}^{K}$, while a subset of the remaining samples act as the query set, $Q = \{x^q_l, y^q_l\}_{l=1}^{L}$. An auxiliary set is formed by randomly selecting N sample pairs from the auxiliary training set, $A = \bigcup_{n=1}^{N}\{x^a_{n,1}, x^a_{n,2}\}$. A set of sample pairs consists of two different samples pertaining to the same intents.

4.2 Semantic Encoder

Semantic Encoder (SE) aims to learn the semantic information of sentences and encode it into the high-dimensional space. Recently, the method of pre-training on a large corpus and then fine-tuning on target tasks has achieved great success. In FSL tasks, although it is unable to provide enough training samples for fine-tuning, the task-agnostic knowledge learned by pre-training is still helpful for semantic extraction [Deng *et al.*, 2020]. Different from the previous work [Nguyen *et al.*, 2020], we hope our model can benefit from the advantage of pre-trained language models and use BERT [Devlin *et al.*, 2019] as the Semantic Encoder.

Instead of using the pre-trained model directly, we further pre-train BERT on the Masked Language Model (MLM) Task with the training dataset, which will greatly improve the quality of embedded representation [Dopierre *et al.*, 2021]. We use the final representation of the [CLS] token as the semantic representation of the whole sentence x, $e = SE(x)$, $e \in \mathbb{R}^d$.

4.3 Diversity Feature Generator

Diversity Feature Generator (DFG) aims to transfer diversity features from the auxiliary set to target intent vectors. Samples from the auxiliary set A, the support set S and the query set Q are encoded as vectors e^a, e^s and e^q. The structure of the DFG is shown in Figure 3, where ϕ_1 and ϕ_2 are two linear layers.

First, sample pair vectors $e^a_{n,1}$, $e^a_{n,2}$ and a support vector $e^s_{c,k}$ are mapped into a new hidden space by ϕ_1, $\hat{e} = \phi_1(e)$, $\hat{e} \in \mathbb{R}^m$. The difference between $\hat{e}^a_{n,1}$ and $\hat{e}^a_{n,2}$ can be regarded as the diversity feature within the same category. Then, the enhanced sample vector is generated by adding this diversity feature to the target sample vector $\hat{e}^s_{c,k}$, $\hat{e}^a_{n,1} - \hat{e}^a_{n,2} + \hat{e}^s_{c,k}$. Finally, after mapping it into the original vector space by ϕ_2, we can get the generated vector $e^{sg}_{c,k}$. More specifically:

$$\begin{aligned} e^{sg}_{c,k} &= DFG(e^a_{n,1}, e^a_{n,2}, e^s_{c,k}) \\ &= \phi_2(\phi_1(e^a_{n,1}) - \phi_1(e^a_{n,2}) + \phi_1(e^s_{c,k})) \end{aligned} \tag{3}$$

The auxiliary set A includes N sample pairs. Therefore, for each support vector, N diversity features are generated by the DFG.

4.4 Feature Fusion

The Feature Fusion (FF) module fuses generated vectors with the original support vector and then gets the prototype vector of each class. The algorithm of the Feature Fusion process is presented in Algorithm 1.

For each generated vectors $e^{sg}_{c,k,n}$, $n \in [1, 2, \cdots, N]$, we employ a residual connection [He *et al.*, 2016] after the DFG,

Algorithm 1 Feature Fusion

Input: sample vector $e^s_{c,k}$ in support set S, generated features $e^{sg}_{c,k,n}$
Parameter: linear layer weights W and bias b
Output: prototype vector z_c

for all samples $k = 1, 2, \cdots, K$ in class c **do**
  for $n = 1, 2, \cdots, N$ **do**
    $\hat{e}^{sg}_{c,k,n} = LayerNorm(e^{sg}_{c,k,n} + e^s_{c,k})$
  end for
  $\hat{e}^{sg}_{c,k} = concat(\frac{1}{N}\sum_{n=1}^{N} \hat{e}^{sg}_{c,k,n}, e^s_{c,k})$
  $z_{c,k} = \tanh(W\hat{e}^{sg}_{c,k} + b)$
end for
$z_c = \frac{1}{K}\sum_{k=1}^{K} z_{c,k}$
return z_c

followed by layer normalization [Ba *et al.*, 2016], and the output is

$$\hat{e}^{sg}_{c,k,n} = LayerNorm(e^{sg}_{c,k,n} + e^s_{c,k}) \tag{4}$$

Next, the average of the N outputs and the original support vector are spliced together, the fused vector is

$$\hat{e}^{sg}_{c,k} = concat(\frac{1}{N}\sum_{n=1}^{N} \hat{e}^{sg}_{c,k,n}, e^s_{c,k}), \hat{e}^{sg}_{c,k} \in \mathbb{R}^{2d} \tag{5}$$

And then it is fed to a linear layer, whose weight is W and bias is b,

$$z_{c,k} = \tanh(W\hat{e}^{sg}_{c,k} + b), W \in \mathbb{R}^{2d\times 2d}, b \in \mathbb{R}^{2d}. \tag{6}$$

Finally, averaging the K enhanced sample vectors to get the prototype representation of the corresponding category,

$$z_c = \frac{1}{K}\sum_{k=1}^{K} z_{c,k}. \tag{7}$$

4.5 Similarity Classifier

In order to learn the interactive knowledge between support and query set vectors, we also enhance query vectors through the above modules. A query vector e^q_l is concerted to the enhanced query vector $\hat{e}^q_l$,

$$\hat{e}^q_l = FF(\bigcup_{n=1}^{N} DFG(e^a_{n,1}, e^a_{n,2}, e^q_l), e^q_l). \tag{8}$$

The article [Snell *et al.*, 2017] notes that Euclidean distance is more suitable for classifier of prototypical networks comparing with cosine distance. Therefore, we decide to use Euclidean distance to measure the similarity between $\hat{e}^q_l$ and z_c, getting the classification score,

$$s^q_{l,c} = Euc(\hat{e}^q_l, z_c). \tag{9}$$

For each training episode, given auxiliary set A, support set S and query set $Q = \{x^q_l, y^q_l\}^L_{l=1}$, the training objective is to minimize the cross-entropy loss on the query set. The loss function is as follows:

$$L(A, S, Q) = -\frac{1}{C}\sum_{c=1}^{C}\frac{1}{L}\sum_{l=1}^{L} y_q \log(\hat{y}_q) \tag{10}$$

where $\hat{y}_q = softmax(-s^q_{l,c})$ is the predicted probabilities of C classes.

	SNIPS	NLUE
Auxiliary intents	1	10
Base intents	4	38
Novel intents	2	16
Joint intents	7	64
Auxiliary samples	1585	1417
Base samples	6302	4976
Novel samples	769	274
Joint samples	2688	1873

Table 1: Detailed division of SNIPS and NLUE (Fold 1) datasets.

5 Experiments

In this section, we conduct experiments on two benchmark intent detection datasets to test the performance of our method in both FSID and GFSID scenarios. we select various outstanding FSL algorithms as baselines, including **MN** [Vinyals *et al.*, 2016], **PN** [Snell *et al.*, 2017], **RN** [Sung *et al.*, 2018], **HATT** [Gao *et al.*, 2019]), **MLMAN** [Ye and Ling, 2019], **HAPN** [Sun *et al.*, 2019], **SMAN** [Nguyen *et al.*, 2020] and **MLADA** [Han *et al.*, 2021]. **MLADA** is one of the state-of-the-art few-shot text classification models. **SMAN** is the state-of-the-art model for the GFSID task.

5.1 Datasets

In this section, we describe the details of two benchmark intent detection datasets used in experiments: SNIPS Natural Language Understanding benchmark (SNIPS) [Coucke *et al.*, 2018] and NLU-Evaluation Dataset (NLUE) [Liu *et al.*, 2019].

Following [Nguyen *et al.*, 2020], we divide these two datasets to evaluate the performance of FSID and GFSID. As shown in Table 1, seen intents are divided into auxiliary intents and base intents randomly in advance. GFSID testing samples (joint samples) include all novel intent samples and 20 percent of seen intent samples. We use auxiliary and base samples to train our model and test it on novel samples (FSID) and joint samples (GFSID).

SNIPS: According to the data partition method in [Xia *et al.*, 2018], the number of novel intents is set to 2. After selecting novel intents, we select one intent from the remaining intents as the auxiliary intent while the last four intents are base intents.

NLUE: Following [Liu *et al.*, 2019], we choose the same 16 intents as novel intents, 10 intents as auxiliary intents and the last 38 intents as base intents.

5.2 Implementation Details

Following [Nguyen *et al.*, 2020], we randomly construct 1000 training episodes with K= 1, 5 and C=2 (FSID) or 4 (GFSID) on SNIPS dataset. For NLUE dataset, we also structure 1000 training episodes with the same K and C=5 (FSID) or 10 (GFSID). In the training stage, we extract 20 samples from each class to form the query set on both datasets. Our model is trained with learning rate equals 2e-5 and the hidden dimension d is 768, m is 1024. The number of auxiliary sample

Model	1-shot						5-shot					
	Non-episodic(noneps)			Episodic(eps)			Non-episodic(noneps)			Episodic(eps)		
	S-J	S-N	h-acc	S-J	S-N	h-acc	S-J	S-N	h-acc	S-J	S-N	h-acc
MN (2016)	73.50	86.99	79.68	82.67	85.97	84.29	77.31	90.12	83.22	84.60	90.12	87.27
PN (2017)	71.61	94.67	81.54	87.04	89.91	88.45	85.31	93.11	89.04	91.05	92.96	92.00
RN (2018)	74.94	88.14	81.01	85.63	87.63	86.62	64.09	87.99	74.16	79.25	83.86	81.49
HATT (2019)	71.54	93.76	81.16	84.51	93.55	88.80	86.53	94.15	90.18	91.85	93.98	92.90
MLMAN (2019)	78.61	94.41	85.79	87.77	92.48	90.06	79.58	95.06	86.64	89.27	94.13	91.64
HAPN (2019)	74.33	91.42	81.99	85.37	91.52	88.34	86.19	92.85	89.40	89.40	94.32	91.79
SMAN (2020)	81.85	95.84	88.29	88.10	95.48	91.64	87.87	97.01	92.21	93.18	96.81	94.96
MLADA (2021)	71.15	95.45	81.53	79.14	95.30	86.47	88.90	95.84	92.24	92.89	96.68	94.75
Ours	**92.30**	**97.66**	**94.90**	**95.41**	**96.40**	**95.90**	**93.42**	**97.92**	**95.62**	**96.27**	**97.86**	**97.06**

Table 2: Comparison of accuracy (%) on SNIPS.

Model	1-shot						5-shot					
	Non-episodic(noneps)			Episodic(eps)			Non-episodic(noneps)			Episodic(eps)		
	S-J	S-N	h-acc	S-J	S-N	h-acc	S-J	S-N	h-acc	S-J	S-N	h-acc
MN (2016)	62.30	35.40	45.15	76.21	58.16	65.97	56.27	52.55	54.35	78.85	73.69	76.18
PN (2017)	62.63	36.86	46.41	80.78	58.44	67.82	66.20	59.49	62.67	85.13	79.39	82.16
RN (2018)	56.75	27.74	37.26	73.57	49.47	59.16	46.50	34.31	39.49	75.23	62.15	68.07
HATT (2019)	64.01	34.67	44.98	81.39	58.47	68.05	67.86	61.15	64.33	78.41	74.74	76.53
MLMAN (2019)	63.12	41.61	51.60	82.65	60.64	69.95	60.70	59.49	60.09	84.45	76.70	80.39
HAPN (2019)	60.44	41.78	49.41	82.00	62.39	70.86	68.34	64.60	66.42	84.75	80.11	82.36
SMAN (2020)	66.10	44.11	52.91	**89.54**	62.81	73.83	72.18	66.96	69.47	87.76	81.12	84.31
MLADA (2021)	45.31	45.99	45.65	67.91	63.36	65.56	61.93	64.20	63.04	82.56	80.18	81.35
Ours	**72.09**	**58.01**	**64.29**	87.35	**77.81**	**82.30**	**82.28**	**77.50**	**79.82**	**93.17**	**89.87**	**91.49**

Table 3: Comparison of accuracy (%) on NLUE.

pairs N is 25. We use BERT models from the Hugging Face [Wolf *et al.*, 2020] team.

We evaluate models on both FSID and GFSID tasks. The S-J accuracy represents the GFSID testing results while the S-N accuracy denotes FSID results. H-acc is the harmonic mean of S-J accuracy and S-N accuracy, used to measure the comprehensive ability of models in both FSID and GFSID scenarios. Besides, these models are evaluated in two different testing procedures: Episodic Evaluation (eps) and Non-episodic Evaluation (noneps).

Episodic Evaluation: Traditional FSL methods follow a principle: test and train conditions must match [Vinyals *et al.*, 2016]. We construct 1000 episodes with 5 samples per class to form the query set in the testing stage.

Non-episodic Evaluation: [Nguyen *et al.*, 2020] points out the above-mentioned episodic evaluation method lacks practicability in realistic applications. They recommend testing unlabeled samples only once and these samples are selected from the whole dataset in advance.

On SNIPS dataset, we use 5 different random seeds to carry out five groups of experiments, and the final average accuracy is shown in Table 2. On NLUE dataset, we use 10-fold cross-validation to test our model and the final average accuracy is shown in Table 3.

5.3 Comparisons with State-of-the-arts

Table 2 and 3 indicate an overall improvement compare to previous baselines.

Our model performs worse than SMAN in only one situation in which testing 10-way 1-shot S-J on NLUE by episodic evaluation ($87.35 < 89.54$). However, when we changed the testing procedure to non-episodic evaluation, our model achieved the best S-J accuracy which is 6 percentage points higher than SMAN. This is mainly because the number of test samples per class in NLUE is much less than that in SNIPS ($29.3 < 384$), leading to a lot of repeated choices when constructing testing episodes. It magnifies the uncertainty of testing. We can see that all models have the same consistent trend of S-J accuracy under two different testing conditions on SNIPS when increasing the number of support set samples from 1 to 5. For example, the S-J accuracy of MN increases under both two testing standards (noneps: 73.50 to 77.31; eps: 82.67 to 84.60). However, only four models (PN, HAPN, MLADA and our model) maintain this consistency on NLUE and it proves the existence of the testing uncertainty mentioned above.

In most cases, the recognition accuracy of all models is decreased when evaluated on non-episodic evaluation instead of episodic evaluation because non-episodic evaluation is more challenging. However, S-N accuracies on SNIPS dataset

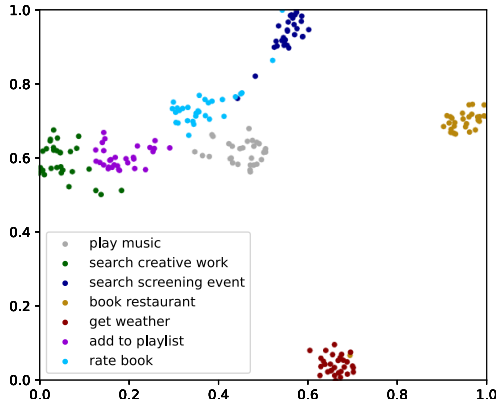

Figure 4: t-SNE visualization of the enhanced vectors.

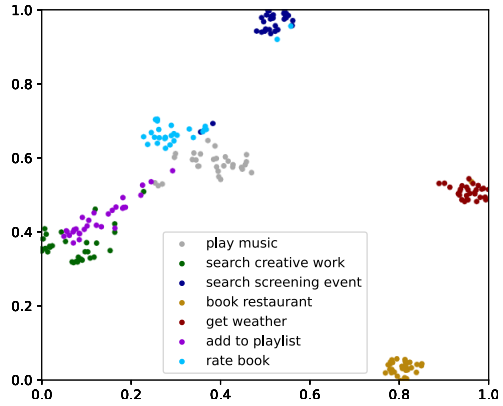

Figure 5: t-SNE visualization of the original vectors.

	1-shot		5-shot	
	noneps	eps	noneps	eps
w/o FP	74.45	82.40	90.02	93.67
w/o DFG	89.00	93.35	92.74	94.92
w/o FF	89.87	93.52	95.58	96.79
w/o QE	93.82	95.85	94.90	96.71
Full Model	**94.90**	**95.90**	**95.62**	**97.06**

Table 4: Ablation study of h-acc (%) on SNIPS

which are evaluated by both two testing procedures are almost same. This is because the FSID task can be seemed as a binary classification problem in both testing conditions.

Our model gets more improvement in non-episodic testing than episodic testing. It shows that our model can be well adapted to the realistic environment. Meanwhile, the performance improvement of our model is more obvious in 1-shot situation. This is because the smaller number of support set samples, the more difficult it is to reflect the diversity of samples. In this case, enhancing the diversity of samples can effectively improve the performance.

5.4 Ablation Study

In this section, we conduct a large number of ablation studies to observe the effect of each individual component in the model. We test the performance of our full model and its ablations on SNIPS dataset. H-acc results are shown in Table 4.

"w/o FP" means that we use the pre-trained BERT model directly instead of further pre-training BERT with SNIPS dataset. The performances in the 1-shot classification and 5-shot classification decrease by 20.45% and 5.6% on non-episodic procedure. It proves that our model needs Semantic Encoder to have a good initial semantic parsing ability, otherwise the difference between two auxiliary vectors is too huge, causing a lot of noise.

"w/o DFG" means that we directly use Prototypical Network with further pre-training BERT. On non-episodic procedure, the performances are reduced by 5.9% and 2.88% in 1-shot and 5-shot scenarios. To further analyze the impact of the DFG module, we feed 210 test samples (30 samples per intent) into models and get these high-dimensional representations. Next, we use t-SNE [Van der Maaten and Hinton, 2008] to reduce the dimension of original vectors and enhanced vectors. As shown in Figure 4, enhanced vectors are much more diverse and the boundary between two classes is also clearer, comparing with original vectors as in Figure 5.

"w/o FF" uses a simple strategy which the enhanced feature is the average of the original vector and diversity features to replace the Feature Fusion module. The performances are decreased under all conditions, especially in the 1-shot situation. It proves that the module has strong correction ability, playing an important role in extreme circumstances.

"w/o QE" means that query vectors are not enhanced by DFG and FF. We connect the query vector with its copy vector instead. The results illustrate that enhanced query vectors can better adapt to the realistic scene.

6 Conclusion

In this paper, we propose an effective Diversity Features Enhanced Prototypical Network for both few-shot and generalized few-shot intent detection. DFEPN enhances the diversity of samples in target domains by transferring diversity features from source domains. Experimental results show that DFEPN is superior to previous state-of-the-art methods on the SNIPS and NLUE datasets for FSID and GFSID tasks. In the future, we will explore the effectiveness of DFEPN in other tasks.

Acknowledgments

This research is supported by the Natural Science Foundation for Distinguished Young Scholars of Xinjiang Uygur Autonomous Region (2022D01E04), the Xinjiang Key Laboratory Fund under Grant No.2020D04050, the West Light Foundation of The Chinese Academy of Sciences (Grant No.2018-XBQNXZ-A-003) and the Xinjiang Science and Technology Major Project (No.2020A02001-1).

References

[Ba *et al.*, 2016] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[Chen *et al.*, 2020] Mengting Chen, Yuxin Fang, Xinggang Wang, Heng Luo, Yifeng Geng, Xinyu Zhang, Chang Huang, Wenyu Liu, and Bo Wang. Diversity transfer network for few-shot learning. In *Proc. of AAAI*, pages 10559–10566, 2020.

[Coucke *et al.*, 2018] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190, 2018.

[Deng *et al.*, 2020] Shumin Deng, Ningyu Zhang, Zhanlin Sun, Jiaoyan Chen, and Huajun Chen. When low resource NLP meets unsupervised language model: Meta-pretraining then meta-learning for few-shot text classification (student abstract). In *Proc. of AAAI*, pages 13773–13774, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, pages 4171–4186, 2019.

[Dopierre *et al.*, 2021] Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. A neural few-shot text classification reality check. In *Proc. of EACL*, pages 935–943, 2021.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. of ICML*, pages 1126–1135, 2017.

[Gao *et al.*, 2019] Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proc. of AAAI*, pages 6407–6414, 2019.

[Han *et al.*, 2021] Chengcheng Han, Zeqiu Fan, Dongxiang Zhang, Minghui Qiu, Ming Gao, and Aoying Zhou. Meta-learning adversarial domain adaptation network for few-shot text classification. In *Proc. of ACL Findings*, pages 1664–1673, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of CVPR*, pages 770–778, 2016.

[Li *et al.*, 2019] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *Proc. of ICCV*, pages 9714–9723, 2019.

[Liu *et al.*, 2019] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. Benchmarking natural language understanding services for building conversational agents. In *International Workshop on Spoken Dialogue Systems*, pages 165–183, 2019.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Nguyen *et al.*, 2020] Hoang Nguyen, Chenwei Zhang, Congying Xia, and Philip S. Yu. Semantic matching and aggregation network for few-shot intent detection. In *Proc. of EMNLP Findings*, pages 1209–1218, 2020.

[Rusu *et al.*, 2018] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *CoRR*, abs/1807.05960, 2018.

[Shi *et al.*, 2019] Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. Relational generalized few-shot learning. *CoRR*, abs/1907.09557, 2019.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Proc. of NeurIPS*, pages 4077–4087, 2017.

[Sun *et al.*, 2019] Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. Hierarchical attention prototypical networks for few-shot text classification. In *Proc. of EMNLP*, pages 476–485, 2019.

[Sun *et al.*, 2021] Pengfei Sun, Yawen Ouyang, Wenming Zhang, and Xinyu Dai. MEDA: meta-learning with data augmentation for few-shot text classification. In *Proc. of IJCAI*, pages 3929–3935, 2021.

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proc. of CVPR*, pages 1199–1208, 2018.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, pages 2579–2605, 2008.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proc. of NeurIPS*, pages 3630–3638, 2016.

[Wolf *et al.*, 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proc. of EMNLP*, pages 38–45, 2020.

[Xia *et al.*, 2018] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. Zero-shot user intent detection via capsule neural networks. In *Proc. of EMNLP*, pages 3090–3099, 2018.

[Xia *et al.*, 2020] Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip S. Yu. CG-BERT: conditional text generation with BERT for generalized few-shot intent detection. *CoRR*, abs/2004.01881, 2020.

[Ye and Ling, 2019] Zhi-Xiu Ye and Zhen-Hua Ling. Multi-level matching and aggregation network for few-shot relation classification. In *Proc. of ACL*, pages 2872–2881, 2019.