Decoupling Pseudo Label Disambiguation and Representation Learning for Generalized Intent Discovery

Yutao Mou[1*], Xiaoshuai Song[1*], Keqing He[2*], Chen Zeng[1], Pei Wang[1]
Jingang Wang[2], Yunsen Xian[2], Weiran Xu[1*]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan, Beijing, China
{myt,songxiaoshuai,chenzeng,wangpei,xuweiran}@bupt.edu.cn
{hekeqing,wangjingang,xianyunsen}@meituan.com

Abstract

Generalized intent discovery aims to extend a closed-set in-domain intent classifier to an open-world intent set including in-domain and out-of-domain intents. The key challenges lie in pseudo label disambiguation and representation learning. Previous methods suffer from a coupling of pseudo label disambiguation and representation learning, that is, the reliability of pseudo labels relies on representation learning, and representation learning is restricted by pseudo labels in turn. In this paper, we propose a decoupled prototype learning framework (DPL) to decouple pseudo label disambiguation and representation learning. Specifically, we firstly introduce prototypical contrastive representation learning (PCL) to get discriminative representations. And then we adopt a prototype-based label disambiguation method (PLD) to obtain pseudo labels. We theoretically prove that PCL and PLD work in a collaborative fashion and facilitate pseudo label disambiguation. Experiments and analysis on three benchmark datasets show the effectiveness of our method.[1]

1 Introduction

Intent classification (IC) is an important component of task-oriented dialogue (TOD) systems. Traditional intent classification models are based on a closed-set hypothesis (Chen et al., 2019; Yang et al., 2021). That is, they rely on a pre-defined intent set provided by domain experts and can only recognize limited in-domain (IND) intent categories. However, users may input out-of-domain (OOD) queries in the real open world. OOD intent detection (Lin and Xu, 2019; Xu et al., 2020; Zeng et al., 2021; Wu et al., 2022a,b) identifies whether a user query falls outside the range of pre-defined IND intent set. Further, OOD intent discovery task (Lin et al., 2020; Zhang et al., 2022; Mou et al., 2022c,a) (also known as new intent discovery) groups unlabeled OOD intents into different clusters. However, all these work cannot expand the recognition scope of the existing IND intent classifier incrementally.

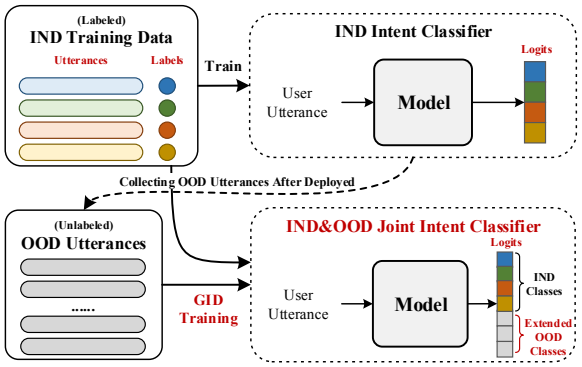

Figure 1: The illustration of GID task.

To solve the problem, Mou et al. (2022b) proposes the Generalized Intent Discovery (GID) task, which aims to simultaneously classify a set of labeled IND intents while discovering and recognizing new unlabeled OOD types incrementally. As shown in Fig 1, GID extends a closed-set IND classifier to an open-world intent set including IND and OOD intents and enables the dialogue system to continuously learn from the open world. Previous GID methods can be generally classified into two types: pipeline and end-to-end. The former firstly performs intent clustering and obtains pseudo OOD labels using K-means (MacQueen, 1967) or DeepAligned (Zhang et al., 2021a), and then mixes labeled IND data with pseudo-labeled OOD data to jointly learn a new classifier. However, pipeline-based methods separate the intent clustering stage from the joint classification stage and these pseudo OOD labels obtained in the intent clustering stage may induce severe noise to the joint classification. In addition, the deep semantic interaction between the labeled IND intents and the unlabeled OOD data is not fully considered in the intent clustering stage. To alleviate these

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

[1]We release our code at https://github.com/songxiaoshuai/DPL

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9661–9675

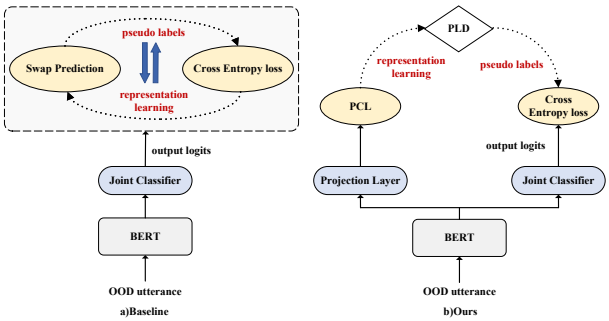

Figure 2: Comparison between baseline E2E and our proposed DPL method.

problems, Mou et al. (2022b) proposes an end-to-end (E2E) framework. It mixes labeled IND data with unlabeled OOD data in the training process and simultaneously learns pseudo OOD cluster assignments and classifies IND&OOD classes via self-labeling (Asano et al., 2020).

E2E framework achieves state-of-the-art results in most scenarios, but there are still two key challenges: (1) **Pseudo Label Disambiguation**. In the GID task, the performance of the joint classifier depends on pseudo labels of unlabeled OOD data, so we need to improve the reliability of pseudo labels during the training process, which is called "pseudo label disambiguation". (2) **Representation Learning**. We hope to form a clear cluster boundary for different IND and OOD intent types, which also benefits pseudo label disambiguation. As shown in Fig 2(a), the state-of-the-art E2E method (Mou et al., 2022b) adopts a self-labeling strategy (Asano et al., 2020; Fini et al., 2021) for pseudo label disambiguation and representation learning. Firstly, it obtains the pseudo label of an OOD query by its augmented view in a swapped prediction way for pseudo label disambiguation. Next, it uses the pseudo labels as supervised signals and adopts a cross-entropy classification loss for representation learning. Therefore, pseudo label disambiguation and representation learning are coupled, which has led to a non-trivial dilemma: the inaccurate pseudo labels will limit the quality of representation learning, and poor representation quality will in turn prevent effective pseudo label disambiguation. We also find that the coupling of pseudo label disambiguation and representation learning leads to slow convergence of the model (see Section 5.1).

To solve this problem, we propose a novel **D**ecoupled **P**rototype **L**earning framework (**DPL**) for generalized intent discovery, which aims to decouple pseudo label disambiguation and representation learning. Different from the previous E2E method, DPL consists of two complementary components: prototypical contrastive representation learning (PCL) to get good intent representations and prototype-based label disambiguation (PLD) to obtain high-quality pseudo labels, as shown in Fig 2(b). In our framework, PCL and PLD work together to realize the decoupling of pseudo label disambiguation and representation learning. Specifically, we firstly employ the output probability distribution of the joint classifier to align samples and corresponding prototypes and perform prototypical contrastive representation learning (Li et al., 2021; Wang et al., 2021; Cui et al., 2022). We aim to pull together similar samples to the same prototype and obtain discriminative intent representations. Secondly, based on the embeddings and class prototypes learned by PCL, we introduce a prototype-based label disambiguation, which gradually updates pseudo labels based on the class prototypes closest to the samples. Finally, we use these pseudo labels to train a joint classifier. We leave the details in the following Section 2. In addition, we theoretically explain that prototypical contrastive representation learning gets closely aligned representations for examples from the same classes and facilitates pseudo label disambiguation (Section 3). We also perform exhaustive experiments and qualitative analysis to demonstrate that our DPL framework can obtain more reliable pseudo labels and learn better representations in Section 5.

Our contributions are three-fold: (1) We propose a novel decoupled prototype learning (DPL) framework for generalized intent discovery to better decouple pseudo label disambiguation and representation learning. (2) We give a theoretical interpretation of prototypical contrastive representation learning to show that it gets better representations to help pseudo label disambiguation. (3) Experiments and analysis on three benchmark datasets demonstrate the effectiveness of our method for generalized intent discovery.

2 Approach

2.1 Problem Formulation

Given a set of labeled in-domain data $\mathbf{D}^{IND} = \{(x_i^{IND}, y_i^{IND})\}_{i=1}^{n}$ and unlabeled OOD data $\mathbf{D}^{OOD} = \{(x_i^{OOD})\}_{i=1}^{m}$, where $y_i^{IND} \in \mathcal{Y}^{IND}, \mathcal{Y}^{IND} = \{1, 2, \ldots, N\}$, GID aims to train a joint classifier to classify an input query to the total label set $\mathcal{Y} = \{1, \ldots, N, N+1, \ldots, N+M\}$ where the first N elements denote labeled IND

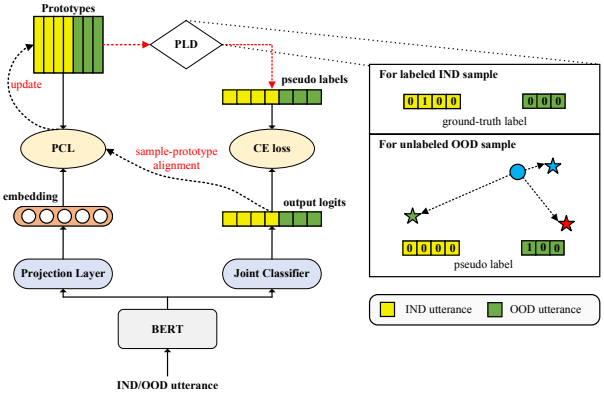

Figure 3: Overall architecture of our DPL method.

classes and the subsequent M ones denote newly discovered unlabeled OOD classes. For simplicity, we assume the number of OOD classes is specified as M. Since OOD training data is unlabeled, how to obtain accurate pseudo labels a key problem.

2.2 Overall Architecture

Fig 3 displays the overall architecture of our proposed decoupled prototype learning (DPL) framework for generalized intent discovery. We firstly get contextual features using BERT encoder (Devlin et al., 2019). To better leverage prior intent knowledge, we first pre-train the encoder on labeled IND data to get intent representations as E2E (Mou et al., 2022b). And then we add a joint classifer and a projection layer [2] on top of BERT. Given an input query, the projection layer maps the intent features of BERT encoder to a hypersphere, and uses prototypical contrastive representation learning (PCL) to further learn discriminative intent representations and class prototypes. Based on the representations and prototypes, we adopt a prototype-based label disambiguation (PLD) method to obtain pseudo labels, and use a cross-entropy(CE) objective to optimize the joint classifier. In the DPL framework, prototypical contrastive representation learning is not limitted by pseudo labels, and decouples pseudo label disambiguation and representation learning. We provide a pseudo-code of DPL in Appendix D.

2.3 Prototypical Contrastive Learning

Sample-prototype alignment We introduce prototypical contrastive representation learning (PCL) in our DPL framework. Firstly, we randomly initialize the L_2-normalized prototype embedding $\mu_j, j = 1, 2, ..., N + M$ of each intent category, which can be seen as a set of representative embedding vectors, and then for each input sample x_i, we need to align it with the corresponding class prototype. Specifically, if x_i belongs to IND intents, we use ground-truth label to align the sample with class prototype. If the input sample belongs to OOD intents, the output logit $l_i^{OOD} = (l_i^{N+1}, ..., l_i^{N+M})$ can be obtained by the joint classifier $f(x_i)$ [3], and we can use l_i^{OOD} to align the sample with class prototype. The alignment relationship is as follows:

$$\boldsymbol{q_i} = \begin{cases} [y_i^{IND}; \mathbf{0}_M] & x_i \in \mathbf{D}^{IND} \\ [\mathbf{0}_N; l_i^{OOD}] & x_i \in \mathbf{D}^{OOD} \end{cases} \quad (1)$$

where y_i^{IND} is a one-hot vector of ground-truth label, $\mathbf{0}_M, \mathbf{0}_N$ are M or N-dimention zero vectors and $q_i^j, j = 1, 2, ..., N + M$ represents the confidence probability that sample x_i belongs to prototype μ_j. After obtaining the alignment relationship between samples and prototypes, we get the L_2-normalized embedding z_i of sample x_i through the projection layer $g(x_i)$, and then perform prototypical contrastive learning as follows:

$$\mathcal{L}_{PCL} = -\sum_{i,j} q_i^j \log \frac{\exp\left(\text{sim}\left(z_i, \mu_j\right)/\tau\right)}{\sum_r \exp\left(\text{sim}\left(z_i, \mu_r\right)/\tau\right)} \quad (2)$$

where τ denotes temperature, and we set it to 0.5 in our experiments. PCL pulls together similar samples to the same prototype and obtain discriminative intent representations. Furthermore, we also add the instance-level contrastive loss to alleviate the problem of incorrect alignment between samples and prototypes caused by unreliable confidence probability at the beginning of training.

$$\mathcal{L}_{ins} = -\sum_{i} \log \frac{\exp\left(\text{sim}\left(z_i, \hat{z}_i\right)/\tau\right)}{\sum_k \mathbf{1}_{[k \neq i]} \exp\left(\text{sim}\left(z_i, z_k\right)/\tau\right)} \quad (3)$$

where $\hat{z}_i$ denotes the dropout-augmented view of z_i. Finally, we jointly optimize $\mathcal{L}_{PCL}$ and $\mathcal{L}_{ins}$ to learn cluster-friendly representation.

Update prototype embedding The class prototype embedding needs to be constantly updated during the training process. The naive way to update prototypes is to calculate the average value of embeddings for samples of the same class at each iteration. However, this will lead to a large amount of computing overhead, which will lead to unbearable training delays. Therefore, we update the prototype vector in a moving-average style:

$$\boldsymbol{\mu}_c = \text{Normalize}\left(\gamma \boldsymbol{\mu}_c + (1 - \gamma)\boldsymbol{z_i}\right) \quad (4)$$

[2] In the experiments, we use a two-layer non-linear MLP to implement the projection layer.

[3] Following (Mou et al., 2022b), we also adopt SK algorithm (Cuturi, 2013) to calibrate the output logits.

where the prototype μ_c of intent class c can be defined as the moving-average of normalized embeddings z_i, if the confidence of sample x_i belonging to category c is the largest. The moving average coefficient γ is a tunable hyperparameter.

2.4 Prototype-based Label Disambiguation

Prototypical contrastive learning gets discriminative intent representations, compact cluster distributions and class prototype embeddings that fall in the center of corresponding clusters. Next, we need to use the learned class prototypes for pseudo label disambiguation. Specifically, if an input sample x_i belongs to IND intents, we use ground-truth label directly, if an input sample belongs to OOD intents, the pseudo target assignment is to find the nearest prototype of the current embedding vector. The pseudo label is constructed as follows:

$$\boldsymbol{y}_i = \begin{cases} \left[y_i^{IND}; \mathbf{0}_M\right] & x_i \in \mathbf{D}^{IND} \\ \left[\mathbf{0}_N; \hat{\boldsymbol{p}}_i^{OOD}\right] & x_i \in \mathbf{D}^{OOD} \end{cases} \tag{5}$$

$$\hat{\boldsymbol{p}}_i^c = \begin{cases} 1 & \text{if } c = \arg\max_{j \in \mathcal{Y}^{OOD}} \boldsymbol{z_i}^\top \boldsymbol{\mu}_j \\ 0 & \text{else} \end{cases} \tag{6}$$

After obtaining pseudo labels, we use cross-entropy loss $\mathcal{L}_{CE}$ to optimize the joint classifier, and learn to classify labeled IND intents and the newly discovered unlabeled OOD intents.

3 Theoretical Analysis

In this section, we provide a theoretical explanation of why prototypical contrastive representation learning can learn cluster-friendly intent representations and class prototypes that facilitate pseudo label disambiguation. PCL essentially draws similar samples towards the same prototype, and forms compact clusters in the representation space, which is consistent with the goal of clustering, so we will explain it from the perspective of EM algorithm.

As defined in Section 2.1, we have n labeled IND samples and m unlabeled OOD samples. In the GID task, our goal is to find suitable network parameters to maximize the log-likelihood function as follows:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n+m} \log P\left(x_i \mid \theta\right) \tag{7}$$

E-step In the supervised learning setting, it is easy to estimate the likelihood probability using ground-truth labels. However, in the GID task, we not only have labeled IND samples, but also have a large number of unlabeled OOD samples, so we need to associate each sample with an implicit variable $j, j = 1, 2, ..., N + M$ (j represents the intent category). In addition, this likelihood function is hard to be directly optimized, so we need to introduce a probability density function $q_i(j)$ to represent the probability that sample x_i belongs to intent category j. Finally, we can use Jensen's inequality to derive the lower bound of the maximum likelihood function as follows (We leave detailed derivation process in appendix C):

$$\begin{aligned} \theta^* &= \operatorname*{argmax}_\theta \sum_{i=1}^{n+m} \log P\left(x_i \mid \theta\right) \\ &\geqslant \arg\max_\theta \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log \frac{P\left(x_i, j \mid \theta\right)}{q_i(j)} \end{aligned} \tag{8}$$

Since $\log(\cdot)$ is a concave function, the inequality holds with equality when $\frac{P(x_i, j|\theta)}{q_i(j)}$ is constant. Thus we can derive $q_i(j)$ as follows:

$$q_i(j) = \frac{P\left(x_i, j \mid \theta\right)}{\sum_{j \in y_{all}} P\left(x_i, j \mid \theta\right)} = P\left(j \mid x_i, \theta\right) \tag{9}$$

We can know that when $q_i(j)$ is a posterior class probability, maximizing the lower bound of the likelihood function is equivalent to maximizing the likelihood function itself. In our GID task, there are both labeled IND data and unlabeled OOD data. Therefore, for labeled IND data, we can directly use ground-truth label to estimate the posterior class probability. For unlabeled OOD data, we can estimate the posterior probability distribution by the joint classifier. This provides theoretical support for the sample-prototype alignment in PCL.

M-step We have estimated $q_i(j)$ in E-step. Next, we need to maximize the likelihood function and find the optimal network parameters under the assumption that $q_i(j)$ is known. The optimization objective is as follows (We leave detailed derivation process in appendix C):

$$\begin{aligned} L(\theta) &= \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log \frac{P\left(x_i, j \mid \theta\right)}{q_i(j)} \\ &\approx \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log P\left(x_i \mid j, \theta\right) \\ &\approx \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_{i(j)} \log \frac{\exp\left(\frac{\boldsymbol{z_i} \cdot \boldsymbol{\mu_j}}{\sigma_j^2}\right)}{\sum_{r \in y_{all}} \exp\left(\frac{\boldsymbol{z_i} \cdot \boldsymbol{\mu_r}}{\sigma_r^2}\right)} \\ &\Leftrightarrow \min \mathcal{L}_{PCL} \end{aligned} \tag{10}$$

Method	GID-SD					GID-CD					GID-MD				
	IND	OOD		ALL		IND	OOD		ALL		IND	OOD		ALL	
	ACC	ACC	F1	ACC	F1	ACC	ACC	F1	ACC	F1	ACC	ACC	F1	ACC	F1
k-means	90.38	62.34	62.44	78.99	78.32	97.70	61.67	60.43	83.20	82.30	97.26	73.00	72.66	87.56	87.08
DeepAligned	91.72	69.11	69.72	82.57	82.10	97.85	78.55	77.81	90.12	89.68	97.85	87.55	87.14	93.70	93.29
DeepAligned-Mix	82.30	54.97	59.79	71.30	69.60	97.33	72.41	71.54	87.36	86.21	92.86	81.70	83.30	88.12	87.42
End-to-End	92.84	72.28	73.28	84.49	84.10	98.00	79.19	79.06	90.46	90.28	98.32	91.92	92.46	95.78	95.73
DPL(ours)	92.89	**74.38**	**75.46**	**85.43**	**85.34**	98.37	**82.40**	**82.37**	**91.98**	**91.85**	98.29	**92.84**	**93.00**	**96.11**	**95.96**

Table 1: Performance comparison on three benchmark datasets. Results are averaged over three random run.($p < 0.01$ under t-test). Here we report the results of 40% OOD ratio. For experimental results of more OOD ratios, we have made further discussion in Section 5.6.

where $P(x_i \mid j, \theta)$ represents the data distribution of the class j in the representation space. We think that the larger the likelihood probability, the more reliable the pseudo labels. We assume that the class j follows a gaussian distribution in the representation space, and can derive that minimizing the PCL objective is equivalent to maximizing the likelihood function, which explains why the prototypical contrastive representation learning facilitates pseudo label disambiguation.

4 Experiments

4.1 Datasets

We conducted experiments on three benchmark datasets constructed by (Mou et al., 2022b), GID-SD(single domain), GID-CD(cross domain) and GID-MD(multiple domain). GID-SD randomly selects intents as the OOD type from the single-domain dataset Banking (Casanueva et al., 2020), which contains 77 intents in banking domain, and the rest as the IND type. GID-CD restricts IND and OOD intents from non-overlapping domains from the multi-domain dataset CLINC (Larson et al., 2019), which covers 150 intents in 10 domains, while GID-MD ignores domain constraints and randomizes all CLINC classes into IND sets and OOD sets. To avoid randomness, we average the results in three random runs. We leave the detailed statistical information of datasets to Appendix A.

4.2 Baselines

Similar with (Mou et al., 2022b), we extensively compare our method with the following GID baselines: k-means (MacQueen, 1967), DeepAligned (Zhang et al., 2021a), DeepAligned-Mix (Mou et al., 2022b), End-to-End (E2E) (Mou et al., 2022b), in which E2E is the current state-of-the-art method for GID task. For a fair comparison, all baselines use the same BERT encoder as the backbone network. We leave the details of the baselines in Appendix B. We adopt two widely used metrics to evaluate the performance of the joint classifier: Accuracy(ACC) and F1-score(F1), in which ACC is calculated over IND, OOD and total(ALL) classes respectively and F1 is calculated over OOD and all classes to better evaluate the ability of methods to discover and incrementally extend OOD intents. OOD and ALL ACC/F1 are the main metrics.

4.3 Implementation Details

For a fair comparison of the various methods, we use the pre-trained BERT model (bert-base-uncased [4], with 12-layer transformer) as our network backbone, and add a pooling layer to get intent representation(dimension=768). Moreover, we freeze all but the last transformer layer parameters to achieve better performance with BERT backbone and speed up the training procedure as suggested in (Zhang et al., 2021a).

The class prototype embedding(dimension=128) is obtained by the representation through a linear projection layer. For training, we use SGD with momentum as the optimizer, with linear warm-up and cosine annealing (lr_{min} = 0.01; for GID-SD, lr_{base} = 0.02, for GID-CD and GID-MD,lr_{base} = 0.1), and weight decay is 1.5e-4.The moving average coefficient γ=0.9.We train 100 epochs and use the Silhouette Coefficient(SC) of OOD data in the validation set to select the best checkpoints. Notably, We use dropout to construct augmented examples and the dropout value is fixed at 0.1.

The average value of the trainable model parameters is 9.1M and the total parameters are 110M which is basically the same as E2E. In the training stage, the decoupling-related components of DPL bring approximately 8% additional training load compared to E2E. In the inference stage, DPL only requires the classifier branch, without additional computational overhead. It can be seen that our DPL method has significantly improved perfor-

[4]https://github.com/google-research/bert

mance compared to E2E, but the cost of time and space complexity is not large. All experiments use a single Nvidia RTX 3090 GPU(24 GB of memory).

4.4 Main Results

Table 1 shows the performance comparison of different methods on three benchmark GID datasets. In general, our DPL method consistently outperforms all the previous baselines with a large margin in various scenarios. Next, we analyze the results from three aspects:

(1) **Comparison of different methods.** We can see that our proposed DPL method is better than all baselines. For example, DPL is superior to E2E by 2.1% (OOD ACC), 2.18% (OOD F1) and 1.24% (ALL F1) on GID-SD dataset, 3.21% (OOD ACC), 3.31% (OOD F1) and 1.57% (ALL F1) on GID-CD dataset, 0.92% (OOD ACC), 0.54% (OOD F1) and 0.23% (ALL F1) on GID-MD dataset. This shows that DPL framework decouples pseudo label disambiguation and representation learning, which makes the pseudo labels and representation learning no longer restrict each other, and effectively improves the reliability of pseudo labels (We give a detailed analysis in section 5.1). Accurate pseudo labels further improve the classification performance of the joint classifier, especially the ability to discover and recognize new OOD intent categories.

(2) **Single-domain scenario** Since IND and OOD intents belong to the same domain in GID-SD, and the difference between intents is smaller than that of multiple-domain dataset GID-MD, so it is more difficult to form clear cluster boundaries, and the performance of joint classifier is relatively low. Interestingly, we observed that the improvement of DPL method in single-domain dataset GID-SD is more significant than that in multiple-domain dataset GID-MD. For example, In GID-MD, DPL only increased by 0.54% (OOD F1) compared with E2E, while in the more challenging GID-SD dataset, it increased by 2.18% (OOD F1). We believe that it is because prototypical contrastive representation learning can draw similar samples to the same prototype to learn cluster-friendly representation, which helps to form a clearer cluster boundary for each intents and improve the accuracy of pseudo labels. We leave more detailed analysis in Section 5.2.

(3) **Cross-domain scenario** Since IND and OOD intents come from different domains in GID-CD, which means that it is more difficult to transfer the prior knowledge of labeled IND intents to help pseudo labeling of unlabeled OOD data. This can be seen from the small improvement (0.64% OOD ACC) of E2E compared with DeepAligned. However, we find that our DPL method increased by 3.31% (OOD F1) and 1.57% (ALL F1) on GID-CD, which is far higher than the previous improvement. We believe that this may be due to the use of prototypical contrastive representation learning to learn the class prototypes of IND and OOD intents at the same time, which more effectively make use of the prior knowledge of labeled IND intents to help the representation learning and obtain more accurate pseudo labels.

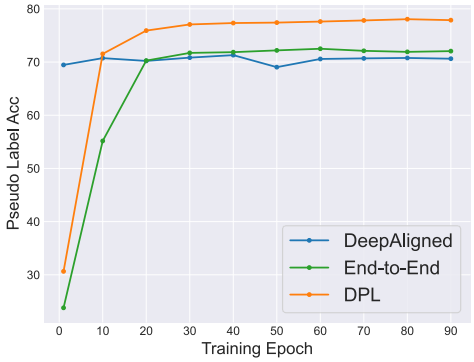

Figure 4: Pseudo label accuracy curves in the training process.

5 Qualitative Analysis

5.1 Pseudo Label Disambiguation

One of the key challenges of generalized intent discovery is pseudo label disambiguation. We compared the pseudo labels accuracy of different methods, as shown in Fig 4. Firstly, we can see that the end-to-end framework (DPL and E2E) has a higher upper bound of pseudo label accuracy than the pipeline framework (DeepAligned). We think that it is because the end-to-end framework fully considers the knowledge interaction between labeled IND intents and unlabeled OOD data in the training process. Next, we analyze the advantages of DPL over E2E in pseudo label disambiguation from two perspectives: (1) Our DPL method converges faster than E2E method. We think that the E2E method converges slower because pseudo label disambiguation and representation learning are coupled. Inaccurate pseudo labels limit the representation learning, while poor intent representation hinders pseudo label disambiguation. In contrast,

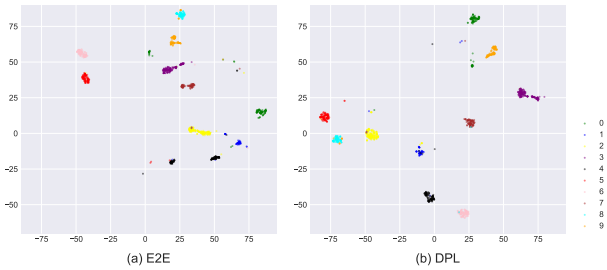

Figure 5: Visualization of different methods.

	IND intents ↑	OOD intents ↑	ALL intents ↑
E2E	5.25	2.55	3.81
DPL(ours)	**5.78**	**2.71**	**4.09**

Table 2: Cluster compactness of different methods.

our DPL method decouples the pseudo-label disambiguation and representation learning, which makes the pseudo labels and intent representation no longer restrict each other and accelerates the convergence. (2) Compared with E2E method, our DPL method can obtain more accurate pseudo labels in the training process, and reach higher upper bound. We believe that there are two reasons for this. First, DPL framework decouples pseudo label disambiguation and representation learning. The quality of pseudo labels will not limit representation learning, so it can obtain more discriminative representation, thus improving the accuracy of pseudo labels. Besides, we use prototype-based contrastive learning for representation learning, which aligns with the subsequent prototype-based label disambiguation.

5.2 Representation Learning

A cluster-friendly intent representation is very important for the pseudo label disambiguation of the generalized intent discovery task. PCL can get closely aligned cluster distribution for similar samples, which is beneficial for prototype-based label disambiguation. Firstly, we quantitatively compare the cluster compactness learned by DPL and E2E. We calculate the intra-class and inter-class distances following Feng et al. (2021). For the intra-class distance, we calculate the mean value of the euclidean distance between each sample and its class center. For the inter-class distance, we calculate the mean value of the euclidean distance between the center of each class and the center of other classes. We report the ratio of inter-class and intra-class distance in Table 2. The higher the value, the clearer the boundary between different intent

Models	OOD ACC	OOD F1	ALL F1
E2E	72.28	73.28	84.10
DPL($\mathcal{L}_{PCL} + \mathcal{L}_{ins}$)	**74.38**	**75.46**	**85.34**
-w/o $\mathcal{L}_{ins}$	73.28	74.25	84.51
-w/o $\mathcal{L}_{PCL}$	73.97	74.16	84.27
$\mathcal{L}_{SCL}$	71.15	70.47	83.10
$\mathcal{L}_{PCL} + \mathcal{L}_{SCL}$	71.39	72.16	83.86

Table 3: Ablation study of different representation learning objective for DPL.

categories. The results show that PCL learns better intent representation, which explains why the DPL method can obtain more accurate pseudo labels. In order to more intuitively analyze the effect of PCL in representation learning, we perform intent visualization of E2E and DPL methods, as shown in Fig 5. We can see that the DPL framework adopts PCL for representation learning, which can obtain compact cluster (see "black" and "blue" points). In addition, we can observe that clusters learned by E2E method are concentrated in the upper right part, while DPL can obtain are more evenly distributed clusters. To see the evolution of our DPL method in the training process, we show a visualization at four different timestamps in Fig 6. We can see that samples of different intents are mixed in the representation space at the begining, and cannot form compact clusters. As the training process goes, the boundary of different intent clusters becomes clearer and the learned class prototypes gradually falls in the center of the corresponding intent cluster.

5.3 Ablation Study

To understand the effect of different contrastive learning objectives on our DPL framework, we perform ablation study in Table 3. In our DPL framework, we jointly optimized $\mathcal{L}_{PCL}$ and $\mathcal{L}_{ins}$ to achieve the best performance. Then we remove $\mathcal{L}_{PCL}$ and $\mathcal{L}_{ins}$ respectively, and find that compared with the joint optimization, the performance drop to a certain extent, but both are better than the baseline. This shows that both prototypical contrastive learning and instance-level contrastive learning can learn discriminative intent representations and facilitate pseudo label disambiguation.

In addition, we also explored the adaptability of the commonly used supervised contrastive learning (SCL) in the DPL framework. We find that the performance of $\mathcal{L}_{SCL}$ is significantly lower than that of $\mathcal{L}_{PCL}$ and $\mathcal{L}_{ins}$. We argue that this is because SCL draws similar samples closer and pushes

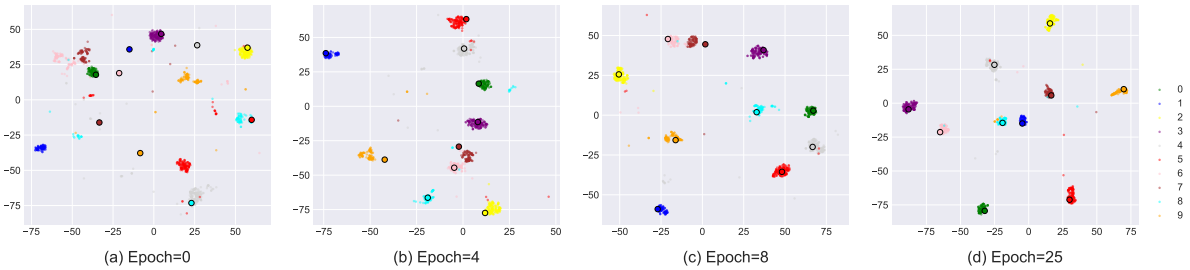

Figure 6: intent and prototypes visualization of different training epochs for our proposed DPL method.

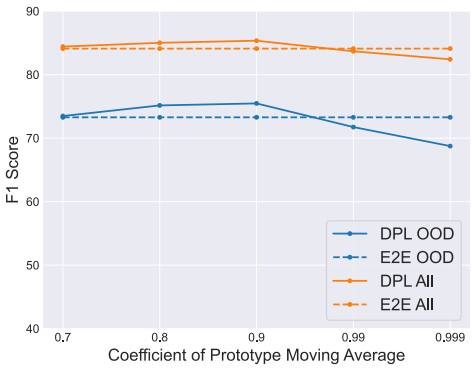

Figure 7: Effect of the prototype moving average coefficient γ on our DPL method. We conduct experiments on GID-SD dataset.

	OOD ACC	OOD F1	ALL F1	K
DeepAligned	69.11	69.72	82.10	31
End-to-End	72.28	73.28	84.10	31
DPL(ours)	**74.38**	**75.46**	**85.34**	31
DeepAligned	62.50	59.74	77.39	26
End-to-End	66.29	61.55	78.57	26
DPL(ours)	**70.81**	**67.57**	**81.17**	26

Table 4: Estimate the number of OOD classes. We take GID-SD as an example. K=26 is the estimated number compared to ground-truth number 31.

apart dissimilar samples, but it lacks the interaction between samples and class prototypes in the training process, and there is gap with the subsequent prototype-based label disambiguation.

5.4 Effect of Moving Average Coefficient

Fig 7 shows the effect of different prototype moving average coefficient γ on our DPL method. Results show that $\gamma = 0.9$ gets the best performance on GID-SD. Our DPL method with γ in (0.7, 0.95) outperforms SOTA baseline, which proves DPL method is robust for different γ. In addition, we observe that when γ is greater than 0.9, the performance of DPL decreases significantly. We argue that this is because a large γ will slow the speed of prototypes moving to the center of the corresponding clusters, resulting in getting poor prototypes, which hinders pseudo label disambiguation.

5.5 Estimate the Number of OOD intents

In standard GID setting, we assume that the number of OOD classes is ground-truth. However, in the real applications, the number of OOD clusters often needs to be estimated automatically. We use the same OOD cluster number estimation strategy as Zhang et al. (2021a); Mou et al. (2022b). The results are showed in Table 4. It can be seen that when the number of OOD clusters is inaccurate, all methods have a certain decline, but our DPL method still significantly outperforms all baselines, and even the improvement is more obvious, which also proves that DPL is more robust and practical.

5.6 Effect of different OOD ratios

In Fig 8, we compare the effect of different OOD ratios on various methods. The larger the OOD ratio means the fewer the IND categories and the more the OOD categories. On the one hand, it reduces the available prior knowledge of IND intents, and on the other hand, it is more difficult to distinguish the unlabeled OOD intents. The experimental results show that the performance of all methods decrease significantly as the OOD ratio increases. However, we find that when the OOD ratio is large, our DPL method has a more obvious improvement compared with other baselines, which shows that our method is more robust to different OOD ratios, and DPL decouples pseudo label disambiguation and representation learning, which can more effectively use the prior knowledge of IND intents and learn the discriminative intent representations, which improves the reliability of pseudo labels.

5.7 Effect of imbalanced OOD data

As mentioned in the previous work (Mou et al., 2022b), E2E introduces a method based on optimal transport (Cuturi, 2013; Caron et al., 2020; Fini et al., 2021) to calibrate the output logits of the joint

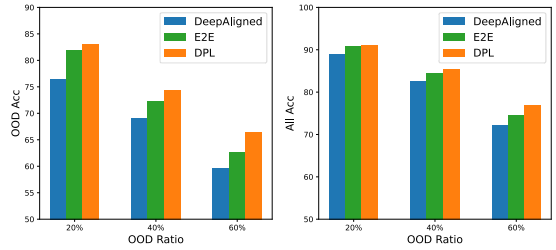

Figure 8: The effect of different OOD ratios on the performance of each GID method.

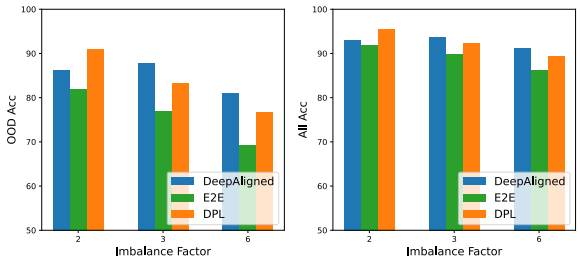

Figure 9: The effect of different imbalance factors of OOD data on the performance of each GID method.

classifier before swapped prediction. This assumes that the unlabeled OOD samples in each batch are evenly distributed to M OOD categories. However, it is hard to ensure that the unlabeled OOD data in the real scene is class-balanced, and sometimes even the long-tailed distribution. The experimental results in Fig 9 show that the E2E method has a significant performance degradation in the case of OOD class-imbalanced. In contrast, our proposed DPL framework adopts a prototype-based label disambiguation method, and it doesn't rely on the assumption of class-distribution assumption. Therefore, it is significantly better than E2E in the OOD class-imbalanced scenario.

However, we also observed that when the imbalance factor increased, the performance of our DPL method decline more significantly compared with DeepAligned. We think that this is because DPL method needs to use unlabeled OOD data to learn the discriminative representations and class prototypes. When the imbalance factor increases, the number of samples for OOD intent categories will become smaller, which is not conducive to learning the cluster-friendly representations and class prototypes. We can alleviate this problem by increasing the number of samples through data augmentation. We will leave this to future work.

6 Related Work

Generalized Intent Discovery Existing intent classification models have little to offer in an open-world setting, in which many new intent categories are not pre-defined and no labeled data is available. These models can only recognize limited in-domain (IND) intent categories. Lin and Xu (2019); Xu et al. (2020) propose the OOD intent detection task to identify whether a user query falls outside the range of a pre-defined intent set. Further, OOD intent discovery task (also known as new intent discovery) (Lin et al., 2020; Zhang et al., 2021a) is proposed to cluster unlabeled OOD data. Mou et al. (2022b) proposes the Generalized Intent Discovery (GID) task, which aims to simultaneously classify a set of labeled IND intents while discovering and recognizing new unlabeled OOD types incrementally.

Prototype-based Learning Prototype-based metric learning methods have been promising approaches in many applications. Snell et al. (2017) first proposes Prototypical Networks (ProtoNet) which introduces prototypes into deep learning. Specifically, ProtoNet calculates prototype vectors by taking the average of instance vectors and makes predictions by metric-based comparisons between prototypes and query instances. Li et al. (2020b) proposes self-supervised prototype representation learning by using prototypes as latent variables. Learning good representations also helps weakly supervised learning tasks, including noisy label learning (Li et al., 2020a), semi-supervised learning (Zhang et al., 2021b), partial label learning (Wang et al., 2022), etc. Inspired by these methods, we propose a decoupled prototype learning framework (DPL) to decouple pseudo label disambiguation and representation learning for GID.

7 Conclusion

In this paper, we propose a decoupled prototype learning (DPL) framework for generalized intent discovery. We introduce prototypical contrastive representation learning and prototype-based label disambiguation method to decouple representation learning and pseudo label disambiguation. Theoretical analysis and extensive experiments prove that our method can learn discriminative intent representations and prototypes, which facilitates pseudo label disambiguation. We will explore broader applications of DPL method in the future.

Limitations

This paper mainly focuses on the generalized intent discovery (GID) task in task-oriented dialogue systems. Our proposed Decoupled Prototype Learning (DPL) framework well decouple pseudo label disambiguation and representation learning through protopical contrastive learning and prototype-based label disambiguation, and achieves SOTA performance on three GID benchmark datasets. However, our work also have several limitations: (1) We only verified the effectiveness of our DPL framework on GID task, but the adaptability of DPL in more unsupervised / semi-supervised settings, such as unsupervised clustering and OOD intent discovery, is worth further exploration. (2) We follow standard experiment settings as previous work, and assume that each OOD sample must belong to a corresponding intent cluster. However, a more realistic scenario is that there may be noise samples in the OOD data. These noise samples do not actually belong to any cluster/category and are some outliers. We leave the noise OOD issue to the future work. (3) Our experiments in Appendix 6 find that the performance of the DPL method decreases significantly when the imbalance factor of unlabeled OOD data increases. How to improve the performance of GID model on the long tailed unlabeled data is also a problem worthy of attention in the future.

Acknowledgements

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

References

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024, Dublin, Ireland. Association for Computational Linguistics.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2292–2300.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yutong Feng, Jianwen Jiang, Mingqian Tang, Rong Jin, and Yue Gao. 2021. Rethinking supervised pretraining for better downstream transferring. *arXiv preprint arXiv:2110.06014*.

Enrico Fini, E. Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. 2021. A unified objective for novel class discovery. *ArXiv preprint*, abs/2108.08536.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. 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)*.

Junnan Li, Caiming Xiong, and Steven C. H. Hoi. 2020a. Mopro: Webly supervised learning with momentum prototypes. *ArXiv*, abs/2009.07995.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. 2021. Prototypical contrastive learning of unsupervised representations. *ICLR*.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C. H. Hoi. 2020b. Prototypical contrastive learning of unsupervised representations. *ArXiv*, abs/2005.04966.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5496, Florence, Italy. Association for Computational Linguistics.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *AAAI*.

J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations.

Yutao Mou, Keqing He, Pei Wang, Yanan Wu, Jingang Wang, Wei Wu, and Weiran Xu. 2022a. Watch the neighbors: A unified k-nearest neighbor contrastive learning framework for ood intent discovery. *arXiv preprint arXiv:2210.08909*.

Yutao Mou, Keqing He, Yanan Wu, Pei Wang, Jingang Wang, Wei Wu, Yi Huang, Junlan Feng, and Weiran Xu. 2022b. Generalized intent discovery: Learning from open world dialogue system. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 707–720, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yutao Mou, Keqing He, Yanan Wu, Zhiyuan Zeng, Hong Xu, Huixing Jiang, Wei Wu, and Weiran Xu. 2022c. Disentangled knowledge transfer for OOD intent discovery with unified contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 46–53, Dublin, Ireland. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. *ArXiv*, abs/1703.05175.

Haobo Wang, Rui Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Jake Zhao. 2022. Pico: Contrastive label disambiguation for partial label learning. *ArXiv*, abs/2201.08984.

Liwen Wang, Xuefeng Li, Jiachi Liu, Keqing He, Yuanmeng Yan, and Weiran Xu. 2021. Bridge to target domain by prototypical contrastive learning and label confusion: Re-explore zero-shot learning for slot filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9474–9480, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanan Wu, Keqing He, Yuanmeng Yan, QiXiang Gao, Zhiyuan Zeng, Fujia Zheng, Lulu Zhao, Huixing Jiang, Wei Wu, and Weiran Xu. 2022a. Revisit overconfidence for OOD detection: Reassigned contrastive learning with adaptive class-dependent threshold. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4165–4179, Seattle, United States. Association for Computational Linguistics.

Yanan Wu, Zhiyuan Zeng, Keqing He, Yutao Mou, Pei Wang, Yuanmeng Yan, and Weiran Xu. 2022b. Disentangling confidence score distribution for out-of-domain intent detection with energy-based learning. *ArXiv*, abs/2210.08830.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1452–1460, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2021. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 870–878, Online. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lv. 2021a. Discovering new intents with deep aligned clustering. In *AAAI*.

Yuhang Zhang, Xiaopeng Zhang, Robert Caiming Qiu, Jie Li, Haohang Xu, and Qi Tian. 2021b. Semi-supervised contrastive learning with similarity co-calibration. *ArXiv*, abs/2105.07387.

Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2022. New intent discovery with pre-training and contrastive learning. *ArXiv*, abs/2205.12914.

A Datasets

Table 5 shows the statistics of the original datasets Banking and CLINC, where each class in CLINC has the same number of samples but Banking is class-imbalanced. For the three GID datasets GID-SD, GID-CD and GID-MD, We show the detailed statistics in Table 6. Due to Banking is class-imbalanced and we conducted three random partitions, we report the average of the sample number of GID-SD.

Dataset	Classes	Training	Validation	Test	Vocabulary	Length (max / mean)
Banking	77	9003	1000	3080	5028	79/11.91
CLINC	150	18000	2250	2250	7283	28/8.31

Table 5: Statistics of Banking and CLINC datasets.

Dataset	IND/OOD Classes	IND/OOD Domains	IND/OOD Training	IND/OOD Validation	IND/OOD Test
GID-SD	46/31	1/1	5346/3657	593/407	1840/1240
GID-CD	90/60	6/4	7200/4800	1350/900	1350/900
GID-MD	90/60	10/10	7200/4800	1350/900	1350/900

Table 6: Statistics of GID-SD, GID-CD and GID-MD datasets.

B Baselines

The details of baselines are as follows:

k-means is a pipeline method which first uses kmeans (MacQueen, 1967) to cluster OOD data and obtains pseudo OOD labels, and then trains a new classifier together with IND data.

DeepAligned is similar to k-means, the difference is that the clustering algorithm adopts DeepAligned (Zhang et al., 2021a), which uses an alignment strategy to tackle the label inconsistency problem during clustering assignments.

DeepAligned-Mix (Mou et al., 2022b) is an extended method from DeepAligned for GID task. In each iteration, it firstly mix up IND and OOD data together for clustering using k-means and an alignment strategy and then uses a unified cross-entropy loss to optimize the model. In the inference stage, instead of using k-means for clustering, DeepAligned-Mix use the classification head of the new classifier to make predictions.

E2E (Mou et al., 2022b) mixes IND and OOD data in the training process and simultaneously learns pseudo OOD cluster as signments and classifies all classes via self-labeling. Given an input query, E2E connects the encoder output through two independent projection layers, IND head and OOD head, as the final logit and optimize the model through the unified classification loss, where the OOD pseudo label is obtained through swapped prediction(Caron et al., 2020).

C Details of derivation process

C.1 Derivation process of equation 8

In the GID task, likelihood function is hard to be directly optimized, so we need to introduce a probability density function $q_i(j)$ to represent the probability that sample x_i belongs to intent j. The detailed derivation process is as follows:

$$\begin{aligned}
\theta^* &= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n+m} \log P\left(x_i \mid \theta\right) \\
&= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n+m} \log \sum_{j \in y_{\text{all}}} P\left(x_i, j \mid \theta\right) \\
&= \arg\max \sum_{i=1}^{n+m} \log \sum_{j \in y_{all}} q_i(j) \frac{P\left(x_i, j \mid \theta\right)}{q_i(j)} \\
&\geqslant \arg\max_{\theta} \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log \frac{P\left(x_i, j \mid \theta\right)}{q_i(j)}
\end{aligned} \tag{11}$$

C.2 Derivation process of equation 10

$$\begin{aligned}
L(\theta) &= \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log \frac{P\left(x_i, j \mid \theta\right)}{q_i(j)} \\
&= \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log P\left(x_i, j \mid \theta\right) \\
&\approx \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log P\left(x_i \mid j, \theta\right) \\
&= \max \sum_{i,j} q_i(j) \log \frac{\exp\left(\frac{-(z_i-\mu_j)^2}{2\sigma_j^2}\right)}{\sum_{r \in y_{all}} \exp\left(\frac{-(z_i-\mu_r)^2}{2\sigma_r^2}\right)} \\
&\approx \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_i(j) \log \frac{\exp\left(\frac{2z_i \cdot \mu_j}{2\sigma_j^2}\right)}{\sum_{r \in y_{all}} \exp\left(\frac{2z_i \cdot \mu_r}{2\sigma_r^2}\right)} \\
&\approx \max \sum_{i=1}^{n+m} \sum_{j \in y_{all}} q_{i(j)} \log \frac{\exp\left(\frac{\boldsymbol{z_i} \cdot \boldsymbol{\mu_j}}{\sigma_j^2}\right)}{\sum_{r \in y_{all}} \exp\left(\frac{\boldsymbol{z_i} \cdot \boldsymbol{\mu_r}}{\sigma_r^2}\right)} \\
&\Leftrightarrow \min \mathcal{L}_{PCL}
\end{aligned} \tag{12}$$

Algorithm 1 : Decoupled Prototype Learning

Input: training dataset $\mathbf{D}^{IND} = \left\{\left(x_i^{IND}, y_i^{IND}\right)\right\}_{i=1}^{n}$ and $\mathbf{D}^{OOD} = \left\{\left(x_i^{OOD}\right)\right\}_{i=1}^{m}$, IND label set $\mathcal{Y}^{IND} = \{1, 2, \ldots, N\}$, ground-truth number of OOD intents M, training epoch E, batch size B

Output: a new intent classification model, which can classify an input query to the total label set $\mathcal{Y} = \{1, \ldots, N, N+1, \ldots, N+M\}$.

randomly initialize the L_2-normalized prototype embedding $\mu_j, j = 1, 2, ..., N+M$.
for epoch = 1 to E **do**
    mix $\mathbf{D}^{IND}$ and $\mathbf{D}^{OOD}$ to get $\mathbf{D}^{ALL}$
    for $iter$ = 0, 1, 2, ... **do**
        sample a mini-batch **B** from $\mathbf{D}^{ALL}$
        get the L_2-normalized embedding z_i of sample x_i through the projection layer
        align sample x_i with prototypes μ_j by equation 1
        compute $\mathcal{L}_{PCL}$ and $\mathcal{L}_{ins}$ ▷ **prototypical contrastive representation learning**
        estimate the pseudo label $\boldsymbol{y}_i$ by equation 5 and 6 ▷ **pseudo label disambiguation**
        compute $\mathcal{L}_{CE}$ on the joint classifier
        add $\mathcal{L}_{PCL}$, $\mathcal{L}_{ins}$ and $\mathcal{L}_{CE}$ together, and jointly optimize them
        update the prototype vector by equation 4
    end for
end for

D Algorithm

We summarize the pseudo-code of our DPL method in Algorithm 1.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
We discuss the limitations of our work in Limitations Section.

☑ A2. Did you discuss any potential risks of your work?
We discuss them in Limitations Section.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
We summarize the paper's main claims in Line 6-14 in abstract, and Line 93-106 in introduction.

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

B ☑ Did you use or create scientific artifacts?

In section 4 and Appendix A/B, we cited all baselines and datasets we use in this paper.

☑ B1. Did you cite the creators of artifacts you used?
In section 4 and Appendix A/B, we cited all baselines and datasets we use in this paper.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4.1 and 4.2 Appendix A and B

☑ 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)?
Section 4.1 and 4.2 Appendix A and B

☑ 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?
Section 4.1 and 4.2 Appendix A and B

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1 and 4.2 Appendix A and B

☑ 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 and 4.2 Appendix A and B

C ☑ Did you run computational experiments?

Section 4.3, Section 5 Appendix C and D

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

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?
Appendix C, Section 5.4

☑ 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?
Section 4.3, Appendix C

☑ 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.)?
Appendix C

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.