Generalized Intent Discovery: Learning from Open World Dialogue System

Yutao Mou[1]*, Keqing He[2]*, Yanan Wu[1], Pei Wang[1], Jingang Wang[2]
Wei Wu[2], Yi Huang[3], Junlan Feng[3], Weiran Xu[1]*
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group, Beijing, China
[3]China Mobile Research Institute, Beijing, China
{myt,yanan.wu,wangpei,xuweiran}@bupt.edu.cn
{hekeqing,wangjingang,wuwei}@meituan.com
{huangyi,fengjunlan}@chinamobile.com

Abstract

Traditional intent classification models are based on a pre-defined intent set and only recognize limited in-domain (IND) intent classes. But users may input out-of-domain (OOD) queries in a practical dialogue system. Such OOD queries can provide directions for future improvement. In this paper, we define a new task, Generalized Intent Discovery (GID), which aims to extend an IND intent classifier to an open-world intent set including IND and OOD intents. We hope to simultaneously classify a set of labeled IND intent classes while discovering and recognizing new unlabeled OOD types incrementally. We construct three public datasets for different application scenarios and propose two kinds of frameworks, pipeline-based and end-to-end for future work. Further, We conduct exhaustive experiments and qualitative analysis to comprehend key challenges and provide new guidance for future GID research. [1]

1 Introduction

Intent classification (IC) in a dialogue system aims to identify the goal of a user query, such as *BookFlight* or *AddToPlaylist*. Recent neural-based models (Liu and Lane, 2016; Goo et al., 2018; E et al., 2019; Chen et al., 2019; He et al., 2020) have achieved satisfying performance under the availability of large-scale labeled data. However, these methods face the challenge of data scarcity and poor scalability. They rely on a pre-defined intent set and supervised labels, which is limitted in some practical scenarios.

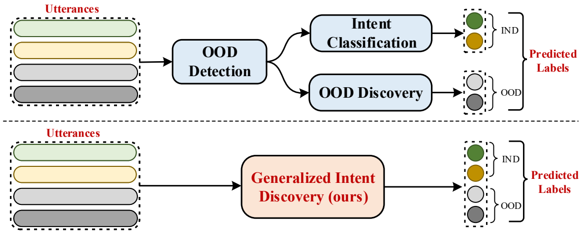


Figure 1: Illustration of our proposed GID task. The above subfig shows a practical intent classification system where an OOD detection module firstly identifies whether a test intent belongs to OOD, then an in-domain classifier and an OOD discoverer respectively recognize IND and OOD intents. In contrast, our proposed GID can simultaneously classify a set of labeled IND intent classes and new OOD types in an end-to-end manner.

Existing intent classification models have little to offer in an open-world setting, in which many new intent categories are not defined apriori and no labeled data is available. These models rely on the pre-defined intent set, making it only recognize limited in-domain (IND) intent categories. But plenty of input queries may be outside of the fixed intent set, which we call Out-of-Domain (OOD) intents (Xu et al., 2020; Zeng et al., 2021a,b). In recent years, OOD intent detection (Hendrycks and Gimpel, 2017; Larson et al., 2019a; Lin and Xu, 2019; Ren et al., 2019; Xu et al., 2020; Zheng et al., 2020) has been well studied, which identifies whether a user query falls outside the range of pre-defined intent set to avoid performing wrong operations. But it can only safely reject OOD intents thus ignore these valuable OOD concepts for future development. Further, OOD intent discovery task (also known as new intent discovery) (Lin et al., 2020; Zhang et al., 2021b) is proposed to cluster unlabeled OOD data. The adopted clustering method can only group those OOD intents into clusters, but cannot further expand the recognition scope of the existing IND intent classifier incrementally.

Inspired by the above issues, we introduce a new task of extending and recognizing intent categories automatically, **G**eneralized **I**ntent **D**iscovery(**GID**). GID aims to extend an existing IND intent classifier to an open-world OOD intent set, as shown in Fig 1. The main motivation is that we hope to

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

[1] We release our code at https://github.com/myt517/GID_benchmark.

Proceedings of the 29th International Conference on Computational Linguistics, pages 707–720
October 12–17, 2022.

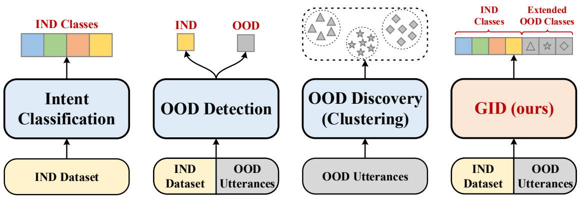


Figure 2: The comparison of GID to other related tasks.

train a network that can simultaneously classify a set of labeled IND intent classes while discovering new ones in an unlabeled OOD set. In this way, we can enhance the capability of an IC system by expanding its recognition scope incrementally. We show a comparison of GID and existing OOD tasks in Fig 2. Since the practical OOD intents are unsupervised, neither the OOD labels nor OOD intent schema make it different from zero-shot learning (Yan et al., 2020; Siddique et al., 2021) and continual learning (Xu et al., 2019) which both rely on a given label ontology, like label descriptions. Therefore, to explore unique characteristics of GID, we construct three kinds of GID benchmarks, including single domain, multiple domain, and cross-domain settings (Section 3). These settings denote different application scenarios which we will discuss later.

Subsequently, we propose two kinds of frameworks for GID, pipeline and end-to-end. A straightforward idea is pipeline-based methods which firstly learn OOD cluster assignments and get pseudo OOD labels, then jointly classify labeled IND data and pseudo labeled OOD data. However, pipeline-based methods separate OOD clustering and classification process, which ignores the interaction between labeled IND data and unlabeled OOD data. Besides, these pseudo OOD labels may induce severe noise to the joint classification, limiting the performance of the joint IND and OOD classifiers. Therefore, we further propose an end-to-end framework to simultaneously learn pseudo OOD cluster labels and classify IND&OOD classes along with ground truth IND labels via a unified objective. We obtain the pseudo label of an OOD query by its augmented view in a swapped prediction way (Caron et al., 2020; Asano et al., 2020; Fini et al., 2021) and employ the Sinkhorn-Knopp (SK) algorithm (Cuturi, 2013) to solve the optimization problem. We leave the details to Section 4. We also perform exhaustive experiments (Section 5.2) and qualitative analysis (Section 5.3) to shed light on the challenges that current approaches face with GID. We find fine-grained OOD types, domain gap, data imbalance, real OOD noise and estimating the number of OOD types are the main challenges (Section 6), which provide insightful guidance for future GID work.

Our contributions are four-fold: (1) We introduce a new task, Generalized Intent Discovery (GID) which aims to extend an IND intent classifier to an open-world OOD intent set. GID helps expand the model's recognition scope and develop new skills for improving dialogue systems. (2) We construct three kinds of public GID benchmarks for different application scenarios, which help to explore the key challenges of GID comprehensively. (3) We propose an end-to-end GID framework to jointly learn clustering and classification, and extensive baselines of two frameworks, pipeline-based and end-to-end for future work. (4) We conduct exhaustive experiments and qualitative analysis to comprehend key challenges and provide new guidance for future GID research.

2 Problem Formulation

In this section, we first briefly introduce the traditional intent classification (IC) task, then dive into the details of our proposed Generalized Intent Discovery (GID) task.

Intent Classification Given a labeled in-domain (IND) dataset $\mathbf{D}^{IND} = \left\{\left(x_1^{IND}, y_1^{IND}\right), \ldots, \left(x_n^{IND}, y_n^{IND}\right)\right\}$, IC aims to predict the intent class of a test query by training an IND classifier, based on the assumption that all the queries belong to a pre-defined fixed set $\mathcal{Y}^{IND} = \{1, \ldots, N\}$ of N intent categories.

Generalized Intent Discovery In contrast, GID is to classify queries corresponding to both labeled IND and unlabeled OOD classes. Apart from the above labeled IND dataset $\mathbf{D}^{IND}$, an unlabeled OOD dataset $\mathbf{D}^{OOD} = \left\{\left(x_1^{OOD}\right), \ldots, \left(x_m^{OOD}\right)\right\}$ is also given. For simplicity, we assume the number of OOD classes is specified as M. In practical scenarios, we can estimate the number of clusters following previous work (Zhang et al., 2021b) (see Section 5.3.4). Since these OOD intents are usually collected from an online IC system by rejecting low confident queries [2], the set of N IND classes

[2] For example, given a test query, if an IC model predicts an output with low confident probability, we can assume the query doesn't belong to any IND type but OOD intents. Please refer to related OOD detection work (Xu et al., 2020; Zeng et al., 2021c; Zheng et al., 2020) for details. In this paper, we focus on the joint classification of unlabeled OOD and labeled IND. Thus, we suppose the two sets of IND classes and OOD classes are disjoint from each other.

is assumed to be disjoint from the set of M OOD classes. We also provide a discussion about real OOD noise in Section 5.3.2. The final goal of GID is 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 classes and the subsequent M ones denote unlabeled OOD classes. The challenges of GID come from two aspects, discovering the semantic concepts from unlabeled OOD data and jointly classifying IND&OOD intents. On the one hand, models need to automatically cluster OOD concepts which is more difficult than supervised classification tasks. On the other hand, they require jointly recognizing IND&OOD intents using these noisy cluster signals which may harm the final performance.

3 Dataset

To explore the practical significance and key challenges of GID task, we need to construct the GID dataset. However, we found that in some related tasks such as OOD intent discovery (Zhang et al., 2021b) and zero-shot intent detection (Siddique et al., 2021), the commonly used construction methods are to randomly divide the intent classification dataset into IND and OOD subset. This may not reflect real online intent classification scenarios.

We design more diverse GID dataset construction strategies, mainly in order to be able to discuss the practical significance and key challenges of GID more comprehensively. we construct three kinds of benchmark datasets GID-SD (single-domain), GID-MD (multiple-domain) and GID-CD (cross-domain) based on the two widely used intent datasets, CLINC (Larson et al., 2019b) and Banking (Casanueva et al., 2020). The three settings denote different real-world application scenarios in dialogue systems. Besides, we also construct two dataset variants GID-noise and GID-imbalance to explore more severe challenges of GID tasks in real scenes. We first briefly introduce original CLINC and Banking datasets, then elaborate on GID dataset construction, and display the statistic of GID benchmarks. Finally, we introduce evaluation metrics for the GID task, accuracy and F1 score both for IND and OOD data.

3.1 Original Intent Datasets

CLINC contains 22,500 queries covering 150 intents across 10 domains and Banking is a fine-grained dataset in a single domain, which contains 13,083 user queries with 77 intents. We show the detailed statistics of the two original datasets in Appendix A.1.

3.2 GID Dataset Construction

GID Benchmarks For CLINC and Banking datasets, we randomly choose the specified ratio (20%, 40%, 60%) of all intent classes as OOD types, and the rest are IND, similar to Xu et al. (2020); Zhang et al. (2021b).[3] The original train/val/test split is fixed. We only keep IND queries with their labels and the queries belonging to OOD classes in the original train and val data. Note that GID assumes OOD training data is unlabeled so we remove OOD queries' labels in the original train and val data. In the test set, we keep all the original IND and OOD intents and labels for evaluating metrics.[4] Considering different scenarios of dialogue systems, we construct three benchmarks, GID-SD (single-domain), GID-MD (multiple-domain) and GID-CD (cross-domain). Specifically, for the single-domain Banking dataset, we randomly select the specified ratio of all intent classes as OOD types, and the rest are IND to construct GID-SD. Since Banking has a large intent set in a single domain, we find these fine-grained OOD types are difficult to recognize (see Section 5.2). For the multiple-domain CLINC dataset, we propose two split strategies: (1) **Overlapping** (for GID-MD): We neglect the domain constraint and randomly split all the intent classes into the IND set and OOD set as above, which means intent categories from a domain may be divided to the two sets, which we call Domain Overlapping [5]. The situation occurs where an online IC system can hardly cover all the intent classes in a domain and OOD intents may come from the same domain as IND. (2) **Non-Overlapping** (for GID-CD): We restrict IND intent classes and OOD classes are from different domains, so we select a ratio of all domains as IND and the rest as OOD. Once a domain is chosen as IND, all the intents in this domain belong to IND intent classes and vice versa. The non-overlapping setting is more practical in a real scenario where we need to transfer a business to another.

[3] To avoid randomness, we report the averaged experiment results of three runs for each ratio. And for each run, all the models are based on the same dataset IND/OOD split.

[4] Although CLINC contains a real unlabeled OOD set, we can't use it because not able to evaluate the performance of models. We use the set for constructing a noisy GID dataset.

[5] Please mind IND intent classes and OOD classes are still disjoint from each other, but may belong to the same domain.

Dataset	IND classes	OOD classes	IND domains	OOD domains	Train samples	Val samples	Test samples
GID-SD-40%	46	31	1	1	5414/3589	600/400	1840/1240
GID-MD-40%	90	60	10	10	10,800/7200	1350/900	1350/900
GID-CD-40%	90	60	6	4	10,800/7200	1350/900	1350/900

Table 1: Statistics of GID-SD-40%, GID-MD-40% and GID-CD-40%.

GID Dataset Variants To explore more severe challenges of GID tasks in real applications, we construct two variants based on GID-MD-40%, GID-noise and GID-imbalance. (1) **GID-noise**: In the standard GID setting, we suppose the OOD data in the training set is "clean", that is, each OOD query must belong to a specific intent category. However, in practice, some OOD queries may be meaningless and not belong to any intent cluster, which we call OOD noise. We use 1350 real out-of-scope(oos) samples in CLINC, which semantically do not belong to any intent category in the training set, and add these noisy samples into the OOD train set to see if performance changes (see Section 5.3.2). Specifically, we add different numbers of oos samples according to 5%, 10% and 15% of the number of OOD samples in the training set of GID-MD-40%. (2) **GID-imbalance**: Data imbalance is a common issue in practice. To explore the impact of OOD data imbalance, we construct imbalanced GID datasets with different imbalance ratios ($\rho = 2, 3, 6$) by sampling each class of OOD samples in the GID-MD-40% training set. Following (Zhang et al., 2021c; Hong et al., 2021), we first sort the OOD classes of GID-MD-40% and each class is assigned an index $j (j = 1, 2, 3, ..., M)$, where M denotes the total number of OOD intent categories. Then we sample from each OOD class according to $n_j = n_{min}\rho^{(j-1)/M}$, $j = 1, 2, 3, ..., M$, where n_{min} is the least number of samples across all OOD classes. We adjust different imbalance ratios $\rho = n_{max}/n_{min}$ to simulate the degree of imbalance. $n_{max} = 120$ is the max number of samples per class in GID-MD-40%. We put the detailed statistics of GID-imbalance in Appendix A.2.

3.3 Statistic of GID Datasets and Evaluation

Since different proportions of OOD intents have different statistics, here we only display the results of 40% OOD for brevity. Table 1 shows the statistics of GID-SD-40%, GID-MD-40%, GID-CD-40%.

We use intent accuracy (ACC) and macro F1 as evaluation metrics for GID task. We report all IND, OOD and total (ALL) metrics where OOD and ALL ACC/F1 are the main metrics. Following Zhang et al. (2021b), we use the Hungarian algorithm (Kuhn, 1955) to obtain the mapping between the predicted OOD classes and ground-truth classes in the test set.

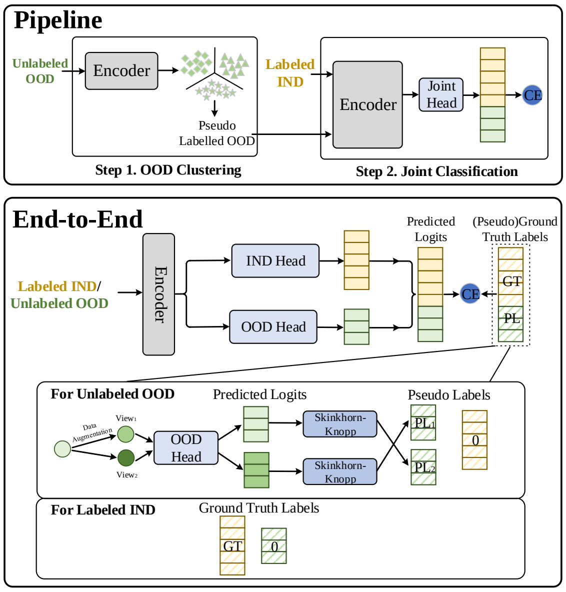


Figure 3: Overall architecture of our proposed pipeline and end-to-end methods.

4 Method

Overall Architecture We extend the idea of traditional intent classification models by using pseudo OOD labels. IC calculates the N-dimension cross-entropy (CE) loss for labeled IND data (Qin et al., 2019; He et al., 2020). Similarly, we can compute (N+M)-dimension CE loss both for labeled IND and unlabeled OOD data where IND labels are given but OOD (pseudo) labels are estimated (Zhang et al., 2021b; Han et al., 2020; Fini et al., 2021). Thus, the key challenge is to estimate OOD pseudo cluster labels by transferring prior IND knowledge. We propose two kinds of frameworks, pipeline and end-to-end, shown in Fig 3.

Pipeline A simple idea is pipeline-based methods which firstly learn OOD cluster assignments, then jointly classify labeled IND data and pseudo labeled OOD data. Specifically, we use the same BERT intent encoder as DeepAligned (Zhang et al., 2021b) to cluster OOD data. To transfer prior knowledge, we first pre-train the encoder on IND data to get intent representations. Then, we respectively use two OOD clustering methods, k-means (MacQueen, 1967) and DeepAligned to obtain pseudo OOD labels $\hat{\boldsymbol{y}}^{OOD}$. Finally, we mix up all the IND and OOD data and construct the new

(N+M)-dimension intent label $\boldsymbol{y}$ as follows:

$$\boldsymbol{y} = \begin{cases} \left[\boldsymbol{y}^{IND}; \mathbf{0}_M\right] & \mathbf{x} \in \mathbf{D}^{IND} \\ \left[\mathbf{0}_N; \hat{\boldsymbol{y}}^{OOD}\right] & \mathbf{x} \in \mathbf{D}^{OOD} \end{cases} \tag{1}$$

where $\boldsymbol{y}^{IND}, \hat{\boldsymbol{y}}^{OOD}$ are one-hot labels and $\mathbf{0}_M, \mathbf{0}_N$ are M or N-dimention zero vectors. We use the original CE loss to train a (N+M)-class open-set intent classifier.

End-to-End The main drawback of pipeline methods is the lack of deep semantic interaction between IND and OOD data in the clustering stage, leading to poor pseudo cluster labels. To alleviate the issue, we adopt an end-to-end framework to simultaneously learn pseudo OOD cluster labels and classify IND&OOD classes, shown in Fig 3. Our motivation is that each view of an OOD intent query after data augmentation can predict the other's pseudo labels, following swapped prediction (Caron et al., 2020). And we can learn the simple pseudo-labeling process via the unified classification loss instead of extra clustering objectives. Specifically, we use the same pre-trained BERT encoder in IND data as pipeline and two independent projection layers, IND head I and OOD head O. Given an input query, we concat the outputs of two heads as the final logit. For labeled IND intents, the ground-truth labels are easily obtained by Eq 1. We now discuss how to get the pseudo labels of unlabeled OOD intents. Inspired by Caron et al. (2020); Asano et al. (2020); Fini et al. (2021), we use the following swapped prediction way:

$$\ell_{CE}\left(\boldsymbol{x}_1, \hat{\boldsymbol{y}}_2\right) + \ell_{CE}\left(\boldsymbol{x}_2, \hat{\boldsymbol{y}}_1\right) \tag{2}$$

where $\boldsymbol{x}_1, \boldsymbol{x}_2$ are two dropout-augmented (Gao et al., 2021) views from an OOD intent query and $\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2$ are corresponding pseudo labels. We use $\boldsymbol{x}_1$ to compute $\hat{\boldsymbol{y}}_1$ and $\boldsymbol{x}_2$ for $\hat{\boldsymbol{y}}_2$. A simple way of obtaining $\hat{\boldsymbol{y}}_1$ from $\boldsymbol{x}_1$ is to regard the predicted softmax logits after OOD head of $\boldsymbol{x}_1$ as $\hat{\boldsymbol{y}}_1$. But Asano et al. (2020) observes this strategy easily leads to degenerate solutions where all the intents predict the same pseudo label and are grouped into the same cluster. Therefore, we add an entropy penalty to avoid all the pseudo labels are equal to each other and keep more uniform distribution of the pseudo-labels over all the M OOD clusters. We formulate the new optimization way:

$$\hat{\mathbf{Y}}^* = \arg\max_{\hat{\mathbf{Y}} \in \Gamma} \operatorname{Tr}(\hat{\mathbf{Y}}\boldsymbol{L}) + \epsilon \mathrm{H}(\hat{\mathbf{Y}}) \tag{3}$$

where $\hat{\mathbf{Y}} = [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_B]^\top$ is the matrix whose columns are the unknown pseudo-labels of the current batch B and $\boldsymbol{L} = [\boldsymbol{l}_1, \ldots, \boldsymbol{l}_B]$ is the predicted logits by the OOD head. H is the entropy function and ϵ is an hyper-parameter(we set it to 0.05 in the experiments). The goal is to obtain the best pseudo-labels $\hat{\mathbf{Y}}^*$ by maximizing Eq 3. And $\hat{\mathbf{Y}}$ must meet the following constraints similar to Caron et al. (2020); Fini et al. (2021):

$$\Gamma = \{\hat{\mathbf{Y}} \in \mathbb{R}_+^{M \times B} \mid \hat{\mathbf{Y}}\mathbf{1}_B = \frac{1}{M}\mathbf{1}_M, \hat{\mathbf{Y}}^\top \mathbf{1}_M = \frac{1}{B}\mathbf{1}_B\} \tag{4}$$

where $\mathbf{1}_B$ denotes the vector of all ones with B dimensions. Essentially, Eq 3&4 can be regarded as an optimal transport problem and we use the Sinkhorn-Knopp (SK) algorithm (Cuturi, 2013) to solve it.[6] After we get the pseudo OOD labels in a mini-batch, we can use Eq 1 to compute the CE loss. Note that the losses of IND and OOD data in a batch are jointly optimized. Compared to pipeline methods, our end-to-end method can simultaneously learn pseudo OOD cluster labels and distinguish IND&OOD classes via a CE loss. Joint optimization enables semantic interaction between IND and OOD data for better knowledge transfer and to reduce noisy clustering signals. For inference, we forward the input query (including IND and OOD) to the model and obtain its prediction.

5 Experiments and Analysis

5.1 Baselines

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

DeepAligned Similar to k-means, the difference is that the clustering algorithm adopts DeepAligned (Zhang et al., 2021b), which is the current state-of-the-art method for OOD discovery task.

DeepAligned-Mix This is an end-to-end approach where we extend DeepAligned for GID. DeepAligned is an iterative clustering method. In each iteration, it firstly uses k-means and an alignment strategy to cluster and label the OOD data and then computes the cross-entropy classification for representation learning. Our proposed DeepAligned-Mix mainly improves two points: (1) We mix up IND and OOD data together for iterative clustering, and the model is optimized with

[6] We recommend referring to Cuturi (2013) for more details about the theoretical explanation of optimal transport and SK algorithm.

Method	GID-SD-20%					GID-SD-40%					GID-SD-60%				
	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	91.29	70.50	71.43	87.21	86.90	90.38	62.34	62.44	78.99	78.32	90.40	51.58	51.96	67.08	66.70
DeepAligned	92.00	76.44	77.40	88.94	88.60	91.72	69.11	69.72	82.57	82.10	90.97	59.55	59.51	72.05	71.42
DeepAligned-Mix	85.62	56.28	60.26	79.90	78.20	82.30	54.97	59.79	71.30	69.60	80.70	52.66	54.66	63.95	61.92
End-to-End	92.82	**81.78**	**83.53**	**90.67**	**90.64**	92.84	**72.28**	**73.28**	**84.49**	**84.10**	**92.45**	**62.63**	**62.65**	**74.59**	**73.99**

Table 2: Performance on GID-SD (single-domain). 20%, 40% and 60% denotes the ratio of OOD intents. Results are averaged over three random run.($p < 0.01$ under t-test)

Method	GID-MD-20%					GID-MD-40%					GID-MD-60%				
	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	97.22	76.22	75.03	93.02	92.74	97.26	73.00	72.66	87.56	87.08	95.00	65.11	63.68	77.02	76.09
DeepAligned	97.83	90.89	91.08	96.43	96.32	97.85	87.55	87.14	93.70	93.29	97.67	83.38	82.78	89.10	88.52
DeepAligned-Mix	95.91	81.93	83.93	93.11	92.54	92.86	81.70	83.30	88.12	87.42	92.59	78.34	79.88	84.05	82.74
End-to-End	98.17	**95.26**	**96.08**	**97.58**	**97.59**	98.32	**91.92**	**92.46**	**95.78**	**95.73**	98.26	**87.63**	**87.84**	**91.88**	**91.78**

Table 3: Performance on GID-MD (multiple-domain).

Method	GID-CD-20%					GID-CD-40%					GID-CD-60%				
	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	97.39	75.78	75.79	92.98	92.72	97.70	61.67	60.43	83.20	82.30	96.44	54.67	53.69	71.38	70.57
DeepAligned	97.83	84.81	84.22	95.23	95.01	97.85	78.55	77.81	90.12	89.68	97.33	76.15	74.80	84.62	83.60
DeepAligned-Mix	97.15	77.41	77.7	93.20	92.53	97.33	72.41	71.54	87.36	86.21	93.89	75.63	74.29	82.93	81.37
End-to-End	97.92	**87.41**	**87.55**	95.81	**95.75**	98.00	**79.19**	**79.06**	90.46	**90.28**	98.22	**78.01**	**77.48**	**86.09**	**85.63**

Table 4: Performance on GID-CD (cross-domain).

a unified cross-entropy loss; (2) In the inference stage, instead of using k-means for clustering, we use the classification head of the new classifier to make predictions.

5.2 Main Results

We conduct experiments on three benchmark GID datasets GID-SD, GID-MD and GID-CD with different OOD ratios, shown in Table 4. In general, Our proposed end-to-end (E2E) method consistently outperforms all the baselines with a large margin. We analyze the results from three aspects:
Comparison of different methods We see E2E significantly outperforms all the baselines under the three datasets and different OOD ratio settings. For example, E2E outperforms previous state-of-the-art DeepAligned by 3.14%(OOD F1) and 2.57%(ALL F1) on GID-SD-60%, 5.06%(OOD F1) and 3.26%(ALL F1) on GID-MD-60%, 2.68%(OOD F1) and 2.03%(ALL F1) on GID-CD-60%. These prove that joint clustering and classification helps to perform more interaction between IND and OOD and obtain accurate pseudo OOD labels. We also observe E2E achieves slightly better IND ACC than pipeline methods, which means joint classification doesn't sacrifice IND performance while improving OOD recognition.
Comparison of different datasets To explore the effect of different practical scenarios, we compare the performance of the same method on different datasets. Results show metrics on GID-SD are the lowest, GID-CD is in the middle and GID-MD is the best for almost all the methods, which denotes the difficulty order is single-domain>cross-domain>multiple-domain. We argue GID-SD contains more fine-grained intent types in a single domain which makes it challenging to recognize OOD intents. Comparing CD and MD, IND and OOD types from the same domain makes it easier to transfer prior knowledge, so MD gets higher scores.
Effect of different OOD ratios We compare the results of different OOD ratios on the same dataset. We find with the increase of OOD ratio, the performance consistently drops. For example, E2E achieves 95.26% OOD ACC on GID-MD-20%, but OOD ACC decreases by 3.34% on GID-MD-40% and 7.63% on GID-MD-60%. Intuitively, the increase in the number of OOD intents makes it more difficult to distinguish them.

5.3 Qualitative Analysis

5.3.1 Cross-Domain Transferability

For GID-CD, cross-domain knowledge transfer is important and challenging. To study the effect of domain similarity on knowledge transfer, we perform a cross-domain transferability analysis in Fig

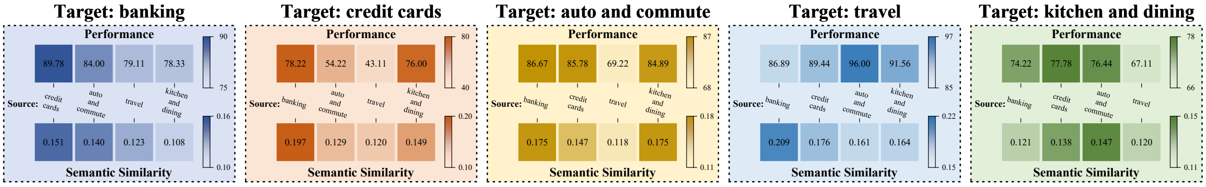


Figure 4: Cross-domain transferability from source IND to target OOD. We display OOD ACC and domain similarity scores. The larger the number is, the deeper the color is.

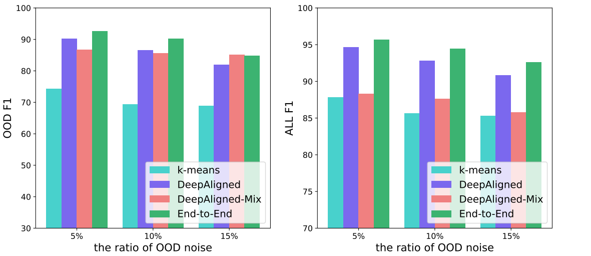


Figure 5: The impact of adding different numbers of noisy OOD samples to the training set on the performance of each GID model.

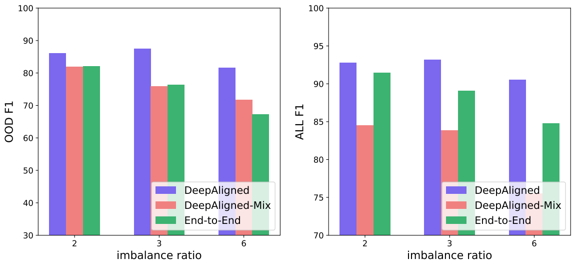


Figure 6: The impact of different imbalance ratios of OOD data on the performance of each GID model.

4. We select five domains (banking, credit_card, auto_and_commute, travel, kitchen_and_dining) and perform the one-to-one transfer. To measure domain similarity, we first train an IND intent classifier, then perform k-means using extracted representations of OOD samples to calculate Silhouette Coefficient (SC) values (Rousseeuw, 1987) [7]. We can see that the larger the SC value is, the higher the similarity between IND&OOD domains is, resulting in better OOD metrics. The results prove good cross-domain transferability comes from semantically similar domains, such as banking and credit_card.

5.3.2 Effect of OOD noise

In the real world, OOD data may not necessarily belong to a certain OOD cluster, and there is often some OOD noise. We use the constructed dataset variant GID-noise to examine the impact of noisy OOD in the training set on model performance. Fig 5 shows the impact of different amounts of OOD noise in the training set on model performance. The results show that as the amount of OOD noise increases, the OOD performance drops. Our proposed E2E still achieves the best performance over all baselines. We argue that this is because the presence of OOD noise makes it difficult for the model to learn a clear cluster boundary for unlabeled OOD.

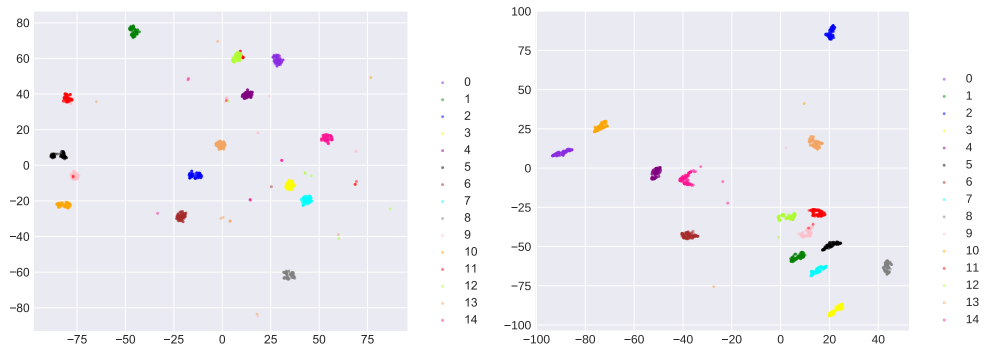
5.3.3 Effect of imbalanced OOD data

Fig 6 shows the impact of class imbalance degree of OOD data on model performance. The results show that when the imbalance degree of OOD categories increases, the performance of all models decreases significantly. We also find an interesting phenomenon that our proposed end-to-end method drops more significantly than pipeline-based DeepAligned. We argue that there are two reasons for this. (1) When our end-to-end method obtains OOD pseudo-labels, the SK algorithm is based on a strong assumption, the number of pseudo-labels for each category in a batch is uniform, which is obviously invalid in the class-imbalanced scenario. (2) E2E uses IND and OOD to jointly train the classifier. Since the number of samples in each class of IND keeps fixed to 120(equal to the number of OOD samples in the majority class of OOD), this will exacerbate the degree of imbalance and affect the accuracy of pseudo-labels for long-tail categories. Therefore, we need to further explore better pseudo-label methods in the future and how to improve the class-imbalanced defect of end-to-end methods.

5.3.4 Estimate the Number of Cluster K

All the results we showed so far assume that the number of OOD classes is pre-defined. However, in real-world applications, this often needs to be estimated automatically. Table 5 shows the results using the same automatic K-value estimation strategy

[7] Please see more details about SC in Appendix A.6.



(a) DeepAligned

(b) End-to-End

Figure 7: IND & OOD intents visualazation of DeepAligned and E2E method, we select 9 IND intents and 6 OOD intents in GID-MD-40% (index 0-8 denotes IND intents, index 9-14 denotes OOD intents)

	OOD ACC	OOD F1	ALL ACC	K
DeepAligned	87.55	87.14	93.70	60
DeepAligned-Mix	82.70	84.65	88.12	60
End-to-End	91.92	92.46	95.78	60
DeepAligned	72.89	66.75	87.91	47
DeepAligned-Mix	69.56	62.32	85.29	47
End-to-End	74.89	67.23	88.58	47

Table 5: Estimate the number of OOD clusters. K=47 is the estimated number compared to original 60.

[8]. We find that our method both achieves the best performance under the fixed or auto K settings. It should be noted that no matter the end-to-end methods or the pipeline methods, the performance drops significantly when the number of OOD classes is unknown. Therefore, how to estimate an accurate K value and how to design a more robust GID method is a great challenge.

5.3.5 Visualization

To further visually compare the performance of end-to-end and pipeline methods in classifying IND and clustering OOD, we performed a visualization of IND & OOD intent representations for E2E and DeepAligned, as shown in Fig 7. Comparing E2E to DeepAligned, we can observe DeepAligned gets some mixed OOD clusters (see greenyellow and red dots in Fig a) while E2E method successfully separates them. We also find that many OOD intents in DeepAligned that cannot be clustered into single cluster, but are scattered into multiple clusters (see deeppink dots in Fig a), but E2E method can form compact clusters for them. We argue this is because the pipeline method introduces serious error propagation in the OOD clustering stage; while the E2E method jointly learns OOD cluster assignments and classification of IND & OOD, which helps to get clear cluster boundary.

5.3.6 Noise of IND

In the general GID setting, we assume that the IND and OOD categories do not overlap, however the OOD data collected in practical application scenarios may have some IND noise due to the error propagation of OOD detection. We analyze the performance changes of each GID method when mixing different proportions of IND noise in OOD data, as shown in Fig 10. The results show that our E2E method still significantly outperforms the pipeline baseline under IND noise scenarios. The performance of IND classification and OOD clustering for all methods decrease significantly, and the IND performance decrease is more significant for DeepAligned and E2E. We argue that this is due to the inclusion of a small amount of IND data in the OOD data, which causes these IND data to be incorrectly labeled, and severely impairs the performance of IND classification, making it difficult to form clear IND class boundaries. We also found that when the IND noise ratio reached 15%, the OOD clustering performance of the E2E method was worse than DeepAligned. We argue that this is because the E2E method jointly learns to classify IND intents and discover OOD intents, which needs to leverage IND prior knowledge to enhance OOD clustering. However, When there is more IND noise to be mixed with OOD data, it will affect the effectiveness of the knowledge interaction between IND and OOD. In practical applications, when the performance of OOD detection is improved, this IND noise problem can be relieved naturally, which is not within the scope of this papar.

[8] Here we use the same estimation algorithm as Zhang et al. (2021b). We leave the details in Appendix A.4.

6 Challenges

Based on the above analysis, we summarize the current challenges faced by the GID task:

Fine-grained OOD data When OOD intents are fine-grained like GID-SD, the OOD performance of existing GID methods decreases significantly. We argue fine-grained OOD intents make it hard to construct clear boundary while clustering.

Cross-domain transfer When IND and OOD intent types are from different distant domains, the knowledge learned from IND is hard to transfer to OOD due to the semantic gap in different domains.

OOD noise OOD data collected in practical applications are usually noisy, and there may be some OOD samples that do not belong to a certain intent type. The performance of each GID method degrades when trained with these noisy OOD data.

imbalanced OOD data The OOD data in real-world scenarios is often class-imbalanced, and our analysis in section 5.3.3 proves that the performance of current methods drops significantly under imbalanced data, especially end-to-end methods.

Inaccurate estimation of the number of OOD categories Most previous work assume the number of OOD categories is known. However, in practical applications, we usually need to estimate the number of categories K, which is often inaccurate. We propose a preliminary analysis in Section 5.3.4 which shows significant performance drop when the estimation is not totally accurate.

7 Related Work

OOD Detection aims to know when a query falls outside the range of pre-defined supported intents (Zeng et al., 2021a; Lin and Xu, 2019; Xu et al., 2020; Zeng et al., 2021b; Wu et al., 2022) to avoid performing wrong operation. OOD detection has attracted more and more attention in recent years, so various similar names are derived, such as anomaly detection, open world classification (Shu et al., 2021), open-world learning (Xu et al., 2019), open intent classification(Zhang et al., 2021a) and so on. However, all of them are essentially to distinguish whether a query belongs to IND or OOD intents, without further discovering new semantic categories from unsupervised OOD data.

OOD Discovery aims to discover new intent concepts from unlabeled OOD data and form OOD intent clusters (Lin et al., 2020; Zhang et al., 2021b; Mou et al., 2022), which focuses more on how to cluster OOD data, while ignoring the fusion of IND and OOD, which makes the model only recognize OOD intents. For example, (Zhang et al., 2021b) design an iterative clustering algorithm DeepAligned, which iteratively learns intent representations then cluster assignments. **Open Intent Extraction** also aims to extract unknown intents from unlabelled user queries (Vedula et al., 2020), and is a completely unsupervised task. However, in terms of method, open intent extraction is more about extracting intent names through sequence annotation methods. In contrast, GID aims to train a network that can simultaneously classify a set of labeled IND intent classes while discovering and recognizing unlabeled OOD intents.

Incremental/Continual Learning There is currently some work on extending closed-set classifier to new classes in the open world incrementally, such as (Xu et al., 2019). But all these works follow a traditional incremental learning setting, which requires new category data with labels. In practical applications, we can only obtain these unlabeled OOD data from the dialogue system logs, and these data are often updated continuously, and human annotation of these data is very labor-intensive. Therefore, we propose a more human-free task GID, which aims to automatically discover new categories from the unlabeled OOD data, and further expand the recognition scope of the existing IND intent classifier incrementally.

Zero-shot Intent Detection Zero-shot intent detection (Yan et al., 2020; Siddique et al., 2021) assumes that no target domain data is available during training, but the category and category descriptions from target domain are given, but in practical applications we often have access to a large amount of unlabeled dialogue logs, and we need to consider how to discover new intent categories from them for system development.

8 Conclusion

In this paper, we introduce a new task, Generalized Intent Discovery (GID), which aims to extend an IND intent classifier to an open-world intent set. Then we provide three public datasets for different application scenarios and establish a benchmark for the GID task. We also propose extensive baselines of two frameworks, pipeline-based and end-to-end for future work. Further, We conduct exhaustive experiments and qualitative analysis to comprehend key challenges and provide new guidance for future GID research.

Acknowledgements

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

Broader Impact

Task-oriented dialogue systems have demonstrated remarkable performance in a wide range of applications, and have significant positive impact on human production mode and lifeway. Intent classification is an important component of task-oriented dialogue system. Existing intent classification models can only identify a limited number of predefined in-domain (IND) intents, however, out-of-domain (OOD) or unknown intents will appear continually when the dialogue system is deployed online. If we can group these OOD samples into different clusters, we can discover new intents, guide future development of the system, and expand the classification capabilities of the system. We note that OOD intent detection and OOD intent discovery tasks have been widely studied recently. The former focuses on identifying whether a sample is IND or OOD, while the latter focuses on how to cluster OOD data. The generalized intent discovery (GID) task proposed in this paper focuses on an incremental setting, that is simultaneously classifying a set of labeled IND intent classes while discovering and recognizing new unlabeled OOD types incrementally. GID aims to provide the model with the ability to automatically learning according to known knowledge in the open world, which is a new attempt for scalable dialogue system and open world learning.

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*, abs/1902.10909.

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.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471, Florence, Italy. Association for Computational Linguistics.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv preprint*, abs/2104.08821.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Kai Han, Sylvestre-Alvise Rebuffi, Sébastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. 2020. Automatically discovering and learning new visual categories with ranking statistics. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Keqing He, Shuyu Lei, Yushu Yang, Huixing Jiang, and Zhongyuan Wang. 2020. Syntactic graph convolutional network for spoken language understanding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2728–2738, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. 2021. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6626–6636.

H. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

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. 2019a. 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)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

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. 2019b. 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)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8360–8367. AAAI Press.

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*.

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

Yutao Mou, Keqing He, Yanan Wu, Zhiyuan Zeng, Hong Xu, Huixing Jiang, Wei Wu, and Weiran Xu. 2022. 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.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China. Association for Computational Linguistics.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14680–14691.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Lei Shu, Yassine Benajiba, Saab Mansour, and Yi Zhang. 2021. Odist: Open world classification via distributionally shifted instances. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3751–3756.

AB Siddique, Fuad Jamour, Luxun Xu, and Vagelis Hristidis. 2021. Generalized zero-shot intent detection via commonsense knowledge. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1925–1929.

Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020. Open intent extraction from natural language interactions. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2009–2020. ACM / IW3C2.

Yanan Wu, Keqing He, Yuanmeng Yan, QiXiang Gao, Zhiyuan Zeng, Fujia Zheng, Lulu Zhao, Huixing Jiang, Wei Wu, and Weiran Xu. 2022. Revisit overconfidence for ood detection: Reassigned contrastive learning with adaptive class-dependent threshold. In *NAACL*.

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.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. Open-world learning and application to product classification. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 3413–3419. ACM.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y.S. Lam. 2020. Unknown intent detection using Gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1050–1060, Online. Association for Computational Linguistics.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021a. 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.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Hong Xu, and Weiran Xu. 2021b. Adversarial self-supervised learning for out-of-domain detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5631–5639, Online. Association for Computational Linguistics.

Zhiyuan Zeng, Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2021c. Adversarial generative distance-based classifier for robust out-of-domain detection. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7658–7662.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021a. Deep open intent classification with adaptive decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14374–14382.

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

Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. 2021c. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *ArXiv preprint*, abs/2107.09249.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1198–1209.

A Appendix

A.1 Original Intent Dataset Statistics

We show the detailed statistics of CLINC and BANKING datasets in Table 6. Banking is class-imbalanced, and the number of samples for each class is shown in Fig 8. The three GID datasets GID-SD GID-MD and GID-CD we constructed in this paper, all maintain the same train/dev/test split as the original dataset. Table 7 shows the number of intents divided into IND and OOD per domain for GID-MD-40%. Since CLINC and BANKING are open source datasets, there is no license problem.

A.2 GID-imbalanced

For our imbalanced dataset GID-imbalance, we show the distribution of the number of samples per OOD category under the influence of different imbalance ratio($\rho = 2, 3, 6$) in Figure9. The larger the imbalance ratio, the more significant the class imbalance degree of the corresponding imbalanced dataset.

A.3 Implementation Details

For a fair comparison of the various methods, we use the pre-trained BERT model (bert-base-uncased [9], 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., 2021b). Firstly, we use labeled IND data to pretrain BERT model. For pipeline method(k-means and DeepAligned), we use the official implementation and hyperparameters offered by (Zhang et al., 2021b) to realize it, and the batch size is 512 and learning rate is 5e-5 for joint classification stage. For DeepAligned-Mix, the training batch size is 512 and the learning rate is 5e-5. For end-to-end method, IND head and OOD head are two symmetrical MLPs (input dimension is 768 and output dimension is the number of categories for IND/OOD), and we select $tanh$ as activation function as previous work. We use SGD with momentum as optimizer, with linear warm-up and cosine annealing (lr_{base} = 0.4, lr_{min} = 0.01), and weight decay 10^{-4}. The batch size is always set to 512 for all experiments. Notably, We use dropout (Gao et al., 2021) to construct augmented examples and the dropout value is fixed at

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

Dataset	Classes	Training	Validation	Test	Vocabulary	Length (max / mean)
CLINC	150	18,000	2,250	2,250	7,283	28 / 8.31
BANKING	77	9,003	1,000	3,080	5,028	79 / 11.91

Table 6: Statistics of CLINC and BANKING datasets.

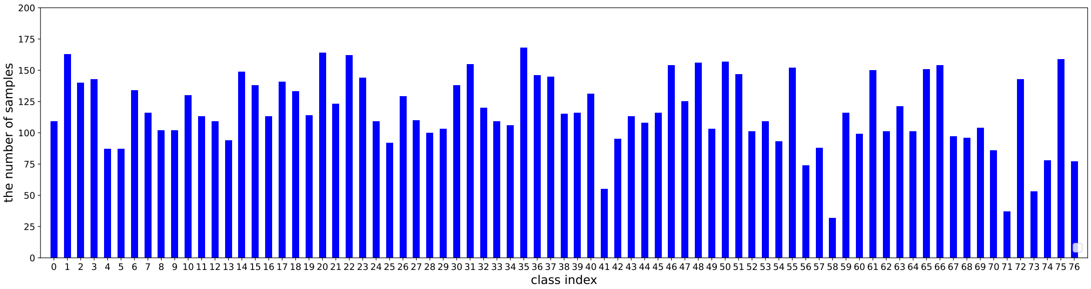


Figure 8: The number of samples for each class in Banking dataset.

Domains	IND intents	OOD intents
banking	10	5
credit_cards	8	7
kitchen_and_dining	9	6
home	6	9
work	10	5
utility	8	7
travel	9	6
auto_and_commute	10	5
small_talk	11	4
meta	9	6

Table 7: The number of intents divided into IND and OOD per domain for GID-MD-40%

0.5. For what concerns pseudo-labeling, we use the implementation of the Sinkhorn-Knopp algorithm provided by (Caron et al., 2020) and we inherit all the hyperparameters from (Caron et al., 2020), e.g. $n\_iter = 3$ and $\epsilon = 0.05$. We use the SC value of the validation data to select the best checkpoints. All experiments use a single Tesla T4 GPU(16 GB of memory). Table8 shows the comparison of the epoch and training time required for the convergence of the End-to-End method and DeepAligned. We can see that the E2E method takes fewer epochs to converge than the DeepAligned method. This is because the DeepAligned method first performs OOD clustering, and then uses the obtained OOD pseudo-labels and IND ground-truth labels for joint classification, which will lead to The OOD pseudo-labels have serious errors, and these label errors cannot be corrected in classification process, resulting in difficulty in model convergence. In addition, we can also see that the E2E method only increases the time required for each epoch by 1.8s compared to the classification stage of DeepAligned, which indicates the efficiency of the SK algorithm.

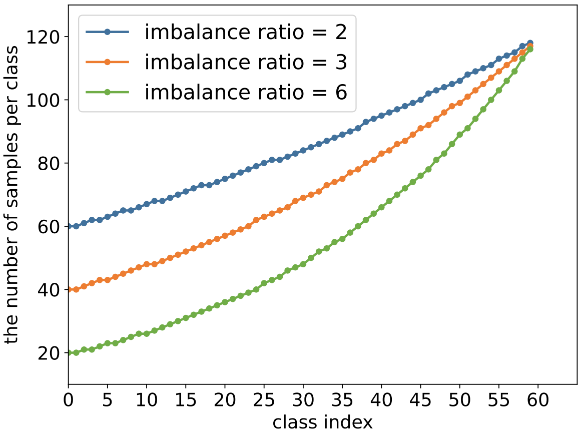


Figure 9: The distribution of the number of samples for GID-imbalance

Method	training epoch	training time
End-to-End	51	30s/epoch
DeepAligned(two-stages)		
- clustering	67	27.6s/epoch
- classification	91	28.2s/epoch

Table 8: Comparison of training efficiency between pipeline and End-to-End methods. We use the same hardware.

A.4 Estimate K

Since we may not know the exact number of OOD clusters, we use the following K estimation method (Zhang et al., 2021b) to determine the number of

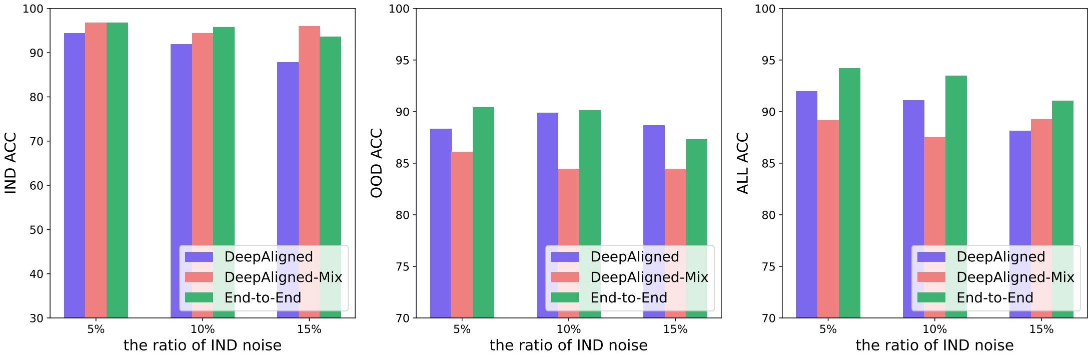


Figure 10: The impact of adding different ratios of IND noise samples to the OOD training data on the performance of each GID model.

clusters K before clustering. The method estimates K with the aid of the well-initialized intent features. We assign a big K' as the number of clusters at first. As a good feature initialization is helpful for partition-based methods (e.g., k-means), we use the well pre-trained model to extract intent features. Then, we perform k-means with the extracted features. We suppose that real clusters tend to be dense even with K', and the size of more confident clusters is larger than some threshold t. Therefore, we drop the low confidence cluster whose size is smaller than t, and calculate K with:

$$K = \sum_{i=1}^{K'} \delta\left(|S_i| >= t\right) \tag{5}$$

where $|S_i|$ is the size of the i^{th} produced cluster, and $\delta(\cdot)$ is an indicator function. It outputs 1 if condition is satisfied, and outputs 0 if not. Notably, we assign the threshold t as the expected cluster mean size $\frac{N}{K'}$ in this formula.

A.5 Effect of IND Data

We analyze the impact of the number of samples per IND class on the performance of each model. Fig 11 shows the trend of model performance as the number of IND samples for each class decreases. Overall, the performance of our end-to-end method is much better than the baselines. Moreover, with the decrease of the amount of in-domain data, all methods show varying degrees of performance fluctuation. We observe the changes of IND F1 and OOD F1, and find IND F1 generally shows a downward trend, especially for DeepAligned-Mix. We believe that this is because the number of IND samples in each category is reduced, resulting in the biased joint classification of IND&OOD towards the OOD category. DeepAligned-Mix learns both IND and OOD by clustering, which will lead to inaccurate pseudo-labels obtained by IND, further degrading the performance. As for OOD F1, due to the reduced number of IND samples, the model can learn less IND prior knowledge, thus affecting the performance of OOD. Therefore, GID in the small labeled IND scenario is also a challenge worthy of attention.

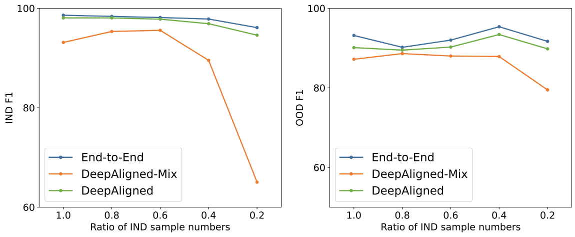


Figure 11: Effect of IND data for GID. The left subfig denotes IND F1 and the right subfig denotes OOD F1.

A.6 Silhouette Coefficient (SC)

Following Zhang et al. (2021b), we use the cluster validity index (CVI) to evaluate the quality of clusters obtained during each training epoch after clustering. Specifically, we adopt an unsupervised metric Silhouette Coefficient (Rousseeuw, 1987) for evaluation:

$$SC = \frac{1}{N} \sum_{i=1}^{N} \frac{b\left(\boldsymbol{I}_i\right) - a\left(\boldsymbol{I}_i\right)}{\max\left\{a\left(\boldsymbol{I}_i\right), b\left(\boldsymbol{I}_i\right)\right\}} \tag{6}$$

where $a\left(\boldsymbol{I}_i\right)$ is the average distance between $\boldsymbol{I}_i$ and all other samples in the i-th cluster, which indicates the intra-class compactness. $b\left(\boldsymbol{I}_i\right)$ is the smallest distance between $\boldsymbol{I}_i$ and all samples not in the i-th cluster, which indicates the inter-class separation. The range of SC is between -1 and 1, and the higher score means the better clustering results.