# Towards Open Environment Intent Prediction

**Yunhua Zhou, Jiawei Hong, Xipeng Qiu**[*]
School of Computer Science, Fudan University
{zhouyh20,xpqiu}@fudan.edu.cn
hongjw21@m.fudan.edu.cn

## Abstract

*Out-of-Domain (OOD) Intent Classification* and *New Intent Discovery* are as two basic and critical tasks in the Task-Oriented Dialogue System, which are typically treated as two independent tasks. *Classification* focuses on identifying intents beyond the predefined set of the dialog system, but it will not further differentiate detected OOD intents in fine granularity. *Discovery* focuses on how to cluster unlabeled samples according to their semantic representation, which relies heavily on prior knowledge and can not provide label information for the formed clusters. To be closer to the real user-facing scenarios, we strengthen a combined generative task paradigm to extend *Classification* with *Discovery* referred to as Open Environment Intent Prediction, which is to make a further fine-grained discovery of OOD based on OOD Intent Classification. Using various widely-used generative models as an archetype, we propose a general scheme for Open Environment Intent Prediction. In a nutshell, we first perform intent detection to identify the In-domain (IND) samples and then generate labels for those identified as OOD. With these generated labels, we can discover new general intents and provide label information for them. We develop a suite of benchmarks on the existing intent datasets and present a simple yet effective implementation. Extensive experiments demonstrate that our method establishes substantial improvement compared to the baselines. Codes is publicly available.[1]

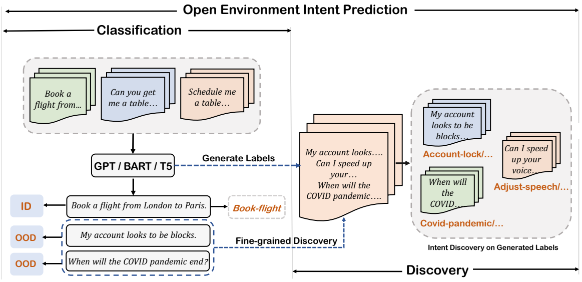

Open Environment Intent Prediction
Classification
Book a flight from...
Can you get me a table…
Schedule me a table…
GPT / BART / T5
Generate Labels
ID
Book a flight from London to Paris.
Book-flight
OOD
My account looks to be blocks.
OOD
When will the COVID pandemic end?
Fine-grained Discovery
My account looks…. Can I speed up your… When will the COVID pandemic….
My account looks to be blocks…
Account-lock/…
Can I speed up your voice…
Adjust-speech/…
When will the COVID…
Covid-pandemic/…
Intent Discovery on Generated Labels
Discovery

Figure 1: Illustration of Open Environment Intent Prediction. (a) *Classification* is to distinguish IND (give specific labels for IND samples) and OOD. (b) *Discovery* is to make a further fine-grained discovery of OOD based on generated labels. See text for details.

## 1 Introduction

OOD Intent Classification, also known as OOD Intent Detection (OID), and New Intent Discovery (NID), as two basic tasks of the Task-Oriented Dialogue System, have been two areas of active research. The purpose of OOD Intent Classification (Zhang et al., 2021b; Zhan et al., 2021; Zhou et al., 2022a) is to identify utterances with not supported intents to prevent them from being wrongly post-processed. However, in the setting of OID, all OOD samples, which contain a lot of valuable corpus with different meaningful intents, are just grouped into one rejected class and are not distinguished in a fine-grained way. At the same time, how to effectively identify intents under the generative paradigm has been underdeveloped.

New Intent Discovery (Zhang et al., 2021c; Zhou et al., 2022b) focuses on how to cluster unlabeled data according to their learned semantic representation. However, existing research on New Intent Discovery needs strong prior knowledge (Zhang et al., 2022) to learn the semantic representation that can adapt to subsequent clustering, which also depends on unacceptable assumptions in real scenarios, such as knowing the number of categories of OOD intents in advance. In addition, its processing procedure is usually cumbersome with multiple dependent processing stages, which often suffers the dilemma that the knowledge learned in the previous is often forgotten in the follow-up as demonstrated in Zhou et al. (2022b) and the generated clusters usually lack semantic label information. Further, since unlabeled data usually contains a large number of samples with known intents, a closer look at

[*] Corresponding author.

[1] https://github.com/zyh190507/Open-Enviroment-Intent-Prediction

*Findings of the Association for Computational Linguistics: ACL 2023*, pages 2226–2240
July 9-14, 2023 ©2023 Association for Computational Linguistics

the process of NID will reveal that it pays a lot of costs, but in many cases, it is just gathering a large number of samples with definite intents into clusters but cannot provide labels and not fully commit to discovering new intents.

To be closer to the realistic scenarios, we first strengthen a combined generative task paradigm based on the characteristics of the above two tasks–Open Environment Intent Prediction, which is to make a further fine-grained discovery of OOD based on OOD Intent Classification and not only gives the specific categories of IND samples but also further gives the label information of OOD. This paradigm can reduce the “burden” of existing NID tasks by avoiding clustering a large number of known intent samples and focusing on discovering new intents while giving specific label information. Compared with OID and NID, our proposed task paradigm is more general and practical, whose whole process is shown in Figure 1. Then we offer a general implementation based on the generative models. Specifically, with a generative model in hand, we carry out OID according to the learned semantic representation and give the corresponding predefined label for IND. At the same time, labels are generated for the samples identified as OOD, which also can help to discover more general intents in fine granularity.

For more general and practical, we expect to not rely on any assumptions or prior about OOD and directly provide high-quality OOD labels, which also becomes more challenging. Especially for the label generation, since only IND samples are available in the training set, fine-tuning the model directly (Model-tuning) will cause the generated labels to overfit training labels, making it a poor choice for the Open Environment Intent Prediction. Therefore, we adopt prefix-tuning (Li and Liang, 2021) to retain the general knowledge learned during pretraining to avoid shifting toward the training labels to generate more diverse labels. On this basis, to discover more general intents, we reformulate the intent discovery as a minimum cost *Multi-Cut* problem, which can automatically divide samples belonging to a general intent into a cluster according to the similarity of labels. Further, to mitigate the impact of Inherent Label Uncertainty (Wang et al., 2022) on the Open Environment Intent Prediction, with the help of large pre-trained models such as GPT-3 (Brown et al., 2020) or ChatGPT[2], we introduce a simple yet effective method of enriching the expression of intents and generate multiple related labels for each intent in the training set.

The contributions can be summarized as follows: Firstly, this paper strengthens a combined generative task paradigm, which can not only give the specific category of IND but also give the label information of OOD and can further discover more general intents. Secondly, this paper offers an effective implementation for such a paradigm without relying on any prior about OOD and provides a novel solution for enriching the expression of intents and a general method for intent discovery. Thirdly, to evaluate the effectiveness and generality of our method, we establish a suite of benchmarks across widely-used generative models and datasets. The experimental results demonstrate our method not only performs better *classification* but also makes an effective *discovery*.

## 2 Related Work

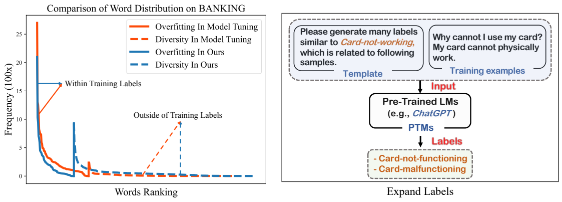
**OOD Intent Detection (OID)** OID is a field of concern recently, and many excellent related pieces of research have emerged. According to whether there are additional OOD samples involved in the training process, these works can be broadly categorized into two main groups, namely supervised and unsupervised. The supervised approaches (Zheng et al., 2020; Zhan et al., 2021; Lang et al., 2022) focus on how to help distinguish IND and OOD by using additional collected or synthesized OOD samples. The unsupervised methods usually constrain decision boundaries through specific training paradigms (Zeng et al., 2021; Zhou et al., 2022a) or post-processing methods (Zhang et al., 2021b). The existing work usually groups all OOD samples into one rejected class without further fine-grained distinction. At the same time, there is less research on generative models for OOD Intent classification. This work explores how to carry out OOD classification on the generative models and expand the OOD Intent Classification.

**New Intent Discovery (NID)** This name may be a bit misleading (it is called Generalized Category Discovery (Vaze et al., 2022) in the field of computer vision). In the field of natural language processing, unlabeled corpus in the setting of NID include samples with known intents in addition to OOD samples. Zhang et al. (2021c, 2022) learn the clustering friendly-representation by generalizing prior knowledge to the representation of unlabeled

[2] https://openai.com/blog/chatgpt/

samples so that samples with similar representations can be divided into the same cluster. Gao et al. (2021b) discover new intents by a variant of PageRank and Intent rank algorithm and Zhou et al. (2022b) introduce a principled probabilistic framework for this task. Zhang et al. (2021a) provide a tool platform to integrate various existing methods about OID and NID. Vedula et al. (2020); Zheng et al. (2022) can be approximated as two specific implementations of the paradigm proposed in this work. However, they either need to rely on the prior knowledge of OOD or need to make complex category estimations. Further, they need to rely on all samples during discovery and cannot directly provide label information, which is not general. Different from the previous work, we use a model to implement the Open Environment Intent Prediction, and our method does not rely on any prior knowledge or assumptions about OOD while providing effective label information.

**Parameter-Efficient Tuning (PET)** PET aims to optimize as few parameters as possible while achieving the effect as optimizing all parameters (He et al., 2022). To this end, Lester et al. (2021) inject tunable prompts into the input layer. Li and Liang (2021); Liu et al. (2022) go a step further and put tunable prompts on each internal layer of the model to achieve better results.



(a) Generated Words Distribution (b) Label Extension by PTMs

Figure 2: Plots show (a) the comparison of generated word distribution, the solid line represents the distribution of words falling into the training label set (overfitting), and the dotted line represents the distribution of words beyond the label list (diversity). The inside of each part is sorted by word frequency from high to low. Our method (Blue) can not only alleviate overfitting but also increase diversity. (b) label extension by the PTMs. See text for details.

## 3 Proposed Method

A natural solution to solve the Open Environment Intent Prediction is to carry out full model tuning, i.e., fine-tune all the parameters of the generative models, by taking generating labels for IND samples as the downstream task. However, model tuning could lead to a certain degree of "degradation" of the vocabulary generated by the fine-tuned generative model, which means that the generated labels overfit the labels in the training set.

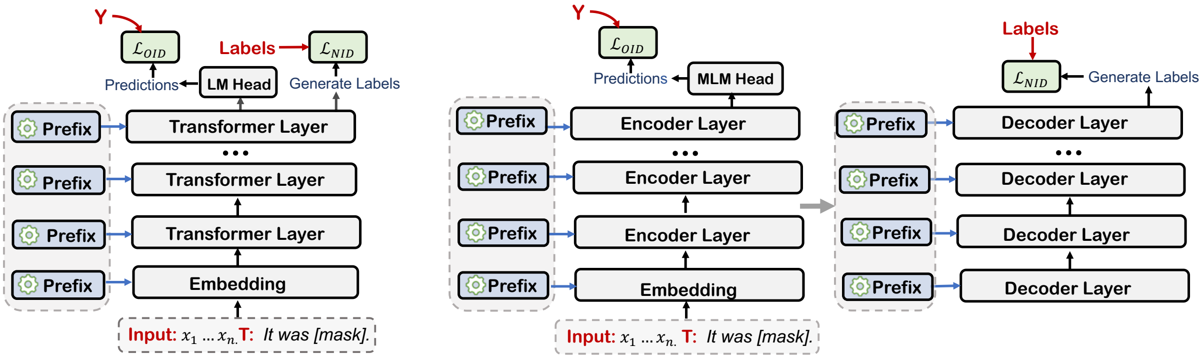
Specifically, as shown in Figure 2(a), almost all the words generated by the fine-tuned model fall in the vocabulary composed of the labels in the training set (solid red line in Figure 2(a)), and few words beyond the vocabulary (dotted red line in Figure 2(a)) can be generated, which will fail to generate correct labels for OOD samples.

**Prompt-based prefix tuning** To retain the general knowledge (avoid shifting towards training labels) obtained by pre-training in large-scale corpus while adapting the model to the downstream task that generates diverse labels, we achieve it by prompting the model with tunable instructions to retain the main parameters of the model unchanged. Specifically, we adopt the prefix-tuning (Li and Liang, 2021; Liu et al., 2022) training paradigm to prepend continuous tunable tokens $p_l \in \mathcal{R}^{n \times d}$ (termed as prefix) to the $l$-th internal layer of the model, denoting $P = [p_1, p_2, ..., p_l]$ as the whole prefixes in all layers. In addition, to steer generative models to generate labels according to the content of samples, we formulate the input $X$ to model with natural language prompts (such as "*It was [Mask]*", which a crafted *template* of *prompt*) into $\mathcal{T}(X) = \{x.\text{It was [Mask]}.|x \in X\}$ to prompt the model to generate appropriate labels for [Mask] during decoding as suggested in Gao et al. (2021a). The optimization objective is formulated as follows:

$$P = \arg\min_{P \in \mathcal{P}} \mathcal{L}_{obj}(\mathcal{F}(\mathcal{T}(X), P; \theta), Y), \qquad (1)$$

where $\mathcal{F}$ is the generative model, $\theta$ is main parameters, $\mathcal{P}$ is the prefix space, $\mathcal{L}_{obj}$ is the tuning loss Eq.(6) and $Y$ is the label space. The whole process of tuning is shown in Figure 3 and the advantages of this proposed method are shown in subsequent experiments.

**Label Extension with Pre-trained Models** Both the OID and NID tasks face a real dilemma. Because of the inherent defects of annotation and diversity of intent expression, only one label given in the dataset usually can not accurately reflect the true intents behind the samples or even is wrong for some samples, which can be called Inherent Label Uncertainty (ILU). ILU not only affects the defi-



(a) Autoregressive Model

(b) Encoder-Decoder Model

Figure 3: An illustration of our implementation based on (a) Autoregressive Model (GPT-2) and (b) Encoder-Decoder Model (BART/T5). The prefix refers to trainable prompt tokens, and the parameters of the pre-trained models are frozen. The input to the model includes a crafted prompt in addition to the original utterance. (M)LM head output logit score on semantic representation. See text for details.

nition of decision boundary with IND intents but also weakens the ability of the model to generate correct labels.

To alleviate the Inherent Label Uncertainty, we extend the label space $Y$ in the dataset and provide multiple candidate labels for each intent. We propose utilizing the emerging generative capacity of large generative language models such as GPT-3 or ChatGPT (used in this paper) to expand labels. For a specific label $y \in Y$ in the training set, we use crafted template $\mathcal{T}$ followed by a certain amount of randomly selected samples $x_{1:n}$ in this category to prompt model $\mathcal{F}$ to expand the label. The extended label space can be denoted as $\mathcal{Y} = \mathcal{F}(\mathcal{T}(y), x_{1:n})$. The process of extension is shown in Figure 2(b).

Unlike the previous work of generating training samples using large models, compared with the number of samples required for training, the number of labels to be expended can be almost negligible. Therefore, our method is extremely efficient and can obtain labels of higher quality than human annotations with the help of general knowledge of large models.

**The Loss Function of OID** Considering the existence of Inherent Label Uncertainty and the waste of generating labels for a large number of IND samples, it is not the best choice to directly use the generated labels for OID (See Appendix B for more discussion). We adopt the previous OID paradigm to detect through the learned discriminative representation of samples. For the semantic representation $z$ of the input $x$, it can be obtained by averaging the hidden vectors outputted by the last layer of the model (decoder-only PTMs, GPT-2 (Radford et al., 2019)) or averaging the hidden vectors outputted by the encoder (encoder-decoder PTMs, BART (Lewis et al., 2020), T5 (Raffel et al., 2020)), which is shown in Figure 3. With the original label space $Y$, a head for the OID task can be trained by cross-entropy loss:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\phi_{y_i}(z_i))}{\sum_{k \in [K]} \exp(\phi_k(z_i))}, \quad (2)$$

where $y_i$ is the gold label for input $x_i$, $\phi$ is a linear classifier and $K$ is the number of IND classes. For each sample, there is also extended label space $\mathcal{Y}$, which can help learn the discriminative representation. Inspired by the Multi-label research, we introduce an additional loss suggested in (Su, 2020) for a specific input $x$ with $\mathcal{Y}$:

$$\mathcal{L}_{\text{ex}}(x) = \log(1 + \sum_{i \in \overline{\Omega}, j \in \Omega} \exp(\phi_i(z) - \phi_j(z))), \quad (3)$$

where $z$ is the representation of input $x$, $\Omega$ is the extended set of the label of $x$, $\overline{\Omega} = \mathcal{Y} - \Omega$ is the set of remaining classes and $\phi_j(z)$ denotes the logit score of the j-th class. Intuitively, the purpose of Eq.(3) is to make the score of each extended class no less than that of each other class, so that the learned representation can be more discriminative. So far, we can train the OID-specific head by the following loss:

$$\mathcal{L}_{\text{OID}} = (1 - \alpha) \cdot \mathcal{L}_{\text{ce}} + \alpha \cdot \mathcal{L}_{\text{ex}}, \quad (4)$$

where $\alpha$ is a hyper-parameter and $\mathcal{L}_{\text{ex}}$ is calculated by $\frac{1}{|X|}\sum_{x \in X} \mathcal{L}_{\text{ex}}(x)$.

After obtaining the representation, in order not to rely on any assumptions or prior knowledge, we perform detection following Zhang et al. (2021c). First, determine a decision boundary in the representation space for each known intent. Those samples falling into the boundary are considered as the intent, and those not within any decision boundary are OOD.

**The Loss Funciton of NID** The whole NID consists of two parts. First, generate labels for the samples identified as OOD, and cluster according to the label similarity to discover general intents. For label generation, we adopt the standard language modeling objective to decode:

$$\mathcal{L}_{\text{NID}} = -\alpha(x) \sum_{(x,u\in\pi(y))} \sum_{j=1}^{|u|} p(u_j | u_{i<j}, \mathcal{T}(x)), \tag{5}$$

where $D$ is the training data, $(x, y)$ is a pair in $D$, $\pi(y)$ is a set just containing extended labels (not original labels), $\mathcal{T}$ is the template of prompts and $p$ is the conditional probability calculated by *softmax* function, whose input is the hidden vector output at the corresponding position of the last layer of the decoder and output is the probability of token $u_j$. The $\alpha(x)$ is set as $1/N_{\pi(y)}$.

Combined with Eq.(4) and Eq.(5), the finetune optimization objective are as:

$$\mathcal{L}_{\text{OBJ}} = (1-\lambda) \cdot \mathcal{L}_{\text{OID}} + \lambda \cdot \mathcal{L}_{\text{NID}}, \tag{6}$$

where $\lambda$ is a hyper-parameter to balance the loss of two tasks.

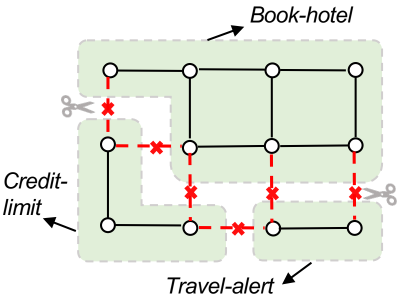


Figure 4: Illustration (inspired by Abbas and Swoboda (2022)) of formulating new intent discovery into minimum cost multi-cut. A graph is automatically cut into three segmentations (green), representing the general intents of *Book-hotel*, *Credit-limit* and *Travel-alert*. The red dotted line indicates a (minimum cost) multi-cut.

Since multiple similar labels can be generated for the same intent, to discover a more general new intent, samples with labels belonging to the same intent should be divided into one group as an intent set. To this end, we establish a weighted association network (graph) with nodes as samples and the weights of edges as the similarity (ROUGE (Lin, 2004) adopted in this paper, see Appendix A for details and more discussion) between labels of the linked samples. We reformulate the new intent discovery as a minimum cost *Multi-cut* problem on a graph. Samples belonging to the same intent will be automatically divided into the same cluster due to the high label similarity (see Figure 4), and thus do not rely on any prior about OOD.

For a specific weighted association graph $G = (V, E, W)$, a multi-cut refers to a subset of edges dividing the graph into distinct clusters, which satisfies the following constraints:

$$\text{P} := \{p(V_1, \ldots, V_n) | \bigcup_{i}^{n} V_i = V; V_i \cap V_j = \emptyset\}, \tag{7}$$

where $V_i$ is a node set, $P$ is the space of all multi-cut and $p$ is a specific cut.

The minimum cost multi-cut problem takes the weight of the edge $W \in R^{E\times E}$ into account. Intuitively, a greater weight of an edge $(u, v) \in E$ suggests a higher likelihood that $u$ and $v$ are in the same cluster and more cost is needed to remove the edge. The minimum cost muli-cut is to find a cut with the lowest cost, which can be defined as $\min_{p\in P}$ <W, p> suggested in Abbas and Swoboda (2022). In this paper, we find the minimum cost muli-cut by the implementation in Abbas and Swoboda (2022), which is an algorithm that can be run in GPU. See Appendix A for more discussion.

## 4 Experiments

### 4.1 Evaluation Datasets and Backbones

We conduct extensive experiments across two challenging real-world datasets and three widely used generative models.

**CLINC** (Larson et al., 2019) This is a widely studied intent dataset, which covers a wide range of intent categories. Specifically, This dataset includes 150 classes distributed across 10 different domains, consisting of 22500 utterances totally.

**BANKING** (Casanueva et al., 2020) This is a dataset related to the banking business, which is notable for its imbalanced distribution of samples across different categories. The dataset includes 77 intents, consisting of 9003 training samples and 3080 test samples. Appendix C summarizes detailed statistics of each dataset.

To verify the generality and effectiveness of our proposed method, we set up benchmarks on the widely used generative models across various architectures, i.e., autoregressive language model (decoder-only, **GPT-2** (Radford et al., 2019)), and encoder-decoder architecture (**BART** (Lewis et al., 2020), **T5** (Raffel et al., 2020)), and make a comprehensive comparison with our proposed method.

## 4.2 Evaluation Protocol and Baselines

We follow the generally accepted metrics used in the previous work of OID and NID tasks. In the task of OID, as suggested in Zhang et al. (2021b); Zhou et al. (2022a), we calculate macro F1-score for IND and OOD classes donated as **F1-IND** and **F1-OOD** respectively. Calculate accuracy score (**ACC-ALL**) and F1-score (**F1-ALL**) on all classes meanwhile.

For the task of NID, following Zhang et al. (2021c, 2022), we adopt the two metrics: Adjusted Mutual Information (**AMI**) and Adjusted Rand Index (**ARI**), to measure the quality of clustering (new intents found). In particular, we use the Hungarian algorithm (consistent with the previous methods) to align predicted classes and gold classes to calculate Accuracy (**ACC**). Finally, we calculate the macro average (**AVG.**) of these metrics to comprehensively measure the performance of different methods.

Based on the above evaluation metrics of different tasks, we use two baseline methods (**Model Tuning** and **Prefix Tuning**) to establish comparable benchmarks on the above datasets. Model Tuning refers to fine-tuning the full parameters of models. Prefix-tuning is a variation based on Li and Liang (2021) and our method is introduced in Section 3. In particular, the cluster-based method is a common method in the NID field, so we also made a comparison with it. We perform **K-means** with the representations identified as OOD to discover new intents as Zhang et al. (2021c, 2022) do.

## 4.3 Experimental Setting

Following the general setting in OID and NID tasks, we randomly select 75% of the intent classes given in the dataset as known intents (IND intents), and the rest are regarded as unknown intents (OOD intents). The OOD samples in training and validation sets are discarded. In the OID task, the disposed of classes in the test set are grouped into one rejected class (remarked as OOD), while in the NID task, the disposed of labels are retained in the test set to evaluate the quality of the predicted new class.

The details about the used models and hyperparameters are listed in Appendix D. Baselines and our method use the same experimental settings. Whether it is the main experiment or the analysis experiments, we use multiple different random seeds to conduct at least three rounds of experiments and report the average results.

## 4.4 Main Results

The comparison results of our methods and baselines across different generative models and datasets are shown in Table 1 (See Appendix D for the statistics of experimental parameters and the standard deviation). On the whole, our method obtain substantial improvements across various metrics in different datasets compared with baselines, which shows that our method can not only distinguish IND and OOD better but also better further distinguish OOD in fine granularity.

A closer look at Table 1, for the OOD Intent Detection, it can be observed from the table that BART and T5 are better than GPT-2 on the whole, and T5 performs better than BART on the CLINC dataset, but the opposite is true on the BANKING dataset. Interestingly, we observe that the effect of Prefix-tuning is better than that of Model-tuning, especially in the BANKING dataset, which shows that overfitting not only affects the generation of labels but also affects the learning of representations. Furthermore, our method is better than Prefix-tuning, which shows that expended labels and prompts can help to learn discriminative representations.

Further observation of the comparison results on the New Intent Discovery task shows that the results of intent discovery based on label similarity are better than those based on cluster-based (K-means), reflecting the advantages of our proposed method. The comparison between different models shows that T5 performs better than other models in different datasets (across different training methods), which relies on the excellent generation ability of T5. The Prefix-based training methods are better than the Model-tuning, which shows that the Prefix-based training method can well alleviate the generated labels overfitting to the labels in the training set and is also in line with our expectations. At the same time, by comparing our method with Prefix-tuning, we can further show that prompts and extended labels can help the model generate

| Model | Methods | CLINC | | | | BANKING | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OOD Intent Detection | | | | | | | |
| | | F1-ALL | ACC-ALL | F1-OOD | F1-IND | F1-ALL | ACC-ALL | F1-OOD | F1-IND |
| **GPT-2** | Model-tuning | 86.83 | 81.39 | 66.49 | 87.01 | 81.75 | 75.73 | 57.23 | 82.17 |
| | Prefix-tuning | 91.61 | 88.47 | 80.11 | 91.72 | 86.06 | 81.92 | 70.44 | 86.33 |
| | *Ours* | **92.69** | **89.44** | **80.68** | **92.80** | **86.93** | **82.57** | **70.75** | **87.21** |
| **BART** | Model-tuning | 93.55 | 90.50 | 82.25 | 93.65 | 87.62 | 82.77 | 66.95 | 87.98 |
| | Prefix-tuning | 93.94 | 90.90 | 82.66 | 94.04 | 87.94 | **83.88** | 72.24 | 88.21 |
| | *Ours* | **94.21** | **91.33** | **83.57** | **94.30** | **88.00** | 83.83 | **72.40** | **88.27** |
| **T5** | Model-tuning | 93.04 | 90.13 | 82.18 | 93.13 | 86.71 | 82.16 | 69.81 | 87.00 |
| | Prefix-tuning | 93.05 | 90.33 | 83.02 | 93.14 | 87.11 | 82.90 | 71.43 | 87.38 |
| | *Ours* | **94.52** | **91.74** | **84.36** | **94.61** | **87.85** | **83.63** | **72.13** | **88.11** |

| Model | Methods | New Intent Discovery | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | ACC | ARI | AMI | AVG. | ACC | ARI | AMI | AVG. |
| **GPT-2** | K-means | 28.49 | 6.22 | 12.79 | 15.83 | 21.58 | 6.75 | 16.46 | 14.93 |
| | Model-tuning | 25.13 | 8.40 | 26.95 | 20.16 | 26.21 | 11.15 | 32.28 | 23.21 |
| | Prefix-tuning | 32.86 | 16.34 | 32.48 | 27.23 | 27.10 | 13.71 | 29.68 | 23.49 |
| | *Ours* | **36.30** | **18.25** | **34.15** | **29.56** | **29.54** | **16.88** | **34.33** | **26.91** |
| **BART** | K-means | 30.81 | 14.32 | 29.86 | 25.00 | 31.61 | 19.16 | 41.76 | 30.84 |
| | Model-tuning | 28.52 | 14.10 | 41.30 | 27.97 | 35.53 | 21.18 | 42.08 | 32.92 |
| | Prefix-tuning | 35.72 | 18.76 | 32.76 | 29.08 | 36.38 | 23.11 | 42.11 | 33.86 |
| | *Ours* | **39.57** | **23.99** | **45.29** | **36.28** | **36.77** | **23.42** | **43.41** | **34.53** |
| **T5** | K-means | 33.98 | 19.36 | 36.33 | 29.88 | 33.61 | 26.37 | 50.76 | 36.91 |
| | Model-tuning | 42.17 | 25.51 | 51.13 | 39.61 | 32.27 | 21.22 | 45.01 | 32.83 |
| | Prefix-tuning | 47.96 | 33.22 | 50.42 | 43.87 | 37.82 | 24.56 | 45.01 | 35.80 |
| | *Ours* | **48.78** | **35.61** | **53.06** | **45.82** | **41.51** | **29.43** | **50.77** | **40.57** |

Table 1: Overall comparison results across various models and datasets. The upper part is the comparison result of the OID task, the lower part is the result of the NID. All exhibited results are percentages and the average of the results over different random seeds. See Appendix D for the standard deviation and text for details.

better labels (Appendix A).

## 5 Analysis

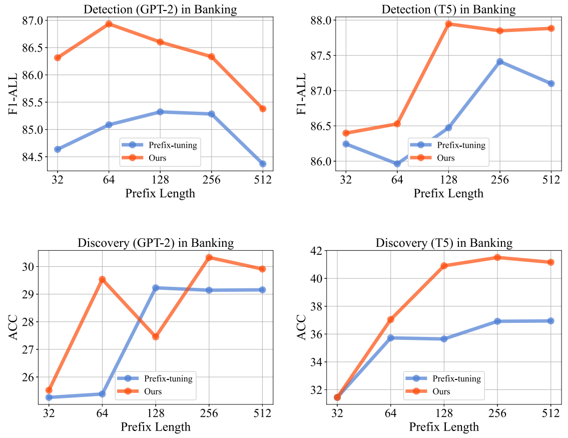


Figure 5: Plots show the impact of the length of the prefix. Our method (Red) is also better than Prefix-tuning (Blue) under various prefix lengths.

### 5.1 Impact of Prefix Length

In this section, we explore the specific impact of the length of the prefix. From Figure 5, we can observe that both tasks seem to be sensitive to the length of the prefix. A small prefix will not play its advantages. Further, the performance of *Detection* may decline with the increase of prefix length (especially for GPT-2). For the *Discovery*, similar phenomena will be observed, but the downward trend will be postponed. The above phenomenon may be attributed to the fact that the increase of the prefix length brings more fine-tuned parameters, which results in the model shifting toward the limited IND data, which not only weakens the ability to generate labels but also affects the learning of discriminative representations. Under various prefix lengths (other parameters remain the same), Our method is better than Prefix-tuning.

### 5.2 Towards a Win-win Training

We adopt hyper-parameter $\lambda$ to balance the losses of two tasks in Eq.(6) during training. In this sec-

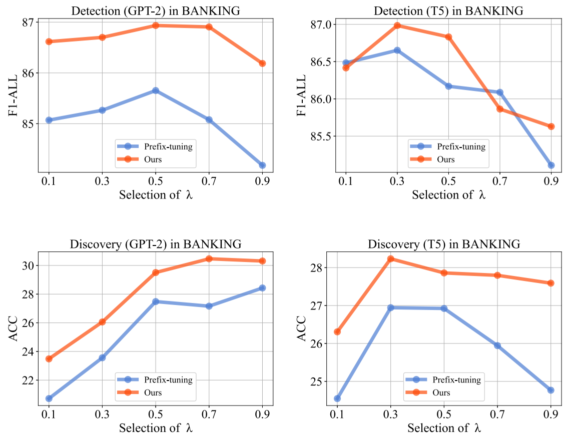


Figure 6: Plots show the effect of $\lambda$. The $\lambda$ can balance different losses to make the model achieve satisfactory performance in the Open Environment Intent Prediction, and our method (Red) is better than the baseline (Blue) under different settings.

tion, we evaluate the benefits of $\lambda$ in the training process. Specifically, we vary the value to obtain the changing trend of the performance of two tasks, and the results are shown in Figure 6. When the $\lambda$ at around 0.5, the two tasks can achieve a win-win situation across the different models and different training methods, which demonstrates the rationality of extending *Classification* with *Discovery*. In addition, by varying the value of the $\lambda$ while keeping other parameters unchanged, our method is always better than the baseline method. See Appendix B for more related discussion.

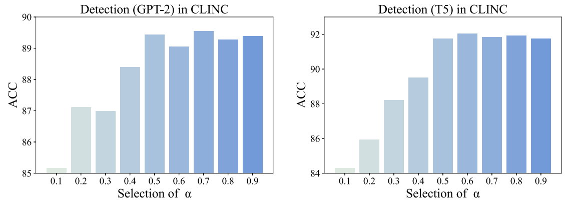


Figure 7: Effect of extended labels in detection. The extended labels can help different models GPT-2 (left) and T5 (right) better distinguish IND and OOD.

### 5.3 Effect of Extended Labels

In this section, we explore the effect of extended labels in the Open Environment Intent Prediction. The extended labels affect the *Detection* in the form of $L_{ex}$ in Eq.(3). By varying $\alpha$ in Eq.(4), we can observe the effect of extended labels. The results are shown in Figure 7, which shows that with the increase of weight $\alpha$, the model can learn better discriminative representations (F1-ALL rises), but if the $\alpha$ continues to increase, the recognition accuracy may be affected due to the influence of uncertainty between labels. More experiments in Appendix B can demonstrate its effect is general. For the *Discovery*, it has been proved that extended labels can alleviate the degradation of the generated vocabulary(Section 3) and help to discover new intents(Section 4.4). We also evaluate the quality of labels generated with the help of extended labels in Appendix A.

| Template | GPT-2 |  | T5 |  |
|---|---|---|---|---|
|  | Dete.<br>(F1-ALL) | Disc.<br>(ACC) | Dete.<br>(F1-ALL) | Disc.<br>(ACC) |
| <x>. (w/o template) | 86.29 | 25.91 | 87.65 | 35.05 |
| ($*$) <x>.*It was [Mask].* | **86.65** | **26.77** | **88.15** | **36.11** |
| (†) <x>.*Refer to [Mask].* | **86.68** | **28.80** | **87.86** | **37.80** |
| (†) <x>.*This is [Mask].* | **86.40** | **30.15** | **87.89** | **37.69** |

Table 2: Effect of prompts. $*$ is a crafted template and † represents the templates are generated automatically.

### 5.4 Necessity of Prompts

To steer the model to generate high-quality labels, we task the model with natural language prompts in the input. In this section, we explore the specific effect of prompts. The experimental results in BANKING are listed in Table 2, where the input in the first row is without prompt and the inputs in the following three rows are with templates generated in different ways. From Table 2, it can be observed that the existence of prompts can not only help with detection (Dete.) but also has an obvious effect on new intent discovery (Disc.). At the same time, the help of prompts is general. In addition to manual design, we also try to automatically generate templates based on Gao et al. (2021a) (Appendix E). Compared with only inputting utterances to the model, formulating input with these generated templates $\mathcal{T}(X)$ shows a certain degree of help.

## 6 Conclusion

In this paper, we strengthen a combined generative task paradigm to expand the two basic tasks of the Task-Oriented Dialogue system, which is more general and practical. Further, without relying on prior knowledge about OOD, we provide an effective and efficient implementation based on the generative model. At the same time, we introduce an effective method of intent expansion to alleviate Inherent Label Uncertainty and provide a method for constructing multi-label intent datasets to inspire further research. Extensive experiments across different models and different datasets have verified effectiveness and generality.

## Limitations

To better enlighten the follow-up research, we conclude the limitations of our method as follows:
1) Although the method we proposed can help improve the quality of generated labels, there is still room for further improvement; 2) Because our detection is not perfect, it will lead to inaccurate labels of some samples. We look forward to better methods to improve detection in the future; 3) This work has verified that the extended labels can effectively help the performance of models and proposed a method of label extension, but has not tried other extension methods or whether it is helpful to extend more labels. 4) This work focuses on solving Open Environment Intent Prediction with different generative models, without exploring other types of models.

## Acknowledgements

We thank Dr.Guo qipeng for his patient and valuable feedback in this work. This work was supported by the National Key Research and Development Program of China (No.2022CSJGG0801), National Natural Science Foundation of China (No.62022027) and CAAI-Huawei MindSpore Open Fund.

## References

Ahmed Abbas and Paul Swoboda. 2022. Rama: A rapid multicut algorithm on gpu. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8193–8202.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Xibin Gao, Radhika Arava, Qian Hu, Thahir Mohamed, Wei Xiao, Zheng Gao, and Mohamed AbdelHady. 2021b. Graphire: Novel intent discovery with pretraining on prior knowledge using contrastive learning. In *KDD 2021 Workshop on Pretraining: Algorithms, Architectures, and Applications*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Hao Lang, Yinhe Zheng, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022. Estimating soft labels for out-of-domain intent detection. *CoRR*, abs/2211.05561.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 1311–1316. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

*International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Jianlin Su. 2020. Extend softmax and multi-label cross entropy to multi-label classification.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2022. Generalized category discovery. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Nikhita Vedula, Rahul Gupta, Aman Alok, and Mukund Sridhar. 2020. Automatic discovery of novel intents & domains from text utterances. *CoRR*, abs/2006.01208.

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. 2022. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*.

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, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 870–878. Association for Computational Linguistics.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert Y. S. Lam. 2021. Out-of-scope intent detection with self-supervision and discriminative training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3521–3532. Association for Computational Linguistics.

Hanlei Zhang, Xiaoteng Li, Hua Xu, Panpan Zhang, Kang Zhao, and Kai Gao. 2021a. TEXTOIR: An integrated and visualized platform for text open intent recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 167–174, Online. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021b. 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 Lyu. 2021c. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14365–14373.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2022. New intent discovery with pre-training and contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 256–269, Dublin, Ireland. Association for Computational Linguistics.

J. Zheng, W. Li, J. Hong, L. Petersson, and N. Barnes. 2022. Towards open-set object detection and discovery. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3960–3969, Los Alamitos, CA, USA. IEEE Computer Society.

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.

Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 2022a. KNN-contrastive learning for out-of-domain intent classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5129–5141, Dublin, Ireland. Association for Computational Linguistics.

Yunhua Zhou, Peiju Liu, Yuxin Wang, and Xipeng QIu. 2022b. Discovering new intents using latent variables. *arXiv preprint arXiv:2210.11804*.

| Methods | GPT-2 | | | T5 | | |
|---|---|---|---|---|---|---|
| | AMI | ARI | ACC | AMI | ARI | ACC |
| | **GloVe** (Pennington et al., 2014) | | | | | |
| Prefix-tuning | 31.77 | 13.96 | **28.65** | 46.77 | **24.21** | 36.87 |
| *Ours* | **33.94** | **14.14** | 28.30 | **49.00** | 23.71 | **38.06** |
| | **BERTScore** (Zhang et al., 2020) | | | | | |
| Prefix-tuning | 27.20 | 9.57 | 24.58 | 47.47 | 25.81 | 39.50 |
| *Ours* | **33.48** | **11.76** | **27.10** | **50.62** | **28.23** | **40.12** |
| | **ROUGE** (Lin, 2004) | | | | | |
| Prefix-tuning | 28.20 | 11.66 | 25.87 | 46.47 | 25.85 | 39.20 |
| *Ours* | **34.33** | **16.88** | **29.54** | **48.97** | **27.95** | **41.53** |

Table 3: Results of the New Intent Discovery by labels generated in different ways in the BANKING dataset. Under different similarity measures, our method has achieved better results, which reflects that our methods can generate better labels.

## A More Discussion on Generated Labels and New Intent Discovery

In this section, we evaluate the quality of generated labels. Because we discover general intents based on generated labels (Section 3), the better effect of intent discovery suggests the better quality of generated labels. In this paper, For efficiency and effect, we use **ROGUE** (Lin, 2004) to measure the similarity between two labels. Specifically, we calculate the average of the ROGUE-1, ROGUE-2, and ROGUE-L[3] F1-scores of two labels as the similarity score. In addition, for the sake of generality, we try two additional widely-used similarity measures: **GloVe** (Pennington et al., 2014) and **BERTScore** (Zhang et al., 2020). We use labels generated in different ways to discover intents and compare the effects in Table 3. Under different similarity measures, our methods have achieved better results, which shows that our methods can generate better labels.

At the same time, it should be emphasized that the scheme we proposed for new intent discovery based on generated labels in Section 3 is a general framework that can be flexibly implemented. In addition to the way to establish graphs described in Section 3, we can also build a weighted association graph with labels as nodes, whose edges are the similarity between linked labels, then perform minimum cost multi-cut on this graph, where the segmentations (composed by similar labels) divided are also regarded as more general intents, and the whole process also does not depend on any prior or assumptions about OOD. We leave more and broader exploration for future research.

For the label for discovered general intents, you can either pick the label with the highest frequency in the corresponding segmentation as the label of the intent, or the labels with the top $k$ highest frequency, which depends on the purpose of using the data.

| Methods | CLINC | | | |
|---|---|---|---|---|
| | F1-ALL | ACC-ALL | F1-OOD | F1-IND |
| Cluster-based | 76.22 | 72.96 | 64.66 | 76.33 |
| Detection-based | 93.79 | 90.96 | 83.31 | 93.88 |
| +Expended labels | 94.47 | 91.67 | 84.27 | 94.57 |
| Label-based | 71.25 | 67.76 | 62.40 | 71.33 |
| +Expended labels | 73.68 | 72.30 | 62.55 | 73.78 |
| *Ours* | **94.68** | **92.07** | **85.07** | **94.77** |

Table 4: Comparison results of different paradigms of detection. The results are obtained with T5 on CLINC dataset.

## B More Comprehensive Comparison of Detection

As mentioned in Section 3, considering the existence of Inherent Label Uncertainty and the waste of generating labels (or clusters) for a large number of IND samples, we conduct the OID task based on the learned representation. To learn the discriminative representations, we enrich the expression of intent by multi labels and train together with the loss of generation Eq.(6) during training (The effectiveness is proved in Section 5.2).

In this section, in order to further verify the effectiveness of our method, we make a comprehensive comparison with various paradigms. **Cluster-based** refers to the paradigm adopted by previous work in the NID (Zhang et al., 2021c, 2022), all samples are directly clustered by **K-means** for intent discovery after learning representation, **Detection-based** means that only the OID loss $\mathcal{L}_{OID}$ ($\lambda = 0.0$ in Eq.(6)) is used for training to obtain representations of samples, which is a paradigm in the OID task (Zhang et al., 2021b), and **Label-based** means that only the NID loss $\mathcal{L}_{NID}$ ($\lambda = 1.0$ in Eq.(6)) is used for training then

[3] https://pypi.org/project/rouge/

discovery intents based on labels (same as that we used in 3). The experimental parameters of all methods are consistent.

We show the comparison results of different methods in Table 4, which demonstrates our method is superior to other methods. In addition, many meaningful observations can be obtained from the table. The introduction of extended labels can improve the effect of detection under different paradigms, which reflects the generality. The effect of the Cluster-based method is significantly lower than that of representation-based detection, which also shows that the previous paradigms in the NID not only waste a lot of costs to cluster IND samples but also may have very limited effect. The above comparison results can fully demonstrate the rationality and effectiveness of our method.

## C Statistics of Datasets

The detailed statistics of the datasets described in the Section 4.1 are summarized in Table 5.

## D Details of the Models and Hyper-parameters

In this paper, experiments are conducted on models with different architectures, i.e., decoder-only (GPT-2 (Radford et al., 2019)), and encoder-decoder architecture (BART (Lewis et al., 2020), T5 (Raffel et al., 2020)), whose details are shown in 6. The implementations of GPT-2[4], BART[5] and T5[6] are based on the Huggingface Transformer models. We tried learning rate in {1e-4, 2e-4, 3e-4, 4e-4}, training batch size in {64,128}, the length of tunable prefix in {64,128,256} and trained 100 epochs with an AdamW optimizer. We utilized four extended labels for each intent during the experiment (In fact, for certain intents, the number of labels was expanded to five.). In the **K-means** setting, we set $k$ to three times the ground truth number of intent categories. Baselines and our method use the same experimental settings. Whether it is the main experiment or the analysis experiments, we use multiple different random seeds to conduct multiple rounds of experiments and report the average results. We list the standard deviation of the main experiment results (Table 1) in Table 7. Our experiments are conducted on a single NVIDIA A100 Tensor Core GPU. We also have tried to conduct experiments on a single NVIDIA GTX 3090 with small batchsizes.

## E Automatic Generation of Templates

To verify the generality of the benefits of the prompts, in addition to manually designing templates, we use **T5** to automatically generate models based on Gao et al. (2021a). The difference is that to maintain the semantics of labels, we have not pruned the generated vocabulary set. To generate templates, we formalize the input $(x, y) \in \mathcal{D}_{train}$ to **T5** as $x$.<s1>$y$<s2> (The <s1> and <s2> are the mask tokens) and let T5 automatically fill in <s1> and <s2> (i.e., templates) during decoding. We select the templates with higher beam search scores as the candidates then use $D_{dev}$ to pick templates with better performance. See Gao et al. (2021a) for more details.

[4] https://huggingface.co/gpt2

[5] https://huggingface.co/facebook/bart-base

[6] https://huggingface.co/t5-base

| Dataset | Classes | \|Training\| | \|Validation\| | \|Test\| | Vocabulary | Length (Avg.) |
|---|---|---|---|---|---|---|
| CLINC-FULL (Larson et al., 2019) | 150 | 18000 | 2250 | 2250 | 7283 | 8.32 |
| BANKING (Casanueva et al., 2020) | 77 | 9003 | 1000 | 3080 | 5028 | 11.91 |

Table 5: Statistics of CLINC-FULL, BANKING datasets. || denotes the total number of utterances. Length indicates the average length of each utterance in the dataset. The vocabulary is drawn from (Zhang et al., 2021c)

| Model | Magnitude | Encoder | Decoder | DIM.(hidden) | Parameters |
|---|---|---|---|---|---|
| GPT-2 (Radford et al., 2019) | Base | / | 12-layer | 768 | 117M |
| BART (Lewis et al., 2020) | Base | 6-layer | 6-layer | 768 | 139M |
| T5 (Raffel et al., 2020) | Base | 12-layer | 12-layer | 768 | 220M |

Table 6: Details of the model adopted in this paper. Dim.(hidden) refers to the dimension of the hidden vector.

| Model | Methods | CLINC | | | | BANKING | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OOD Intent Detection | | | | | | | |
| | | F1-ALL | ACC-ALL | F1-OOD | F1-IND | F1-ALL | ACC-ALL | F1-OOD | F1-IND |
| **GPT-2** | Model-tuning | 0.99 | 1.11 | 1.47 | 0.99 | 1.48 | 1.60 | 2.97 | 1.47 |
| | Prefix-tuning | 0.98 | 1.52 | 2.66 | 0.97 | 0.42 | 0.67 | 1.91 | 0.41 |
| | *Ours* | 0.60 | 0.92 | 1.75 | 0.59 | 0.10 | 0.24 | 0.85 | 0.09 |
| **BART** | Model-tuning | 0.33 | 0.78 | 1.77 | 0.31 | 0.79 | 1.44 | 4.52 | 0.74 |
| | Prefix-tuning | 0.10 | 0.33 | 0.85 | 0.10 | 0.96 | 1.73 | 3.92 | 0.91 |
| | *Ours* | 0.59 | 1.08 | 2.27 | 0.58 | 0.55 | 0.79 | 1.65 | 0.53 |
| **T5** | Model-tuning | 0.28 | 0.69 | 1.65 | 0.27 | 0.77 | 0.90 | 2.04 | 0.75 |
| | Prefix-tuning | 0.36 | 0.18 | 0.58 | 0.36 | 0.23 | 0.64 | 2.53 | 0.23 |
| | *Ours* | 0.51 | 0.91 | 1.91 | 0.50 | 0.61 | 1.14 | 2.82 | 0.57 |
| **Model** | **Methods** | **New Intent Discovery** | | | | | | | |
| | | ACC | ARI | AMI | - | ACC | ARI | AMI | - |
| **GPT-2** | K-means | 0.34 | 0.79 | 1.46 | - | 0.69 | 0.90 | 1.46 | - |
| | Model-tuning | 0.54 | 1.19 | 0.80 | - | 2.28 | 2.91 | 1.54 | - |
| | Prefix-tuning | 1.74 | 1.32 | 1.81 | - | 0.47 | 2.06 | 2.86 | - |
| | *Ours* | 4.24 | 3.42 | 5.07 | - | 1.21 | 2.19 | 1.83 | - |
| **BART** | K-means | 1.87 | 2.65 | 3.05 | - | 1.87 | 3.54 | 4.78 | - |
| | Model-tuning | 1.61 | 2.49 | 1.89 | - | 3.34 | 2.76 | 2.10 | - |
| | Prefix-tuning | 6.67 | 9.69 | 12.69 | - | 4.32 | 3.71 | 3.79 | - |
| | *Ours* | 3.23 | 3.83 | 5.19 | - | 1.26 | 1.37 | 1.19 | - |
| **T5** | K-means | 1.97 | 2.99 | 3.15 | - | 2.20 | 2.29 | 2.49 | - |
| | Model-tuning | 3.02 | 4.09 | 2.49 | - | 1.32 | 1.02 | 0.66 | - |
| | Prefix-tuning | 0.69 | 2.40 | 1.14 | - | 2.25 | 2.74 | 1.54 | - |
| | *Ours* | 4.38 | 4.75 | 3.59 | - | 3.07 | 2.39 | 1.71 | - |

Table 7: The standard deviation corresponding to each mean result in Table 1.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Section "Limitations" (7th Section)*

☑ A2. Did you discuss any potential risks of your work?
*Section "Limitations" (7th Section)*

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

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

### B ☑ Did you use or create scientific artifacts?

*Section 4.1 and Appendix C*

☑ B1. Did you cite the creators of artifacts you used?
*Section 4.1*

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*These datasets are available for all researchers in the NLP community.*

☒ 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)?
*These datasets are only for scientific research and are available for all members of the NLP research community. We have adhered to the typical method of utilizing these resources.*

☒ 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?
*These datasets are only for scientific research and are available for all members of the NLP research community. We have adhered to the typical method of utilizing these resources.*

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

☑ 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.
*Appendix C*

### C ☑ Did you run computational experiments?

*Section 4*

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

*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?
*Section 4.3 and Appendix D*

☑ 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 and Appendix D*

☑ 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 A and Appendix D*

**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.*