# Deep Open Intent Classification with Adaptive Decision Boundary

**Hanlei Zhang,**[1, 2] **Hua Xu,**[1, 2*] **Ting-En Lin**[1, 2]
[1]State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China,
[2] Beijing National Research Center for Information Science and Technology (BNRist), Beijing 100084, China
zhang-hl20@mails.tsinghua.edu.cn, xuhua@tsinghua.edu.cn, ting-en.lte@alibaba-inc.com

## Abstract

Open intent classification is a challenging task in dialogue systems. On the one hand, it should ensure the quality of known intent identification. On the other hand, it needs to detect the open (unknown) intent without prior knowledge. Current models are limited in finding the appropriate decision boundary to balance the performances of both known intents and the open intent. In this paper, we propose a post-processing method to learn the adaptive decision boundary (ADB) for open intent classification. We first utilize the labeled known intent samples to pre-train the model. Then, we automatically learn the adaptive spherical decision boundary for each known class with the aid of well-trained features. Specifically, we propose a new loss function to balance both the empirical risk and the open space risk. Our method does not need open intent samples and is free from modifying the model architecture. Moreover, our approach is surprisingly insensitive with less labeled data and fewer known intents. Extensive experiments on three benchmark datasets show that our method yields significant improvements compared with the state-of-the-art methods. The codes are released at https://github.com/thuiar/Adaptive-Decision-Boundary.

## Introduction

Identifying the user's open intent plays a significant role in dialogue systems. As shown in Figure 1, we have two known intents for specific purposes, such as book flight and restaurant reservation. However, there are also utterances with irrelevant or unsupported intents that our system cannot handle. It is necessary to distinguish these utterances from the known intents as much as possible. On the one hand, effectively identifying the open intent can improve customer satisfaction by reducing false-positive error. On the other hand, we can use the open intent to discover potential user needs.

We regard open intent classification as an (n+1)-class classification task as suggested in (Shu, Xu, and Liu 2017; Lin and Xu 2019a), and group open classes into the (n+1)$^{\text{th}}$ class . Our goal is to classify the n-class known intents into their corresponding classes correctly while identifying the (n+1)$^{\text{th}}$ class open intent. To solve this problem, Scheirer et al. (2013) propose the concept of open space risk as the measure of open classification. Fei and Liu (2016) reduce the open space risk by learning the closed boundary of each positive class in the similarity space. However, they fail to capture high-level semantic concepts with SVM. Bendale and Boult (2016) manage to reduce the open space risk through deep neural networks (DNNs), but need to sample open classes for selecting the core hyperparameters. Hendrycks and Gimpel (2017) use the softmax probability as the confidence score, but also need to select the confidence threshold with negative samples. Shu, Xu, and Liu (2017) replace softmax with the sigmoid activation function, and calculate the confidence thresholds of each class based on statistics. However, the statistics-based thresholds can not learn the essential differences between known classes and the open class. Lin and Xu (2019a) propose to learn the deep intent features with the margin loss and detect the unknown intent with local outlier factor (Breunig et al. 2000). However, it has no specific decision boundaries for distinguishing the open intent, and needs model architecture modification.

| User utterances | Intent Label |
|---|---|
| Book a flight from LA to Madrid. | Book flight |
| Can you get me a table at Steve's? | Restaurant reservation |
| Book Delta ticket Madison to Atlanta. | Book flight |
| Schedule me a table at Red Lobster. | Restaurant reservation |
| ... | ... |
| Can you tell me the name of this song? | **Open** |
| Look up the calories in an apple. | **Open** |

Figure 1: An example of open intent classification. Book flight and Restaurant reservation are two known intents. We should identify them correctly while detecting the sentences with the open intent.

Most of the existing methods need to design specific classifiers for identifying the open class (Bendale and Boult 2016; Shu, Xu, and Liu 2017; Lin and Xu 2019a), and perform poorly with the common classifier (Hendrycks and Gimpel 2017). Moreover, the performance of open classi-

*Hua Xu is the corresponding author.

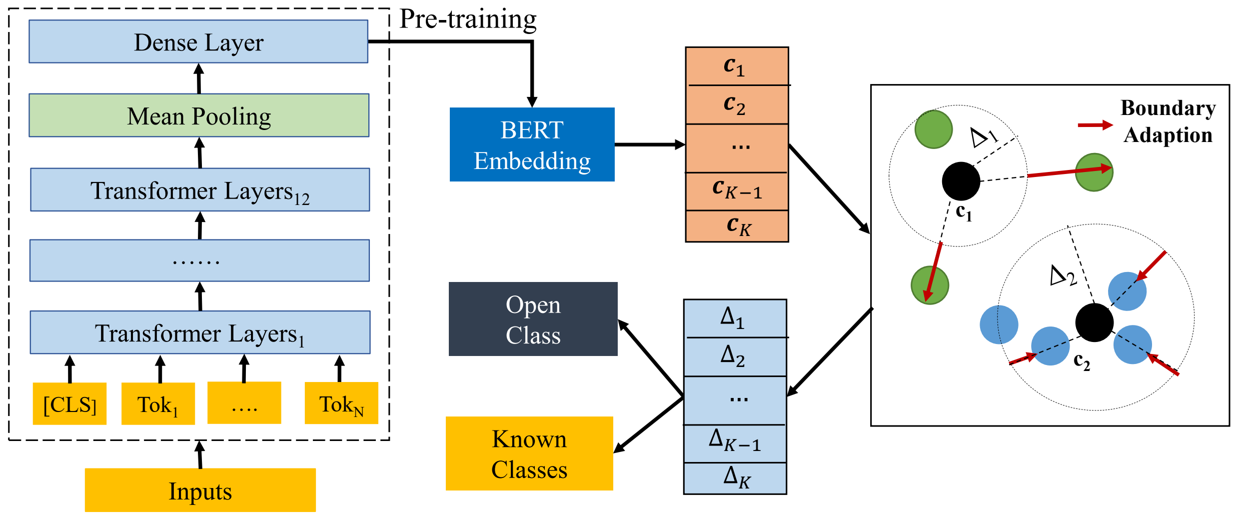


Figure 2: The model architecture of our approach. Firstly, we use BERT to extract intent features and pre-train the model with labeled samples. Then, we initialize the centroids $\{\boldsymbol{c}_i\}_{i=1}^K$ and the radius of decision boundaries $\{\Delta_i\}_{i=1}^K$ for each known class. Next, we propose the boundary loss to learn tight decision boundaries adaptive to the known intent feature space. Finally, we perform open classification with the learned decision boundaries to both identify known classes and detect the open class.

fication largely depends on the decision conditions. Most of these methods need negative samples for determining the suitable decision conditions (Scheirer et al. 2013; Fei and Liu 2016; Hendrycks and Gimpel 2017; Liang, Li, and Srikant 2018). It is also a complicated and time-consuming process to manually select the optimal decision condition, which is not applicable in real scenarios.

To solve these problems, we use known intents as prior knowledge, and propose a novel post-processing method to learn the adaptive decision boundary (ADB) for open intent classification. As illustrated in Figure 2, we first extract intent representations from BERT (Devlin et al. 2019). Then, we pre-train the model under the supervision of the softmax loss. We define centroids for each known class and suppose intent features of each known class are constrained in a closed ball area. Next, we aim to learn the radius of the spherical area to obtain the decision boundaries. Specifically, we initialize the boundary parameters with standard normal distribution and use a learnable activation function as a projection to get the radius of each decision boundary.

The suitable decision boundaries should satisfy two conditions. On the one hand, they should be broad enough to surround known intent samples as much as possible. On the other hand, they need to be tight enough to prevent the open intent samples from being identified as known intents. To address these issues, we propose a new loss function, which optimizes the boundary parameters by balancing both the open space risk and the empirical risk (Scheirer et al. 2013). The decision boundaries can automatically learn to adapt to the intent feature space with the boundary loss until balance. We find our post-processing method can learn discriminative decision boundaries for detecting the open intent even without modifying the original model architecture.

We summarize our contribution as follows. Firstly, we propose a novel post-processing method for open classification, with no need for prior knowledge of the open intent. Secondly, we propose a new loss function to automatically learn tight decision boundaries adaptive to the feature space. To the best of our knowledge, this is the first attempt to adopt deep neural networks to learn the adaptive decision boundaries for open classification. Thirdly, extensive experiments conducted on three challenging datasets show that our approach yields consistently better and more robust results compared with the state-of-the-art methods.

## The Proposed Approach

### Intent Representation

We use the BERT model to extract deep intent features. Given $i^{th}$ input sentence $\boldsymbol{s}_i$, we get all its token embeddings $[CLS, T_1, \cdots, T_N] \in \mathbb{R}^{(N+1)\times H}$ from the last hidden layer of BERT. As suggested in (Lin, Xu, and Zhang 2020), we perform mean-pooling on these token embeddings to synthesize the high-level semantic features in one sentence, and get the averaged representation $\boldsymbol{x}_i \in \mathbb{R}^H$:

$$\boldsymbol{x}_i = \text{mean-pooling}([CLS, T_1, \cdots, T_N]), \tag{1}$$

where $CLS$ is the vector for text classification, $N$ is the sequence length and $H$ is the hidden layer size. To further strengthen feature extraction capability, we feed $\boldsymbol{x}_i$ to a dense layer $h$ to get the intent representation $\boldsymbol{z}_i \in \mathbb{R}^D$:

$$\boldsymbol{z}_i = h(\boldsymbol{x}_i) = \sigma(W_h \boldsymbol{x}_i + b_h), \tag{2}$$

where $D$ is the dimension of the intent representation, $\sigma$ is a ReLU activation function, $W_h \in \mathbb{R}^{H\times D}$ and $b_h \in \mathbb{R}^D$ respectively denote the weights and the bias term of layer $h$.

| Dataset | Classes | #Training | #Validation | #Test | Vocabulary Size | Length (max / mean) |
|---|---|---|---|---|---|---|
| BANKING | 77 | 9,003 | 1,000 | 3,080 | 5,028 | 79 / 11.91 |
| OOS | 150 | 15,000 | 3,000 | 5,700 | 8,376 | 28 / 8.31 |
| StackOverflow | 20 | 12,000 | 2,000 | 6,000 | 17,182 | 41 / 9.18 |

Table 1: Statistics of BANKING, OOS and StackOverflow datasets. # indicates the total number of sentences.

## Pre-training

As the decision boundaries learn to adapt to the intent feature space, we need to learn intent representations at first. Due to lack of open intent samples, we use known intents as prior knowledge to pre-train the model. In order to reflect the effectiveness of the learned decision boundary, we use the simple softmax loss $\mathcal{L}_s$ to learn the intent feature $\boldsymbol{z}_i$:

$$\mathcal{L}_s = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\phi(\boldsymbol{z}_i)^{y_i})}{\sum_{j=1}^{K}\exp(\phi(\boldsymbol{z}_i)^{j})}, \tag{3}$$

where $\phi(\cdot)$ is a linear classifier and $\phi(\cdot)^j$ are the output logits of the $j^{th}$ class. Then, we use the pre-trained model to extract intent features for learning decision boundaries.

## Adaptive Decision Boundary Learning

In this section, we propose our approach to learning the adaptive decision boundary (ADB) for open intent classification. First, we introduce the formulation of the decision boundary. Then, we propose our boundary learning strategy for optimization. Finally, we use the learned decision boundary to perform open classification.

**Decision Boundary Formulation** It has been shown the superiority of the spherical shape boundary for open classification (Fei and Liu 2016). Compared with the half-space binary linear classifier (Schölkopf et al. 2001) or two parallel hyper-planes (Scheirer et al. 2013), the bounded spherical area greatly reduces the open space risk. Inspired by this, we aim to learn the decision boundary of each class constraining the known intents within a ball area.

Let $S = \{(\boldsymbol{z}_i, y_i), \ldots, (\boldsymbol{z}_N, y_N)\}$ be the known intent examples with their corresponding labels. $S_k$ denotes the set of examples labeled with class $k$. The centroid $\boldsymbol{c}_k \in \mathbb{R}^D$ is the mean vector of embedded samples in $S_k$:

$$\boldsymbol{c}_k = \frac{1}{|S_k|}\sum_{(\boldsymbol{z}_i, y_i)\in S_k}\boldsymbol{z}_i, \tag{4}$$

where $|S_k|$ denotes the number of examples in $S_k$. We define $\Delta_k$ as the radius of the decision boundary with respect to the centroid $\boldsymbol{c}_k$. For each known intent $\boldsymbol{z}_i$, we aim to satisfy the following constraints:

$$\forall \boldsymbol{z}_i \in S_k, \|\boldsymbol{z}_i - \boldsymbol{c}_k\|_2 \leq \Delta_k, \tag{5}$$

where $\|\boldsymbol{z}_i - \boldsymbol{c}_k\|_2$ denotes the Euclidean distance between $\boldsymbol{z}_i$ and $\boldsymbol{c}_k$. That is, we hope examples belonging to class $k$ are constrained in the ball area with centroid $\boldsymbol{c}_k$ and radius $\Delta_k$. As radius $\Delta_k$ needs to be adaptive to the intent feature space, we use the deep neural network to optimize the learnable boundary parameter $\widehat{\Delta_k} \in \mathbb{R}$. As suggested in (Tapaswi, Law, and Fidler 2019), we use $\mathrm{Softplus}$ activation function as the mapping between $\Delta_k$ and $\widehat{\Delta_k}$:

$$\Delta_k = \log\left(1 + e^{\widehat{\Delta_k}}\right). \tag{6}$$

The $\mathrm{Softplus}$ activation function has the following advantages. First, it is totally differentiable with different $\widehat{\Delta_k} \in \mathbb{R}$. Second, it can ensure the learned radius $\Delta_k$ is above zero. Finally, it achieves linear characteristics like ReLU and allows for bigger $\Delta_k$ if necessary.

**Boundary Learning** The decision boundaries should be adaptive to the intent feature space to balance both empirical and open space risk (Bendale and Boult 2015). For example, if $\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 > \Delta_{y_i}$, the known intent samples are outside their corresponding decision boundaries, which may introduce more empirical risk. Therefore, the decision boundaries need to expand to contain more samples from known classes. If $\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 < \Delta_{y_i}$, though more known intent samples are likely to be identified with broader decision boundaries, it may introduce more open intent samples and increase the open space risk. Thus, we propose the boundary loss $\mathcal{L}_b$:

$$\begin{aligned}\mathcal{L}_b = \frac{1}{N}\sum_{i=1}^{N}&\left[\delta_i\left(\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 - \Delta_{y_i}\right)\right.\\ &\left.+ \left(1-\delta_i\right)\left(\Delta_{y_i} - \|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2\right)\right],\end{aligned} \tag{7}$$

where $y_i$ is the label of the $i^{th}$ sample and $\delta_i$ is defined as:

$$\delta_i := \begin{cases} 1, \text{ if } & \|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 > \Delta_{y_i}, \\ 0, \text{ if } & \|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 \leq \Delta_{y_i}. \end{cases} \tag{8}$$

Then, we update the boundary parameter $\widehat{\Delta_k}$ regarding to $\mathcal{L}_b$ as follows:

$$\widehat{\Delta_k} := \widehat{\Delta_k} - \eta\frac{\partial\mathcal{L}_b}{\partial\widehat{\Delta_k}}, \tag{9}$$

where $\eta$ is the learning rate of the boundary parameters $\widehat{\Delta}$ and $\frac{\partial\mathcal{L}_b}{\partial\widehat{\Delta_k}}$ is computed by:

$$\frac{\partial\mathcal{L}_b}{\partial\widehat{\Delta_k}} = \frac{\sum_{i=1}^{N}\delta'\left(y_i = k\right)\cdot(-1)^{\delta_i}}{\sum_{i=1}^{N}\delta'\left(y_i = k\right)}\cdot\frac{1}{1 + e^{-\widehat{\Delta_k}}}, \tag{10}$$

where $\delta^{'}(y_i = k) = 1$ if $y_i = k$ and $\delta^{'}(y_i = k) = 0$ if not. We only update the radius $\Delta_{y_i}$ belonging to class $k$ in a mini-batch, which ensures the denominator is not zero.

With the boundary loss $\mathcal{L}_b$, the boundaries can adapt to the intent feature space and learn suitable decision boundaries. The learned decision boundaries can not only effectively surround most of the known intent samples, but also not be far away from each known class centroid, which is effective to identify the open intent samples.

| | Methods | BANKING | | OOS | | StackOverflow | |
|---|---|---|---|---|---|---|---|
| | | Accuracy | F1-score | Accuracy | F1-score | Accuracy | F1-score |
| 25% | MSP | 43.67 | 50.09 | 47.02 | 47.62 | 28.67 | 37.85 |
| | DOC | 56.99 | 58.03 | 74.97 | 66.37 | 42.74 | 47.73 |
| | OpenMax | 49.94 | 54.14 | 68.50 | 61.99 | 40.28 | 45.98 |
| | DeepUnk | 64.21 | 61.36 | 81.43 | 71.16 | 47.84 | 52.05 |
| | ADB | **78.85** | **71.62** | **87.59** | **77.19** | **86.72** | **80.83** |
| 50% | MSP | 59.73 | 71.18 | 62.96 | 70.41 | 52.42 | 63.01 |
| | DOC | 64.81 | 73.12 | 77.16 | 78.26 | 52.53 | 62.84 |
| | OpenMax | 65.31 | 74.24 | 80.11 | 80.56 | 60.35 | 68.18 |
| | DeepUnk | 72.73 | 77.53 | 83.35 | 82.16 | 58.98 | 68.01 |
| | ADB | **78.86** | **80.90** | **86.54** | **85.05** | **86.40** | **85.83** |
| 75% | MSP | 75.89 | 83.60 | 74.07 | 82.38 | 72.17 | 77.95 |
| | DOC | 76.77 | 83.34 | 78.73 | 83.59 | 68.91 | 75.06 |
| | OpenMax | 77.45 | 84.07 | 76.80 | 73.16 | 74.42 | 79.78 |
| | DeepUnk | 78.52 | 84.31 | 83.71 | 86.23 | 72.33 | 78.28 |
| | ADB | **81.08** | **85.96** | **86.32** | **88.53** | **82.78** | **85.99** |

Table 2: Results of open classification with different known class proportions (25%, 50% and 75%) on BANKING, OOS and StackOverflow datasets. "Accuracy" and "F1-score" respectively denote the accuracy score and macro F1-score over all classes.

## Open Classification with Decision Boundary

After training, we use the centroids and the learned decision boundaries of each known class for inference. We suppose known intent samples are constrained in the closed ball area produced by their corresponding centroids and decision boundaries. On the contrary, the open intent samples are outside any of the bounded spherical areas. Specifically, we perform open intent classification as follows:

$$\hat{y} = \begin{cases} \text{open, if } d(\boldsymbol{z}_i, \boldsymbol{c}_k) > \Delta_k, \forall k \in \mathcal{Y}; \\ \arg\min_{k \in \mathcal{Y}} d(\boldsymbol{z}_i, \boldsymbol{c}_k), \text{otherwise}, \end{cases} \tag{11}$$

where $d(\boldsymbol{z}_i, \boldsymbol{c}_k)$ denotes the Euclidean distance between $\boldsymbol{z}_i$ and $\boldsymbol{c}_k$. $\mathcal{Y} = \{1, 2, \cdots, K\}$ denote the known intent labels.

# Experiments

## Datasets

We conduct experiments on three challenging real-world datasets to evaluate our approach. The detailed statistics are shown in Table 1.

**BANKING** A fine-grained dataset in the banking domain (Casanueva et al. 2020). It contains 77 intents and 13,083 customer service queries.

**OOS** A dataset for intent classification and out-of-scope prediction (Larson et al. 2019). It contains 150 intents, 22,500 in-domain queries and 1,200 out-of-domain queries.

**StackOverflow** A dataset published in Kaggle.com. It contains 3,370,528 technical question titles. We use the processed dataset (Xu et al. 2015), which has 20 different classes and 1,000 samples for each class.

## Baselines

We compare our method with the following state-of-the-art open classification methods: OpenMax (Bendale and Boult 2016), MSP (Hendrycks and Gimpel 2017), DOC (Shu, Xu, and Liu 2017) and DeepUnk (Lin and Xu 2019a).

As OpenMax is an open set detection method in computer vision, we adapt it for open intent classification. We firstly use the softmax loss to train a classifier on known intents, then fit a Weibull distribution to the classifier's output logits. Finally, we recalibrate the confidence scores with the OpenMax Layer. Due to lack of open intent for tuning, we adopt default hyperparameters of OpenMax. We use the same confidence threshold (0.5) as in (Lin and Xu 2019a) for MSP. For a fairness comparison, we replace the backbone network of these methods with the same BERT model as ours.

## Evaluation Metrics

Following previous work (Shu, Xu, and Liu 2017; Lin and Xu 2019a), we regard all the open classes as one rejected class. To evaluate the overall performance, we use accuracy score (Accuracy) and macro F1-score (F1-score) as metrics. They are calculated over all classes (known classes and open class). We also calculate macro F1-score over known classes and open class respectively, which better evaluates the fine-grained performance.

## Experimental Settings

Following the same settings as in (Shu, Xu, and Liu 2017; Lin and Xu 2019a), we keep some classes as unknown (open) and integrate them back during testing. All datasets are divided into training, validation and test sets. The number of known classes are varied with the proportions of 25%, 50%, and 75% in the training set. The remaining classes are regarded as one open class and removed from the training set. Both known classes and open class are used for testing.

| | Methods | BANKING | | OOS | | StackOverflow | |
|---|---|---|---|---|---|---|---|
| | | Open | Known | Open | Known | Open | Known |
| 25% | MSP | 41.43 | 50.55 | 50.88 | 47.53 | 13.03 | 42.82 |
| | DOC | 61.42 | 57.85 | 81.98 | 65.96 | 41.25 | 49.02 |
| | OpenMax | 51.32 | 54.28 | 75.76 | 61.62 | 36.41 | 47.89 |
| | DeepUnk | 70.44 | 60.88 | 87.33 | 70.73 | 49.29 | 52.60 |
| | ADB | **84.56** | **70.94** | **91.84** | **76.80** | **90.88** | **78.82** |
| 50% | MSP | 41.19 | 71.97 | 57.62 | 70.58 | 23.99 | 66.91 |
| | DOC | 55.14 | 73.59 | 79.00 | 78.25 | 25.44 | 66.58 |
| | OpenMax | 54.33 | 74.76 | 81.89 | 80.54 | 45.00 | 70.49 |
| | DeepUnk | 69.53 | 77.74 | 85.85 | 82.11 | 43.01 | 70.51 |
| | ADB | **78.44** | **80.96** | **88.65** | **85.00** | **87.34** | **85.68** |
| 75% | MSP | 39.23 | 84.36 | 59.08 | 82.59 | 33.96 | 80.88 |
| | DOC | 50.60 | 83.91 | 72.87 | 83.69 | 16.76 | 78.95 |
| | OpenMax | 50.85 | 84.64 | 76.35 | 73.13 | 44.87 | 82.11 |
| | DeepUnk | 58.54 | 84.75 | 81.15 | 86.27 | 37.59 | 81.00 |
| | ADB | **66.47** | **86.29** | **83.92** | **88.58** | **73.86** | **86.80** |

Table 3: Results of open classification with different known class ratios (25%, 50% and 75%) on BANKING, OOS and StackOverflow datasets. "Open" and "Known" denote the macro f1-score over open class and known classes respectively.

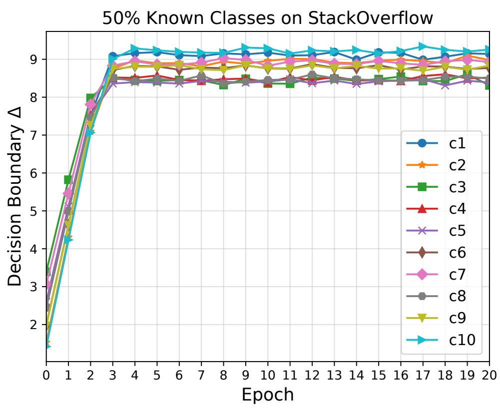


Figure 3: The boundary learning process.

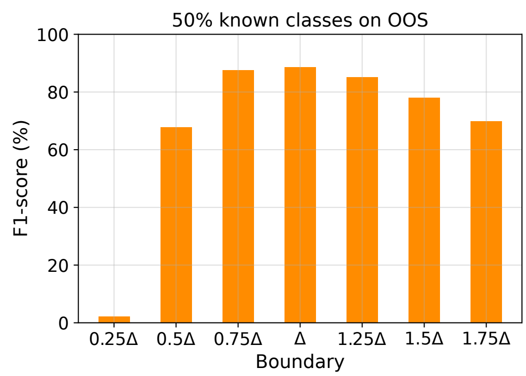


Figure 4: Influence of the learned decision boundary.

For each known class ratio, we report the average performance over ten runs of experiments.

We employ the BERT model (bert-uncased, with 12-layer transformer) implemented in PyTorch (Wolf et al. 2019) and adopt most of its suggested hyperparameters for optimization. To speed up the training procedure and achieve better performance, we freeze all but the last transformer layer parameters of BERT. The training batch size is 128, and the learning rate is 2e-5. For the boundary loss $\mathcal{L}_b$, we employ Adam (Kingma and Ba 2014) to optimize the boundary parameters at a learning rate of 0.05.

## Results

Table 2 and Table 3 show the performances of all compared methods, where the best results are highlighted in bold. Firstly, we observe the overall performance. Table 2 shows accuracy score and macro F1-score over all classes. With 25%, 50%, and 75% known classes, our approach consistently achieves the best results and outperforms other baselines by a significant margin. Compared with the best results of all baselines, our method improves accuracy score (Accuracy) on BANKING by 14.64%, 6.13%, and 2.56%, on OOS by 6.16%, 3.19%, and 2.61%, on StackOverflow by 38.88%, 27.42%, and 10.45% in 25%, 50% and 75% settings respectively, which demonstrates the priority of our method.

Secondly, we notice that the improvements on StackOverflow are much more drastic than the other two datasets. We suppose the improvements mainly depend on the characteristics of datasets. Most baselines lack explicit or suitable decision boundaries for identifying the open intent, so they are more sensitive to different datasets. For example, they are limited to distinguish difficult semantic intents (e.g., technical question titles in StackOverflow) without prior knowledge. By contrast, our method learns specific and tight decision boundaries for each known class, which is more effective for open intent classification.

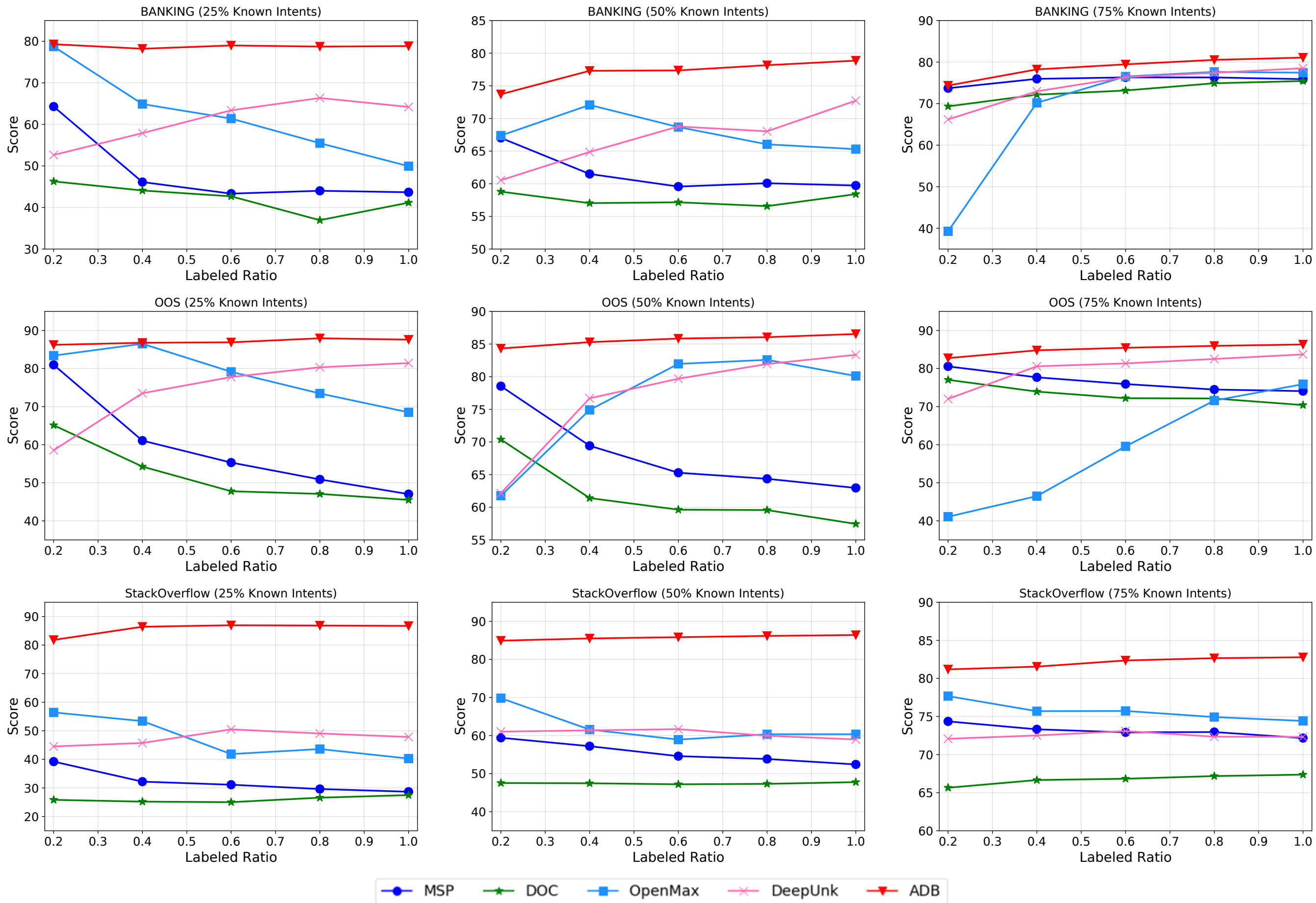


Figure 5: Influence of the labeled ratio on three datasets with different known class proportions (25%, 50%, 75%).

Thirdly, we observe the fine-grained performance. Table 3 shows the macro F1-score on open intent and known intents respectively. We notice that our method not only achieves substantial improvements on open class, but also largely enhances the performances on known classes compared with baselines. That is because our method can learn specific and tight decision boundaries for detecting open class while ensuring the quality of known intent classification.

# Discussion

## Boundary Learning Process

Figure 3 shows the decision boundary learning process. At first, most parameters are assigned small values near zero after initialization, which leads to small radius with the $\mathrm{Softplus}$ activation function. As the initial radius is too small, the empirical risk plays a dominant role. Therefore, the radius of each decision boundary expands to contain more known intent samples belonging to its class. As the training process goes on, the radius of the decision boundary learns to be large enough to contain most of the known intents. However, the large radius will also introduce redundant open intent samples. In this case, the open space risk plays a dominant role, which prevents the radius from enlarging. Finally, the decision boundaries converge with a balance between empirical risk and open space risk.

## Effect of Decision Boundary

To verify the effectiveness of the learned decision boundary, we use different ratios of $\Delta$ as boundaries during testing. As shown in Figure 4, ADB achieves the best performance with $\Delta$ among all assigned decision boundaries, which verifies the tightness of the learned decision boundary. Moreover, we notice that the performance of open classification is sensitive to the size of the decision boundaries. Overcompact decision boundaries will increase the open space risk by misclassifying more known intent samples to the open intent. Correspondingly, overrelaxed decision boundaries will increase the empirical risk by misclassifying more open intent samples as known intents. As shown in Figure 4, both of these two cases perform worse compared with $\Delta$.

## Effect of Labeled Data

To investigate the influence of labeled data, we vary the labeled ratio in the training set in the range of 0.2, 0.4, 0.6, 0.8 and 1.0. We use Accuracy as the score to evaluate the performance. As shown in Figure 5, ADB outperforms all the other baselines on three datasets on almost all settings. Besides, it keeps a more robust performance under different labeled ratios compared with other methods.

Notably, the statistic-based methods (e.g., MSP and DOC) show better performances with less labeled data. We suppose the reason is that the predicted scores are in low-confidence with less prior knowledge for training, which is helpful to reject the open intent with the threshold. However, as the number of labeled data increases, these methods tend to be biased towards the known intents, with the aid of strong feature extraction capability of DNNs (Nguyen, Yosinski, and Clune 2015). Therefore, the performances drop dramatically.

In addition, we notice that OpenMax and DeepUnk are two competitive baselines. We suppose the reason is that they both leverage the characteristics of intent feature distribution to detect the open class. However, OpenMax computes centroids of each known class with only corrective positive training samples. The qualities of centroids are easily influenced by the number of training samples. DeepUnk adopts a density-based novelty detection algorithm to perform open classification, which is also limited to the prior knowledge of labeled data. Thus, their performances all drop dramatically with less labeled data, as shown in Figure 5.

### Effect of Known Classes

We vary the known class ratio between 25%, 50% and 75%, and show the results in Table 2 and Table 3. Firstly, we observe the overall performance in Table 2. Compared with other methods, our method achieves huge improvements on all settings of three datasets. All baselines drop dramatically as the known class ratio decreases. By contrast, our method still achieves robust results on accuracy score with fewer training samples.

Then, we observe the fine-grained performance in Table 3. We notice that all baselines achieve high scores on known classes, but they are limited to identify the open intent and suffer poor performance. However, our method still yields the best results on both known classes and the open class. It further demonstrates that the suitable learned decision boundaries are helpful to both balance the empirical risk and the open space risk.

## Related Work

### Intent Detection

There are many works for intent detection in dialogue systems in recent years (Min et al. 2020; Qin et al. 2020; Zhang et al. 2019; E et al. 2019; Qin et al. 2019). Nevertheless, they all make assumptions of closed world classification without the open intent. Srivastava, Labutov, and Mitchell (2018) perform intent detection with the zero-shot learning (ZSL) method. However, ZSL is different from our task because it only contains novel classes during testing.

Unknown intent detection is a specific task for detecting the unknown intent. Brychcin and Král (2017) propose an unsupervised approach to modeling intents, but fail to utilize the prior knowledge of known intents. Kim and Kim (2018) jointly train the in-domain (ID) classifier and out-of-domain (OOD) detector but need to sample OOD utterances. Yu et al. (2017) adopt adversarial learning to generate positive and negative samples for training the classifier. Ryu et al. (2018) use a generative adversarial network (GAN) to train on the ID samples and detect the OOD samples with the discriminator. However, it has been shown that deep generative models fail to capture high-level semantics on real-world data (Nalisnick et al. 2019; Mundt et al. 2019). Recent methods try to learn friendly features for detecting the unknown intent (Lin and Xu 2019a; Gangal et al. 2020; Yan et al. 2020), but they need to modify the model architecture, and fail to construct specific decision boundaries.

### Open World Classification

At first, researchers use SVM to solve open set problems. One-class classifiers (Schölkopf et al. 2001; Tax and Duin 2004) find the decision boundary based on the positive training data. For multi-class open classification, One-vs-all SVM (Rifkin and Klautau 2004) trains the binary classifier for each class and treats the negative classified samples as the open class. Scheirer et al. (2013) extend the method to computer vision and introduce the concept of open space risk. Jain, Scheirer, and Boult (2014) estimate the unnormalized posterior probability of inclusion for open set problems. They fit the probability distributions to statistical Extreme Value Theory (EVT) by using a Weibull-calibrated multi-class SVM. Scheirer, Jain, and Boult (2014) propose a Compact Abating Probability (CAP) model, which further improves the performance of Weibull-calibrated SVM by truncating the abating probability. However, all these methods need negative samples for selecting the decision boundary or probability threshold. Moreover, SVM cannot capture advanced semantic features of intents (Lin and Xu 2019b).

Recently, researchers use deep neural networks for open classification. OpenMax (Bendale and Boult 2016) fits Weibull distribution to the outputs of the penultimate layer, but still needs negative samples for selecting the best hyperparameters. MSP (Hendrycks and Gimpel 2017) calculates the softmax probability of known samples and rejects the low confidence unknown samples with the threshold. ODIN (Liang, Li, and Srikant 2018) uses temperature scaling and input preprocessing to enlarge the differences between known and unknown samples. However, both of them (Hendrycks and Gimpel 2017; Liang, Li, and Srikant 2018) need unknown samples to select the confidence threshold artificially. DOC (Shu, Xu, and Liu 2017) uses the sigmoid function and calculates the confidence threshold based on Gaussian statistics, but it performs worse when the output probabilities are not discriminative.

## Conclusion

In this paper, we propose a novel post-processing method for open intent classification. After pre-training the model with labeled samples, our model can automatically learn specific and tight decision boundaries adaptive to the known intent feature space. Our method has no require for open intent or model architecture modification. Extensive experiments on three benchmark datasets show that our method yields significant improvements over the compared baselines, and is more robust with less labeled data and fewer known intents.

## Acknowledgments

This work is supported by seed fund of Tsinghua University (Department of Computer Science and Technology)-Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things.

## References

Bendale, A.; and Boult, T. 2015. Towards Open World Recognition. In *Proceedings of CVPR*, 1893–1902.

Bendale, A.; and Boult, T. E. 2016. Towards open set deep networks. In *Proceedings of CVPR*, 1563–1572.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, volume 29, 93–104.

Brychcin, T.; and Král, P. 2017. Unsupervised Dialogue Act Induction using Gaussian Mixtures. In *Proceedings of EACL*, 485–490.

Casanueva, I.; Temcinas, T.; Gerz, D.; Henderson, M.; and Vulic, I. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of ACL WorkShop*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

E, H.; Niu, P.; Chen, Z.; and Song, M. 2019. A Novel Bidirectional Interrelated Model for Joint Intent Detection and Slot Filling. In *Proceedings of ACL*, 5467–5471.

Fei, G.; and Liu, B. 2016. Breaking the Closed World Assumption in Text Classification. In *Proceedings of NAACL-HLT*, 506–514.

Gangal, V.; Arora, A.; Einolghozati, A.; and Gupta, S. 2020. Likelihood Ratios and Generative Classifiers for Unsupervised Out-of-Domain Detection in Task Oriented Dialog. In *Proceedings of AAAI*, 7764–7771.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *Proceedings of ICLR*.

Jain, L. P.; Scheirer, W. J.; and Boult, T. E. 2014. Multi-Class Open Set Recognition Using Probability of Inclusion. In *Proceedings of ECCV*.

Kim, J.-K.; and Kim, Y.-B. 2018. Joint Learning of Domain Classification and Out-of-Domain Detection with Dynamic Class Weighting for Satisficing False Acceptance Rates. In *Proceedings of INTERSPEECH*, 556–560.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Larson, S.; Mahendran, A.; Peper, J. J.; Clarke, C.; Lee, A.; Hill, P.; Kummerfeld, J. K.; Leach, K.; Laurenzano, M. A.; Tang, L.; and Mars, J. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of EMNLP-IJCNLP*, 1311–1316.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *Proceedings of ICLR*.

Lin, T.-E.; and Xu, H. 2019a. Deep Unknown Intent Detection with Margin Loss. In *Proceedings of ACL*, 5491–5496.

Lin, T.-E.; and Xu, H. 2019b. A post-processing method for detecting unknown intent of dialogue system via pre-trained deep neural network classifier. *Knowledge-Based Systems* 186: 104979.

Lin, T.-E.; Xu, H.; and Zhang, H. 2020. Discovering New Intents via Constrained Deep Adaptive Clustering with Cluster Refinement. In *Proceedings of AAAI*, 8360–8367.

Min, Q.; Qin, L.; Teng, Z.; Liu, X.; and Zhang, Y. 2020. Dialogue State Induction Using Neural Latent Variable Models. In *Proceedings of IJCAI*, 3845–3852.

Mundt, M.; Pliushch, I.; Majumder, S.; and Ramesh, V. 2019. Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers? In *Proceedings of ICCV Workshops*.

Nalisnick, E.; Matsukawa, A.; Teh, Y. W.; Gorur, D.; and Lakshminarayanan, B. 2019. Do Deep Generative Models Know What They Don't Know? In *Proceedings of ICLR*.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of CVPR*, 427–436.

Qin, L.; Che, W.; Li, Y.; Ni, M.; and Liu, T. 2020. DCR-Net: A Deep Co-Interactive Relation Network for Joint Dialog Act Recognition and Sentiment Classification. In *Proceedings of AAAI*, 8665–8672.

Qin, L.; Che, W.; Li, Y.; Wen, H.; and Liu, T. 2019. A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding. In *Proceedings of EMNLP*, 2078–2087.

Rifkin, R. M.; and Klautau, A. 2004. In Defense of One-Vs-All Classification. *Journal of Machine Learning Research* 5: 101–141.

Ryu, S.; Koo, S.; Yu, H.; and Lee, G. G. 2018. Out-of-domain Detection based on Generative Adversarial Network. In *Proceedings of EMNLP*, 714–718.

Scheirer, W. J.; de Rezende Rocha, A.; Sapkota, A.; and Boult, T. E. 2013. Toward Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7(35): 1757–1772.

Scheirer, W. J.; Jain, L. P.; and Boult, T. E. 2014. Probability Models for Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)* 36.

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13(7): 1443–1471.

Shu, L.; Xu, H.; and Liu, B. 2017. DOC: Deep Open Classification of Text Documents. In *Proceedings of EMNLP*, 2911–2916.

Srivastava, S.; Labutov, I.; and Mitchell, T. 2018. Zero-shot Learning of Classifiers from Natural Language Quantification. In *Proceedings of ACL*, 306–316.

Tapaswi, M.; Law, M. T.; and Fidler, S. 2019. Video Face Clustering with Unknown Number of Clusters. In *Proceedings of ICCV*, 5026–5035.

Tax, D.; and Duin, R. 2004. Support Vector Data Description. *Machine Learning* 54: 45–66.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; and Brew, J. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771* .

Xu, J.; Wang, P.; Tian, G.; Xu, B.; Zhao, J.; Wang, F.; and Hao, H. 2015. Short Text Clustering via Convolutional Neural Networks. In *Proceedings of NAACL-HLT*, 62–69.

Yan, G.; Fan, L.; Li, Q.; Liu, H.; Zhang, X.; Wu, X.-M.; and Lam, A. Y. 2020. Unknown Intent Detection Using Gaussian Mixture Model with an Application to Zero-shot Intent Classification. In *Proceedings of ACL*, 1050–1060.

Yu, Y.; Qu, W.-Y.; Li, N.; and Guo, Z. 2017. Open-Category Classification by Adversarial Sample Generation. In *Proceedings of IJCAI*, 3357–3363.

Zhang, C.; Li, Y.; Du, N.; Fan, W.; and Yu, P. 2019. Joint Slot Filling and Intent Detection via Capsule Neural Networks. In *Proceedings of ACL*, 5259–5267.