Learning Discriminative Representations and Decision Boundaries for Open Intent Detection

Hanlei Zhang, Hua Xu, Shaojie Zhao, Qianrui Zhou

***Abstract*—Open intent detection is a significant problem in natural language understanding, which aims to identify the unseen open intent while ensuring known intent identification performance. However, current methods face two major challenges. Firstly, they struggle to learn friendly representations to detect the open intent with prior knowledge of only known intents. Secondly, there is a lack of an effective approach to obtaining specific and compact decision boundaries for known intents. To address these issues, this paper presents an original framework called DA-ADB, which successively learns distance-aware intent representations and adaptive decision boundaries for open intent detection. Specifically, we first leverage distance information to enhance the distinguishing capability of the intent representations. Then, we design a novel loss function to obtain appropriate decision boundaries by balancing both empirical and open space risks. Extensive experiments demonstrate the effectiveness of the proposed distance-aware and boundary learning strategies. Compared to state-of-the-art methods, our framework achieves substantial improvements on three benchmark datasets. Furthermore, it yields robust performance with varying proportions of labeled data and known categories. The full data and codes are available for use at https://github.com/thuiar/TEXTOIR.**

***Index Terms*—Intent detection, open classification, natural language understanding, representation learning, deep neural network.**

I. Introduction

INTENT detection plays a critical role in natural language understanding (NLU), aiming to mine user purposes behind the text utterances. The traditional intent detection task is restricted to closed-world classification. It assumes all the intent categories are accessible and has achieved great progress with a booming of effective methods for supervised classification [1], [2].

Nonetheless, due to the variety and uncertainty of the user needs, it is usually inapplicable to cover all intent categories. Taking Figure 1 as an example, there are two task-specific intents of booking flight and restaurant reservation. Ideally, we hope to identify each utterance within the two known intent categories. However, in the real application, some unexpected utterances may exist with unknown intents that have never been seen before, such as asking about the time or place. Effectively detecting these unknown intents helps reduce false-positive errors and makes the system more robust. Moreover, we can leverage them to explore more potential user needs and improve customer satisfaction.

Hanlei Zhang, Hua Xu, Shaojie Zhao and Qianrui Zhou are with the State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail:zhang-hl20@mails.tsinghua.edu.cn; xuhua@tsinghua.edu.cn; murrayzhao@163.com; zgr22@mails.tsinghua.edu.cn).

Shaojie Zhao is with School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China (e-mail:murrayzhao@163.com).

User utterances	Intent Label
Book a flight from LA to Madrid.	Booking flight
Can you get me a table at Steve's?	Restaurant reservation
Book Delta ticket Madison to Atlanta.	Booking flight
Schedule me a table at Red Lobster.	Restaurant reservation
.....	.....
What time is it now?	**Open**
Where is the nearest school?	**Open**

Fig. 1. An example of open intent detection. Booking flight and Restaurant reservation are two known intents. We should identify them correctly while detecting the utterances with the open intent.

Our previous works first proposed the open (unknown) intent detection task [3]–[5] to solve this problem. As we do not have any prior knowledge of unknown intents (e.g., the specific categories and the class number), all the unknown classes are regarded as one open class. Specifically, the goal of this task is to use the prior knowledge of only K-class known intents to detect the unseen $(K{+}1)^{\text{th}}$ class open intent while ensuring the known intent identification performance. It requires no need to collect labeled data for each fine-grained open intent category during training or evaluation, saving much time and workforce for practical application.

Similar open set problems were first explored in computer vision [6]. Among these problems, open set recognition (OSR) [7] has the closest setting to our task, which also aims to identify known classes and reject the unknown class that does not appear in the training set. However, the setting of OSR can use unknown-class data for tuning parameters [6]. In contrast, the open intent data are unavailable during validation in our task, which is more suitable in real-world scenarios. Besides, OSR methods are only applied in visual tasks, which may not work well on discrete text data [8], [9]. Fei and Liu [10] extended this problem to the text classification but also used unknown-class data for obtaining SVM boundaries. Shu et al. [11] adopted the convolution neural network (CNN) to extract deep features and obtained tight confidence thresholds for each known class based on the statistics information. However, separating known classes and the open class with dense confidence scores may be hard.

Out-of-distribution (OOD) detection is a task that involves identifying outliers that differ from in-distribution (ID) data

during testing. The goal is to generate discriminative scores that can distinguish between ID and OOD samples [12]–[17]. While our task is similar, the main difference is that we aim to design appropriate decision criteria to balance the performance of multi-class ID and the one-class OOD samples. Our experiments have shown that using OOD detection methods directly after training a model on IND data can lead to a degradation in performance compared to open intent detection methods.

The research on open intent detection is just beginning in recent years. Lin and Xu [3] made the first trial on this task with the feature-based methods. In particular, the large margin cosine loss [18] was first adopted to learn representations with intra-class compactness and inter-class separation properties. Then, the local outlier factor (LOF) [19] was used to detect the low-density examples as the open class. Nevertheless, the embeddings optimized in cosine space may be less suitable for LOF [20]. Yan et al. [20] leveraged the Gaussian mixture model incorporating the class label information to obtain more suitable representations for anomaly detection. However, the performance drops dramatically when the intent categories have complicated semantics. Recent works tried to construct pseudo open class samples for (K+1)-way training [21], [22], yet the samples from the open class may not follow the same distribution in the semantic space.

There are two main difficulties in current open intent detection methods. Firstly, the representations trained on known intents may not be robust enough for detecting the unseen open class. Secondly, the decision conditions (e.g., confidence or density thresholds) need to be selected manually and are implicit based on the prior knowledge of only known intents.

We propose a novel framework for open intent detection to solve these problems, which learns discriminative representations and decision boundaries with only known intents. It first extracts deep intent representations from the pre-trained language model BERT [23] at the sentence level and calculates the centroids by averaging the samples of each known class. Then, it aims to perceive the distance information to learn representations with distinguishing capability. Specifically, to compute the distance-aware coefficient, each sample compares the Euclidean distances between its nearest and next-nearest centroids. The coefficient is incorporated into the original intent representation to produce the meta-embedding. The magnitude of its meta-embedding reflects the hardness of each sample. That is, the larger magnitude corresponds to the easier sample with more confidence to differentiate the two nearest centroids. To achieve this goal, a cosine classifier [24] is adopted after the meta-embedding to consider the effect of the magnitude information, which helps focus on harder samples with smaller magnitudes during training. In this way, the intent representations are calibrated to distance-aware concepts for more robust performance.

After representation learning, we aim to obtain specific and compact decision boundaries in the intent feature space. We suppose each known intent cluster is constrained in a spherical decision boundary to its centroid, which helps reduce the open space risk [10]. The decision boundaries are determined by the radius of each ball area and should be flexibly adaptive to different feature distributions. In particular, the boundary parameters are first initialized with standard normal distribution and then projected with a learnable activation function to get the radius of each decision boundary.

The key factor is how to control the radius to learn compact decision boundaries for open intent detection, which 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. A new loss function is designed to address these issues, optimizing the boundary parameters by balancing both the open space and empirical risk with known intent samples inside and outside the decision boundaries. With the boundary loss, the decision boundaries can automatically adapt to the intent feature space until balance. The boundary learning process is a post-processing method that requires no modifying the original model architecture. It works even with the features trained on the simple softmax loss. The distance-aware strategy can further facilitate learning more discriminative representations for better performance.

Our contributions are summarized as follows:

- We clarify the definition of the open intent detection problem and propose a novel and effective framework DA-ADB for the main challenges.
- A distance-aware strategy is designed to capture the distinguishing ability of each sample, which helps learn discriminative intent features.
- A novel post-processing method is proposed to learn tight decision boundaries adaptive to the feature space. To the best of our knowledge, it is the first attempt to automatically learn adaptive decision boundaries for detecting the unseen open class.
- Extensive experiments conducted on three challenging datasets show that our approach achieves consistently better and more robust results than state-of-the-art methods.

The idea of adaptive decision boundary (ADB) was presented in a preliminary version of this paper published in the proceeding of the thirty-fifth AAAI conference (AAAI-21) [5]. In this paper, we extend the preliminary version in the following aspects:

- A novel method is introduced to incorporate the distance information into the intent representations. It can capture the hardness of each sample with the distance-aware coefficient for effective training.
- A series of experiments are conducted to show the advantages of injecting distance-aware concepts into intent representations for open intent detection.
- We formulate the open intent detection problem and enrich the introduction and related work. More baselines are reproduced and added to our experiments with detailed analysis.
- The experimental results are updated with our TEXTOIR platform [25], which has standard and unified interfaces for a fair comparison.

II. Related Works

This section reviews the related works in open set recognition, out-of-distribution, and open intent detection.

A. Open Set Recognition

OSR is a pioneering work related to us, aiming to reject the negative samples while identifying positive samples. At first, researchers used SVM-based methods as the solutions. One-class SVM [26] was designed for binary classification, which found the plane based on the positive training data and regarded the origin as the only member of the negative class. One-vs-all SVM [27] was designed for multi-class open classification, which trained the binary classifier for each class and treated the negative classified samples as the open class. Scheirer et al. [7] extended the method to computer vision and introduced the concept of open space risk. They introduced the one-vs-set machine to improve generalization ability by compressing the decision space of one-class SVM. Jain et al. [28] used a Weibull-calibrated multi-class SVM to estimate the posterior probability satisfying the statistical Extreme Value Theory (EVT). Scheirer et al. [29] presented a Compact Abating Probability (CAP) model, which further improved the performance of Weibull-calibrated SVM by truncating the abating probability.

However, the SVM-based methods have difficulties in capturing advanced pattern semantic concepts [4]. Thus, researchers used deep neural networks for OSR. For example, Bendale and Boult [8] designed an OpenMax layer after the penultimate layer of deep neural networks (DNNs) to estimate the open class probability. Zhou et al. [30] calibrated the closed-set classifier by learning the classifier and data placeholders, which are used to distinguish between known and unknown data and simulate open-class data, respectively. Chen et al. [9] constructed reciprocal points to reduce the empirical risk and further introduced a bounded adversarial mechanism to reduce the open space risk. Nevertheless, these methods only verified the effectiveness of benchmarks in computer vision. Shu et al. [11] explored this task in natural language processing. They used the output layer of sigmoids and calculated the confidence thresholds based on Gaussian statistics, but the method performs worse when the output probabilities are not discriminative.

B. Out-of-distribution Detection

OOD detection is a popular task that has received much attention in recent years, which goal is to detect the samples in the testing set that exhibit distribution shifts [31]. A line of works used both ID and ground-truth/generated OOD samples during training. For instance, Kim and Kim [15] jointly trained an in-domain classifier and an out-of-domain detector with both ID and OOD annotated utterances. Hendrycks et al. [14] proposed the use of outlier exposure (OE) to train an OOD detector with external OOD samples. However, annotating OOD samples can be time-consuming and labor-intensive. To address this issue, researchers used the generative adversarial network (GAN) to generate OOD samples. Yu et al. [32] adopted adversarial learning to generate positive and negative samples for training the classifier. Ryu et al. [33] used a GAN to train on the ID samples and detected OOD samples with a discriminator. Zheng et al. [34] used a GAN-based generator to produce pseudo OOD samples of discrete token sequences. Nevertheless, it has been shown that deep generative models have limitations in learning high-level semantics on discrete text data [35].

Another line of works used only ID samples during training. For example, Hendrycks et al. [12] calculated the softmax probability from ID samples and rejected the low-confidence OOD samples with a threshold. Liang et al. [13] used temperature scaling and input pre-processing to enlarge the differences between ID and OOD samples. Xu et al. [16] first fine-tuned the pre-trained transformer with the objective of masked language modeling (MLM) and then utilized the distance information of features from all layers for detecting outliers. Moreover, a series of methods have been developed to design the score function without modifying the network architecture [36], [37]. However, all of the above methods mainly focus on binary classification and may suffer a performance decrease when adapting them to our K+1 classification task.

C. Open Intent Detection

Intent detection is a fundamental task in NLU. For example, the joint slot filling and intent detection task [38]–[40] has been extensively studied and achieved outstanding performance on standard benchmark datasets [41], [42]. Although more challenging intent benchmark datasets have been proposed in recent years [43], [44], powerful pre-trained language models still perform well under the assumption of closed world classification [43].

However, in real-world scenarios, user needs are diverse and unpredictable, and it is almost impossible to know all intent categories in advance. Therefore, several approaches have been proposed to detect the open (unknown) intent. For example, Brychcin and Kr'al [45] proposed an unsupervised method for intent modeling, but it failed to leverage the information of known intents. Zhang et al. [46] explored intent detection in few-shot scenarios by learning relations between synthesized positive and negative samples. However, this method needs to augment a large amount of data pairs, which is impractical with more labeled data and leads to unacceptable training and inference time. Xia et al. [47] performed intent detection under the zero-shot setting, which assumes the class number and side information of open intents are known. However, there is usually no prior knowledge of open intents in real applications. To address these challenges, we proposed the open intent detection task [3]–[5]. In this task, only known intents are used for training and validation. The goal is to identify known intents and detect the one-class open intent during testing.

Lin and Xu [3] first tackled this problem by learning deep intent features with the margin loss and detected the open intent with LOF. Yan et al. [20] replaced the margin loss with the Gaussian mixture loss to learn better embeddings, but the performance largely depends on the class-label semantics. Moreover, the density-based algorithm is unable to construct specific decision boundaries. Zhan et al. [21] and Cheng et al. [22] regarded different (pseudo) open intents as one class during training, but it may lead to the collapse with intents of disparate semantics.

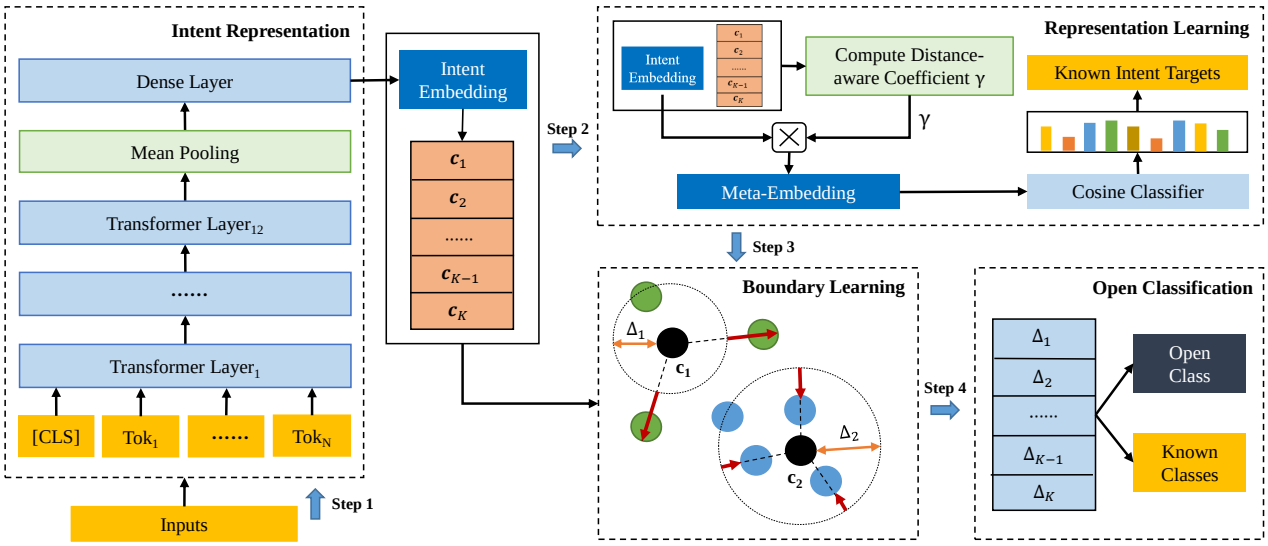


Fig. 2. The overall architecture of the proposed framework. Firstly, we use the pre-trained language model BERT to get intent embeddings and average them for each known class to obtain the centroids $\{\boldsymbol{c}_i\}_{i=1}^{K}$. Then, the intent embeddings and centroids are leveraged to compute distance-aware coefficients, further multiplied over the original intent embeddings to yield the meta-embeddings. A cosine classifier learns the distance information with the known intent targets. Next, a new loss function is proposed to learn the radii of the decision boundaries $\{\Delta_i\}_{i=1}^{K}$ adaptive to the intent feature space. Finally, the centroids and decision boundaries are used for open intent detection.

III. Problem Formulation

In open intent detection, we are given an intent label set I and a data set D. The intent label set $I = \{I^{\text{Known}}, \text{Open}\}$, where $I^{\text{Known}} = \{I_1, \ldots, I_K\}$ is the known intent label set and K is the number of known intents. Notably, there may be multiple remaining intent labels in the initial label set $I \backslash \{I^{\text{Known}}\}$, as indicated in [48], [49]. In this problem, the non-known intents are all assigned the unified Open label.

The data set $D = \{D^{\text{Train}}, D^{\text{Valid}}, D^{\text{Test}}\}$ consists of training, validation and testing sets. Each subset (e.g., D^{Train}) contains a set of labeled samples $(\boldsymbol{s}_i, y_i)$, where $\boldsymbol{s}_i$ is the i^{th} utterance, and y_i is its intent label.

The intent label set for both D^{Train} and D^{Valid} is I^{Known}, while for D^{Test} is I. The training and validation sets contain merely known intent samples, and the unseen open intent samples only exist in the testing set. The goal of open intent detection is to leverage the K-class known intents as prior knowledge to both identify known intents and detect the $(K{+}1)^{\text{th}}$ class open intent.

IV. The Proposed Approach

This section presents a novel framework for learning friendly intent representations and appropriate decision boundaries for open intent detection. It contains four main steps, intent representation, distance-aware representation learning, adaptive decision boundary learning, and open classification. Figure 2 shows the overall architecture of our proposed approach.

A. Intent Representation

The pre-trained BERT language model is adopted to extract deep intent features. Given the i^{th} input utterance $\boldsymbol{s}_i$, we get all its token embeddings $[CLS, Tok_1, \cdots, Tok_M] \in \mathbb{R}^{(M+1)\times H}$ from the last hidden layer of BERT. As suggested in [48], we perform mean-pooling on these token embeddings to synthesize the semantic features in the utterance and get the averaged representation $\boldsymbol{x}_i \in \mathbb{R}^H$:

$$\boldsymbol{x}_i = \text{mean-pooling}([CLS, Tok_1, \cdots, Tok_M]), \quad (1)$$

where CLS is a special classification token, M is the sequence length and H is the hidden layer size 768. To further strengthen feature extraction capability, we feed $\boldsymbol{x}_i$ to a dense layer h to get the intent representation $\boldsymbol{z}_i \in \mathbb{R}^D$:

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

where D is the feature dimension, σ is the 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.

B. Distance-aware Representation Learning

A new distance-aware representation learning strategy is introduced to learn discriminative intent features. In this method, the centroids of each known class are first calculated and then used to compute the distance-aware coefficient of each sample to obtain the meta-embedding. After the meta-embedding, a cosine classifier enables each sample to perceive the distance information.

1) Centroids Calculation: To calculate the centroids of each known class, let $S = \{(\boldsymbol{z}_i, y_i), \ldots, (\boldsymbol{z}_N, y_N)\}$ be N known intent labeled examples. S_k denotes the set of feature vectors labeled with class k. The centroid $\boldsymbol{c}_k \in \mathbb{R}^D$ is the mean vector of embedded examples in S_k:

$$\boldsymbol{c}_k = \frac{1}{|S_k|} \sum_{(\boldsymbol{z}_i, y_i)\in S_k} \boldsymbol{z}_i, \quad (3)$$

where $|S_k|$ denotes the number of examples in S_k.

2) Meta Embedding with Distance-aware Concept: The initial intent representations have limitations in identifying whether an example is "easy" or "hard" during training, which is unfavorable for discriminative representation learning. To address this issue, we leverage the distance-aware concept to obtain the meta-embedding for enhancing the distinguishing ability.

For each example, the confidence of which known class it belongs depends on the Euclidean distances between it and known-class centroids in the feature space. The index of the nearest centroid k_a is most likely to be its corresponding class, while the index of the next-nearest centroid k_b is the most confusing category to be classified. Thus, k_a and k_b are the two most informative centroid indexes to evaluate the distinguishing ability, and they are computed by:

$$k_a = \underset{k}{\operatorname{argmin}} \left\{ \|\boldsymbol{z}_i - \boldsymbol{c}_k\|_2 \right\}_{k \in I^{\text{Known}}}, \tag{4}$$

$$k_b = \underset{k}{\operatorname{argmin}} \left\{ \|\boldsymbol{z}_i - \boldsymbol{c}_k\|_2 \right\}_{k \in I^{\text{Known}} \setminus \{k_a\}}, \tag{5}$$

where $\|\boldsymbol{z}_i - \boldsymbol{c}_k\|_2$ denotes the Euclidean distance between $\boldsymbol{z}_i$ and $\boldsymbol{c}_k$. A discriminative example should be close to its nearest centroid and far away from the next nearest centroid. Thus, the difference between $\|\boldsymbol{z}_i - \boldsymbol{c}_{k_b}\|_2$ and $\|\boldsymbol{z}_i - \boldsymbol{c}_{k_a}\|_2$ is used to reflect the separating capacity of $\boldsymbol{z}_i$. In particular, the distance-aware coefficient γ_i is defined as:

$$\gamma_i = \exp(\|\boldsymbol{z}_i - \boldsymbol{c}_{k_b}\|_2 - \|\boldsymbol{z}_i - \boldsymbol{c}_{k_a}\|_2) \text{ s.t. } \gamma_i \geq 1, \tag{6}$$

where $\exp(\cdot)$ enables an exponentially large reception field. It enhances the effect of differentiation and avoids the trivial solution when $\|\boldsymbol{z}_i - \boldsymbol{c}_{k_b}\|_2$ is close to $\|\boldsymbol{z}_i - \boldsymbol{c}_{k_a}\|_2$. Particularly, γ_i also suggests the difficulty of an example. An "easy" example is more confident to distinguish between the two nearest centroids and has a large γ_i. Yet, a "hard" example is more likely to be confused by the next nearest centroid and has a small γ_i.

To leverage the distance-aware concept, the intent representation $\boldsymbol{z}_i$ is multiplied by γ_i to obtain the meta-embedding $\boldsymbol{z}_i^{meta}$:

$$\boldsymbol{z}_i^{meta} = \gamma_i \cdot \boldsymbol{z}_i. \tag{7}$$

3) Representation Learning: As the distance-aware coefficient is positively correlated with the magnitude of meta-embedding, it is natural to use the vector length to represent the distance-aware concept. For this purpose, the cosine classifier [24] is adopted to capture the distance information contained in the meta-embedding.

Specifically, the cosine similarity operator is used on the normalized meta-embeddings and weight vectors to compute the classification logits:

$$\phi(\boldsymbol{z}_i^{meta})^k = \alpha \cdot \cos\left(\boldsymbol{z}_i^{meta}, w_k^\star\right) = \alpha \cdot \overline{\boldsymbol{z}_i^{meta}}^\top \overline{w_k^\star}, \tag{8}$$

where $\phi(\cdot)$ is the cosine classifier and $\phi(\cdot)^k$ are the output logits of the k^{th} class, α is a scalar hyper-parameter (detailed discussion can be seen in section VI-A2). $\cos(\cdot)$ is the cosine similarity operator, which first normalizes the meta-embedding $\boldsymbol{z}_i^{meta}$ and the k^{th} class weight vector $w_k^\star$ and then performs dot product operation.

In particular, $\boldsymbol{z}_i^{meta}$ and $w_k^\star$ are applied by a non-linear squashing function [50] and L2 normalization to obtain $\overline{\boldsymbol{z}_i^{meta}}$ and $\overline{w_k^\star}$, respectively:

$$\overline{\boldsymbol{z}_i^{meta}} = \frac{\|\boldsymbol{z}_i^{meta}\|^2}{1 + \|\boldsymbol{z}_i^{meta}\|^2} \frac{\boldsymbol{z}_i^{meta}}{\|\boldsymbol{z}_i^{meta}\|}, \tag{9}$$

$$\overline{w_k^\star} = \frac{w_k^\star}{||w_k^\star||}, \tag{10}$$

where $\|\boldsymbol{z}_i^{meta}\|$ and $\|w_k^\star\|$ denote the magnitudes of $\boldsymbol{z}_i^{meta}$ and $w_k^\star$. The non-linear squashing function is helpful to reflect the magnitude information of a vector. It shrinks $\boldsymbol{z}_i^{meta}$ with a large magnitude to the length slightly below 1 and with a short magnitude to the length of almost 0. The L2 normalization eliminates the effect of the weight vector magnitudes on classification logits.

The cosine classifier converts the magnitude information to discriminative classification outputs and facilitates paying more attention to the "hard" example, which outputs lower classification scores with a smaller magnitude.

Finally, we use the softmax loss $\mathcal{L}_s$ to train the meta-embeddings under the supervision of known intent targets:

$$\mathcal{L}_s = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\phi(\boldsymbol{z}_i^{meta})^{y_i})}{\sum_{j=1}^{K} \exp(\phi(\boldsymbol{z}_i^{meta})^j)}, \tag{11}$$

where y_i is the label of the i^{th} example. The initial intent representations can be calibrated to distance-aware concepts during training, and they are further used for learning decision boundaries.

C. Adaptive Decision Boundary Learning

An original approach to learning the adaptive decision boundary (ADB) is designed for open intent detection. We first formulate the decision boundary and then propose our boundary learning strategy for optimization.

1) Decision Boundary Formulation: It has been shown the superiority of the spherical shape boundary for open world classification [10], which greatly reduced the open space risk compared with the half-space binary linear classifier [26] and two parallel hyper-planes [7]. Inspired by this, we hope to construct ball-like decision boundaries in the deep feature space for open intent detection.

Due to the specificity of different intent categories, we aim to learn the corresponding decision boundaries for each known class. Concretely, for the k^{th} known class, the spherical decision boundary is determined by its corresponding centroid $\boldsymbol{c}_k$ and radius Δ_k, where $k \in \{1, 2, \cdots, K\}$.

The centroid $\boldsymbol{c}_k$ is the average intent representation in class k, as defined in section IV-B1. As the decision boundaries need to be adaptive to the intent feature space, the radius should be learnable to control the space range of the closed ball area. For this purpose, Δ_k is learned by the neural network with a boundary parameter $\widehat{\Delta_k} \in \mathbb{R}$. As suggested in [51], the $\mathrm{Softplus}$ activation function is utilized as the mapping between Δ_k and $\widehat{\Delta_k}$:

$$\Delta_k = \log\left(1 + \exp(\widehat{\Delta_k})\right). \tag{12}$$

TABLE I
STATISTICS OF BANKING, OOS AND STACKOVERFLOW DATASETS. # INDICATES THE TOTAL NUMBER OF UTTERANCES.

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

The $\mathrm{Softplus}$ activation function is selected for the following reasons. Firstly, it guarantees differentiability with different $\widehat{\Delta_k} \in \mathbb{R}$ and supports stable optimization. Secondly, it ensures the radius Δ_k is above zero. Finally, it achieves linear characteristics like ReLU and allows for bigger Δ_k if necessary.

2) Boundary Learning: After formulating the centroid $\boldsymbol{c}_k$ and the radius Δ_k, a critical problem is how to find the suitable decision boundary with the prior knowledge of only known intent feature distributions. For each known class, a tight decision boundary should balance both empirical and open space risks [7]. That is, a tradeoff between both inside and outside examples belonging to its class is required.

For each example $(\boldsymbol{z}_i, y_i)$, if $\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 > \Delta_{y_i}$, the decision boundaries are too small to contain their corresponding known intent examples, which may increase the empirical risk. In contrast, if $\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 < \Delta_{y_i}$, though larger decision boundaries are beneficial to identify more known intent examples, they are more likely to introduce more open intent examples, which may increase the open space risk. Thus, the boundary loss $\mathcal{L}_b$ is proposed to make a tradeoff:

$$\begin{aligned}\mathcal{L}_b = \frac{1}{N}\sum_{i=1}^{N}\big[\delta_i\left(\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 - \Delta_{y_i}\right) \\ + \left(1-\delta_i\right)\left(\Delta_{y_i} - \|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2\right)\big],\end{aligned} \tag{13}$$

where δ_i is defined as:

$$\delta_i := \begin{cases} 1, \text{ if } & \|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 > \Delta_{y_i}, \\ 0, \text{ if } & \|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 \le \Delta_{y_i}. \end{cases} \tag{14}$$

The boundary parameter $\widehat{\Delta_k}$ is updated regarding to $\mathcal{L}_b$ as follows:

$$\widehat{\Delta_k} := \widehat{\Delta_k} - \eta\frac{\partial \mathcal{L}_b}{\partial \widehat{\Delta_k}}, \tag{15}$$

where η is the learning rate and $\frac{\partial \mathcal{L}_b}{\partial \widehat{\Delta_k}}$ is computed by:

$$\frac{\partial \mathcal{L}_b}{\partial \widehat{\Delta_k}} = \frac{\sum_{i=1}^{N}\delta'\left(y_i = k\right)\cdot(-1)^{\delta_i}}{\sum_{i=1}^{N}\delta'\left(y_i = k\right)} \cdot \frac{1}{1+e^{-\widehat{\Delta_k}}}, \tag{16}$$

where $\delta'(y_i = k) = 1$ if $y_i = k$ and $\delta'(y_i = k) = 0$ if not. The denominator is guaranteed to be not zero by updating only $\widehat{\Delta_k}$ that has examples belonging to class k in a mini-batch.

The intrinsic properties of the boundary loss $\mathcal{L}_b$ are in favor of learning adaptive decision boundaries. It calculates the Euclidean distance between each example and its centroid $\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2$ and uses the distance to compare with the radius of the corresponding decision boundary Δ_{y_i}, yielding the inside loss or outside loss. Specifically, the inside (or outside) loss is computed by the sum of the distances between the k^{th} class boundary-inside (or boundary-outside) examples and the boundary Δ_k. The suitable decision boundary is a balance between the inside and outside losses. When the inside loss is larger, the cumulative gradients are positive to move inward the decision boundary. Similarly, when the outside loss is larger, the cumulative gradients are negative to make the decision boundary expand outward. This process enables the decision boundaries to be adaptive to the known intent feature space until balance.

D. Open Classification with Decision Boundary

After boundary learning, the learned decision boundaries and centroids are used for inference. For each example $\boldsymbol{z}_i$, it is first recognized as the index k_a of the nearest centroid. Then, the corresponding decision boundary Δ_{k_a} is utilized to detect whether it belongs to the open intent:

$$k_a = \underset{k}{\operatorname{argmin}}\left\{d(\boldsymbol{z}_i, \boldsymbol{c}_k)\right\}_{k \in I^{\text{Known}}}, \tag{17}$$

$$\hat{y} = \begin{cases} \text{Open, if } d(\boldsymbol{z}_i, \boldsymbol{c}_k) > \Delta_{k_a}; \\ k_a, \text{ if } d(\boldsymbol{z}_i, \boldsymbol{c}_k) \le \Delta_{k_a}, \end{cases} \tag{18}$$

where I^{known} is the known intent label set as mentioned in section III and $d(\boldsymbol{z}_i, \boldsymbol{c}_k)$ denotes the Euclidean distance between $\boldsymbol{z}_i$ and $\boldsymbol{c}_k$.

V. EXPERIMENTS

This section introduces the benchmark datasets, baselines, evaluation metrics, experimental settings, and results.

A. Datasets

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

BANKING: A fine-grained dataset in the banking domain [43]. It contains 77 intents and 13,083 customer service queries. We split a validation set of 1,000 samples from the original training set.

OOS: A dataset for intent classification and out-of-scope prediction [44]. It contains 150 intents, 22,500 in-domain queries and 1,200 out-of-domain queries.

StackOverflow: The dataset was published on Kaggle.com. It contains 3,370,528 technical question titles. We use the processed dataset [52], which has 20 different classes and 1,000 samples for each class.

TABLE II
OVERALL PERFORMANCE OF OPEN INTENT DETECTION WITH DIFFERENT KNOWN CLASS RATIOS (25%, 50% AND 75%) AND THEIR MEAN SCORES ON THREE DATASETS. THE PROPOSED METHOD DA-ADB AND ITS VARIANT ADB ARE SIGNIFICANTLY BETTER THAN OTHERS WITH p-VALUE < 0.05 (†) AND p-VALUE < 0.1 (*) USING T-TEST.

Datasets	Methods	25%		50%		75%		Mean	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1
BANKING	MSP	41.84†	50.03†	59.80†	71.40†	75.90†	83.49†	59.18†	68.31†
	SEG	49.73†	52.03†	54.66†	62.86†	64.54†	69.37†	56.31†	61.42†
	OpenMax	47.76†	53.18†	65.53†	74.64†	78.32†	84.95†	63.87†	70.92†
	MDF	77.17†	46.85†	60.18†	64.10†	64.59†	74.76†	67.31†	61.90†
	LOF	66.73†	63.38†	71.13†	76.26†	77.21†	83.64†	71.69†	74.43†
	DOC	70.31†	65.74†	74.60†	78.24†	78.94†	83.79†	74.62†	75.92†
	DeepUNK	70.68†	65.57†	71.01†	75.41†	74.73†	81.12†	72.14†	74.03†
	(K+1)-way	75.43†	68.31†	74.66†	78.13†	79.90†	85.22†	76.66†	77.22†
	ARPL	76.50†	63.77†	75.29†	78.24†	79.26†	85.18†	77.02†	75.73†
	ADB	79.33	71.63	79.61†	81.34†	**81.39**	**86.11**	80.11*	79.69*
	DA-ADB	**81.09**	**73.65**	**81.64**	**82.60**	81.18	85.68	**81.30**	**80.64**
OOS	MSP	49.78†	49.42†	62.71†	70.33†	72.86†	81.61†	61.78†	67.12†
	SEG	53.34†	47.57†	60.54†	62.51†	42.97†	42.49†	52.28†	50.86†
	OpenMax	70.27†	63.03†	80.22†	79.86†	75.36†	71.17†	75.28†	71.35†
	MDF	76.56†	50.34†	60.72†	61.61†	63.98†	72.02†	67.09†	61.32†
	LOF	87.77	78.13	85.22†	83.86†	85.07†	87.20†	86.02†	83.06†
	DOC	86.08†	75.86†	85.19†	83.89†	85.93†	87.87†	85.73†	82.54†
	DeepUNK	87.18*	77.32*	84.95†	83.35†	84.61†	86.53†	85.58†	82.40†
	(K+1)-way	86.98†	76.58†	83.71†	82.85†	85.31†	87.90†	85.33†	82.44†
	ARPL	82.25†	71.62†	78.33†	79.59†	82.66†	86.80†	81.08†	79.34†
	ADB	88.30	78.23	86.54†	85.16	86.99	**88.94**	87.28†	84.11
	DA-ADB	**89.49**	**79.95**	**87.96**	**85.64**	**87.46**	88.47*	**88.30**	**84.69**
StackOverflow	MSP	27.91†	37.49†	53.23†	62.70†	73.20†	78.70†	51.45†	59.63†
	SEG	23.35†	34.59†	43.04†	55.10†	62.63†	69.86†	43.01†	53.18†
	OpenMax	38.97†	45.35†	60.27†	67.72†	75.78†	80.90†	58.34†	64.66†
	MDF	74.10†	53.95†	56.46†	61.47†	62.98†	71.12†	64.51†	62.18†
	LOF	25.02†	35.29†	44.56†	56.57†	65.05†	71.87†	44.88†	54.58†
	DOC	57.75†	57.34†	73.88†	76.80†	80.55†	84.37†	70.73†	72.84†
	DeepUNK	40.03†	45.64†	55.46†	64.78†	71.56†	77.63†	55.68†	62.68†
	(K+1)-way	53.05†	53.12†	63.54†	69.26†	74.72†	79.47†	63.77†	67.28†
	ARPL	62.60†	60.12†	76.04†	78.15†	79.82†	83.97†	72.82†	74.08†
	ADB	86.75	79.85†	86.49†	85.54†	82.89	86.11*	85.38†	83.83†
	DA-ADB	**89.03**	**82.81**	**87.79**	**86.92**	**83.63**	**86.89**	**86.82**	**85.54**

B. Baselines

We compare our approach with state-of-the-art methods in open set recognition: OpenMax [8], DOC [11], ARPL [9] and open intent detection: DeepUnk [3], SEG [20], (K+1)-way discriminative training [21], ADB [5]. Besides, three OOD detection baselines are built for this task: MSP [12], LOF [19], MDF [16]. The detailed information is as follows:

1) MSP: It is a simple baseline that predicts known classes with the maximum softmax probabilities and rejects the negative samples with the threshold of 0.5.

2) LOF: It is a density-based method to detect the low-density outliers as the open-class samples. The hyper-parameters are set as in DeepUnk [3].

3) MDF: It is an unsupervised OOD detection method adapted to our task. We first train the model on ID data by applying both cross-entropy and MLM losses. Once the model is well-trained, we utilize it to identify known intents and extract averaged representations from each pre-trained transformer layer. These representations are then used to calculate Mahalanobis distances, which are concatenated as features. Finally, the features are fed into a one-class SVM to detect OOD samples. For better performance, we choose the RBF kernel and set the lower bound on the fraction of support vectors to 0.1.

4) OpenMax: It first uses the softmax loss to train a classifier on the known intents and then fits a Weibull distribution to the classifier's output logits. The confidence scores are finally calibrated with the OpenMax Layer. The default hyper-parameters in [8] are adopted (Weibull tail size is 20).

5) DOC: It rejects the open class by calculating different probability thresholds of each known class with Gaussian fitting. The default hyper-parameters in [11] are adopted (the number of standard deviations away from the mean is 3).

6) ARPL: It learns representations of reciprocal points by maximizing the variance between them and known-class samples. It combines Euclidean and cosine distances as the metric and sets a learnable margin to constrain the open space. However, the performance collapses after directly applying the approach in computer vision to our task. Thus, we alleviate

TABLE III
FINE-GRAINED PERFORMANCE OF OPEN INTENT DETECTION WITH DIFFERENT KNOWN CLASS RATIOS (25%, 50%, 75%) AND THEIR MEAN SCORES ON THREE DATASETS. THE PROPOSED METHOD DA-ADB AND ITS VARIANT ADB ARE SIGNIFICANTLY BETTER THAN OTHERS WITH p-VALUE < 0.05 (†) AND p-VALUE < 0.1 (*) USING T-TEST.

Datasets	Methods	25%		50%		75%		Mean	
		Open	Known	Open	Known	Open	Known	Open	Known
BANKING	MSP	38.84†	50.62†	42.13†	72.17†	41.64†	84.21†	40.87†	69.00†
	SEG	52.97†	51.98†	42.35†	63.40†	37.58†	69.92†	44.30†	61.77†
	OpenMax	48.52†	53.42†	55.03†	75.16†	53.02†	85.50†	52.19†	71.36†
	MDF	85.70	44.80†	57.72†	64.27†	33.43†	75.47†	58.95†	61.51†
	LOF	72.64†	62.89†	66.81†	76.51†	54.19†	84.15†	64.55†	74.52†
	DOC	76.64†	65.16†	72.66†	78.38†	63.51†	84.14†	70.94†	75.89†
	DeepUNK	76.98†	64.97†	67.80†	75.61†	50.57†	81.65†	65.12†	74.08†
	(K+1)-way	81.52†	67.61†	72.38†	78.29†	62.13†	85.62	72.01†	77.17†
	ARPL	83.17†	62.75†	73.55†	78.36†	59.34†	85.63†	72.02†	75.58†
	ADB	85.05	70.92	79.43†	81.39†	67.34*	**86.44**	77.27†	79.58
	DA-ADB	**86.49**	**72.97**	**82.10**	**82.61**	**69.51**	85.69	**79.37**	**80.51**
OOS	MSP	54.74†	49.28†	57.49†	70.50†	56.26†	81.83†	56.16†	67.20†
	SEG	60.59†	47.23†	61.13†	62.52†	41.60†	42.50†	54.44†	50.75†
	OpenMax	77.51†	62.65†	82.15†	79.83†	75.18†	71.14†	78.28†	71.21†
	MDF	84.89†	49.43†	62.31†	61.60†	51.33†	72.21†	66.18†	61.08†
	LOF	91.96	77.77	87.57†	83.81†	82.81†	87.24†	87.45†	82.94†
	DOC	90.78†	75.46†	87.45†	83.84†	83.87†	87.91†	87.37†	82.40†
	DeepUNK	91.61†	76.95†	87.48†	83.30†	82.67†	86.57†	87.25†	82.27†
	(K+1)-way	91.44†	76.19†	85.84†	82.82†	82.39†	87.95*	86.56†	82.32†
	ARPL	87.83†	71.19†	79.48†	79.59†	77.23†	86.89†	81.51†	79.22†
	ADB	92.36	77.85	88.60†	85.12	84.85†	**88.97**	88.60†	83.98
	DA-ADB	**93.20**	**79.60**	**90.14**	**85.58**	**86.09**	88.49*	**89.81**	**84.56**
StackOverflow	MSP	11.66†	42.66†	26.94†	66.28†	37.86†	81.42†	25.49†	63.45†
	SEG	4.36†	40.63†	4.72†	60.14†	6.38†	74.09†	5.15†	58.29†
	OpenMax	34.52†	47.51†	46.11†	69.88†	49.69†	82.98†	43.44†	66.79†
	MDF	83.03†	48.13†	50.19†	62.60†	28.52†	73.96†	53.91†	61.56†
	LOF	7.14†	40.92†	5.18†	61.71†	5.22†	76.31†	5.85†	59.65†
	DOC	62.50†	56.30†	71.18†	77.37†	65.32†	85.64†	66.33†	73.10†
	DeepUNK	36.87†	47.39†	35.80†	67.67†	34.38†	77.63†	35.68†	65.19†
	(K+1)-way	56.31†	52.48†	53.68†	70.81†	47.57†	81.60†	52.52†	68.30†
	ARPL	68.49†	58.45†	74.49†	78.52†	63.45†	85.34†	68.81†	74.10†
	ADB	90.96	77.62†	87.70†	85.32†	74.10	86.91*	84.25†	83.28†
	DA-ADB	**92.61**	**80.84**	**88.86**	**86.72**	**74.66**	**87.71**	**85.38**	**85.09**

the issue by pre-training with known intents in advance. The probability threshold is set at 0.5 for detecting the open class.

7) DeepUnk: It first uses the margin loss to learn deep features and then detects the unknown class with LOF. The cosine margin and scaling factor are set as 0.35 and 30, respectively.

8) SEG: It incorporates the semantic information of each class into the large margin Gaussian mixture loss [53] for feature representation, followed by a LOF detector.

9) (K+1)-way: It samples OOD data from a different domain (e.g., the SQuaD 2.0 dataset [54]), and treats it as the (K+1)$^{\text{th}}$ open class. The ID and OOD data are jointly trained using soft labels. The temperature parameter is set to 0.2.

10) ADB: It is a variant of DA-ADB, which also learns adaptive decision boundaries based on the known intent feature space. The difference is that it uses softmax loss rather than leveraging the distance-aware concepts during pre-training.

In particular, all methods are used the same BERT model as backbones for a fair comparison.

C. Evaluation Metrics

For open intent detection, the accuracy score (ACC) and the macro F1-score over all classes (F1) are used to evaluate the overall performance. The macro F1-score over known classes (F1_{known}) and over the open class (F1_{open}) are used to evaluate the fine-grained performance.

Given a set of classes $C = \{C_1, \cdots, C_K, C_{K+1}\}$, where K is the number of known classes and C_{K+1} is the open class. The macro F1-score over all classes (F1) is computed by:

$$\text{F1} = 2 \times \frac{\text{P} \times \text{R}}{\text{P} + \text{R}}, \tag{19}$$

$$\text{P} = \frac{\sum_{i=1}^{K+1} \text{P}_{C_i}}{K+1}, \quad \text{R} = \frac{\sum_{i=1}^{K+1} \text{R}_{C_i}}{K+1}, \tag{20}$$

$$\text{P}_{C_i} = \frac{\text{TP}_{C_i}}{\text{TP}_{C_i} + \text{FP}_{C_i}}, \quad \text{R}_{C_i} = \frac{\text{TP}_{C_i}}{\text{TP}_{C_i} + \text{FN}_{C_i}}, \tag{21}$$

where P and R are the macro precision score and the macro recall score over K+1 classes. P_{C_i} and R_{C_i} are the precision

score and recall score on the C_i class. TP_{C_i}, FP_{C_i} and FN_{C_i} are the true positives, false positives and false negatives of the C_i class, respectively. Similarly, the macro f1-score over known classes (F1_{known}) and the open class (F1_{open}) are computed by:

$$\text{F1}_{\text{known}} = 2 \times \frac{\text{P}_{\text{known}} \times \text{R}_{\text{known}}}{\text{P}_{\text{known}} + \text{R}_{\text{known}}}, \tag{22}$$

$$\text{P}_{\text{known}} = \frac{\sum_{i=1}^{K} \text{P}_{C_i}}{K}, \quad \text{R}_{\text{known}} = \frac{\sum_{i=1}^{K} \text{R}_{C_i}}{K}, \tag{23}$$

$$\text{F1}_{\text{open}} = 2 \times \frac{\text{P}_{C_{K+1}} \times \text{R}_{C_{K+1}}}{\text{P}_{C_{K+1}} + \text{R}_{C_{K+1}}}, \tag{24}$$

where P_{C_i} and R_{C_i} are computed the same as in Eq. 21.

D. Experimental Settings

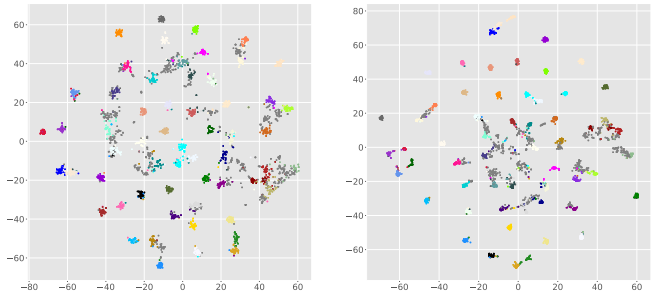
Open intent detection follows the open-world setting [11], which keeps some classes as unknown (open) and integrates them back during testing. Specifically, the proportions of known classes to total categories are varied with 25%, 50%, and 75%. The remaining classes are regarded as one open class. All the datasets are divided into training, validation, and testing sets. The samples from the open class are removed from the training and evaluation sets and only exist in the testing set, as mentioned in section III. To reduce the impact of different selected known intent categories on the performance, we report the average performance over ten runs of experiments for each known class ratio with random seeds of 0-9.

We employ the pre-trained language model BERT (bert-uncased, with 12-layer transformer) implemented in PyTorch [55] and adopt most of its suggested hyper-parameters for optimization. To improve the training efficiency and achieve better performance, we freeze all but the last transformer layer parameters of BERT. The feature dimension D is 768, the training batch size is 128, and the learning rate is 2e-5. For DA-ADB, the scalar α of the cosine classifier is 4, which is searched from $\{2, 4, 8, 16, 32, 64\}$ by combining both feature learning and open classification performance on the evaluation set. The boundary loss $\mathcal{L}_b$ uses Adam [56] to optimize the boundary parameters at a learning rate of 0.05.

E. Results

The main experimental results of open intent detection are presented in Table II and Table III. The best results are highlighted in bold, and Student's t-test is conducted to measure the significance of performance difference between the best-performing method and other methods for each evaluation metric.

Table II shows the overall performance of the accuracy score (ACC) and macro F1-score (F1) over all classes. The proposed approach DA-ADB and its variant ADB achieve the best results in all settings and outperform other baselines significantly. Compared with ADB, DA-ADB yields substantial improvements with fewer known intents (25% and 50%) and achieves competitive results with more known intents (75%), which indicates the effectiveness of the distance-aware concept for representation learning. Compared with the state-of-the-art methods, the average performance of three known-class ratios shows that DA-ADB improves ACC by 4.28% on BANKING, 2.28% on OOS, and 14.07% on StackOverflow, respectively.



(a) Vanilla intent representations. (b) Intent representations learned with distance-aware concepts.

Fig. 3. Visualization of learned intent representations on the BANKING dataset. The open intent samples are marked in gray, and known intents are marked in other colors.

Table III shows the fine-grained performance of the macro F1-score over known classes (F1_{known}) and the open class (F1_{open}). Our approach achieves significant improvements in detecting the open intent and largely enhances the known intent identification performance. We notice that ADB gains the best results over all baselines. It indicates that the learned decision boundaries are suitable to balance both the empirical and open space risks. On this basis, DA-ADB learns more friendly intent representations with distance information, which helps increase $1\% \sim 2\%$ scores for open intent detection.

Moreover, it is worth noting that the improvements on the StackOverflow dataset are much more drastic than the other two datasets. We suppose the reason is that the characteristics of StackOverflow put forward higher requirements for open intent detection. Existing methods are limited to distinguishing the difficult semantic intents as technical question titles in StackOverflow without learning discriminative intent representations and decision boundaries.

VI. Discussions

This section investigates the effect of distance-aware representation, adaptive decision boundary learning, and labeled data. The first subsection visualizes the intent representations to verify the effectiveness of the distance-aware representation learning strategy and analyzes the influence of the hyper-parameter α mentioned in IV-B3. The second subsection compares the performance with different radii to demonstrate the compactness of the learned decision boundaries and visualizes the boundary learning process. The final subsection compares the robustness of different methods with less labeled data.

A. Effect of Distance-aware Representation Learning

1) Visualization of Intent Representations: In Figure 3, we use t-SNE [57] to visualize the learned intent representations on the testing set of the BANKING dataset. It is shown that the vanilla intent representations of known intents are

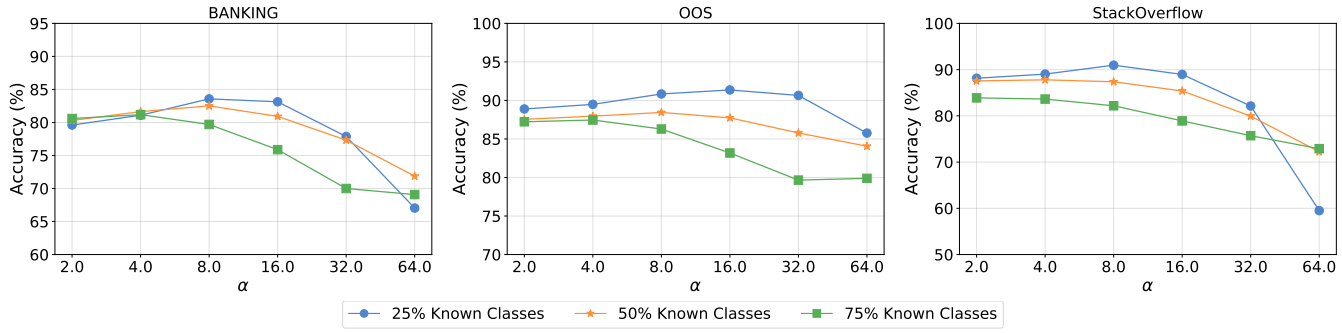


Fig. 4. Influence of α on three datasets with different known class ratios.

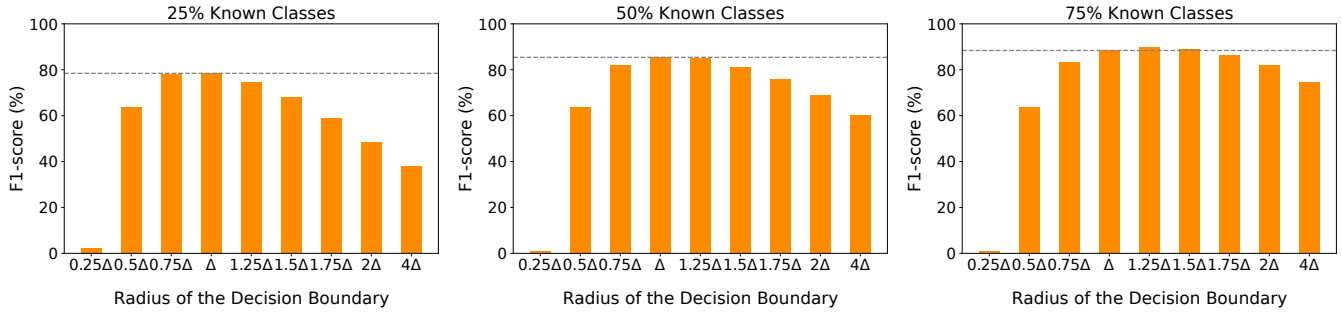


Fig. 5. Influence of the decision boundary on the OOS dataset with different known class ratios.

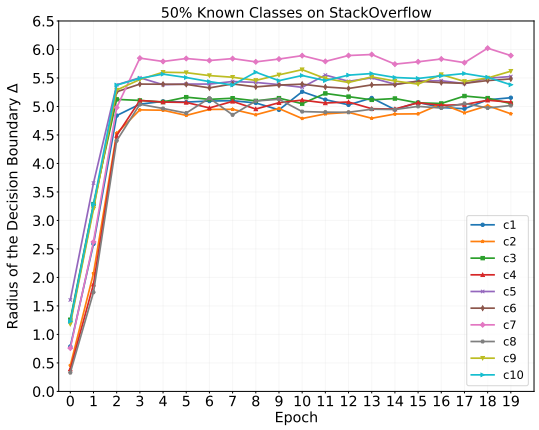


Fig. 6. The boundary learning process.

dispersed, which may result in fuzzy boundaries between adjacent clusters. The open intent representations are distributed haphazardly throughout the feature space, which easily leads to confusion between open and known intent samples.

In contrast, the intent representations learned with distance-aware concepts are more discriminative. The known intent samples are with intra-class compactness and inter-class separation properties. Surprisingly, the distributions of open intent samples are more concentrated to be far away from many known intents, which is beneficial to be better detected.

2) Analysis of Hyper-parameter α: In Figure 4, we use the accuracy score as the metric and show the effect of α on three datasets with different known class ratios. The hyper-parameter α is used to control the logits of the cosine classifier within a desirable range. Intuitively, a large α is better as it can scale the cosine similarity scores and further increase the peakiness of the softmax distribution [24] to enhance the discrimination of the intent representations.

However, it is interesting to observe that though a larger α may achieve higher performance with fewer known intents (25%), the performance drops rapidly with more known intents (75%). We suppose the reason is that the discriminative feature distributions help learn compact decision boundaries, which works better when there are fewer known intent samples. Nevertheless, it may not be good to identify more known intent samples with tight decision boundaries.

B. Effect of Adaptive Decision Boundary Learning

1) Analysis of Radius of Decision Boundary Δ: To verify the discrimination of the learned decision boundary, we use different ratios of Δ (radius of the decision boundary) for open intent detection on the BANKING dataset and show the results in Figure 5. The dotted lines indicate the performance with our learned Δ.

DA-ADB achieves the best or competitive performance with Δ among all assigned decision boundaries, which verifies the tightness of the learned decision boundary. Though 1.25Δ is slightly better on 75% known classes, it is lower on the other two settings. We notice that the performance of open intent detection is sensitive to the size of the decision boundaries. Overcompact decision boundaries will increase the open space risk by misclassifying more known intent samples as the open intent. Correspondingly, overrelaxed decision boundaries will increase the empirical risk by misclassifying more open intent

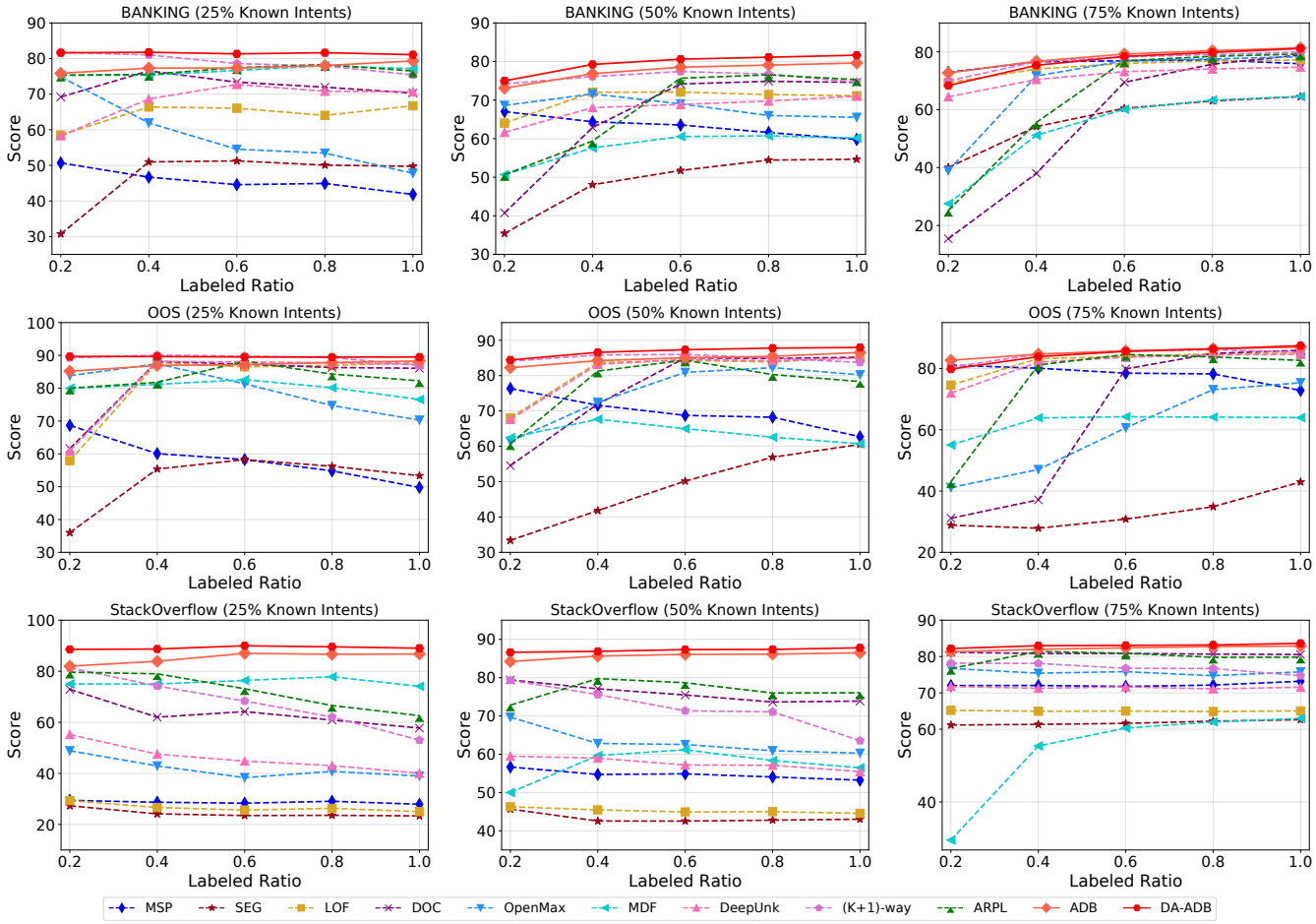


Fig. 7. Influence of the labeled ratio on the three datasets with different known class ratios.

samples as known intents. The two cases both perform worse compared with Δ.

2) Boundary Learning Process: Figure 6 shows the decision boundary learning process. At first, most parameters are assigned small values near zero after initialization, which leads to a small radius with the Softplus activation function. As the initial radius is too small, the empirical risk is dominant. 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 and open space risks.

C. Effect of Labeled Data

To investigate the influence of labeled data, we vary the labeled ratio in training set to 0.2, 0.4, 0.6, 0.8, and 1.0. The accuracy score is used to evaluate the performance. As shown in Figure 7, DA-ADB outperforms all the other baselines on three datasets in almost all settings. Besides, it keeps a more robust performance under different labeled ratios than other methods.

Notably, the probability-based methods (e.g., MSP, DOC, OpenMax, and (K+1)-way) show better performance with less labeled data in many cases. 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 the strong feature extraction capability of DNNs [58] and suffer performance degradation. We also notice that the performance of OpenMax and ARPL is unstable. The former computes centroids of each known class with only corrective positive training samples, and the labeled ratio may easily influence the qualities of centroids. The latter faces a larger open space risk when learning reciprocal points with less prior knowledge of labeled data. Compared with the methods mentioned above, (K+1)-way achieves more robust performance by using pseudo samples as the open class.

In addition, the feature-based methods (e.g., SEG, LOF, DeepUnk) adopt a density-based novelty detection algorithm to perform open intent detection. These methods largely depend on the prior knowledge of labeled data, and their performance all drop dramatically with less labeled data. MDF employs an SVM-based method to detect outliers using Mahalanobis distance information, but it is also sensitive to the amount of labeled data and has limitations in leveraging the

prior knowledge, particularly with more known intents. ADB also shows excellent and robust performance with appropriate learned decision boundaries, but it performs worse in many settings without utilizing the distance information during feature learning.

VII. Conclusions

This paper focuses on a substantial problem, open intent detection in NLU. This problem uses only known intents as prior knowledge, and the goal is not only to identify these known intents but also to detect the open intent. Obtaining friendly intent representations and appropriate decision boundaries are two critical challenges for open intent detection. To solve these problems, we propose a novel pipeline framework, DA-ADB, which learns discriminative features with distance-aware concepts and learns suitable decision boundaries by balancing both empirical and open space risks. We conducted extensive experiments on three benchmark datasets to show the superiority of the proposed method. Our approach yields significant improvements over state-of-the-art methods and achieves robust performance with different known intents and labeled data ratios.

Acknowledgments

This paper is founded by National Natural Science Foundation of China (Grant No. 62173195) and Beijing Academy of Artificial Intelligence (BAAI). Sincerely, we thank the help and constructive feedback of Ting-En Lin.

References

[1] L. Qin, T. Xie, W. Che, and T. Liu, "A survey on spoken language understanding: Recent advances and new frontiers," in *Proceedings of IJCAI*, 2021, pp. 4577–4584, survey Track.

[2] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot filling models in natural language understanding," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, 2022.

[3] E. Lin and H. Xu, "Deep unknown intent detection with margin loss," in *Proceedings of ACL*, 2019, pp. 5491–5496.

[4] T.-E. Lin and H. Xu, "A post-processing method for detecting unknown intent of dialogue system via pre-trained deep neural network classifier," *Knowledge-Based Systems*, vol. 186, pp. 104 979–104 989, 2019.

[5] H. Zhang, H. Xu, and T.-E. Lin, "Deep open intent classification with adaptive decision boundary," in *Proceedings of AAAI*, vol. 35, no. 16, 2021, pp. 14 374–14 382.

[6] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, 2021.

[7] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 35, pp. 1757–1772, 2013.

[8] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of CVPR*, 2016, pp. 1563–1572.

[9] G. Chen, P. Peng, X. Wang, and Y. Tian, "Adversarial reciprocal points learning for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[10] G. Fei and B. Liu, "Breaking the closed world assumption in text classification," in *Proceedings of NAACL-HLT*, 2016, pp. 506–514.

[11] L. Shu, H. Xu, and B. Liu, "Doc: Deep open classification of text documents," in *Proceedings of EMNLP*, 2017, pp. 2911–2916.

[12] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proceedings of ICLR*, 2017.

[13] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proceedings of ICLR*, 2018.

[14] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *Proceedings of ICLR*, 2018.

[15] J.-K. Kim and Y.-B. Kim, "Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates," in *Proceedings of INTERSPEECH*, 2018, pp. 556–560.

[16] K. Xu, T. Ren, S. Zhang, Y. Feng, and C. Xiong, "Unsupervised out-of-domain detection via pre-trained transformers," in *Proceedings of ACL-IJCNLP*, 2021, pp. 1052–1061.

[17] Y. Shen, Y.-C. Hsu, A. Ray, and H. Jin, "Enhancing the generalization for intent classification and out-of-domain detection in slu," in *Proceedings of ACL*, 2021, pp. 2443–2453.

[18] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of CVPR*, 2018, pp. 5265–5274.

[19] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of ACM SIGMOD*, 2000, pp. 93–104.

[20] G. Yan, L. Fan, Q. Li, H. Liu, X. Zhang, X.-M. Wu, and A. Y. Lam, "Unknown intent detection using Gaussian mixture model with an application to zero-shot intent classification," in *Proceedings of ACL*, 2020, p. 1050–1060.

[21] L.-M. Zhan, H. Liang, B. Liu, L. Fan, X.-M. Wu, and A. Y. Lam, "Out-of-scope intent detection with self-supervision and discriminative training," in *Proceedings of ACL*, 2021, pp. 3521–3532.

[22] Z. Cheng, Z. Jiang, Y. Yin, C. Wang, and Q. Gu, "Learning to classify open intent via soft labeling and manifold mixup," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 635–645, 2022.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[24] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of CVPR*, 2018, pp. 4367–4375.

[25] H. Zhang, X. Li, H. Xu, P. Zhang, K. Zhao, and K. Gao, "TEXTOIR: An integrated and visualized platform for text open intent recognition," in *Proceedings of ACL: System Demonstrations*, pp. 167–174.

[26] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[27] R. M. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[28] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Proceedings of ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 393–409.

[29] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014.

[30] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Learning placeholders for open-set recognition," in *Proceedings of CVPR*, 2021, pp. 4401–4410.

[31] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv preprint arXiv:2110.11334*, 2021.

[32] Y. Yu, W.-Y. Qu, N. Li, and Z. Guo, "Open-category classification by adversarial sample generation," in *Proceedings of IJCAI*, 2017, pp. 3357–3363.

[33] S. Ryu, S. Koo, H. Yu, and G. G. Lee, "Out-of-domain detection based on generative adversarial network," in *Proceedings of EMNLP*, 2018, pp. 714–718.

[34] Y. Zheng, G. Chen, and M. Huang, "Out-of-domain detection for natural language understanding in dialog systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1198–1209, 2020.

[35] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" in *Proceedings of ICLR*, 2019.

[36] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proceedings of NeurIPs*, vol. 31, 2018.

[37] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proceedings of NeurIPs*, vol. 33, 2020, pp. 21 464–21 475.

[38] C. Zhang, Y. Li, N. Du, W. Fan, and P. Yu, "Joint slot filling and intent detection via capsule neural networks," in *Proceedings of ACL*, 2019, p. 5259–5267.

[39] H. E, P. Niu, Z. Chen, and M. Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proceedings of ACL*, 2019, pp. 5467–5471.

[40] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proceedings of EMNLP*, 2019, p. 2078–2087.
[41] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Proceedings of a Workshop on Speech and Natural Language*, 1990, pp. 96–101.
[42] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.
[43] I. Casanueva, T. Temcinas, D. Gerz, M. Henderson, and I. Vulic, "Efficient intent detection with dual sentence encoders," in *Proceedings of ACL WorkShop*, 2020.
[44] S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee, P. Hill, J. K. Kummerfeld, K. Leach, M. A. Laurenzano, L. Tang, and J. Mars, "An evaluation dataset for intent classification and out-of-scope prediction," in *Proceedings of EMNLP-IJCNLP*, 2019, pp. 1311–1316.
[45] T. Brychcin and P. Král, "Unsupervised dialogue act induction using gaussian mixtures," in *Proceedings of EACL*, 2017, pp. 485–490.
[46] J. Zhang, K. Hashimoto, W. Liu, C.-S. Wu, Y. Wan, P. Yu, R. Socher, and C. Xiong, "Discriminative nearest neighbor few-shot intent detection by transferring natural language inference," in *Proceedings of EMNLP*, 2020.
[47] C. Xia, C. Zhang, X. Yan, Y. Chang, and P. Yu, "Zero-shot user intent detection via capsule neural networks," in *Proceedings of EMNLP*, 2018, pp. 3090–3099.
[48] T.-E. Lin, H. Xu, and H. Zhang, "Discovering new intents via constrained deep adaptive clustering with cluster refinement," in *Proceedings of AAAI*, 2020, pp. 8360–8367.
[49] H. Zhang, H. Xu, T.-E. Lin, and R. Lyu, "Discovering new intents with deep aligned clustering," in *Proceedings of the AAAI*, 2021, pp. 14 365–14 373.
[50] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proceedings of NeurIPs*, vol. 30, 2017, pp. 3856–3866.
[51] M. Tapaswi, M. T. Law, and S. Fidler, "Video face clustering with unknown number of clusters," in *Proceedings of ICCV*, 2019, pp. 5026–5035.
[52] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, and H. Hao, "Short text clustering via convolutional neural networks," in *Proceedings of NAACL-HLT*, 2015, pp. 62–69.
[53] W. Wan, Y. Zhong, T. Li, and J. Chen, "Rethinking feature distribution for loss functions in image classification," in *Proceedings of CVPR*, 2018, pp. 9117–9126.
[54] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," in *Proceedings of ACL*, 2018, pp. 784–789.
[55] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of EMNLP: System Demonstrations*, 2020, pp. 38–45.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[57] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.
[58] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of CVPR*, 2015, pp. 427–436.

.

Hanlei Zhang received the B.S. degree from the Department of Computer Science and Technology, Beijing Jiaotong University, in 2020. He is currently working toward the Ph.D. degree in the Department of Computer Science and Technology, Tsinghua University. He has published five papers in international conferences, including AAAI, ACM MM, and ACL. His research interests include intent detection and discovery, clustering, multimodal machine learning, natural language processing, etc.

Hua Xu received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 1998, and the M.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 2000 and 2003, respectively. He is a Tenured Associate Professor in the Department of Computer Science and Technology, Tsinghua University. His research interests include data mining, intelligent information processing, and advanced process controllers for IC manufacturing equipment. He has published over 130 peer-reviewed publications in top-tier international journals and conferences. Prof. Xu received Second Prize from the National Science and Technology Progress of China, First Prize from Beijing Science and Technology, and Third Prize from Chongqing Science and Technology.

Shaojie Zhao received the B.S. degree from Hebei University of Science and Technology, in 2020. He is currently working toward the M.S. degree with the the School of Information Science and Engineering, Hebei University of Science and Technology. He has published one paper in the ACM MM international conference. His research interests include intent detection, open world classification, and natural language processing.

Qianrui Zhou received the B.S. degree from the Department of Computer Science and Technology, Tsinghua University, in 2022. He is currently working toward the Ph.D. degree in the Department of Computer Science and Technology, Tsinghua University. His research interests include intent detection, intent discovery, and multimodal machine learning. He has published two papers in international conferences, including ACM MM and ACL. His research interests include natural language processing and multimodal machine learning.