Out-of-Scope Intent Detection with Supervised Deep Metric Learning

Youwen Zhang[1], Xudong Wang*[1], Linlin Wang[1], Ke Yan[2], and Huan Chen[1]
[1]Hello Inc., China
[2]National University of Singapore, Singapore
{zhangyouwen924,wangxudong108,wanglinlin231,shiwan}@hellobike.com
yanke@nus.edu.sg

***Abstract*—Detecting Out-of-Scope(OOS) intents in dialogue systems is a challenging technique with practical applications. As for OOS intent detection, it not only ensures the accuracy of classifying known intents but detecting OOS intents is also crucial. Current related models are limited in learning decision boundaries or setting the threshold of confidence score, which all neglect that a well-formed intent representation is a key point. Meanwhile, text extractors trained by traditional cross-entropy loss merely focus on reducing the error rate of the class to which the sample is classified. In this paper, we propose an effective feature extraction method based on deep metric learning to construct the triplet network with prior knowledge. With the constructed triplet loss, mining hard samples, which refers to the far-apart intents between the same class and close intent representations among different classes, can further obtain discriminative intent representations. In addition, we also introduce adversarial training to make intent representations more robust. Experiments on three public datasets prove the effectiveness of our proposed method of learning discriminative intent representations.**

***Index Terms*—Deep metric learning, Out-of-Scope intent detection, Intent representation**

I. Introduction

As the core module of the dialogue system, most intent recognition methods adopt a classification model, which locates user utterances to a certain type of intent class and gives the corresponding replies. However, the existing knowledge base cannot cover all the intents that users want to ask. Taking Fig.1 as an example, the robot misclassifies the user's utterance ”give this novel 5 stars” as AddToPlayList which belongs to the known intent knowledge base. In effect, the utterance is supposed to be properly identified as an open intent. That is to say, traditional dialogue systems will map text to known intents with a high probability in known designed class, resulting in a situation of ”wrong answer to the question that the user asks”, which will bring a poor user experience. Therefore, it is a significant task of researchers and operations to detect Out-of-Scope intents, which can improve user experience and reduce false-positive error rate.

OOS intent detection is quite challenging, the prominent problem of which is that the class and quantity of OOS intent samples can not be known [1]. In view of this situation, the common way is to train an N-class classification and obtain the maximum softmax probability as a confidence score when testing a sample. If the score of the test sample is lower than a predefined threshold, it is identified as an OOS sample. However, the predefined threshold needs to be manually set. Further, some related methods are proposed for the learnable decision boundary such as LOF [2], and ADB [3]. However, in the step of text feature extraction, previous works employ such as BiLSTM or large pre-trained model BERT [4] and adopt simple cross-entropy loss. The intent embedding space trained by cross-entropy loss may not have much space for new intents and tends to be long and narrow simultaneously, making learning decision boundaries hard. Thus, only using the single cross-entropy loss is not ideal for text encoders of user intents when handling the new intent detection task [5], [6].

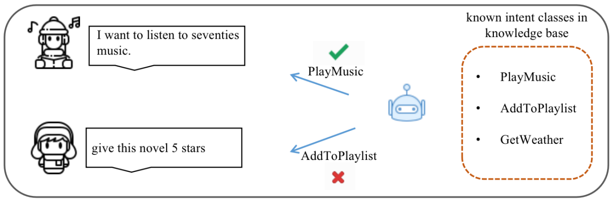


Fig. 1: An example of intent classification. The robot classifies the user utterances as existing known intent classes and lacks the ability to detect open intents.

Theoretically, if a discriminative text representation can be obtained, it will be of great help to the classification of known and open intents when testing. Even though previous works [6], [7] improve the cross-entropy loss, they still do not inherently solve the problem of hard intents interfering with learning decision boundaries. Specifically, SCL [7] improves large margin cosine loss proposed in LMCL with the guidance of contrastive learning. SEG [6] trains with large margin loss and assumes that intent representation follows a Gaussian mixture distribution.

In order to solve this problem, we propose an effective feature extractor based on deep metric learning by constructing informative triplets in the stage of learning intent representation. The essence of deep metric learning is similarity learning, bringing similar samples closer and pushing away dissimilar samples. The form of triplet is a widely used loss function

*Corresponding author. The first two authors contribute equally.

based on deep metric learning. An efficient sampling strategy is introduced for selecting the combination of positive and negative samples in triplets.

Concretely, we use BERT to encode text embedding by training an N-class classifier with known intent as prior knowledge. One part structure of the feature extractor trains the N-class classifier by minimizing the cross-entropy loss with known intents so that roughly similar intent representations within the same class can be learned. Nevertheless, hard intents in the same class and different classes still exist, which will severely interfere with the classification of known and unknown intents in the reasoning phase. Then, we construct triplets in the form of (anchor, positive, negative) for each target anchor intent and minimize the triplet loss. Via pulling hard positive intent feature vectors in the same class and pushing away hard negative intent feature vectors among different classes, the embedding space learned from the synergy of cross-entropy loss and triplet loss shows the phenomenon that intent clusters between different classes are more dispersed in embedding space, which leaves enough space for OOS intents. Our code is available at https://github.com/developer36/Out-of-Scope-Intent-Detection-with-Supervised-Deep-Metric-Learning.git. We summarize our main contributions of the paper as follows:

- We propose a novel feature extractor for learning intent representations by introducing deep metric learning, solving the problem that ”hard” intents in both the same and different classes lead to decision boundaries difficult to learn.
- We are the first to apply the Fast Gradient Method in the task of OOS intent detection.
- We conduct extensive experiments on three public dialogue datasets to demonstrate the effectiveness of our proposed method when conducting OOS detection.

II. Related Work

A. Out-of-Scope Intent Detection

Out-of-Scope(OOS) intent detection, which is also referred to as open intent detection [3], unknown intent detection [8], or Out-of-Domain(OOD) detection [9] in some related works, trains a model to identify known intents and unknown intents without labeled data of unknown intents.

There are mainly three kinds of OOS intent detection works. The first group is to set the threshold for OOS intent rejection according to the confidence value. DOC trains each binary classifier for m known intent classes and then rejects OOS intents that have lower confidence values than all the thresholds. D2U [10] assumes that OOS samples obey a uniform distribution, and then compares it with the threshold by calculating the distance between the output distribution of the test intent sample and the uniform distribution.

The second category of OOS detection work is data augmentation that produces pseudo OOS samples, which strongly rely on the data. For example, (K+1)-way generates OOS samples by combing inliers in a self-supervised manner and sampling outliers. Ryu et al. [9] proposes a GAN-based OOS detector but the model uses in-domain samples.

The third group mainly focuses on optimizing text representation. LMCL replaces softmax loss with large margin cosine loss, making the model distinguishes the relationship between different classes. Inspired by LMCL, Zeng et al. [7] proposed a contrastive learning-based model to maximize inter-class variance and minimize intra-class variance. SEG [6] proposes a novel semantic-enhanced Gaussian mixture model which forms ball-like dense clusters for enforcing feature embeddings and making it easier for OOS intent detection.

Besides, some work performs OOS detection from other novel perspectives. Cavalin et al. [11] utilized the training corpus to construct a word graph and then learned the intent class embedding according to the deepwalk [12] algorithm so that mapping the intent embedding to the class embedding decides whether to reject or not with the threshold. Although there are many angles to improve OOS intent detection, learning well discriminative text representation is of great importance.

B. Deep Metric Learning

Traditional metric learning measures the similarity between samples by an optimal distance metric, which is limited in solving non-linear characteristics [13]. The combination of deep learning and metric learning avoids the process of feature engineering, which is benefited from the non-linear fit ability of activation functions, making similarities among samples more accurate. Deep metric learning [14] [15] has three main parts, which are selecting informative samples, model structure, and a suitable metric loss function.

FaceNet [16] proposes triplet loss which has been extensively studied in the field of person re-identification [17] [18]. Besides the field of computer vision, there are some researches conducted on text either. [19] trains on paired sentences by using the similarity function of Manhattan distance based on a siamese adaptation. [20] solves text clustering by utilizing the triplet network, training sentences within the same section closer than the different sections. [14] shows that the triplet network has more effective performance than Siamese networks, even though they are both belonging to metric learning.

Recently, research on contrastive learning is very prevalent. It is similar to the idea of metric learning, but contrastive learning is an unsupervised or self-supervised learning method, while metric learning is generally a supervised learning method. Moreover, the loss of contrastive learning is formed of a single positive example and multiple negative examples, the key point of which is to construct effective positive examples. For example, SimCSE [21] obtains positive examples by encoding a given input twice with a pretrained language model. Metric learning is mostly in the form of binary or triplets.

III. Methodology

A. Overall Architecture

Fig.2 shows the overall architecture of our proposed feature extraction method for N-class intents. The part of Triplet Loss

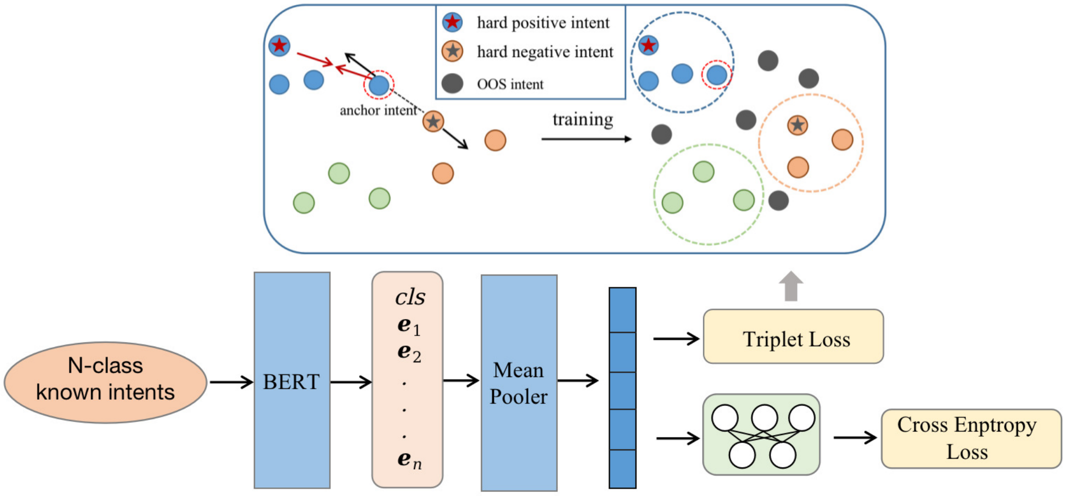


Fig. 2: The overall architecture of our proposed intent feature extractor for open intent detection. We acquire discriminative intent representations by utilizing text encoder BERT trained with the synergy of cross-entropy loss and triplet loss on N-class known intents. The detailed sampling strategy of triplet loss is shown in the upper part.

shows the strategy of how to select hard positive and hard negative samples for triplet. We use BERT as text encoder to obtain low-dimensional embeddings of user utterances.

Simple Cross Entropy Loss. Taking the user utterance $s_i = [t_1, t_2, ...t_n]$ as an example. The output $o = [CLS, \mathbf{e}_1, \mathbf{e}_2, ...\mathbf{e}_n]$ can be obtained in the last hidden layer of BERT. We obtain the intent representation $\mathbf{x}_i$ by performing mean-pooling on o followed by a dense layer to extract deep features:

$$\mathbf{x}_i = \text{mean-pooling}(o) \in R^d \tag{1}$$

where d=768. Then, we minimize the cross-entropy loss function, which is one of the steps in our feature extracting for learning discriminative intent representations:

$$L_{CE} = \frac{1}{N} \sum_{i=1}^{N} -\log \frac{\mathbf{e}^{W_{y_i}^T x_i}}{\sum_{k=1}^{K} \mathbf{e}^{W_{y_k}^T x_i}} \tag{2}$$

where y_i is the true class of the i-th intent. N and W_{y_i} are the number of training samples and weight vector of the k-th class respectively.

Efficient Triplet Loss. Cross entropy is the most effective and simple loss function for classification. However, it only learns the maximum probability of the target true class and does not take into account the similarities and differences of samples between different classes. The open intent classification task uses the known intent as prior knowledge and uses cross entropy as the loss function, which inevitably has deficiencies in learning decision boundaries.

Therefore, we employ deep metric learning when extracting intent features for its better measuring of the distance between intents, which is vital to learn discriminative intent representations. Triplet Loss is a common form based on deep metric learning, and it is widely used especially in the field of face recognition and clustering. Specifically, triplet loss is often presented in the form of a triplet (anchor, positive, negative). Positive represents a positive sample of the same class as the anchor, and negative represents a negative sample of a different class from the anchor sample. By setting a reasonable margin, the goal of the triplet loss is that the distance $d(r_a, r_n)$ between the anchor and negative sample is larger than the distance $d(r_a, r_p)$ between the anchor and the positive sample representation.

By constructing triplets (anchor, positive, negative) to mine hard positive and negative intents, the key point lies in how to select informative positive and negative samples. Above all, in view of the quality of the triplet, selecting very similar positive samples and very different negative samples makes the network always learn simple samples and reduces the generalization. In the view of the number of triplets, taking N classes as an example, each class has N_i samples, then the triplet combination has a total of ways. So we need to dig out effective triplets. Inspired by [22], we traverse each anchor intent in the batch and find the intent that is farthest from it in the same class as the hardest positive sample. Similarly, the closest one in different classes is the hardest negative intent.

$$\begin{aligned} L_{ML}(\theta; x) = \sum_{i=1}^{M_B} m + \max_{j=C_i} d\left(f_\theta\left(x_a^{C_i}\right), f_\theta\left(x_p^{C_i}\right)\right) \\ - \min_{\substack{j=1...N \\ j \neq C_i}} d(f_\theta(x_a^{C_i}), f_\theta(x_n^{C_j})) \end{aligned} \tag{3}$$

where m denotes the margin, which controls the distance between positive and negative samples. M_B is the number of anchor samples in each batch and C_i or C_j is the class that the sample belongs to. $d(.,.)$ is the squared Euclidean distance that measures the distance between the anchor intent and positive(negative) intent.

The final loss of the feature extractor is the sum of the cross-entropy loss L_{CE} and the efficient triplet loss based on deep metric learning L_{ML}:

$$L = L_{CE} + L_{ML} \tag{4}$$

B. Adversarial Augmentation

Adversarial training, originally proposed by Ian Goodfellow et al. [23], is an efficient method to defend against adversarial attacks, which can improve model robustness and generalization. Since the input of NLP is one-hot vectors, Goodfellow et al. [24] proposed that adversarial training in NLP is to add perturbation to the word embeddings. Actually, in NLP tasks, adversarial training is no longer to defend against gradient-based malicious attacks, but more as a regularization to improve the generalization ability of the model. Therefore, we apply Fast Gradient method(FGM) [24] to make metric learning more diverse and obtain robust intent representation simultaneously. The perturbation added to the intent representation $\mathbf{x}$ is the following:

$$\boldsymbol{r}_{\text{adv}} = \epsilon \boldsymbol{g} / \|\boldsymbol{g}\|_2 \text{ where } \boldsymbol{g} = \nabla_{\boldsymbol{x}} \log p(y \mid \boldsymbol{x}; \hat{\boldsymbol{\theta}}) \tag{5}$$

Thus, adversarial intent representation $\mathbf{x}_{adv} = x + \mathbf{r}_{adv}$ achieves the situation where perturbation goes along the direction of the gradient to the maximum value of the loss function.

C. Open Classification with Decision Boundary

We adopt ADB [3] to learn an adaptive decision boundary for each class of known intents. Obviously, a smaller radius affects known intent classification, meanwhile, if the radius is too large, open intent can not be detected. Therefore, it is necessary to ensure the balance between the risk of known intention experience and the risk of unknown intention open space. The boundary loss for learning radius is as follows:

$$\begin{aligned} L_b = \frac{1}{N} \sum_{i=1}^{N} \big[& \delta_i \left(\|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 - \Delta_{y_i} \right) \\ & + (1 - \delta_i) \left(\Delta_{y_i} - \|\boldsymbol{z}_i - \boldsymbol{c}_{y_i}\|_2 \right) \big] \end{aligned} \tag{6}$$

$$\boldsymbol{c}_k = \frac{1}{|S_k|} \sum_{(\boldsymbol{x}_i, y_i) \in S_k} \boldsymbol{x}_i \tag{7}$$

Centroid representation c_k is the average of intent representations within the corresponding class. When the pretraining is done, we classify which class the intent belongs to according to the centroids and the learned radius Δ_{y_i}:

$$\hat{y} = \begin{cases} \text{open, if } d(\boldsymbol{x}_i, \boldsymbol{c}_k) > \Delta_k, \forall k \in \mathcal{Y}; \\ \arg\min_{k \in \mathcal{Y}} d(\boldsymbol{x}_i, \boldsymbol{c}_k), \text{ otherwise} \end{cases} \tag{8}$$

In the inference phase, the intent is classified as an open intent if its representation is none within the radius of a known intent. Otherwise, it will be classified into the class with the closest Euclidean distance to the centroid.

TABLE I: The detailed information about the three datasets. #indicates the total number of samples

Dataset	Classes	#Training	#Validation	#Test
Snips	7	13,084	700	700
Banking	77	9,003	1,000	3080
OOS	150	15,000	2,000	6,000

We evaluate the overall performance of all baseline methods and ours by comparing the accuracy and macro-f1 metric. Suppose we have N classes of known intentions, so there are a total of (N+1)-classes, for the (N+1)-th class is the open intent. We denote the (N+1)-classes as $C = (C_1, C_2, ...C_{N+1})$. The accuracy and macro-f1 over C are computed by:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N+1} TP_{C_i}}{\# \text{ test}} \tag{9}$$

$$\text{F1-all} = \frac{2 \times P \times R}{P + R} \tag{10}$$

Meanwhile, to analyze the performance of known and unknown intents respectively, we use the macro-f1 metric for comparison. The calculation of the macro-f1 score for open and known intent is similar to F1-all.

D. Baselines

We extensively compare our proposed method with the blew state-of-the-art open intent detection methods.

- ADB [3] uses a simple cross-entropy loss to pre-train known intentions, and it proposes adaptive decision boundary that can effectively distinguish between known and unknown intentions.
- (K+1)-way [25] trains a (K+1)-class classifier by constructing two different types of pseudo outliers, which come from unrelated intent classes and convex combinations of two random known intents respectively.
- MSP [26] utilizes softmax probability distribution to classify lower maximum confidence as unknown intent.
- SEG [6] is based on the Gaussian model enforcing to learn ball-like intent representations and also adopts LOF to detect open intent like LMCL.
- DOC [27] replaces softmax with 1-vs-m sigmoids and therefore formulates rejection policy.

IV. Experiments

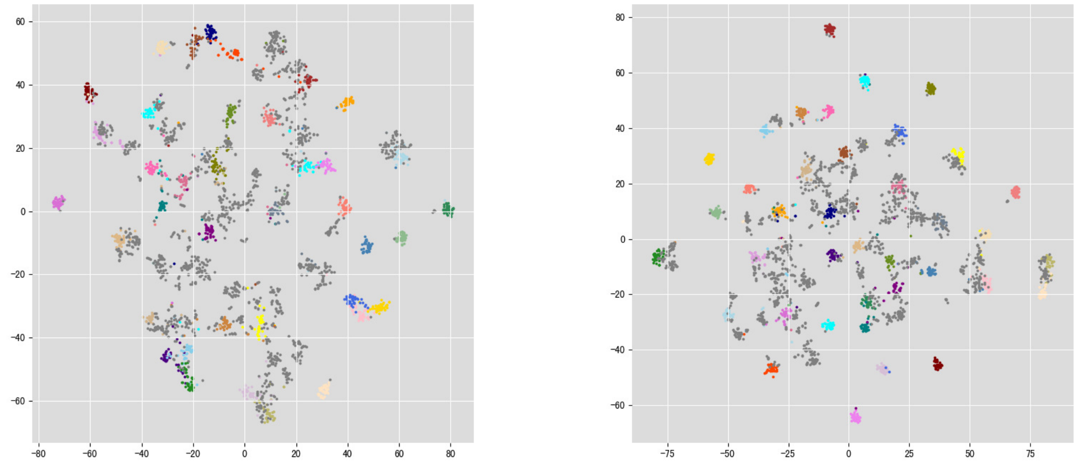
A. Datasets

To demonstrate the improvements of the proposed approach, we perform our experiments on three public dialogue datasets. The detailed information is shown in Table 1.

- **Banking** [28] Banking is a fine-grained intent detection dataset in the banking domain.
- **OOS** [29] OOS is commonly used in text classification and open intent detection with a relative large number intent classes which is 150.
- **Snips** [30] Snips is collected by personal voice assistant which contains 7 types of user intents across different domains.

TABLE II: Overall accuracy and macro-f1 score for open intent detection with three different proportions of known classes on Banking, OOS, and Snips datasets.

	Methods	Banking		OOS		Snips	
		Accuracy	F1-all	Accuracy	F1-all	Accuracy	F1-all
25%	MSP	43.88	51.27	56.27	53.06	28.56	37.80
	DOC	71.87	66.31	86.44	76.18	37.73	47.88
	SEG	51.04	51.97	56.44	49.46	61.56	67.62
	(K+1)-way	76.83	69.53	87.56	77.07	63.77	69.90
	ADB	78.33	70.94	87.71	77.50	66.34	72.43
	Ours	**81.97**	**74.30**	**89.63**	**79.97**	**67.89**	**72.95**
50%	MSP	61.54	72.45	66.90	72.71	58.97	64.59
	DOC	74.42	78.10	85.03	83.84	70.72	76.70
	SEG	55.21	63.15	59.89	62.31	61.13	67.13
	(K+1)-way	75.98	78.89	86.40	84.88	78.66	83.39
	ADB	79.10	81.04	86.25	84.93	79.88	83.70
	Ours	**82.99**	**83.56**	**88.41**	**86.58**	**81.80**	**85.05**
75%	MSP	77.55	84.54	77.04	83.88	72.33	73.79
	DOC	78.84	83.61	85.91	87.87	81.33	84.37
	SEG	64.90	69.77	48.01	48.19	70.94	73.70
	(K+1)-way	80.20	85.26	85.90	88.09	83.29	86.28
	ADB	80.86	85.75	87.11	88.96	85.35	87.88
	Ours	**82.69**	**86.94**	**88.20**	**89.62**	**86.08**	**88.53**



(a) Vanilla intent representations

(b) Learned with deep metric learning

Fig. 3: 2-D visualization of learned intent representations of the Banking dataset with the known ratio of 50%. Gray markers are open intents, and the others are known intents marked with other colors.

TABLE III: Macro-f1 score of unknown(open) intents and known intents with three different proportions of known classes on Banking, OOS, and Snips datasets.

	Methods	Banking		OOS		Snips	
		Open	Known	Open	Known	Open	Known
25%	MSP	42.17	51.75	62.83	52.81	0.00	56.71
	DOC	78.17	65.68	91.05	75.79	22.57	60.53
	SEG	54.93	51.81	64.29	49.07	59.90	71.48
	(K+1)-way	82.77	68.83	91.89	76.68	65.78	71.51
	ADB	84.24	70.24	91.96	77.12	70.17	73.56
	Ours	**87.18**	**73.62**	**93.31**	**79.62**	**70.82**	**74.62**
50%	MSP	45.91	73.14	64.04	72.82	10.03	78.24
	DOC	72.42	78.25	87.26	83.79	49.46	83.51
	SEG	43.73	63.66	59.94	62.34	25.73	77.48
	(K+1)-way	74.44	79.01	88.55	84.83	70.63	86.59
	ADB	78.84	81.10	88.35	84.88	72.32	86.54
	Ours	**83.55**	**83.56**	**90.40**	**86.53**	**75.46**	**87.45**
75%	MSP	48.15	85.16	65.81	84.04	10.12	86.52
	DOC	63.79	83.95	83.30	87.91	55.23	90.20
	SEG	38.41	70.31	45.46	48.21	18.73	84.69
	(K+1)-way	63.48	85.63	83.45	88.13	70.67	90.76
	ADB	66.26	86.09	85.09	88.99	71.45	91.16
	Ours	**71.06**	**87.21**	**86.61**	**89.65**	**74.28**	**91.37**

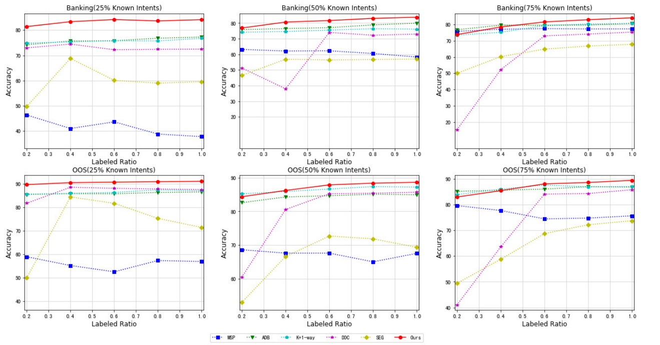
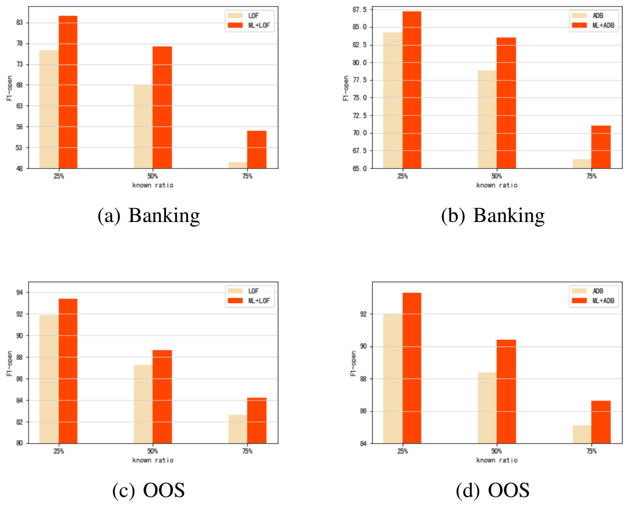


Fig. 4: Effect of labeled ratio on the Banking and OOS datasets with three different proportions of known classes.

B. Experimental Settings and Evaluation Metrics

For a fair comparison with the above baseline methods, we follow the same setting in LMCL. We randomly select 25%, 50%, and 75% of classes from the training set as known intents, and the remaining classes are used as open intent in the test set. We set 10 different random seeds when each training set is randomly divided, and use the average of the ten results as the final evaluation metric. The experiments are conducted by using Pytorch [31] and the intent encoder BERT uses huggingface's transformers [32](bert-base-uncaed). The output of the last hidden layer of BERT is 768 dimension by default. We try both $[CLS]$ which stands for special classification token and average embedding of $[CLS, t_1, t_2, ...t_n]$. We finally adopted the latter as the intent representation for its better performance.



(a) Banking

(b) Banking

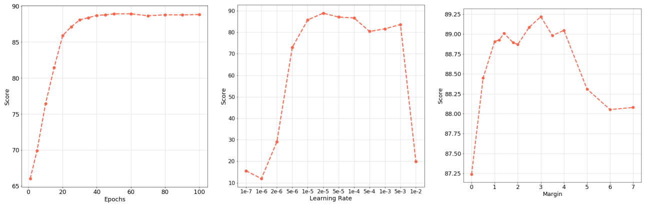
(c) OOS

(d) OOS

Fig. 5: Influence of introducing deep metric learning in the pretraining stage on Banking and OOS datasets.

C. Results and discussion

Result Analysis. The main results of all baseline methods and ours are shown in Table II and Table III. Obviously, our work achieves great progress which is presented in bold in both tables. Specifically, Table II reflects the overall accuracy and macro-f1 score on three different ratios of known intent classes. Compared to the best results among baseline methods, our work significantly improves F1-all on Banking by 3.38%, 2.52%, 1.19%, on OOS by 2.47%, 1.65%, and 0.66%, on Snips



(a) (b) (c)

Fig. 6: Analysis of hyperparameters on the Banking dataset with the known ratio of 25%.

by 0.49%, 1.35%, and 0.65% in the ratio of 25%, 50%, and 75% respectively.

As can be seen from Table III, our method can effectively detect open intentions and simultaneously guarantee classification accuracy on known intents. Taking the ratio of 25% known intents on the Banking dataset as an example, we promote 2.94%, 4.71%, and 4.8% when detecting OOS intents. Comparing the three datasets, our method performs best on Banking. We suppose this is because Banking has a medium size of intent classes, which is 77.

Visualization of known and unknown intent representations. In Fig.3, we use t-SNE [33] to visualize the known and unknown intent representations on the Banking test dataset. Intents of the same class are visualized in the same color. We can see from Fig.3(b) that the intents of the same class are clustered close, and the intents of different classes are also well separable. Moreover, open intents are farther away from known intents, which is beneficial for (N+1)-class classification. Obviously, the introduction of deep metric learning in the stage of extracting text features is of great contribution to learning discriminative known and unknown intent representations. On the contrary, it can be seen from the left visualization without metric learning that the intents of the same class are scattered, and open intentions are mixed with known intents.

Effect of deep metric learning. We further analyze whether introducing deep metric learning in pre-training improves OOS intent detection. We conduct experiments on Banking and OOS datasets using adaptive decision boundary (ADB) and LOF two outlier detection methods. The specific comparison results on classifying open intents are shown in Fig.4. In view of using LOF, our proposed deep metric learning-based method can improve F1-open(F1-unknown) on Banking by around 8%, 9%, and 7%, on OOS by1%, 1.5%, and 1% respectively under the known ratio of 25%, 50%, and 75%. Furthermore, our method can improve F1-open(F1-unknown) on OOS by around 3%, 5%, and 5%, on OOS by 1%, 2%, and 1.5% respectively under the same settings.

Effect of labeled ratio. To analyze the impact of the number of labeled data, we vary the labeled ratio of the training dataset in the range of [0.2, 1.0] with the interval of 0.2. We use the metric of Accuracy to evaluate the performance of the baselines and our proposed method on the known and unknown test intents. In Fig.3, our method outperforms all the baselines on the Banking and OOS datasets and is not sensitive to the labeled ratio, which is plotted in red.

Obviously, the performance of MSP decreases with more labeled intents. The best explanation for this trend is that MSP uses the common softmax, thus DNN's strong non-linear ability to fit known intent leads to bad performance on open intents. SEG is a density-based method for using classic LOF to detect novelty intents, so the number of prior knowledge is quite vital.

In addition, we notice that (K+1)-way is competitive among all baseline methods. It's because (K+1)-way constructs the (K+1)-th pseudo outliers, and well assumes the embedding space of open intents. Meanwhile, the novel adaptive decision boundary method ADB is generally robust, but it is still second to our method and (K+1)-way in terms of specific accuracy score.

Analysis of hyperparameters. We perform hyperparameter analysis on Banking with known intent ratios of 25% and 50%. First, as for the influence of the number of training epochs on detecting open intents, we plot by using the F1-open metric which can be seen in Fig.6(a). Our proposed method achieves the best score when the epoch is around 40 and then tends to steady. Second, as for the analysis of the learning rate, it achieves the best performance of 2e-5. Intuitively, no matter learning rate is too large or too small, it is not facilitated for intent classification. Finally, we vary the value of the margin m to observe its impact on detecting open intents, as shown in Fig.6(c). We notice that better performance can be achieved by setting the margin in the range of 1.2 to 4. Theoretically, a larger margin can enhance the model's discrimination of different classes of intents, but if the margin is set relatively large in the early stage of training, it may increase the difficulty of model training, and then the network will not converge.

V. Conclusion

In this paper, we regard Out-of-Scope intent detection as an N-class classification task during training and propose a novel model for extracting intent representation. Informative hard positive and negative samples are sampled for triplets based on deep metric learning, which can better acquire discriminative intent representations. In addition, adversarial perturbation training in the word embedding space enables our method to generalize well during testing. All the experiments and analysis can confirm the efficiency of our proposed method.

The OOS intent detection method proposed in this paper has been practically applied in the customer service bot of Hello Inc. The number of known intent classes in the existing knowledge base is limited. Users ask related questions according to updated products and the emergence of new policies so that the accuracy of robot intent classification will decrease. Detected valuable open intents can be analyzed to help operators update the classes of known intents, which saves time and promote efficiency.

REFERENCES

[1] C. Liang, P. Huang, W. Lai, and Z. Ruan, "Gan-based out-of-domain detection using both in-domain and out-of-domain samples," in *Proceedings of ICASSP*, 2021, pp. 7663–7667.

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, p. 93–104.

[3] H. Zhang, H. Xu, and T.-E. Lin, "Deep open intent classification with adaptive decision boundary," in *Proceedings of AAAI*, 2021.

[4] 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.

[5] W. Wan, Y. Zhong, T. Li, and J. Chen, "Rethinking feature distribution for loss functions in image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9117–9126.

[6] 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, pp. 1050–1060.

[7] Z. Zeng, K. He, Y. Yan, Z. Liu, Y. Wu, H. Xu, H. Jiang, and W. Xu, "Modeling discriminative representations for out-of-domain detection with supervised contrastive learning," in *Proceedings of ACL*, 2021, pp. 870–878.

[8] 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, p. 104979, 2019.

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

[10] E. Yilmaz and C. Toraman, "D2U: Distance-to-uniform learning for out-of-scope detection," in *Proceedings of ACL*, 2022, pp. 2093–2108.

[11] P. Cavalin, V. Alves Ribeiro, A. Appel, and C. Pinhanez, "Improving out-of-scope detection in intent classification by using embeddings of the word graph space of the classes," 11 2020.

[12] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of KDD*, 2014, p. 701–710.

[13] Kaya and Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

[14] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition*, Cham, 2015, pp. 84–92.

[15] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of CVPR*, 2014, pp. 1875–1882.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of CVPR*, 2015, pp. 815–823.

[17] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proceedings of CVPR*, 2016, pp. 1288–1296.

[18] J. Zhang, J.-P. Ainam, W. Song, L.-h. Zhao, X. Wang, and H. Li, "Learning global and local features using graph neural networks for person re-identification," *Image Commun.*, 2022.

[19] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity." in *Proceedings of AAAI*, 2016, pp. 2786–2792.

[20] L. Ein Dor, Y. Mass, A. Halfon, E. Venezian, I. Shnayderman, R. Aharonov, and N. Slonim, "Learning thematic similarity metric from article sections using triplet networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 49–54.

[21] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6894–6910.

[22] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of ICLR*, 2015.

[24] T. Miyato, A. M. Dai, and I. J. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv: Machine Learning*, 2017.

[25] L.-M. Zhan, H. Liang, B. Liu, L. Fan, X.-M. Wu, and A. Lam, "Out-of-scope intent detection with self-supervision and discriminative training," 06 2021.

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

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

[28] I. Casanueva, T. Temčinas, D. Gerz, M. Henderson, and I. Vulić, "Efficient intent detection with dual sentence encoders," in *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020, pp. 38–45.

[29] 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.

[30] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *ArXiv*, vol. abs/1805.10190, 2018.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019.

[32] 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 the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[33] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," 2008, pp. 2579–2605.