Adversarial Self-Supervised Learning for Out-of-Domain Detection

Zhiyuan Zeng[1], **Keqing He**[1,2], **Yuanmeng Yan**[1], **Hong Xu**[1], **Weiran Xu**[1*]
[1]Pattern Recognition & Intelligent System Laboratory
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group, Beijing, China
{zengzhiyuan,yanyuanmeng,xuhong,xuweiran}@bupt.edu.cn
{hekeqing}@meituan.com

Abstract

Detecting out-of-domain (OOD) intents is crucial for the deployed task-oriented dialogue system. Previous unsupervised OOD detection methods only extract discriminative features of different in-domain intents while supervised counterparts can directly distinguish OOD and in-domain intents but require extensive labeled OOD data. To combine the benefits of both types, we propose a self-supervised contrastive learning framework to model discriminative semantic features of both in-domain intents and OOD intents from unlabeled data. Besides, we introduce an adversarial augmentation neural module to improve the efficiency and robustness of contrastive learning. Experiments on two public benchmark datasets show that our method can consistently outperform the baselines with a statistically significant margin.[1]

1 Introduction

Task-oriented dialog systems (Sarikaya, 2017; Akasaki and Kaji, 2017; Gnewuch et al., 2017; Shum et al., 2018; Tulshan and Dhage, 2018) such as Google's DialogFlow or Amazon's Lex have become ubiquitous to make people interact with machines using natural language. In the architecture of a dialogue system, detecting unknown or OOD (Out-of-Domain) intents from user queries is an essential component that aims to know when a user query falls outside their range of predefined supported intents. Different from traditional intent detection tasks, we do not know the exact number of unknown intents in practical scenarios and can barely annotate extensive OOD samples. Lack of real OOD examples always leads to poor prior knowledge about these unknown intents, making it challenging to identify OOD samples in the task-oriented dialog system.

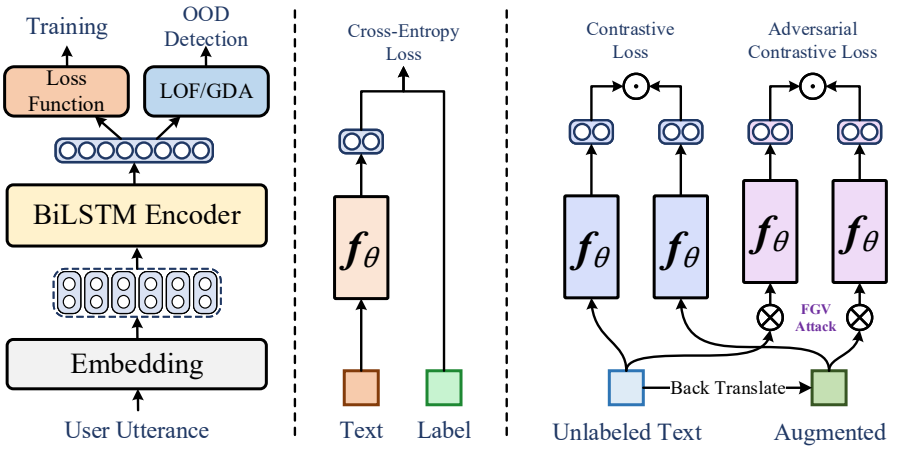
Previous methods of detecting OOD intents can be generally classified into two types: unsupervised and supervised OOD detection. Unsupervised OOD detection (Breunig et al., 2000; Bendale and Boult, 2016; Hendrycks and Gimpel, 2017; Shu et al., 2017; Lee et al., 2018; Ren et al., 2019a; Lin and Xu, 2019; Snell et al., 2017; Finn et al., 2017; Xu et al., 2020) means no labeled OOD samples except for labeled in-domain data. By contrast, supervised OOD detection (Scheirer et al., 2013; Fei and Liu, 2016; Kim and Kim, 2018; Larson et al., 2019; He et al., 2020b; Zheng et al., 2020) represents that there are extensive labeled OOD samples in the training data.

Most of unsupervised OOD detection methods follow a two-stage framework: training and detecting. They first train an in-domain intent classifier to extract intent representations, then detect whether the test query belongs to OOD by estimating its probability density. For example, Hendrycks and Gimpel (2017); Shu et al. (2017) simply use a threshold on the in-domain classifier's probability estimate. Lin and Xu (2019) employs an unsupervised density-based novelty detection algorithm, local outlier factor (LOF) to detect unseen intents. However, such neural models can only extract discriminative features of different in-domain intents since they are trained on the in-domain data without access to OOD data. Therefore, these methods are known to produce highly overconfident posterior distributions even for such abnormal OOD samples (Guo et al., 2017; Liang et al., 2017, 2018). For supervised OOD detection, classical methods such as (Fei and Liu, 2016; Larson et al., 2019), form a $(N+1)$-class classification problem where the $(N+1)$-th class represents the unseen intents. Further, Zheng et al. (2020) uses labeled OOD data to generate an entropy regularization term to enforce the predicted distribution of OOD inputs closer

*Weiran Xu is the corresponding author.

[1]Our code is available at https://github.com/parZival27/Adversarial-Self-Supervised-Out-of-Domain-Detection.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5631–5639
June 6–11, 2021.



a) overall architecture b) supervised training c) self-supervised contrastive training

Figure 1: The overall architecture of our proposed framework. We first train an intent representation extractor using two kinds of objectives: supervised cross-entropy loss on the in-domain data and self-supervised contrastive loss on the unlabeled data. Then we extract the representation of the test query to detect OOD using MSP (Maximum Softmax Probability) (Hendrycks and Gimpel, 2017), LOF (Lin and Xu, 2019) or GDA (Xu et al., 2020).

to the uniform distribution. However, collecting large-scale labeled OOD data is usually difficult and expensive. These drawbacks limit the broad application of supervised OOD detection. In this paper, we aim to capitalize on the benefits of both self-supervised and supervised OOD detection: (1) simultaneously modeling semantic features of both in-domain and OOD data; (2) inducing no labor-intensive OOD annotation.

In this paper, we propose a self-supervised contrastive learning framework to model discriminative semantic features of both in-domain intents and OOD intents from unlabeled data. Without access to labeled OOD data, our method aims to learn representations that discriminate between all unlabeled intents in the instance level. When combined with supervised in-domain training, our method learns features that are both rich and semantically discriminative. Besides, to replace the stochastic data augmentation mechanisms like random cropping, random color distortions in the image processing field (Chen et al., 2020a), we propose an adversarial augmentation neural module to improve the diversity and complexity of pre-defined transformation functions. Specifically, we compute model-agnostic adversarial worst-case perturbations to the inputs in the direction that significantly increases the original contrastive loss. Intuitively, adversarial learning can generate pseudo hard positive pairs thus improve the efficiency and robustness of contrastive learning. Our contributions are three-fold: (1) We propose a self-supervised learning framework to simultaneously modeling semantic features of both in-domain and OOD data. (2) We apply an adversarial augmentation mechanism to improve the efficiency and robustness of self-supervised learning. (3) Experiments conducted on two benchmark OOD datasets show the effectiveness of our proposed method.

2 Approach

Overall Architecture Fig 1(a) shows the overall architecture of our proposed two-stage framework. We first train an in-domain intent classifier to extract intent representations using two objectives then use the detection algorithms MSP (Hendrycks and Gimpel, 2017), LOF (Lin and Xu, 2019) or GDA (Xu et al., 2020) to detect OOD. In the training stage, we first train a BiLSTM in-domain intent classifier similar to Lin and Xu (2019) using labeled in-domain data. Then we apply an adversarial contrastive objective to continue training on the unlabeled data.

Self-Supervised Contrastive Learning To simultaneously model semantic features of both in-domain and OOD data, we propose a self-supervised contrastive learning framework to utilize unlabeled data. Following (Chen et al., 2020a; He et al., 2020a; Chen et al., 2020b; Winkens et al., 2020; Jiang et al., 2020), we formulate the contrastive loss for a positive pair of examples (i, j) as:

$$\ell_{i,j} = -\log \frac{\exp\left(\text{sim}\left(z_i, z_j\right)/\tau\right)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp\left(\text{sim}\left(z_i, z_k\right)/\tau\right)} \quad (1)$$

where z_i represents the feature vector of i-th sentence sample extracted by concatenating the first

and final hidden states of BiLSTM, and $1_{[k \neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if $k \neq i$. τ denotes a temperature parameter. The final loss is computed across all positive pairs, both (i, j) and (j, i) in a mini-batch of N examples. Here we use back-translation as data augmentation to generate positive pairs. Previous work (Chen et al., 2020a) has shown the necessity of more data augmentations, thus we propose an adversarial neural augmentation as follows.

Adversarial Neural Augmentation To improve the diversity of data augmentation and avoid handcrafted engineering, we apply adversarial attack (Goodfellow et al., 2015; Kurakin et al., 2016; Miyato et al., 2016; Jia and Liang, 2017; Zhang et al., 2019; Ren et al., 2019b) to generate pseudo positive samples. It should be noted that samples obtained by adversarial attack is in the form of embedding to ensure end-to-end training. Specifically, we need to compute the worst-case perturbation $\boldsymbol{\delta}$ that maximizes the original contrastive loss $\mathcal{L}$: $\boldsymbol{\delta} = \underset{\|\boldsymbol{\delta}'\| \leq \epsilon}{\arg\max} \; \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\delta}'\right)$, where $\boldsymbol{\theta}$ represents the parameters of a model and $\boldsymbol{x}$ denotes a given sample. ϵ is the norm bound of the perturbation $\boldsymbol{\delta}$. In practical implementation, we apply Fast Gradient Value (FGV) (Rozsa et al., 2016) to approximate the perturbation $\boldsymbol{\delta}$:

$$\boldsymbol{\delta} = \epsilon \frac{g}{||g||}; \text{where } g = \nabla_{(\boldsymbol{x}_i, \boldsymbol{x}_j)} \mathcal{L}(f(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})) \tag{2}$$

where $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represents the original positive pair generated by back-translation. We perform normalization to g and then use a small ϵ to ensure the approximate is reasonable. Finally, we can obtain the pseudo adversarial sample $\boldsymbol{x}_i^{adv} = \boldsymbol{x}_i + \boldsymbol{\delta}$ as well as $\boldsymbol{x}_j^{adv}$. Therefore, we get $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_i^{adv}, \boldsymbol{x}_j^{adv})$ from the original positive pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$. We implement four different contrastive settings: (1) Standard-to-Standard (S2S): the original contrastive loss using $(\boldsymbol{x}_i, \boldsymbol{x}_j)$; (2) Adversarial-to-Adversarial (A2A): the adversarial contrastive loss using $(\boldsymbol{x}_i^{adv}, \boldsymbol{x}_j^{adv})$; (3) Standard-to-Adversarial (S2A): the mixed contrastive loss using $(\boldsymbol{x}_i, \boldsymbol{x}_i^{adv})$ or $(\boldsymbol{x}_j, \boldsymbol{x}_j^{adv})$; (4) Dual Stream (DS): combining S2S and A2A as Fig 1(c) shows. Experiment 3.4 shows that the last setting works best. We argue that DS capture better feature alignment in the latent space. [2] Besides, we find only applying the contrastive loss leads to the worse in-domain intent detection metrics, therefore we mix up the two kinds of objectives during training to avoid catastrophic forgetting (Kirkpatrick et al., 2017). We present an Algorithm section in the appendix.

3 Experiments

3.1 Setup

CLINC	OOS+	Small
Avg utterance length	9	9
Intents	150	150
Training set size	15250	7600
Development set size	3100	3100
Testing Set Size	5500	5500

Table 1: Full statistics of the datasets.

Datasets We perform experiments on two variants of the OOD benchmark dataset CLINC[3] (Larson et al., 2019), namely CLINC-OOS+ and CLINC-Small. Table 1 shows the detailed statistics of two datasets. They both contain 150 in-domain intents across 10 domains where CLINC-OOS+ contains 100 samples for each intent and CLINC-Small has 50 training samples for each intent. Besides, CLINC-OOS+ has 250 OOD examples in training set, while CLINC-Small contains 100.

To construct the unlabeled data, we mix up 10% of in-domain data and all of the OOD data in the training set. The total amount of unlabeled data is equal to 1500 in CLINC-OOS+ and 750 in CLINC-Small, where the number of OOD data is 250 and 100, respectively. Note that during the self-supervised learning phase, we don't utilize label information of the unlabeled data and only perform contrastive learning at the instance-level. During the supervised learning phase, we use the other in-domain training data for cross-entropy loss.

Metrics We report both in-domain metrics: Accuracy(ACC) and F1-score(F1), and OOD metrics: Recall and F1-score(F1). OOD Recall and F1-score are the main metrics in this paper.

3.2 Baseline Details

We compare our proposed self-supervised methods to two types of OOD detection methods, which are supervised and fully unsupervised. The former applies a supervised OOD entropy regularization. We use this setting as the reference upper bound for OOD detection results. The latter represents that we train the sentence feature extractor using only in-domain data. We treat this setting as the

[2] We leave the comprehensive theoretical analysis to future work.

[3] https://github.com/clinc/oos-eval

Model		CLINC-OOS+				CLINC-Small			
		in-domain		OOD		in-domain		OOD	
		ACC	F1	Recall	F1	ACC	F1	Recall	F1
Supervised OOD	N+1	88.6	91.46	19.12	32.00	85.23	88.58	17.46	29.99
	Entropy+MSP(oracle)	87.38	85.71	44.82	57.48	84.52	84.07	27.23	36.81
	Entropy+LOF(oracle)	84.08	85.12	60.44	61.89	82.16	82.83	60.72	61.39
	Entropy+GDA(oracle)	86.53	87.57	70.20	71.22	84.56	84.68	66.98	67.07
Self-Supervised OOD	MSP	83.61	84.05	24.28	36.57	81.84	82.20	19.12	29.79
	MSP+S2S(w/o adv)	84.11	84.93	37.36	45.52	83.98	83.65	22.40	33.06
	MSP+DS(ours)	84.85	84.91	**41.76***	**47.62***	83.93	83.21	**25.62***	**34.82***
	LOF	84.20	85.08	57.40	58.78	82.22	82.73	57.20	58.10
	LOF+S2S(w/o adv)	85.62	85.99	59.12	59.41	82.84	83.67	57.92	59.04
	LOF+DS(ours)	85.87	86.06	**59.96***	**61.20***	82.89	83.85	**59.68***	**60.77***
	GDA	86.34	87.73	63.70	65.23	84.24	84.30	60.40	61.07
	GDA+S2S(w/o adv)	88.56	88.10	64.92	67.22	85.76	86.20	62.80	64.20
	GDA+DS(ours)	88.71	88.98	**67.24***	**69.17***	85.78	86.69	**64.52***	**65.55***

Table 2: Performance comparison between our method and baselines on CLINIC-OSS+ and CLINIC-Small datasets. * indicates significant improvements over the corresponding baselines (p <0.05).

reference lower bound. For each training method, we use different OOD detection models to verify its performance. Therefore, the model proposed in this paper can be divided into two stages. Firstly, the feature extractor training is completed in the training stage, and then the OOD detection is conducted by using different models in detection stage.

Training Stage On the basis of fully unsupervised setting, our proposed four types of adversarial self-supervised learning settings are added, respectively. **Standard-to-Standard** (S2S): Original setting. The contrastive loss is computed between origin and augmented data. The adversarial attack is not involved. **Adversarial-to-Adversarial** (A2A): The setting injecting two adversarial attacks to origin data and augmented data first, then compute contrastive loss between them. **Standard-to-Adversarial** (S2A): This setting divide contrastive loss into two parts. One uses origin data with adversarial attack and augmented data, the other uses augmented data with adversarial attack and origin data. **Dual Stream** (DS): The setting combining S2S and A2A. The contrastive loss contains two parts. One uses origin data and augmented data, the other uses corresponding data with adversarial attacks.

Detection Stage As mentioned above, we compare three OOD detection models: **MSP** (Maximum Softmax Probability)(Hendrycks and Gimpel, 2017) applies a threshold on the maximum softmax probability where the threshold is set as 0.5. **LOF** (Local Outlier Factor)(Lin and Xu, 2019) uses the local outlier factor to detect unknown intents. **GDA** (Gaussian Discriminant Analysis)(Xu et al., 2020) is a generative distance-based classifier for out-of-domain detection with Euclidean and Mahalanobis distances.

In this paper, the experiments and analysis are mainly conducted around the training stage. Different detection models are used to verify the generalization of our proposed method.

3.3 Main Results

Table 2 displays the experiment results. Our method consistently outperforms all the unsupervised baselines in all settings, even close to the supervised oracles. Under the GDA setting, our proposed method outperforms the unsupervised method by 3.94%(OOD F1), 3.54%(OOD Recall) in CLINC-OOS+ and 4.48%(OOD F1), 4.12%(OOD Recall) in CLINC-Small. We also observe similar improvements on the MSP and LOF settings. The results confirm the effectiveness of our self-supervised learning method. Considering the effect of adversarial augmentation, our GDA+DS outperforms the standard contrastive learning (GDA+S2S(w/o adv)) by 1.95%(OOD F1), 2.32%(OOD Recall) in CLINC-OOS+ and 1.35%(OOD F1), 1.72%(OOD Recall) in CLINC-Small. The results demonstrate that adversarial attack can improve the efficiency and robustness of contrastive learning. For in-domain ACC and F1, our method also achieves slightly better performance, even close to N+1 which suffers from a severe drop in OOD metrics for unbalanced data.

3.4 Qualitative Analysis

Effect of Unlabeled Data Size. Fig 2 shows the effect of different sizes of unlabeled data for contrastive learning. We extract each subsets of the total CLINC-OOS+ unlabeled dataset through random sampling, so that the expectation of OOD

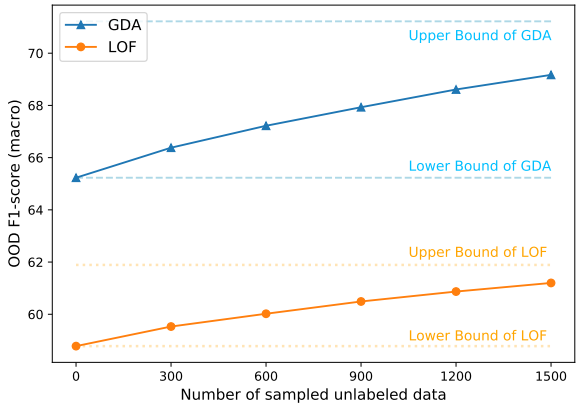


Figure 2: Relation between unlabeled data size and OOD detection F1-score.

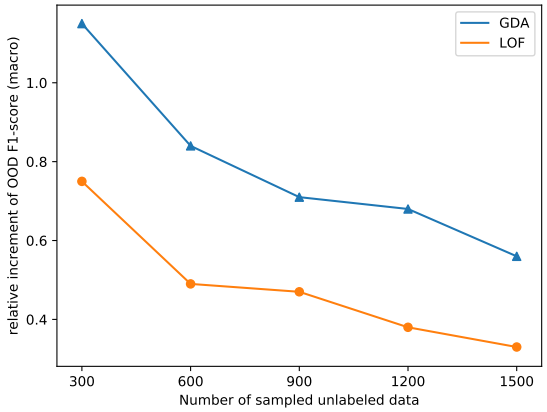


Figure 3: Relation between unlabeled data size and relative increment of OOD detection F1-score.

proportion in every subset is close to the full set (16.67%). We choose LOF and GDA for comparison. The lower bound and upper bound respectively represent unsupervised and supervised OOD. Our method achieves superior performance along with the increase of unlabeled data under two settings. It confirms that our proposed method can learn rich and semantically discriminative features via unlabeled data to facilitate OOD detection.

Fig 3 shows the relative increment of the F1-score during the uniform increase of unlabeled data. Specifically, the difference between the current F1-score and the previous state F1-score is recorded for every 300 samples added. As the amount of data increases uniformly, the extent of increment of OOD F1-score decrease. It confirms that our proposed method can optimize the performance of OOD detection by taking full advantage of unlabeled data and achieve impressive performance with only a small amount of data. Generally, our proposed methods have strong robustness and generalization capability.

Ablation Study of Contrastive Learning Settings. Table 3 shows the results of different contrastive learning setting on CLINC-OOS+. DS achieves the best performance in both in-Domain and OOD metrics. Comparing A2A and S2A to S2S, we observe adversarial augmentation improves OOD performance but decreases in-domain metrics. Therefore, by combining S2S and A2A, our DS can get the benefits of OOD and in-domain improvements from both settings.

Model	in-domain		OOD	
	Acc	F1	Recall	F1
S2S	88.56	88.10	64.92	67.22
S2A	88.21	87.90	65.00	67.63
A2A	87.78	87.41	66.40	68.53
DS	**88.71**	**88.98**	**67.24**	**69.17**

Table 3: Ablation study of contrastive learning settings.

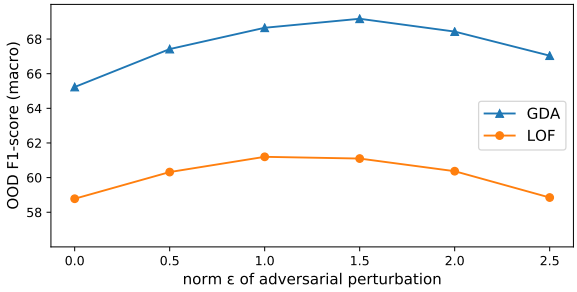


Figure 4: Effect of norm ϵ of adversarial perturbation.

Analysis of Norm of Adversarial Perturbation. Fig 4 displays the effect of norm ϵ of adversarial noise. ϵ controls the range of adversarial perturbation $\boldsymbol{\delta}$. In both LOF and GDA, $\epsilon \in (1.0, 1.5)$ achieves better performances. A smaller or larger value both impair the capability of contrastive learning. We argue that small noise can not improve the complexity of augmentation and large noise may hurt the alignment of positive example pairs.

4 Conclusion

In this paper, we focus on combining the benefits of both unsupervised and supervised OOD detection: simultaneously modeling semantic features of both in-domain and OOD data without requiring labor-intensive OOD annotation. We propose a self-supervised contrastive learning framework to learn rich and semantically discriminative representations from unlabeled data. Besides, we propose an adaptive end-to-end adversarial augmentation neural module to improve the diversity and complexity of pre-defined transformation functions. Experiments show that our method achieves better performance than unsupervised OOD baselines, even close to supervised OOD oracles.

5 Broader Impact

Task-oriented dialog systems have demonstrated remarkable performance across a wide range of applications, with the promise of a significant positive impact on human production mode and lifeway. However, in scenarios where information is complex and rapidly changing, models usually face input that is meaningfully different from typical examples encountered during training. Current models are prone to make unfounded but overconfident predictions on these inputs, which may affect human judgment and thus impair the safety of models in practical applications. In domains with the greatest potential for societal impacts, such as navigation or medical diagnosis, models should be able to detect potentially agnostic OOD and be robust to high-entropy inputs to avoid catastrophic errors. This work proposes a new adversarial self-supervised learning method for OOD detection. The overall robustness of the model is significantly improved by making full use of unlabeled data with potential threats through contrastive learning and adversarial attacks, which takes a step towards the ultimate goal of enabling the safe real-world deployment of task-oriented dialog systems in safety-critical domains. The experimental results have been reported on standard benchmark datasets for considerations of reproducible research.

Acknowledgments

This work was partially supported by National Key RD Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

References

Satoshi Akasaki and Nobuhiro Kaji. 2017. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. *ArXiv*, abs/1705.00746.

Abhijit Bendale and Terrance E. Boult. 2016. Towards open set deep networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *SIGMOD '00*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020a. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. *ArXiv*, abs/2003.04297.

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *HLT-NAACL*.

Chelsea Finn, P. Abbeel, and S. Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Ulrich Gnewuch, S. Morana, and A. Maedche. 2017. Towards designing cooperative and social conversational agents for customer service. In *ICIS*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *ICML*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020a. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735.

Keqing He, Yuanmeng Yan, and Weiran Xu. 2020b. Learning to tag OOV tokens by integrating contextual representation and background knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 619–624, Online. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ArXiv*, abs/1610.02136.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. 2020. Robust pre-training by adversarial contrastive learning. *ArXiv*, abs/2010.13337.

Joo-Kyung Kim and Young-Bum Kim. 2018. Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates. *ArXiv*, abs/1807.00072.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, G. Desjardins, Andrei A. Rusu, K. Milan,

John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, C. Clopath, D. Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Stefan Larson, Anish Mahendran, Joseph Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *EMNLP/IJCNLP*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *ArXiv*, abs/1807.03888.

Shiyu Liang, Yixuan Li, and R. Srikant. 2017. Principled detection of out-of-distribution examples in neural networks. *ArXiv*, abs/1706.02690.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv: Learning*.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *ACL*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. 2019a. Likelihood ratios for out-of-distribution detection. *ArXiv*, abs/1906.02845.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019b. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

Andras Rozsa, Ethan M Rudd, and Terrance E Boult. 2016. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32.

Ruhi Sarikaya. 2017. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine*, 34:67–81.

Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. 2013. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1757–1772.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. In *EMNLP*.

H. Shum, X. He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology Electronic Engineering*, 19:10–26.

J. Snell, Kevin Swersky, and R. Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS*.

Amrita S. Tulshan and S. Dhage. 2018. Survey on virtual assistant: Google assistant, siri, cortana, alexa.

J. Winkens, R. Bunel, Abhijit Guha Roy, Robert Stanforth, V. Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, T. cemgil, S. Eslami, and O. Ronneberger. 2020. Contrastive training for improved out-of-distribution detection. *ArXiv*, abs/2007.05566.

H. Xu, Keqing He, Yuanmeng Yan, Si hong Liu, Z. Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *COLING*.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569.

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

A Appendix

A.1 Implementation Details

We sample the augmentation data proportional to the size of the training set of each dataset. We use public APIs from multiple platforms to complete the back-translation process. Considering the availability of augmented data obtained through the back-translation process, we only sample back translated sequences that are more than 70% and less than 90% similar to the origin text in words overlapping. The total amount of data we sampled is equal to 10% of the volume of in-domain training data. We use the pre-trained GloVe embeddings (Pennington et al., 2014) as the word embedding matrix. For the BiLSTM encoder, we set the dimension of hidden states to 128 and use a dropout rate of 0.5. We use Adam optimizer (Kingma and Ba, 2014) to train our model and use a learning rate of 0.001. In the training stage, 20 epochs of supervised training are first conducted on in-domain labeled data, and then 200 epochs of alternate training are conducted by adding the process of contrastive learning on unlabeled data. The alternate training stage has an early stop setting with patience equalling 20. The algorithm of our proposed training process can be found in the Algorithm.1. We use the best F1-scores on the validation set to calculate the GDA threshold adaptively. Each result of the experiments is tested 5 times under the same setting and gets the average value. The amplitude of adversarial perturbation is obtained by the heuristic method in the range of 0 to 1E-2 (2.5/250), in which MSP and LOF are 4E-3 (1.0/250) and GDA is 6E-3 (1.5/250). In order to fairly compare with other settings, we set the weights of two losses equal in DS ($\alpha = 1$ in Algorithm.1) and S2A ($\beta = 0.5$ in Algorithm.1). The training stage of our model lasts about 15 minutes on a single Tesla T4 GPU(16 Gb of memory). The average value of the model parameters is 2.52M.

Algorithm 1 Algorithm of Proposed Two-Stage Training

Input: A set of clean sentences x and corresponding ground truth label t for labeled data; Feature Extractor g; Similarity algorithm f; $\mathcal{L}_{NT}$ represent contrastive loss; $\mathcal{L}_{CE}$ represent cross entropy loss
Output: Model parameters θ

for number of in-domain pretraining epochs **do**
  for in-domain mini-batch (x, t) **do**
    $\mathcal{L}_i = \mathcal{L}_{CE}(g(x, \theta), t)$
    Update parameters θ to minimize $\mathcal{L}_i$
  end for
end for
for number of mix-up training epochs **do**
  for sampled unlabeled mini-batch x **do**
    augment x_i to be x_j with back translate augmentation
    Generate the corresponding adverasial mini-batch $(x_i + \delta_i, x_j + \delta_j)$

$$\delta_i, \delta_j = \underset{||\delta_i'|| \leq \epsilon, ||\delta_j'|| \leq \epsilon}{\arg\max} \mathcal{L}_{NT}(f(x_i + \delta_i', x_j + \delta_j', \theta))$$

    if mode == S2S **then**
      $\mathcal{L}_m = \mathcal{L}_{NT}(f(x_i, x_j, \theta))$
    end if
    if mode == A2A **then**
      $\mathcal{L}_m = \mathcal{L}_{NT}(f(x_i + \delta_i, x_j + \delta_j, \theta))$
    end if
    if mode == S2A **then**
      $\mathcal{L}_m = \beta \mathcal{L}_{NT}(f(x_i, x_j + \delta_j, \theta)) + (1 - \beta)\mathcal{L}_{NT}(f(x_i + \delta_i, x_j, \theta))$
    end if
    if mode == DS **then**
      $\mathcal{L}_m = \mathcal{L}_{NT}(f(x_i, x_j, \theta)) + \alpha \mathcal{L}_{NT}(g(x_i + \delta_i, x_j + \delta_j, \theta))$
    end if
    Update parameters θ to minimize $\mathcal{L}_m$
  end for
  for in-domain mini-batch (x, t) **do**
    $\mathcal{L}_i = \mathcal{L}_{CE}(g(x, \theta), t)$
    Update parameters θ to minimize $\mathcal{L}_i$
  end for
end for