Open World Classification with Adaptive Negative Samples

Ke Bai[1], **Guoyin Wang**[2], **Jiwei Li**[3], **Sunghyun Park**[2],
Sungjin Lee[2], **Puyang Xu**[2], **Ricardo Henao**[1], **Lawrence Carin**[4]
[1]Duke University [2]Amazon [3]Zhejiang University [4]KAUST
{ke.bai, ricardo.henao}@duke.edu, {guoyiwan, sunghyu, sungjinl, puyax}@amazon.com,
jiwei_li@zju.edu.cn, larry.carin@kaust.edu.sa

Abstract

Open world classification is a task in natural language processing with key practical relevance and impact. Since the open or *unknown* category data only manifests in the inference phase, finding a model with a suitable decision boundary accommodating for the identification of known classes and discrimination of the open category is challenging. The performance of existing models is limited by the lack of effective open category data during the training stage or the lack of a good mechanism to learn appropriate decision boundaries. We propose an approach based on adaptive negative samples (ANS) designed to generate effective synthetic open category samples in the training stage and without requiring any prior knowledge or external datasets. Empirically, we find a significant advantage in using auxiliary one-versus-rest binary classifiers, which effectively utilize the generated negative samples and avoid the complex threshold-seeking stage in previous works. Extensive experiments on three benchmark datasets show that ANS achieves significant improvements over state-of-the-art methods.

1 Introduction

Standard supervised classification assumes that all categories expected in the testing phase have been fully observed while training, *i.e.*, every sample is assigned to one *known* category as illustrated in Figure 1(a). This may not always be satisfied in practical applications, such as dialogue intention classification, where new intents are expected to emerge. Consequently, it is desirable to have a classifier capable of discriminating whether a given sample belongs to a known or an unknown category, *e.g.*, the red samples in Figure 1(a). This problem can be understood as a $(C+1)$ classification problem, where C is the number of known categories and the additional category is reserved for *open* (unknown) samples. This scenario is also known as multi-class open set recognition (Scheirer et al., 2014).

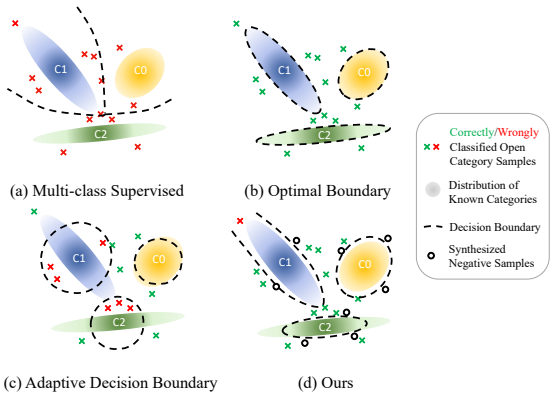

Figure 1: Illustration of previous methods and our proposed one. $C0$, $C1$, and $C2$ are three categories. The boundary is used to discriminate known and open (unknown) categories. (a) Boundary learned by supervised learning. (b) Optimal decision boundary. (c) ADB method which has a closed decision boundary, but may capture irrelevant points. (d) Proposed ANS method.

To discriminate the known from the open samples during inference, it is necessary to create a clear classification boundary that separates the known from the open category. However, the lack of open category samples during training makes this problem challenging. Current research in this setting mainly focuses on two directions. The first direction mainly estimates a tighter decision boundary between known classes to allow for the possibility of the open category. Existing works in this direction include the Local Outlier Factor (LOF) (Breunig et al., 2000; Zhou et al., 2022; Zeng et al., 2021), Deep Open Classification (DOC) (Shu et al., 2017) and Adaptive Decision Boundary (ADB) (Zhang et al., 2021b). LOF and ADB calibrate the decision boundary in the feature space while DOC does it in the probability space.

The second direction deals with learning a better feature representation to make the boundary-seeking problem easier. In this direction, Deep-

Unk (Lin and Xu, 2019a) and SEG (Yan et al., 2020) added constraints to the feature space, SCL (Zeng et al., 2021) and Zhou et al. (2022) fine-tuned the feature extractor backbone with contrastive learning. Zhan et al. (2021) considered introducing open samples from other datasets as negative, and Shu et al. (2021) generated samples with contradictory meanings using a large pretrained model. The latter two deliver large performance gains.

These improvements demonstrate the significance of negative samples in determining the boundary between the known and open categories. To accomplish the same in the absence of additional datasets or knowledge, we propose a novel negative-sample constraint and employ a gradient-based method to generate pseudo open category samples. As shown in Figure 1(d), negative samples are *adaptively* generated for each category to closely bound each category.

Given the generated negative samples, we then empirically find that using auxiliary one-*versus*-rest binary classifiers can better capture the boundary between the known and the open category, relative to a $(C+1)$-way classifier (Zhan et al., 2021), where all the open categories, possibly distributed in multiple modes or arbitrarily scattered over the feature space, are categorized into one class.

Specifically, we first learn a C-category classifier on known category data. Then for each known category, we learn an auxiliary binary classifier, treating this category as positive and others as negative. During inference, one sample is recognized as open if all the binary classifiers predict it as negative, thus not belonging to any known category

Our main contributions are summarized below:

- We propose a novel adaptive negative-sample-generation method for open-world classification problems without the need for external data or prior knowledge of the open categories. Moreover, negative samples can be added to existing methods and yield performance gains.
- We show that synthesized negative samples combined with auxiliary one-versus-rest binary classifiers facilitate learning better decision boundaries and requires no tuning (calibration) on the open category threshold.
- We conduct extensive experiments to show that our approach significantly improves over previous state-of-the-art methods.

2 Related Work

Boundary Calibration The classical local outlier factor (LOF) (Breunig et al., 2000) method is a custom metric that calculates the local density deviation of a data point from its neighbors. However, there is not a principled rule on how to choose the outlier detection threshold when using LOF. Zeng et al. (2021); Zhou et al. (2022) added open category data into the validation set to estimate or grid-search the proper threshold. So motivated, Bendale and Boult (2016) fit the output logits of the classifier to a Weibull distribution, but still use a validation set that contains the open category to select the confidence threshold. Further, Shu et al. (2017) employed one-versus-rest binary classifiers and then calculates the threshold over the confidence score space by fitting it to a Gaussian distribution. This method is limited by the often inaccurate (uncalibrated) predictive confidence learned by the neural network (Guo et al., 2017). Adaptive decision boundary (Zhang et al., 2021b), illustrated in Figure 1(c), was recently proposed to learn bounded spherical regions for known categories to contain known class samples. Though this post-processing approach achieves state-of-the-art performance, it still suffers from the issue that the tight-bounded spheres may not exist or cannot be well-defined in representation space. Due to the fact that high-dimensional data representations usually lie on a low-dimensional manifold (Pless and Souvenir, 2009), a sphere defined in a Euclidean space can be restrictive as a decision boundary. Moreover, the issue can be more severe if certain categories follow multimodal or skewed distributions.

Representation Learning DeepUnk (Lin and Xu, 2019a) trains the feature extractor with Large Margin Cosine Loss (Wang et al., 2018). SEG (Yan et al., 2020) assumes that the known features follow the mixture Gaussian distribution. Zeng et al. (2021) and Zhou et al. (2022) applied supervised contrastive learning (Chen et al., 2020) and further improve the representation quality by using k-nearest positive samples and negative samples collected from the memory buffer of MOCO (He et al., 2020). Getting a better representation trained with known category data only is complementary to our work, since a better pretrained backbone can further improve our results. Recent works found that it may be problematic to learn features solely based on the known classes; thus, it is crucial to pro-

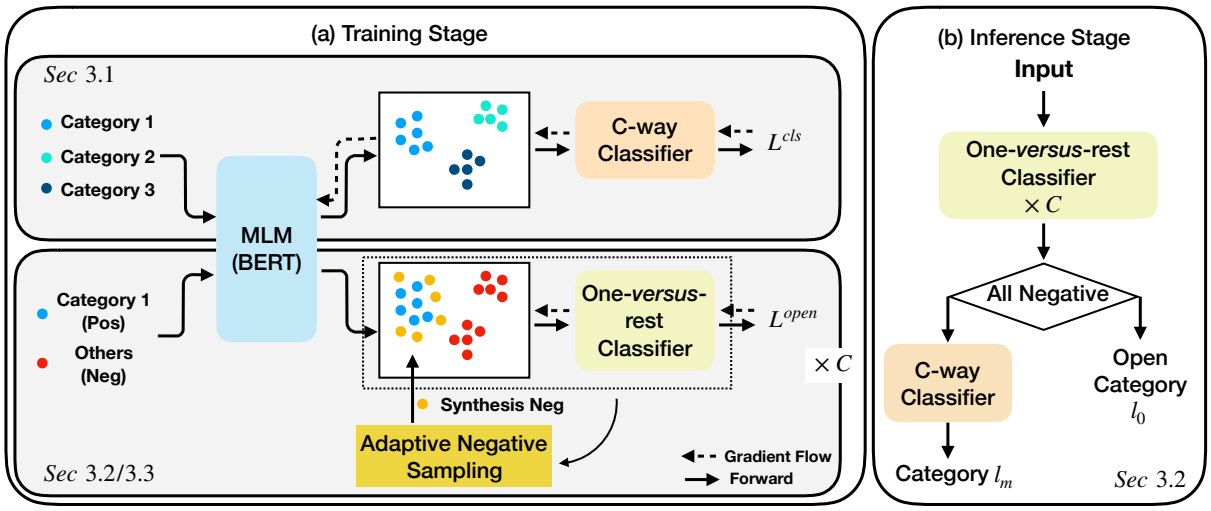

Figure 2: Illustration of the proposed ANS algorithm. (a) The training stage is divided into two blocks. Top: known class classification described in Sec. 3.1. Bottom: one-*versus*-rest classification (Sec. 3.2) on Category 1 as an example. The negative samples are from two sources, namely, known from other classes (in red) and synthesized samples (in yellow). (b) Inference arm described at the end of Section 3.2.

vide samples from unknown classes during training. Specifically, Zhan et al. (2021) creates negative samples with mixup of training data and examples from an external dataset. Shu et al. (2021) generates open class samples using BART (Lewis et al., 2019) and external text entailment information.

3 Methodology

Problem Definition Suppose we are given a training dataset $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ consisting of N examples, where $\boldsymbol{x}_i$ is an input text sequence and y_i its corresponding label, which belongs to a predefined set $\mathcal{L} = \{l_1, l_2, \ldots, l_C\}$ with C categories, thus $y_i \in \mathcal{L}, \forall i \in [N]$, where $[N] := [1, \ldots, N]$. In this paper, we use $[\cdot]$ to represent index sequence. In an open world classification setting, the goal is to learn a model which categorizes a test instance to either one of the predefined C categories or as an open category. In practice, the open category is denoted as a unique category l_0.

3.1 Known Category Classification

Following the setting suggested in Lin and Xu (2019b); Zhang et al. (2021b), we use BERT (Devlin et al., 2018) as our feature extractor $f_{\psi_{enc}}(\cdot)$. For each input text sequence $\boldsymbol{x}$, where $\boldsymbol{x}$ is represented as a sequence of tokens $[t_1, t_2, \ldots, t_n]$, we take the average of features z_{t_i} of each token t_i extracted from the BERT output layer as the sentence representation $\boldsymbol{z}$. - The training of $f_{\psi_{enc}}(\cdot)$ is formulated as a multi-category classification problem by minimizing the cross-entropy loss $\mathcal{L}^{cls}$:

$$\mathcal{L}^{cls}(\psi_{enc}, \psi_{cls}) = - \sum_{i \in [N]} \log \frac{\exp(f^{y_i}_{\psi_{cls}}(\boldsymbol{z}_i))}{\sum_{j=1}^{C} \exp(f^{j}_{\psi_{cls}}(\boldsymbol{z}_i))}, \quad (1)$$

where $f_{\psi_{cls}}(\cdot)$ is a classifier that takes $\boldsymbol{z}$ as input and the output dimension is the number of known categories C. $f^{j}_{\psi_{cls}}(\boldsymbol{z})$ represents the output logit for the j-th category. A well-trained feature extractor $f_{\psi_{enc}}(\cdot)$ and a high-quality classifier $f_{\psi_{cls}}(\cdot)$ can extract representative features of each category and ensure good performance on the classification of known categories during the inference stage.

3.2 Open Category Recognition

Once the classifier for the known categories is available, the task is to recognize samples from the open category *versus* the ones from known categories. As mentioned above, directly using the known category classifier $f_{\psi_{cls}}(\cdot)$ can result in poor performance (Hendrycks and Gimpel, 2016), while using a $(C+1)$ category classifier setting is complicated due to the need to find proper samples to obtain a suitable decision boundary for the open category (Scheirer et al., 2012; Liang et al., 2018; Shu et al., 2021). In this work, building upon ideas from one-class classification (Schölkopf et al., 2001; Ruff et al., 2018) and one-*vs*-all support vector machines (Rifkin and Klautau, 2004),

we propose a one-*versus*-rest framework via training simple binary classifiers for each predefined category. Based on this framework, we then introduce an effective open-sample generation approach to train these classifiers in Section 3.3.

We build an auxiliary one-*versus*-rest binary classifier for each known category, and take m-th category as an example to illustrate. Given a text sequence $\boldsymbol{x}$, we use the BERT pretrained with classification loss as the feature extractor $f_{\psi^{enc}}(\cdot)$ to extract features $\boldsymbol{z} \in \mathbb{R}^d$ to be fed to the binary classifiers, where d is the dimension of the feature space. Each category is provided with a binary classifier denoted as $g_{\theta_m^{cls}}(\boldsymbol{z}) : \mathbb{R}^d \to \mathbb{R}$, such that if $g_{\theta_m^{cls}}(\boldsymbol{z}) > 0$ then the input text $\boldsymbol{x}$ belongs to the m-th category or vice versa belongs to any of the other categories. We parameterize the entire binary classification framework for the m-th category as $\theta_m = (\psi^{enc}, \theta_m^{cls})$.

To learn each binary classifier $g_{\theta_m^{cls}}(\cdot)$ from training data $\mathcal{D}$, we first construct a *positive* set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_m}\}$ using data points with label l_m from $\mathcal{D}$ and a *negative* set $\{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_{N-N_m}\}$ by data points not in category l_m but also from $\mathcal{D}$. The total number of samples within category m is N_m, and $N - N_m$ is the number of remaining samples in $\mathcal{D}$. Each binary classifier is optimized by minimizing the binary cross-entropy loss function $\mathcal{L}^{rest}$:

$$\mathcal{L}^{rest}(\theta_m^{cls}) = \sum_{i \in [N_m]} \log(1 + \exp(-g_{\theta_m}(\boldsymbol{x}_i))) + \sum_{i \in [N-N_m]} \log(1 + \exp(g_{\theta_m}(\hat{\boldsymbol{x}}_i))). \quad (2)$$

During the inference phase, we have

$$\hat{y} = \begin{cases} \text{open}, & \text{if } g_{\theta_m}(\boldsymbol{x}_i) < 0, \forall m \in [C]; \\ \text{known}, & \text{otherwise.} \end{cases}$$

We assign a testing example to the open category l_0 if it is detected as unknown in all C binary classifiers. Otherwise, we pass the example to the known classifier to categorize the known categories. The entire inference pipeline is illustrated in Figure 2(b).

3.3 Adaptive Negative Sampling (ANS)

In practice, it is problematic to learn a good binary classifier with only the aforementioned negative samples from $\mathcal{D}$. The sample space of the open category is complicated, considering that new-category samples could originate from different topics and sources, relative to the known classes. So motivated, some existing methods introduce additional external data as negative samples.

To alleviate the issue associated with the lack of real negative samples, we propose to synthesize negative samples $\tilde{\boldsymbol{x}}$. Considering that it is hard to create actual open-category text examples, we choose to draw *virtual* text examples $\tilde{\boldsymbol{z}}$ in the feature space (Miyato et al., 2016; Zhu et al., 2019). Compared with the token space of text, the feature space is typically smoother (Bengio et al., 2013), which makes it convenient to calculate gradients (Wang et al., 2021).

For all known samples in a category l_m, points that are away from these samples can be recognized as *negative* to classifier $g_{\theta_m^{cls}}(\cdot)$. The entire feature space $\mathbb{R}^d$ contains essentially an uncountable set of such points, among which we are mostly interested in those near the known samples. Consequently, capturing these points will be helpful to characterize the decision boundary.

To give a mathematical description of these points, we assume that data usually lies on a low-dimensional manifold in a high-dimensional space (Pless and Souvenir, 2009). The low-dimensional manifold can be viewed as a local-Euclidean space, thus we can use the Euclidean metric to measure distances locally for each known data $\boldsymbol{z}_i$. Under this assumption, the set of pseudo negative samples $\mathcal{N}_i(r)$ for $\boldsymbol{z}_i$, which we call *adaptive synthesized open set*, can be described as follows,

$$\mathcal{N}_i(r) =: \Big\{\tilde{\boldsymbol{z}} : r \le \|\tilde{\boldsymbol{z}} - \boldsymbol{z}_j\|_2; \|\tilde{\boldsymbol{z}} - \boldsymbol{z}_i\|_2 \le \gamma \cdot r, \forall j : y_j = m\Big\}, \quad (3)$$

where r is the distance radius and $\gamma > 1$ is a hyperparameter. Note that each known sample $\boldsymbol{z}_i$ has an associated adaptive synthesized open set. As defined above, this set is subject to two inequalities. The first keeps synthesized samples away from all known samples within category m. The second implies that the synthesized samples should not be too far from the known samples. An intuitive geometric interpretation is that when $j = i$, the space implied by these two constraints is a spherical shell with inner radius r and outer radius $\gamma \cdot r$.

To get the radius r, we first calculate the covariance matrix Σ of $\boldsymbol{z}$ using known samples from category m and choose r, s.t. $r \le \sqrt{2\,\mathrm{Tr}(\Sigma)}$ and $\gamma r \ge \sqrt{2\,\mathrm{Tr}(\Sigma)}$. This is under the consideration

that $\sqrt{2\,\mathrm{Tr}(\Sigma)}$ is the average Euclidean distance between random samples drawn from a distribution with covariance matrix Σ.

The estimation is supported by the following proposition,

Proposition 1 *The expectation of the euclidean distance between random points sampled from distribution with covariance matrix Σ is smaller than $\sqrt{2\,\mathrm{Tr}(\Sigma)}$, i.e.*

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}(\boldsymbol{\mu},\Sigma)}\sqrt{\|\boldsymbol{x}-\boldsymbol{y}\|^2} \leq \sqrt{2\,\mathrm{Tr}\,\Sigma} \quad (4)$$

The proof can be found in supplementary. In our experiments, we fix $\gamma = 2$ and $r = 8$. The choice of r is relevant to the covariance matrix of the features in representation space. The detailed justification for our selection is provided in Appendix A.3. Ablation studies (Figure 3) show that model performance is not very sensitive to the chosen r.

Binary Classification with ANS According to Equation 3, each sample from a known category m contributes an adaptive synthesized set of open samples $\mathcal{N}_i(r)$. The classifier $g_{\theta_m^{cls}}(\cdot)$ is expected to discriminate them as negative. The corresponding objective function is the binary cross-entropy loss,

$$\mathcal{L}^{syn}(\theta_m^{cls}) = \sum_{i\in[N_m]} \log(1+\exp(g_{\theta_m^{cls}}(\tilde{\boldsymbol{z}}_i))),$$

where $\tilde{\boldsymbol{z}}_i$ is sampled from $\mathcal{N}_i(r)$ and N_m is the total number of known samples with category l_m. However, there exist uncountably many points in $\mathcal{N}_i(r)$. Randomly sampling one example from $\mathcal{N}_i(r)$ is not effective. Alternatively, we choose the most representative ones that are hard for the classifier to classify it as *negative*. Consistent with this intuition, the $\max(\cdot)$ operator is added to select the most challenging synthetic open sample distributed in $\mathcal{N}_i(r)$.

$$\mathcal{L}^{syn}(\theta_m^{cls}) = \sum_{i\in[N_m]} \max_{\tilde{\boldsymbol{z}}_i\in\mathcal{N}_i(r)} \log(1+\exp(g_{\theta_m^{cls}}(\tilde{\boldsymbol{z}}_i))). \quad (5)$$

Finally, the complete loss accounting for open recognition is summarized as:

$$\mathcal{L}^{open} = \mathcal{L}^{rest} + \lambda\mathcal{L}^{syn}, \quad (6)$$

where λ is a regularization hyperparameter.

Algorithm 1 Adaptive Negative Sampling.

1: **Input:** Training data $\mathcal{D} = \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n, \hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_n$. Parameters of current binary classifier θ_m.
2: **Hyper-Parameters:** Radius r, step-size η, number of gradient steps k
3: **for** Batch number $B = 1, \cdots, n_0$ **do**
4: $X_B(\hat{X}_B)$: Collect a batch of positive (negative) samples.
5: Calculate loss $\mathcal{L}_1 = \mathcal{L}^{real}(X_B, \hat{X}_B)$ using Eq. 2
6: Calculate the feature $\boldsymbol{z}_B$ over the positive samples.
7: **Adaptive Negative Sampling:**
8: sample $\boldsymbol{\epsilon} \sim \mathbb{N}(0, 4\cdot diag(\Sigma_{\boldsymbol{z}}))$
9: **for** $i = 1, \cdots, k$ **do**
10: Calculate loss $\ell(\boldsymbol{z}_B + \boldsymbol{\epsilon})$ using Eq. 5
11: $\boldsymbol{\epsilon} = \boldsymbol{\epsilon} + \eta \frac{\nabla_{\boldsymbol{\epsilon}}\ell(\boldsymbol{z}_B+\boldsymbol{\epsilon})}{\|\nabla_{\boldsymbol{\epsilon}}\ell(\boldsymbol{z}_B+\boldsymbol{\epsilon})\|}$ ▷ Gradient Ascend
12: **end for**
13: Calculate α using Eq.7
14: $\boldsymbol{\epsilon} = \frac{\alpha}{\|\boldsymbol{\epsilon}\|}\cdot\boldsymbol{\epsilon}$
15: Calculate loss $\mathcal{L}_2 = \mathcal{L}^{syn}_{\theta'_m}(\boldsymbol{z}_B + \boldsymbol{\epsilon})$ using Eq. 5
16: $\theta_m = \theta_m - \nabla_{\theta_m}(\mathcal{L}_1 + \mathcal{L}_2)$ ▷ Gradient Descend
17: **end for**

Directly minimizing the objective function in Equation 6 subject to the constraint in Equation 3 is challenging. In the experiments, we adopt the projected gradient descend-ascend technique (Goyal et al., 2020) to solve this problem.

Projected Gradient Descend-Ascent We use gradient descent to minimize the open recognition loss $\mathcal{L}^{open}$ and gradient ascent to find the hardest synthetic negative samples $\tilde{\boldsymbol{z}}$. The detailed steps are summarized in Algorithm 1. As illustrated in Figure 2(c), the sample $\tilde{\boldsymbol{z}}_i' = \tilde{\boldsymbol{z}}_i + \epsilon$ directly derived from gradient ascent (line 11 of Algorithm 1) might be out of the constraint area $\mathcal{N}_i(r)$. We then project to the closest $\tilde{\boldsymbol{z}}_i$ within the constraint such that $\tilde{\boldsymbol{z}}_i = \arg\min_{\boldsymbol{u}} \|\tilde{\boldsymbol{z}}_i' - \boldsymbol{u}\|^2, \forall\ \boldsymbol{u} \in \mathcal{N}_i(r)$ (Boyd et al., 2004). Unfortunately, direct search within $\mathcal{N}_i(r)$ defined in Equation 3 requires complex computation over entire training data $\mathcal{D}$. Based on our assumption that the training samples lie on a low-dimensional manifold and the empirical observation that $\tilde{\boldsymbol{z}}_i'$ is always closest to the corresponding positive point $\boldsymbol{z}_i$ relative to other positive points, $\mathcal{N}_i(r)$ can be further relaxed to the sphere shell around sample $\boldsymbol{z}_i$: $\mathcal{N}_i(r) = \{\tilde{\boldsymbol{z}} : r \leq \|\tilde{\boldsymbol{z}} - \boldsymbol{z}_i\|_2 \leq \gamma\cdot r\}$. We can then directly find the synthetic negative sample via a projection along the radius direction, *i.e.*, $\tilde{\boldsymbol{z}}_i = \tilde{\boldsymbol{z}}_i' + \alpha\frac{\tilde{\boldsymbol{z}}_i'-\boldsymbol{z}_i}{\|\tilde{\boldsymbol{z}}_i'-\boldsymbol{z}_i\|}$, where α is adjusted to guarantee $\tilde{\boldsymbol{z}}_i \in \mathcal{N}_i(r)$:

$$\alpha = \begin{cases} 1, & \text{if } r \leq \|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2 \leq \gamma\cdot r \\ \frac{r\gamma}{\|\tilde{\boldsymbol{z}}_i-\boldsymbol{z}_i\|_2}, & \text{if } \gamma\cdot r \leq \|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2 \\ \frac{r}{\|\tilde{\boldsymbol{z}}_i-\boldsymbol{z}_i\|_2}, & \text{if } \|\tilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2 \leq r \end{cases}$$

%		BANKING		CLINC		STACKOVERFLOW	
	METHODS	ACCURACY	F1-SCORE	ACCURACY	F1-SCORE	ACCURACY	F1-SCORE
25	MSP	43.67	50.09	47.02	47.62	28.67	37.85
	DOC	56.99	58.03	74.97	66.37	42.74	47.73
	OPENMAX	49.94	54.14	68.50	61.99	40.28	45.98
	DEEPUNK	64.21	61.36	81.43	71.16	47.84	52.05
	ADB	78.85	71.62	87.59	77.19	86.72	80.83
	SELFSUP*	74.11	69.93	88.44	80.73	68.74	65.64
	OURS	**83.93**	**76.15**	**92.64**	**84.81**	**90.88**	**84.52**
50	MSP	59.73	71.18	62.96	70.41	52.42	63.01
	DOC	64.81	73.12	77.16	78.26	52.53	62.84
	OPENMAX	65.31	74.24	80.11	80.56	60.35	68.18
	DEEPUNK	72.73	77.53	83.35	82.16	58.98	68.01
	ADB	78.86	80.90	86.54	85.05	**86.40**	85.83
	SELFSUP*	72.69	79.21	88.33	86.67	75.08	78.55
	OURS	**81.97**	**83.29**	**90.23**	**88.01**	86.08	**85.90**
75	MSP	75.89	83.60	74.07	82.38	72.17	77.95
	DOC	76.77	83.34	78.73	83.59	68.91	75.06
	OPENMAX	77.45	84.07	76.80	73.16	74.42	79.78
	DEEPUNK	78.52	84.31	83.71	86.23	72.33	78.28
	ADB	81.08	85.96	86.32	88.53	82.78	85.99
	SELFSUP*	81.07	**86.98**	88.08	89.43	81.71	85.85
	OURS	**82.49**	86.92	**88.96**	**89.97**	**84.40**	**87.49**

Table 1: Results of open world classification on three datasets with different known class proportions. * indicates the use of extra datasets during the training. The first five results of the baseline model are from Zhang et al. (2021b). The results for SelfSup are from Zhan et al. (2021).

In the future, we would like to consider relaxing these constraints by only considering the nearest k points instead of all the points within a category.

4 Experiments

We conduct experiments on three datasets: Banking, CLINC and Stackoverflow. Details and examples of the datasets are found in Appendix A.4.

Task Design We apply the same settings as Shu et al. (2017) and Lin and Xu (2019a). For each dataset, we sample 25%, 50%, 75% categories randomly and treat them as the *known category*. Any other categories out of the known categories are grouped into the *open category*. In the training and validation set, only samples within the known category are kept. All the samples in the testing set are retained, and the label of samples belonging to open categories is set to l_0. Importantly, samples from the open category are never exposed to the model in the training and validation process.

Evaluation Metrics The model needs to identify the samples with the open category, as well as classify the known samples correctly. Following Shu et al. (2017); Zhang et al. (2021b), we use accuracy and macro F1 as our evaluation metrics. The accuracy measures the overall performance, considering that open-world classification can be treated as a $C+1$ classification problem. F1 is a binary classification metric mainly used for evaluating the performance of open category recognition. The F1-score reported in this paper is the mean of the macro F1-score per category (including the open category), where the positive category is the corresponding one and negatives are all the other categories. F1-known is the average over F1-score of all known categories. F1-open is the F1-score of the open category.

Experimental setting We use the BERT-base-uncased model to initialize the feature extractor $f_{\phi_{enc}}$ and freeze the first ten layers of BERT during training. Note that all results are the mean of ten trials with different random seeds. Other experimental details are included in Appendix A.5.

4.1 Results

Table 1 compares our approach with previous state-of-the-art methods using accuracy and F1. Our implementation is based on Zhang et al. (2021a).

%	METHODS	ACC	F1	F1-OPEN	F1-KNOWN
25	BASELINE ($\lambda = 0$)	57.05	53.12	63.83	52.84
	+ GAUSSIAN NOISE	90.37	82.09	93.75	81.78
	+ PROJECTION	92.02	83.99	94.91	83.71
	+ ASCEND (OURS)	92.32	84.34	95.11	84.05
50	BASELINE ($\lambda = 0$)	64.60	71.65	60.28	71.80
	+ GAUSSIAN NOISE	88.01	86.90	89.82	86.86
	+ PROJECTION	90.22	88.18	92.01	88.12
	+ ASCEND (OURS)	90.23	88.22	92.02	88.17
75	BASELINE ($\lambda = 0$)	76.17	83.63	63.23	83.81
	+ GAUSSIAN NOISE	88.67	90.45	86.71	90.48
	+ PROJECTION	88.89	89.95	87.52	89.97
	+ ASCEND (OURS)	88.96	89.97	87.62	90.00

Table 2: Ablation study on the negative samples generation. The results are conducted on CLINC with known classes proportion 25%, 50% and 75%. Baseline represents the experiment with no synthesized negative samples under the one-*versus*-rest framework.

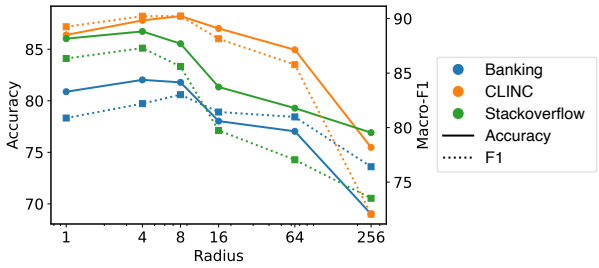

Figure 3: Ablation study on the radius r used in adaptive negative sampling process under the setting with 50% known categories.

The baselines include threshold finding methods, MSP (Hendrycks and Gimpel, 2016), DOC (Shu et al., 2017), OpenMax (Bendale and Boult, 2015), ADB (Zhang et al., 2021b); and feature learning methods, DeepUnk (Lin and Xu, 2019a); and negative data generation method SelfSup (Zhan et al., 2021). We did not include results from ODIST (Shu et al., 2021) because their method relies on MNLI-pretrained BART, which is not currently public and their model performance drops dramatically if not coupled with ADB. Note that SelfSup uses additional datasets, without which the accuracy on 50% CLINC drops from 88.33 to 83.12.

Our approach performs better than most previous methods, even better than the method using additional datasets, with the greatest improvement on CLINC. This is in accordance with SelfSup (Zhan et al., 2021), which also benefits the most on CLINC by adding negative samples. This implies that our synthesized negative samples are of high quality and could possibly be used as extra datasets in other methods.

The average performance gain in these three datasets decreases as the known category ratio increases, *i.e.*, compared to the strongest baseline ADB, our accuracy improvements in the three datasets are $5.08, 3.11, 1.42$ under the setting of $25\%, 50\%, 75\%$. With more known categories available, the more diverse the known negative samples will be, allowing the model to better capture the boundaries of the positive known categories while reducing the impact of synthetic samples.

The comparison with baselines on F1-open and F1-known can be found in Appendix A.3.

METHODS	ADD NEG.	ACC	F1	F1-OPEN	F1-KNOWN
MSP		44.46	51.95	43.22	52.41
	✓	57.07	58.92	61.40	58.79
ADB		78.39	71.53	84.18	70.86
	✓	79.71	73.01	85.18	72.12
ONE-VS -REST		51.33	41.40	50.83	43.54
	✓	80.11	73.35	85.57	72.71

Table 3: Performance comparison when the synthesized negative data are added to different baselines. The experiments are conducted on Banking with 25% known categories.

4.2 Discussion

Synthesized Negative is Beneficial for a Variety of Structures To investigate the contribution of the synthesized samples and the structure of one-*versus*-rest, we performed experiments adding the synthesized samples to two well-known baselines, MSP (Hendrycks and Gimpel, 2016) and ADB (Zhang et al., 2021b) as shown in Table 3. Specifically, the C-way classifier in MSP and ADB is replaced by a $(C + 1)$-way classifier, with an extra head for the synthesized negative samples. See Appendix A.7 for details.

We observe that performance increases on all baselines with synthesized negative samples. The synthesized samples behave like data augmentation, leading to better representation of positive samples.

Further, synthesized samples benefit one-*versus*-rest the most. The difference, we believe, stems from the model's flexibility on boundary learning. The open category may contain sentences with various themes, making it difficult for a single head of a $(C + 1)$-way classifier to catch them all. This one-*versus*-rest flexibility comes at the cost of more classifier parameters. However, compared to the huge feature extractor BERT, the number of additional parameters is relatively small. Distillation techniques can be used to build a smaller model if

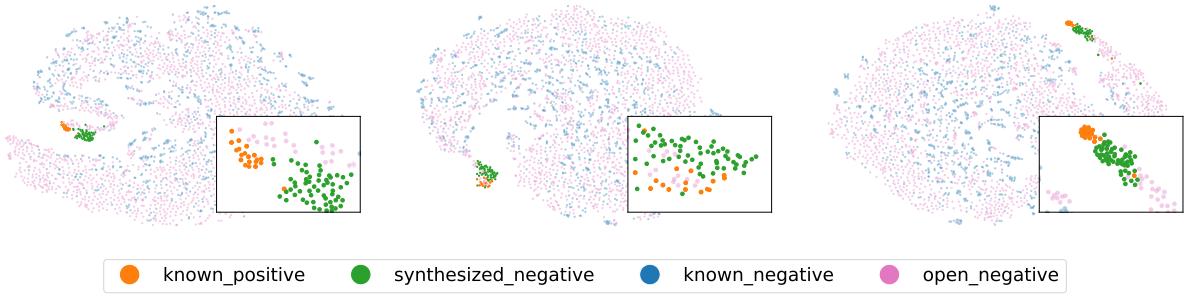

Figure 4: t-SNE plots of the feature extracted from the testing set of CLINC with 50% known categories. Each panel corresponds to a one-*versus*-rest classifier $g_{\theta_m^{cls}}$ with different known category m acting as the positive. Each panel's lower right corner has a square that enlarges the known positive data region to show the effectiveness of the synthesized negative samples. (Best viewed in color).

necessary, for instance, where there are thousands of known categories.

Adaptive Negatives Samples Generation Our adaptive negative sample generation consists of three modules (a) adding Gaussian noise to the original samples (line 8 in Algorithm 1). (b) gradient ascent (line $10 \sim 11$) (c) projection (line $13 \sim 14$). We add each module to the baseline in turn to study their importance. The baseline experiment uses the vanilla one-*versus*-rest framework described in Section 3.2, without the use of synthesized negative samples. Experiments are conducted on CLINC as shown in Table 2.

The following describes our findings from each experiment: (i) Adding samples with noise as negative alleviates the overconfidence problem of the classifier and improves the results significantly. The noise level needs to be designed carefully since small noise blurs the distinction between known and open, while large noise is ineffective.

(ii) Constraining synthesized samples to $\mathcal{N}(r)$ improves performance by keeping synthesized samples from being too close or too far away from positive known samples.

(iii) Adding a gradient ascent step further enhances performance. The improvement over the previous step is marginal. Our hypothesis is that the calculated gradient could be noisy, since the noise we add is isotropic and may be inconsistent with (outside of) the manifold of the original data.

Radius r Analysis In the adaptive negative sample generation process, the radius r and multiplier γ are two hyperparameters that determine the upper and lower bounds of the distance between the synthesized sample and the known sample. To investigate the impact of radius, we fix γ to 2 and increase the r from 1 to 256. Note that 8 is our default setting.

As illustrated in Figure 3, the performance gradually drops when the radius r increases, because the influence of the synthesized negative examples reduces as the distance between them and the positive samples grows. When the radius r decreases, the classifier may be more likely to incorrectly categorize positive as negative because the synthesized negative samples are closer to the known positives, resulting in a decrease in accuracy and F1 score on the banking and CLINC datasets. However, the performance on Stackoverflow improves. We hypothesize that there is a better data-adaptive way to estimate the radius r to improve the performance even further, for example, using k nearest neighbor instead of all the data in a category. We leave this as interesting future direction.

In summary, we observe that the performance is affected by the radius, but comparable results can be obtained for a wide value range. They are all better than the vanilla one-*versus*-rest baseline, which lacks the generated negative samples. The accuracy of baselines on Banking, CLINC and Stackoverflow is 58.09, 71.80 and 64.58, respectively.

Visualization Figure 4 shows the t-SNE representation of the features extracted from the second hidden layer of one-*versus*-rest classifier g_m^{cls}. Randomly chosen three known categories, each corresponds to a one-*versus*-rest classifier, yield three figures. The known positive/negative samples (blue) are clustered because the features are extracted from a pretrained C-way classifier Section 3.1. Open samples (pink) are scattered, some of which overlap with the known positives (see the middle figure). Our synthesized negatives work as expected; they are close to the known positives

and bridge the gap between the positive and other known categories.

5 Conclusions

We have introduced ANS, a pseudo open category sample generation approach for open-world classification problems. The generation process is free from extra datasets or prior knowledge. The synthesized samples are effective for improving existing methods. Combined with one-*versus*-rest framework, significant improvements are observed on three benchmarks. The gradient-based negative sample generation proposed can be applied to other NLP tasks such as out-of-scope discovery, which we leave as future work.

6 Limitations

First, both in terms of the number of training samples and the length of sentences, all the datasets that we employ in this study are relatively small and short. It may not scale well for long texts. Second, further work should be done on the model's ability to handle increasingly complicated data. The training samples from three benchmark datasets might be too sample; many input sentences even include the category descriptions, as shown in Tabel A.1.

References

Abhijit Bendale and Terrance Boult. 2015. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902.

Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572.

Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. 2013. Better mixing via deep representations. In *International conference on machine learning*, pages 552–560. PMLR.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. 2020. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning*, pages 3711–3721. PMLR.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. *ICLR*.

Ting-En Lin and Hua Xu. 2019a. Deep unknown intent detection with margin loss. *arXiv preprint arXiv:1906.00434*.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083.

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

Robert Pless and Richard Souvenir. 2009. A survey of manifold learning for images. *IPSJ Transactions on Computer Vision and Applications*, 1:83–94.

Ryan Rifkin and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR.

Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. 2012. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772.

Walter J Scheirer, Lalit P Jain, and Terrance E Boult. 2014. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324.

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

Lei Shu, Yassine Benajiba, Saab Mansour, and Yi Zhang. 2021. Odist: Open world classification via distributionally shifted instances. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3751–3756.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. *arXiv preprint arXiv:1709.08716*.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274.

Yezhen Wang, Bo Li, Tong Che, Kaiyang Zhou, Ziwei Liu, and Dongsheng Li. 2021. Energy-based open-world uncertainty modeling for confidence calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9302–9311.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert YS Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1050–1060.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. *arXiv preprint arXiv:2105.14289*.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert Lam. 2021. Out-of-scope intent detection with self-supervision and discriminative training. *arXiv preprint arXiv:2106.08616*.

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

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021b. Deep open intent classification with adaptive decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14374–14382.

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

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. *ICLR*.

A Appendix

A.1 Ethical Consideration

The topics of the three datasets we use in this paper are relatively simple, covering only the information needed for classification (check Table A.1). The category labels are either everyday intentions or technical terms in computer science. There are no potentially sensitive topics or contents that we are aware of. All three datasets have been published and are included in our appendix.

A.2 Related work: Adversary Augmentation

The gradient descend-ascend technique has been used successfully in adversarial attacks (Madry et al., 2018; Zeng et al., 2021); however, it differs from ours in terms of motivation and loss formulation.

Madry et al. (2018) sought for samples that were similar to the training sample but had a substantial loss given the paired label. The addition of generated samples during training may strengthen the model's resistance to adversarial attacks by avoiding inputs that are indistinguishable from genuine data but improperly categorised. The associated optimization formula is

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x},y\sim D}[\max_{\delta\in S} l(\theta, \boldsymbol{x}+\delta, y)], \tag{7}$$

where $y, \boldsymbol{x}$ are the training data, and l could be any classification model parameterized by θ. S is the an adversarial perturbation l_∞ ball.

Our work targets on shrinking the decision boundary. We need to treat the samples with positive labels in a specific region $\mathcal{N}(r)$(defined in Equation 3) as negative, *i.e.*

$$\min_{\theta} \mathbb{E}_{\boldsymbol{z}\sim D}[\max_{\delta\in\mathcal{N}(r)} l(\theta, \boldsymbol{z}+\delta, -1)], \tag{8}$$

where $\boldsymbol{z}$ is the positive sample from dataset D, l is a binary classifier with parameter δ. This equation behaves the same as Equation 5.

A.3 Explanation on Radius r

Proposition 2 *The expectation of the euclidean distance between random points sampled from distribution with covariance matrix Σ is smaller than $\sqrt{2\,\mathrm{Tr}(\Sigma)}$, i.e.*

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}(\boldsymbol{\mu},\Sigma)}\sqrt{\|\boldsymbol{x}-\boldsymbol{y}\|^2} \le \sqrt{2\,\mathrm{Tr}\,\Sigma} \tag{9}$$

Proof: Given that we are measuring the distance between samples drawn from the same distribution, we could subtract a constant value from both variables and assume that the distribution's expectation is zero. If $\boldsymbol{x}$ and $\boldsymbol{y}$ are random variables independently sampled from distribution with covariance matrix Σ and zero mean, we could have:

$$\begin{aligned}\mathbb{E}(\|\boldsymbol{x}-\boldsymbol{y}\|_2^2) &= \sum_i \mathbb{E}(\boldsymbol{x}_i^2 - 2\boldsymbol{x}_i\boldsymbol{y}_i + \boldsymbol{y}_i^2)\\ &= 2\sum_i(\mathbb{E}(\boldsymbol{x}_i^2) - \mathbb{E}(\boldsymbol{x}_i)\mathbb{E}(\boldsymbol{y}_i))\\ &= 2\sum_i(\mathbb{E}(\boldsymbol{x}_i^2)\\ &= 2\,\mathrm{Tr}(\Sigma)\end{aligned}$$

For a random variable Z, Jensen's inequality gives us

$$\mathbb{E}(\sqrt{Z}) \le \sqrt{\mathbb{E}[Z]} \tag{10}$$

The combination of the two equations above proves the proposition,

$$\mathbb{E}(\|\boldsymbol{x}-\boldsymbol{y}\|_2) \le \sqrt{2\,\mathrm{Tr}(\Sigma)}$$

In experiment, we choose the mean of the last layer of BERT as the latent representation $\boldsymbol{z} \in \mathbb{R}^{768}$. When calculating the trace, only the variance of each dimension, are required. On three datasets, the predicted distance per category falls primarily into $[8, 12]$, we fix $r = 8$ for all the experiments.

For each positive point, we could sample several adaptive negative samples. The distance between the synthesized negative and the chosen positive is determined by r. Meanwhile, we can also calculate the distance between the synthesized negative and other positive known samples.

We find that even when the radius is set to be less than the average distance, the synthesized negative samples have a much greater distance to other known points. In theory, known samples are on a low-dimensional manifold, whereas synthesized points are in a high-dimensional space, and the probability of sampled points falling into the manifold is zero. We calculated the distance between the synthesized sample and other known samples in the same category empirically, and discovered that their distances are nearly twice the average distance.

A further ablation study on different options of the radius can be found in the main context, where we have comparisons over different radius, *i.e.*, $r \in \{1, 4, 8, 16, 64\}$.

Dataset	Category	Examples
Banking	exchange_via_app	What currences are available for exchange? Does your app allow currency exchange from USD to GBP? I want to exchange USD and GBP with the app
	wrong_exchange_ rate_for_cash _withdrawal	I was given the wrong exchange rate when getting cash I think I was charged a different exchange rate than what was posted at the time. I need an accurate exchange rate, when I make my withdrawals.
	card_payment_ wrong_exchange_rate	Why didn't I receive the correct exchange rate for an item that I purchased? The fee charged when I changed rubles into British pounds was too much. I am being charged the wrong amount on my card.
CLINC	flight_status	what time is this flight supposed to land what time will i be able to board the plane so when is my flight landing
	time	what time is it in adelaide, australia right now please tell me the time how late is it now in ourense
	how_busy	how long will i wait for a table at red lobster can i expect chili's to be busy at 4:30 so how busy is the outback steakhouse at 5 pm
Stackoverflow	wordpress	How Display Recent Posts in all 3 languages at once in Wordpress How secure is Wordpress? Need MySQL Queries to delete WordPress Posts and Post Meta more than X Days Old
	apache	Apache / PHP Disable Cookies for Subdomain ? Increase PHP Memory limit (Apache, Drupal6) is setting the uploads folder 777 permision secure
	excel	Condition to check whether cell is readonly in EXCEL using C# EXCEL XOR multiple bits How I can export a datatable to excel 2007 and pdf from asp.net?

Table A.1: Extracted samples from three main datasets.

Dataset	Avg. Samples per Category	Avg. Length	Classes
Banking	117	11.91	77
CLINIC	100	8.31	150
StackOverflow	600	9.18	20

Table A.2: Statistics of benchmark datasets

A.4 Dataset

All three datasets are in English. The label distributions of each dataset are balanced. **Banking** (Casanueva et al., 2020) is a data set for intent detection in the banking domain with 77 fine-grained categories. It contains 13,083 samples in total. We follow the original splitting and sampling, *i.e.*, 9,003, 1,000, and 3,080 for training, validation and testing respectively.

CLINC (Larson et al., 2019) is an intent classification dataset originally designed for out-of-scope detection. It owns 22,500 in-scope samples and 1,200 out-of-scope samples that existed only in the testing phase. The divisions on the in-scope ones follow the original training, validation and testing splitting, *i.e.*, 15,000, 3,000 and 5,700.

StackOverflow (Xu et al., 2015) consists of 20 categories of technical question titles. We utilize the prepossessed version with 12,000 for training, 2,000 for validation and 6,000 for testing.

The statistics of the datasets are summarized in Table A.2. We also provide raw data samples of each dataset in Table A.1 for an intuitive understanding of the open world recognition task.

A.5 Experimental Details

All experiments are executed on a single NVIDIA Titan Xp GPU with 12,196M memory. All the experiments are done on NVIDIA X

Known Category Classification The classifier $f_{\psi_{cls}}$ is a fully connected neural network composed of one hidden layer with ReLU activation function. The hidden size is 768.

The learning rate of the transformer part and non-transformer part are $5e{-}5$ and $1e{-}4$ respectively. The total number of training epochs is 100. Learning rate decay and early stops are applied.

Training on this part takes about 10-20 minutes, depending on when the early stopping is triggered.

one-*versus*-rest Binary Classification During the training of one-*versus*-rest structure, we fix the

parameters of the feature extractor ψ^{enc}.

For the one-*versus*-rest module, the feature z is chosen as the mean of the output of the BERT model's last layer. The classifier is a fully connected three-layer neural network with ReLU as the activation function. The numbers of hidden neurons are (256,64), respectively. Dropout is added per hidden layer. The learning rate of the classifier is $1e{-}3$ for Banking and CLINC, $3e{-}4$ for Stackoverflow. The total number of epochs for each classifier is $\min(C, 20)$ to avoid overfitting. γ is 0.5.

Currently, we train each one-*versus*-rest classifier individually, and this takes roughly a minute per head. As a result, the total time grows linearly with the number of known categories. Parallel training of multiple heads can increase efficiency if necessary.

Model Size The parameters of the BERT backbone model and the ovr classifier are approximately 109 million and 0.2 million, respectively, implying that the maximum number of parameters from one-*versus*-rest is only about one-fifth of BERT (75% CLINC).

Reproducibility Checklist: hyper-parameter search We didn't include results from the validation set considering there is a huge gap between the current validation set and test set; test set contains open category samples while the validation set does not.

It is difficult to study the hyper-parameter setting because we lack an effective validation set with open category samples and the testing set is unavailable during training. To solve this, we construct a "pseudo dataset" by selecting a subset as "sub-known" from all known categories and treating others as "sub-open". Taking 50% CLINC as an example, we take a quarter of the known category as "sub-known" and the others as "sub-open". We discover that the rules we developed using these synthesized datasets can be transferred to formal experiments. We choose the proper hyper-parameter according to F1.

The hyperparameters we manually tried include training epoch (10, 20), learning rate ($1e-3$,$1e-4$ for the classifier head, $1e-4$, $5e-5$, $1e-5$ for BERT, note that the learning rate of the classifier head is always larger than BERT). The ablation study and the analysis on radius r can be found in Figure 3 and Appendix A.3. The hyper-parameters in gradient ascend are not sensitive to the final experiments.

A.6 More results

F1-known and F1-open The definition of F1-known and F1-open can be found in Section 4. Table A.3 shows the comparisons between the baselines and ours.

Reproducibility Checklist: Differences on Datasets During this process, we found that dataset CLINC is the robustest to change of hyper-parameters while dataset Stackoverflow is the weakest. A similar observation also shows in Zeng et al. (2021). It works the best on CLINC and worst on Stackoverflow.

We hypothesize that the difference comes from the quality of the raw data. As shown in Table Table A.1, the category of the input in Stackoverflow is usually included in the original sentence and we name them "easy". Rare sentences do not follow to this rule and we call them "hard". This leads to an observation in empirical experiments that the number of training epochs should be controlled in a limited range; otherwise many open category samples would be wrongly categorized to the known category.

This is in consistent with the finding in noisy labeling classification. The neural network will first fit the clean label before overfitting the noisy labeled samples. Under the current setting, the "hard" corresponds to the noisy sample. The one-*versus*-rest will first fit the easier one, followed by the harder one. When the harder one is classified correctly, many open categories could also be classified into this known category.

ADB (Zhang et al., 2021b) avoids this problem by working directly on the pre-trained features. It can statistically filter out the influence of noisy samples. Though ADB does not require extra hyper-parameter tuning, we found that the position of features extracted from the model has an impact on the final performances.

In summary, the differences between datasets are an intriguing topic that merits further investigation in the future.

Reproducibility Checklist: Standard Deviation As shown in Table A.4, larger known category ratios are more likely to be associated with lower variance; this is to be expected because more samples make the training more stable.

		BANKING		CLINIC		STACKOVERFLOW	
%	METHODS	OPEN	KNOWN	OPEN	KNOWN	OPEN	KNOWN
25	MSP	41.43	50.55	50.88	47.53	13.03	42.82
	DOC	61.42	57.85	81.98	65.96	41.25	49.02
	OPENMAX	51.32	54.28	75.76	61.62	36.41	47.89
	DEEPUNK	70.44	60.88	87.33	70.73	49.29	52.60
	ADB	84.56	70.94	91.84	76.80	90.88	78.82
	SELFSUP*	80.12	69.39	92.35	80.43	74.86	63.80
	OURS	**86.39**	**72.29**	**95.33**	**84.53**	**93.96**	**82.63**
50	MSP	41.19	71.97	57.62	70.58	23.99	66.91
	DOC	55.14	73.59	79.00	78.25	25.44	66.58
	OPENMAX	54.33	74.76	81.89	80.54	45.00	70.49
	DEEPUNK	69.53	77.74	85.85	82.11	43.01	70.51
	ADB	78.44	80.96	88.65	85.00	**87.34**	85.68
	SELFSUP*	67.26	79.52	90.30	86.54	71.88	79.22
	OURS	**81.06**	**83.52**	**92.16**	**87.95**	86.83	**85.81**
75	MSP	39.23	84.36	59.08	82.59	33.96	80.88
	DOC	50.60	83.91	72.87	83.69	16.76	78.95
	OPENMAX	50.85	84.64	76.35	73.13	44.87	82.11
	DEEPUNK	58.54	84.75	81.15	86.27	37.59	81.00
	ADB	66.47	86.29	83.92	88.58	73.86	86.80
	SELFSUP*	60.71	87.47	86.28	**89.46**	65.44	87.22
	OURS	**70.54**	**88.13**	**87.20**	89.18	**74.82**	**88.34**

Table A.3: Macro-F1 score on known category and open category of open world classification on three datasets with different known class proportions. * means extra datasets are used during the training. This table complements to Table 1

	%	ACCURACY	F1	F1-OPEN	F1-KNOWN
BANKING	25	1.83	1.76	1.47	1.78
	50	1.05	1.00	1.14	1.00
	75	1.05	0.69	2.45	0.67
CLINC	25	1.85	2.60	1.25	2.64
	50	0.86	0.74	0.75	0.74
	75	0.68	0.43	0.91	0.42
STACKOVERFLOW	25	1.48	2.01	1.02	2.22
	50	2.85	2.31	3.13	2.25
	75	1.21	0.54	3.10	0.43

Table A.4: Standard deviation of results collected from different random seeds. This table complements to Table 1

A.7 Ablation Study: Synthesized Negative Samples on other Structures

Negative sample generation for MSP and ADB follow the same process as ours, except that the gradient ascend is removed. The complete version is left for future work.

MSP with negative sampling The original MSP is a C-way classifier f^{cls} trained with cross-entropy loss during the training. In inference, the confidence score $p(\boldsymbol{x}_i) = \text{Softmax}(f^{cls}(\boldsymbol{x}_i))$ is first calculated. If the category with maximum probability p_m is lower than 0.5, the corresponding input is recognized as "open" category. Otherwise, this sample belongs to the category m, *i.e.*,

$$\hat{y} = \begin{cases} \text{open}, & \text{if } \max(p(\boldsymbol{x}_i)) < 0.5 \\ \arg\max p(\boldsymbol{x}_i), & \text{otherwise.} \end{cases}$$

In MSP with negative settings, an extra category l_0 is added for synthesized negatives. The inference

now becomes

$$\hat{y} = \begin{cases} \text{open}, & \text{if } \max(p(\boldsymbol{x}_i)) < 0.5 \\ \text{open}, & \text{if } \arg\max(p(\boldsymbol{x}_i)) == l_0 \\ \arg\max p(\boldsymbol{x}_i), & \text{otherwise} \end{cases}$$

Note that our MSP with synthesized negatives differs from (Zhan et al., 2021) in two aspects, (i), different ways to choose the negative samples. ii), their work added synthesized negative samples to the validation set, while ours uses the origin validation set.

ADB with negative sampling ADB training has two steps. The first step is to learn a good feature extractor using C-way classifier. The second step is to learn the boundary of each category in the pre-trained feature space.

Our modification is the first step. We replace the origin classifier with a $C + 1$-way classifier. The extra head is designed for the synthesized negative samples. The inference step is kept the same as the original method.

A.8 Code

Please find it in supplementary.