Class Name Guided Out-of-Scope Intent Classification

Chandan Gautam*[1], Sethupathy Parameswaran*[3], Aditya Kane[5], Yuan Fang[3], Savitha Ramasamy[1], Suresh Sundaram[4], Sunil Kumar Sahu[6], Xiaoli Li [1,2]

[1]Institute for Infocomm Research, A*STAR, Singapore,
[2]A*STAR Centre for Frontier AI Research, Singapore,
[3]Singapore Management University, [4]Indian Institute of Science, Bengaluru,
[5]Georgia Institute of Technology, [6]G42, Abu Dhabi, UAE

Correspondence: gautamc@i2r.a-star.edu.sg

Abstract

The paper introduces **S**emantics of **C**lass Label-based Unsupervised **O**ut **o**f **S**cope Intent Detection (SCOOS), a novel method aimed at enhancing out-of-scope (OOS) intent classification in task-oriented dialogue systems. Unlike prior approaches that rely solely on in-domain (ID) data features, SCOOS leverages semantic cues embedded in class labels to improve classification accuracy. The method entails forming a compact feature space centered around the semantics of class labels by minimizing losses between ID features and class names. SCOOS achieves this by creating a compact feature space centered around class label semantics, achieved through minimizing losses between in-domain (ID) features and class names. This involves training two spherical variational autoencoders concurrently to learn a shared latent space between ID features and class names, aligning ID feature data based on the corresponding classes in the latent space, and training a classifier for $(m+1)$-class classification using only ID samples, where the $(m+1)^{th}$ class represents OOS samples. Extensive evaluation of three datasets demonstrates that SCOOS outperforms existing methods not only for OOS intent detection but also for ID intent classification. Additionally, an ablation study is conducted to analyze the impact of different components of SCOOS, and we also presented the visualization of the latent space representation providing insights into the influence of semantic information from class labels. [1]

1 Introduction

Task-oriented dialogue systems are prone to encounter out-of-scope (OOS) inquiries during their application in an open-world setting. Such OOS intents need to be detected with the user's utterance, to avoid incorrect responses by the dialogue system. Hence, a critical element of dialogue systems is their ability to identify OOS or unknown intentions from user inquiries. Regardless of the number of classes in the OOS intents, the OOS detection problem is usually solved as a $(m+1)$-class classification problem, where m is the total number of known/In-Domain (ID) classes. Existing methods in the literature address this problem either as a supervised learning framework or as an unsupervised learning framework, depending on the availability of extensive labeled OOS intents during training.

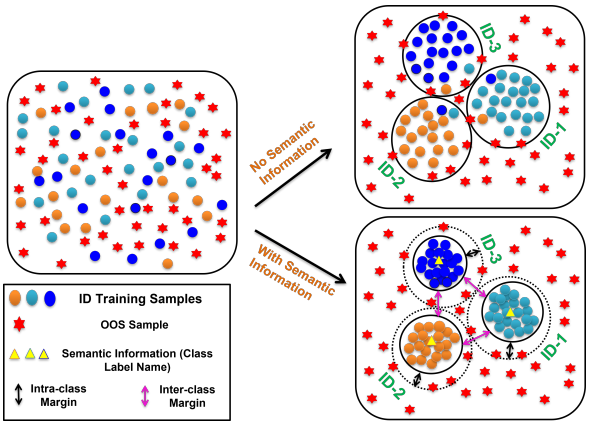


Figure 1: Overview of leveraging the class name for OOS intent detection

The supervised OOS intent recognition (Zheng et al., 2020; Zhan et al., 2021) is preferable when a sufficient amount of labeled OOS samples is available for training the $(m+1)^{th}$ class, where the $(m+1)^{th}$ class denotes the OOS class. In these approaches, the trained model has higher confidence of prediction for the ID classes and lower confidence for the OOS samples. However, these methods need adequate labeled OOS samples, and obtaining these labeled samples is cumbersome. Generative approaches to generate synthetic labeled OOS samples may cause artificial inductive bias and may not work well on real-world data as the

*Equal Contribution
[1]The source code is available at https://github.com/Chandan-IITI/SCOOS

Findings of the Association for Computational Linguistics: EACL 2024, pages 9100–9112

actual OOS data resides in an unbounded space and it is challenging to capture the distribution of such unknown data. On the contrary, unsupervised OOS detection (Lin and Xu, 2019; Yan et al., 2020; Zhang et al., 2021; Zhou et al., 2022) methods require only ID data to train the model and do not require synthetic or real labeled OOS samples. However, these methods do not exploit the semantics of the ID class labels, which provide rich information about the ID class labels that can help to largely improve OOS detection.

This paper proposes an unsupervised OOS intent classification method that uses the complete information about the ID classes, including the semantics of the class labels[2], for OOS detection, namely, **S**emantics of **C**lass Labels-based unsupervised **OOS** Intent Detection (SCOOS). In doing so, it solves OOS detection as a $(m+1)$-class classification problem during inference. The solution comprises of two modules, namely a BERT module and a Spherical Variational Autoencoder (SVAE) (Davidson et al., 2018) module. During training, each module has two sets of BERT Encoder and SVAE, one set for the ID samples and the other set for the corresponding class names. The BERT module learns the representations of the ID samples and the SVAE module uses the representations of the BERT module to perform OOS intent detection. Fig. 1 is a 2D representation of the proposed solution, where each class is cast into a sphere with the class name as the center of the sphere. Thus the SVAE module learns the compact and spherical representation of the ID samples in the latent space by minimizing the intra-class distance between the semantics of the class labels (i.e., class name) and the samples of the corresponding classes. It also maximizes the inter-class distance between the ID samples using class names. It must be noted that although two sets of BERT and SVAE are used during training, only one set of BERT and SVAE is required during inference, as the class labels are not available during inference.

In summary, our contributions are as follows:

- We have observed that despite the readily available semantic information of ID class labels, current methods for OOS intent detection overlook this valuable information. To tackle this problem, we introduce a novel unsupervised method for OOS intent detection (i.e., SCOOS), which harnesses the semantic information contained within ID class labels.

- In SCOOS, we show how sentence representation can be enhanced by aligning it with the class name through concurrent training of two BERT models and SVAEs. Additionally, we provide visualizations of this enhanced representation to offer a clearer insight into the efficacy of our proposed approach.

- Additionally, we conducted extensive experiments on three benchmark datasets for OOS intent classification. The results of these evaluations demonstrate that our proposed method surpasses other unsupervised OOS intent classification methods.

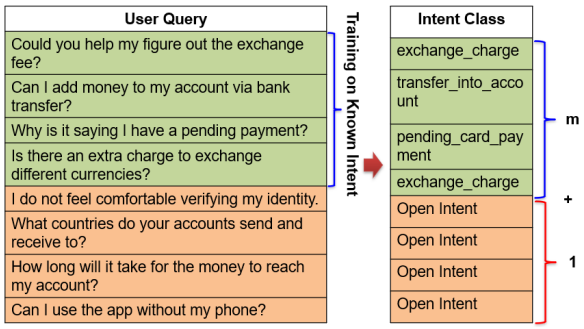


Figure 2: Example to detect the OOS intent without prior knowledge of OOS

2 Related Work

Categorizing user intent in dialogue systems is crucial across various domains such as banking, healthcare, e-commerce, and travel (Wen et al., 2019). Traditionally, deep learning models like convolutional neural networks (Xu and Sarikaya, 2013) and recurrent neural networks (Liu and Lane, 2016) have been used for intent classification with promising results (Yin et al., 2017). However, these methods often assume a closed-world scenario, which doesn't hold in real-world settings where new intent classes can appear. This issue can be addressed using anomaly detection, one-class classification (OCC), or the $(m+1)$-class classification approach. OCC-based OOS detection uses one-class support vector machine (OCSVM) (Schölkopf et al., 2001), support vector data description (SVDD) (Tax and Duin, 2004), local outlier factor (LOF) (Breunig et al., 2000), etc. Of these, LOF is a widely-used method to identify OOS intent (Lin and Xu, 2019; Yan et al., 2020; Zhou et al., 2022). It is

[2]In this paper, 'semantics of the class label(s)' is called as 'class name(s)'

a density-based unsupervised anomaly detection method, which calculates the local density deviation of a particular data point for its neighbors. Samples with significantly lower density compared to its neighbour are deemed as an outliers. Variations of the LoF with additional loss functions to improve the OOS intent detection performance are not uncommon. For example, Lin and Xu (2019) uses large margin cosine loss function with LOF, Yan et al. (2020) uses large margin Gaussian mixture loss with LOF, and Zhou et al. (2022) uses KNN-contrastive loss along with LOF for OOS intent classification.

OOS intent classification is also solved as a $(m+1)$-class classification, where $(m+1)^{th}$ class represents OOS intent. Here, the model is trained such that the $(m+1)^{th}$ class generates a confidence score for the OOS sample. Fei and Liu (2016) employs SVM to develop an one-vs-rest strategy to identify OOS intent. Hendrycks and Gimpel (2017) proposes using the maximum softmax probability to boost the confidence score for ID classes resulting in lower confidence for OOS samples. Softmax is also developed by Prakhya et al. (2017) for open-world learning using Weibull distribution and is popularly known as the OpenMax method. Later, Shu et al. (2017) developed a $(m+1)$-class classifier using a sigmoid function at the classification layer. Recently, Zhang et al. (2021) developed an adaptive circular decision boundary-based $(m+1)$ class classifier to identify the OOS intent in an open-world. Most recently, a self-supervised learning-based $(m+1)$-class classifier has been developed by Zhan et al. (2021) for OOS intent classification.

In the methods discussed above, none leverage the semantic information of the in-domain (ID) class labels during training. These class labels, available beforehand, contain valuable details about their respective ID classes, aiding in effective differentiation among them and, consequently, the out-of-scope (OOS) as well. The work most closely related to the proposed method is by Cavalin et al. (2020), which leverages a graphical network to learn the relationship between the graph embeddings of the input sequences to those of the class names, where class names are used as word graph embedding. On the other hand, the proposed method aims to align the embeddings of the input and class names in the BERT and SVAE embedding spaces themselves (instead of the classification layer) to create tighter boundaries for in-domain classes. Overall, we advocate for utilizing the semantics of class labels for OOS detection.

3 Method

In this work, we propose an unsupervised OOS intent detection method, as shown in Figure 3, by using only ID (known) class samples. It is to be noted here that each class label (categorical number) has its name in the textual form as shown in Figure 2. In this paper, we use the *semantics of these class labels (class name)* to develop an OOS intent classification method as it is readily available for all ID classes during training. We refer to this method as '**S**emantics of **C**lass Name-based Unsupervised **OOS** Intent Detection (SCOOS)'.

3.1 Problem Statement

For a given N^{tr} training data belonging to m ID classes, ID training set consists of $\{(x^i_{tr}, y^i_{tr}, c^i_{text})\}_{i=1}^{N^{id}_{tr}}$, where x^{tr}_i indicates a training ID sample, y^i_{tr} is its corresponding label, and c^i_{text} denotes class name of its corresponding label. It is to be noted that training data consists of only ID samples, and the testing set may consist of ID and OOS data. The goal of the trained OOS model is to classify ID samples accurately and differentiate between ID and OOS samples during inference. Thus, each sample should be assigned to one of the $(m+1)$ classes, where there are m ID classes and the $(m+1)^{th}$ class denotes the OOS class.

3.2 Semantics of Class Label-based Unsupervised Out of Scope Intent Detection (SCOOS)

The framework of the proposed method is illustrated in Figure 3. As shown in the Figure, the framework consists of two sets of transformer-based encoders and two sets of autoencoders. Here, one set of transformer-autoencoders learns the representation from input sequences (Sentences), and the other set of transformer-autoencoders learns the representation from the class names. Thus, the proposed framework aims at aligning the sentence and the class name representations by minimizing suitable loss functions during training. In this paper, we use BERT as the transformer-based encoder and SVAE as the autoencoder. While the BERT transformer helps in learning the better feature representation from the sentence and class names, the autoencoder helps in learning spherical representations towards efficient discrimination of

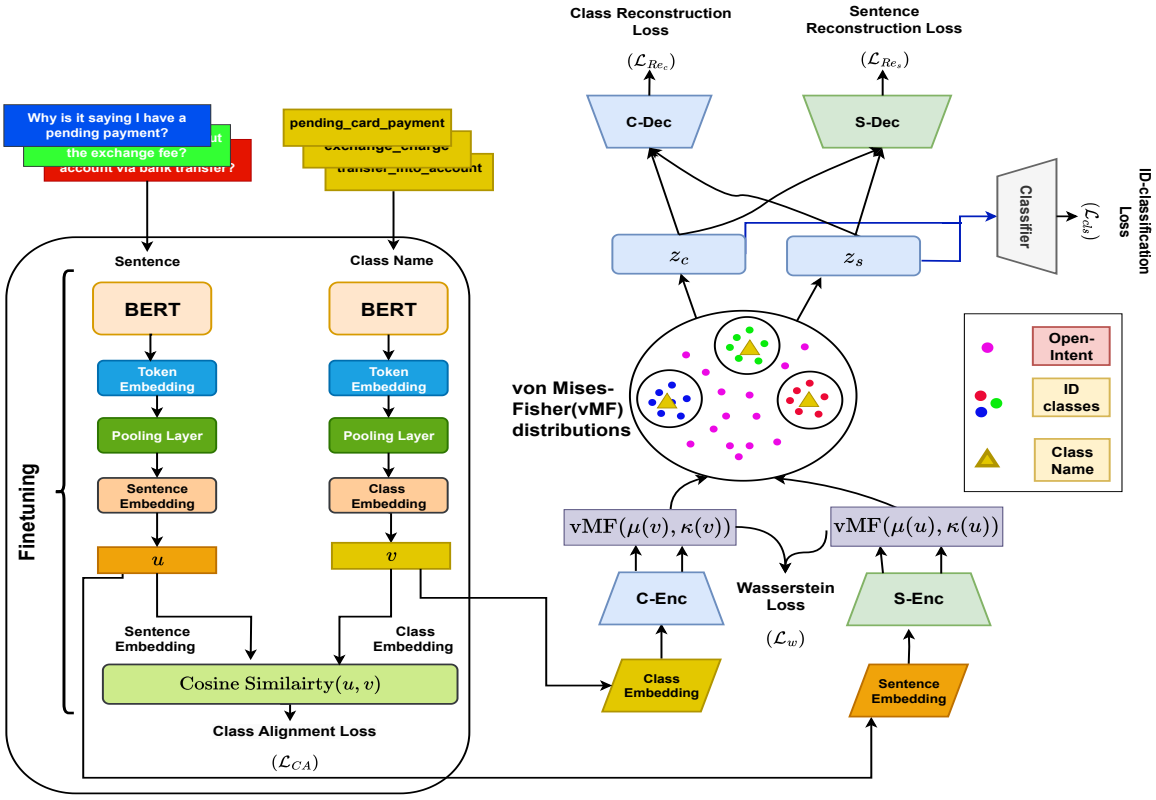


Figure 3: Our proposed architecture for OOS intent classification. Overall, our system performs $(m+1)$-class classification, while only needing examples from m ID classes during training. Here, z_s and z_c denote latent space representations of sentence and class embeddings, respectively; and μ represents the direction on the sphere and κ represents the concentration around μ.

the individual ID classes and the discrimination of OOS by building the manifold border for each ID class. Overall, the proposed method aligns sentences with their corresponding class names and learns the boundary for each class towards identifying the OOS sample. We describe each component of this framework in the subsequent subsections.

3.2.1 Class Alignment using BERT-Encoders

Two sets of BERT encoders, namely, Sentence-BERT (S-BERT) and Class-BERT (C-BERT) are used to align samples belonging to individual classes with their corresponding class names, as shown in Figure 3. The pre-trained BERT encoders are fine-tuned for the dataset of interest to obtain the representations of the sentence and the class names as sentence embeddings (u) and class embeddings (v). A scaled cosine similarity ($scos$) metric between sentence embeddings (u) and class embeddings (v) for all possible combinations of sentence and a unique number of ID classes are computed at the output of the encoder, as shown in Eq. (1).

$$scos(u, v) = \left(\beta \cdot \frac{u}{\|u\|}\right)^{\top} \left(\beta \cdot \frac{v}{\|v\|}\right), \qquad (1)$$

This normalization reduces the variance of the correct pairs of sentence and class embeddings, thereby, improving the accuracies of the ID sample classifications. Here, β is the scaling hyperparameter, which has the same impact as setting a high temperature of β^2 in softmax (Liu et al., 2018). To further align the sentence and the class embeddings of the individual ID classes, these cosine similarity metrics are minimized through a softmax cross-entropy function. We refer to this loss function as class-alignment loss ($\mathcal{L}_{CA}$), which is represented through Eq. (2) for a j^{th} class sample:

$$\mathcal{L}_{CA} = -\log \frac{e^{scos(u, v^j)}}{\sum_{i \in \{C\}} e^{scos(u, v^i)}}. \qquad (2)$$

3.2.2 Class Name based OOS Intent Detection

The sentence embeddings and their corresponding class embeddings that are efficiently aligned using

the two sets of BERT are then used for OOS Intent detection through a Spherical Variational Autoencoder (SVAE) (Davidson et al., 2018). To this end, the two embeddings are used to map the boundaries of individual classes at the latent space of SVAE. As shown in Figure 3, there are two sets of SVAEs, i.e., one each for sentence and class embedding. Thus, there is a pair of sentence-encoder (S-Enc) and sentence-decoder (S-Dec), and another pair of class-encoder (C-Enc) and class-decoder (C-Dec). Each encoder builds a latent space on a unit hypersphere, where each class is approximated by both encoders on a von Mises–Fisher (vMF) distribution. For a sentence embedding u and its corresponding class embedding v obtained through BERT-encoders, S-Enc approximates u as: $q_{\theta_s}(z_s|u) = q(z_s|\mu(u), \kappa(u))$, and C-Enc approximates v as: $q_{\theta_c}(z_c|v) = q(z_c|\mu(v), \kappa(v))$, where z_s and z_c denote latent space representations of sentence and class embeddings, respectively; and μ represents the direction on the sphere and κ represents the concentration around μ. Here, our objective is to minimize the Wasserstein distance (Arjovsky et al., 2017) between the latent spaces of both encoders (S-Enc and C-Enc) to align both representations as follows:

$$\mathcal{L}_w = \inf_{\gamma \in \prod(q_{\theta_s}, q_{\theta_c})} \mathbb{E}_{(z_s, z_c)\sim\gamma}[\| z_s - z_c \|]. \quad (3)$$

Further, to make these latent representations (z_s, z_c) invariant to both SVAE models, we minimize the reconstruction loss to the sentence ($\mathcal{L}_{Re_s}$) and class ($\mathcal{L}_{Re_c}$) embeddings as follows:

$$\begin{aligned}\mathcal{L}_{Re_s} =& |u - \text{S-Dec}(\text{S-Enc}(u))| + \\ & |u - \text{S-Dec}(\text{C-Enc}(v))|.\end{aligned} \quad (4)$$

$$\begin{aligned}\mathcal{L}_{Re_c} =& |v - \text{C-Dec}(\text{C-Enc}(v))| + \\ & |v - \text{C-Dec}(\text{S-Enc}(u))|.\end{aligned} \quad (5)$$

After making the latent spaces invariant, we include a classifier in the latent space to learn more discriminative latent space representation, and it can be defined as follows:

$$\mathcal{L}_{cls} = -\mathbb{E}[p_{z_s} \log q_{z_s}] - \mathbb{E}[p_{z_c} \log q_{z_c}], \quad (6)$$

where p_{z_s} and p_{z_c} are actual label of the latent vector z_s and z_c, respectively. q_{z_s} and q_{z_c} are the predictions made by a linear softmax classifier.

Overall, the SCOOS is trained through two sets of BERT and two sets of SVAEs by minimizing the following loss function:

$$\begin{aligned}\mathcal{L}_{overall} =& \lambda_s \mathcal{L}_{CA} + \lambda_w \mathcal{L}_w + \lambda_{rs} \mathcal{L}_{Re_s} + \\ & \lambda_{rc} \mathcal{L}_{Re_c} + \lambda_{cls} \mathcal{L}_{cls},\end{aligned} \quad (7)$$

where λ_s, λ_w, λ_{rs}, λ_{rc}, and λ_{cls} are the hyperparameters used to weight the different losses.

Once the training of the model is complete with accurate alignment of the sentence and class embeddings and their corresponding mapping into individual spherical distributions (i.e., vMF distribution), a circle on the unit hypersphere may roughly represent the manifold of each class. Thereafter, the center and the boundary may be used to ascertain the existence of the latent variable of any sample within the manifolds. The class centre is estimated using the class name information in the latent space through approximation of the vMF distribution $q(z_c^j|\mu(v^j), \kappa(v^j)); j = 1, \ldots, m$ with $\mu(v^j)$ as the class centre.

On the other hand, the boundary of the sphere is estimated using the statistics of the training data. For j^{th} ID class, the training data is encoded into latent space as $q(z_s^j|\mu(u^j), \kappa(u^j))$. Considering $\mu(u^j)$ as the latent variable for the training samples in the j^{th} ID class, the cosine similarity between each latent variable $\mu(u^j)$ and its corresponding class centre $\mu(v^j)$ are computed using Eq.(1) and stored in Z_{sim}^j. Subsequently, the boundaries are estimated through the estimation of a threshold $\alpha_j \in \alpha$ from Z_{sim}^j such that $\Omega\%$ of samples are ID samples and the remaining samples are OOS, as shown in Eq. (8).

$$\Omega = \frac{|\{\alpha_j \leq z_{sim} | z_{sim} \in Z_{sim}^j\}|}{|Z_{sim}^j|}, \quad (8)$$

where $|.|$ represents the number of elements in a set. Thus, the boundaries of the sphere of individual classes are stored as thresholds α_j.

During inference, a given test sample x^{te} is passed through the sentence embedding BERT (S-BERT) and the Sentence Embedding Spherical Encoder (S-Enc) to obtain the latent variable $\boldsymbol{\mu}(\boldsymbol{u}^{te})$. Then, the cosine similarities of the embeddings for the sample to all class centres are computed and the closest manifold boundary of the class based on the maximum cosine similarity is selected. Finally, by using the estimated threshold for that class (i.e., $\alpha^j; j = 1, \ldots, m$), the sample is deemed to be OOS or ID. If it is an ID, its corresponding class

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

Table 1: Dataset statistics

label is also estimated.

$$L^{oos} = \begin{cases} ID, & \text{if} \quad \max\{cos(\boldsymbol{\mu}(\boldsymbol{u}^{te}), \boldsymbol{\mu}(\boldsymbol{v}^j))|\forall \boldsymbol{v}^j \in \mathcal{ID}\} \geq \alpha^* \\ OOS, & \text{if} \quad \max\{cos(\boldsymbol{\mu}(\boldsymbol{u}^{te}), \boldsymbol{\mu}(\boldsymbol{v}^j))|\forall \boldsymbol{v}^j \in \mathcal{ID}\} < \alpha^* \end{cases} \tag{9}$$

4 Experiments

We evaluate the performance of the proposed SCOOS for unsupervised OOS intent detection on three public benchmark datasets as described in Section 4.1 and the detailed statistics of the datasets are presented in Table 1. As there are no OOS samples in these datasets, we follow the standard practices of present literature (Lin and Xu, 2019; Shu et al., 2017; Zhang et al., 2021) for our experiments. We randomly select 25%, 50%, or 75% of the intent classes as ID (i.e., known) classes in our experiments and keep the remaining classes as OOS classes (i.e., open). The training and validation split consists of only the ID classes, and the testing split consists of ID classes as well as the OOS class. It is to be noted that samples from the OOS class are neither utilized during training nor validation. As previously stated, we adhere to the conventional approach in the literature by grouping all OOS samples into a single class (Zhang et al., 2021; Zhou et al., 2022), resulting in a total of $(m+1)$ classes. To assess overall performance, we compute $(m+1)$-Accuracy (ACC) and Macro-F1 (F1-Score) metrics for comparing the performance of the proposed method with the existing methods. These metrics are computed across $(m+1)$ classes. Additionally, we report F1-Score for ID classes (F1-ID) and F1-Score for OOS/open class (F1-OPEN) separately to provide a comprehensive analysis of our method's performance. The Implementation details are provided in Appendix A.

4.1 Description of Datasets

Banking (Casanueva et al., 2020): It is a fine-grained dataset which consists of 77 intent classes and 13,083 customer service queries. Here, we perform intent classification on the queries.

CLINC (OOS) (Larson et al., 2019): It is an out-of-scope (OOS) dataset, which consists of 150 intent classes and 23700 query samples. There are 22,500 in-domain queries and 1,200 out-of-scope queries.

Stackoverflow (Xu et al., 2015): It consists of 3,370,528 technical question titles. The processed version of this dataset is provided by Xu et al. (2015), which consists of 20 intent classes and each class contains 1,000 samples.

4.2 Baselines

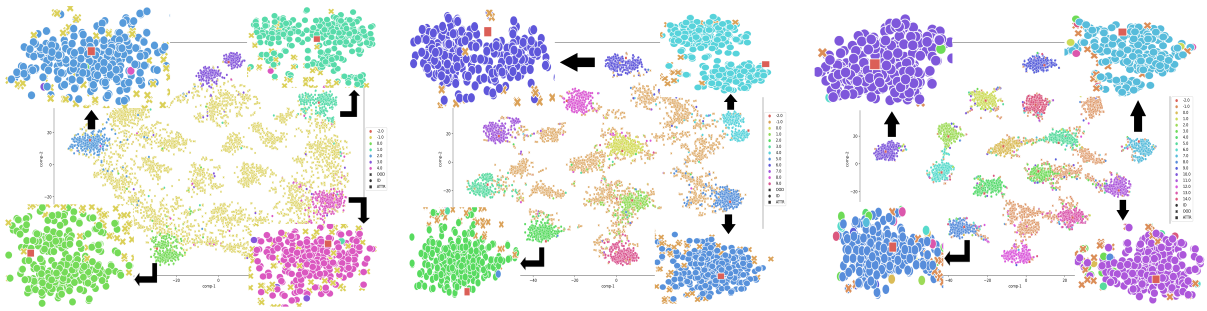
We perform extensive experiments and compare with the following state-of-the-art OSR methods: MSP (Hendrycks and Gimpel, 2017), DOC (Shu et al., 2017), OPENMAX (Bendale and Boult, 2016), DeepUnk (Lin and Xu, 2019), Softmax (Yan et al., 2020), SEG (Yan et al., 2020), ADB (Zhang et al., 2021), SELFSUP (Zhan et al., 2021), and KNN-Contra (Zhou et al., 2022). For a fair comparison, we use the BERT as the backbone network for all these methods.

4.3 Results and Discussion

We present all results in Table 2 for comparison, and the best results are highlighted in bold red. Table 2 shows F1-Score, F1-OPEN and F1-ID for the three benchmark datasets with different proportions (25%, 50% or 75%) of ID classes. As can be observed here, the proposed method SCOOS outperforms other existing OOS intent detection methods for all three settings on all three datasets. It improves the F1-Score without sacrificing the ACC (refer Table 7 in Appendix B) compared to the existing methods among all three datasets (minimum 3.7%, and maximum 7.62% improvement of F1-SCORE for 25% setting). Moreover, it doesn't only improve the OOS detection performance, but it also improves the ID classification performance by a significant margin among all three datasets (minimum 3.68% and maximum 7.76% improvement of F1-ID for 25% setting). It can be noticed that when less number of ID classes are available (i.e., 25%), the performance of the proposed SCOOS is commendable, compared to when more ID classes

% of ID Classes	METHODS	BANKING			CLINC (OOS)			STACKOVERFLOW		
		F1-SCORE	F1-OPEN	F1-ID	F1-SCORE	F1-OPEN	F1-ID	F1-SCORE	F1-OPEN	F1-ID
25%	‡MSP (Hendrycks and Gimpel, 2017)	50.09	41.43	50.55	47.62	50.88	47.53	37.85	13.03	42.82
	‡DOC (Shu et al., 2017)	58.03	61.42	57.85	66.37	81.98	65.96	47.73	41.25	49.02
	‡OPENMAX (Bendale and Boult, 2016)	54.14	51.32	54.28	61.99	75.76	61.62	45.98	36.41	47.89
	†Softmax (Yan et al., 2020)	58.32	62.52	58.1	67.74	83.04	67.34	50.78	45.52	51.83
	†SEG (Yan et al., 2020)	55.68	53.22	55.81	65.44	79.9	65.06	52.83	46.17	54.16
	‡DeepUnk (Lin and Xu, 2019)	61.36	70.44	60.88	71.16	87.33	70.73	52.05	49.29	52.6
	‡ADB (Zhang et al., 2021)	71.62	84.56	70.94	77.19	91.84	76.8	80.83	90.88	78.82
	†SELFSUP (Zhan et al., 2021)	69.93	80.12	69.39	80.73	92.35	80.43	65.64	74.86	63.8
	⋆KNN-Contra (Zhou et al., 2022)	77.13	90.19	76.44	-	-	-	81.61	92.7	79.39
	∗DE-OOD (Zhou et al., 2023)	80.35	-	-	82.82	-	-	87.04	-	-
	SCOOS (Ours)	**84.75**	**95.29**	**84.2**	**84.43**	**96.32**	**84.11**	**87.99**	**95.98**	**86.39**
50%	‡MSP (Hendrycks and Gimpel, 2017)	71.18	41.19	71.97	70.41	57.62	70.58	63.01	23.99	66.91
	‡DOC (Shu et al., 2017)	73.12	55.14	73.59	78.26	79	78.25	62.84	25.44	66.58
	‡OPENMAX (Bendale and Boult, 2016)	74.24	54.33	74.76	80.56	81.89	80.54	68.18	45	70.49
	†Softmax (Yan et al., 2020)	74.19	60.28	74.56	82.86	84.19	82.84	71.94	56.8	73.45
	†SEG (Yan et al., 2020)	76.48	60.42	76.9	79.42	78.02	79.43	74.18	60.89	75.51
	‡DeepUnk (Lin and Xu, 2019)	77.53	69.53	77.74	82.16	85.85	82.11	68.01	43.01	70.51
	‡ADB (Zhang et al., 2021)	80.9	78.44	80.96	85.05	88.65	85	85.83	87.34	85.68
	†SELFSUP (Zhan et al., 2021)	79.21	67.26	79.52	86.67	90.3	86.54	78.55	71.88	79.22
	⋆KNN-Contra (Zhou et al., 2022)	83.87	83.58	83.88	-	-	-	87.18	88.36	87.06
	∗DE-OOD (Zhou et al., 2023)	-	-	-	-	-	-	-	-	-
	SCOOS (Ours)	**86.68**	**88.11**	**86.64**	**88.01**	**92.05**	**87.95**	**88.17**	**88.83**	**88.11**
75%	‡MSP (Hendrycks and Gimpel, 2017)	83.6	39.23	84.36	82.38	59.08	82.59	77.95	33.96	80.88
	‡DOC (Shu et al., 2017)	83.34	50.6	83.91	83.59	72.87	83.69	75.06	16.76	78.95
	‡OPENMAX (Bendale and Boult, 2016)	84.07	50.85	84.64	73.16	76.35	73.13	79.78	44.87	82.11
	†Softmax (Yan et al., 2020)	84.31	56.9	84.78	89.01	83.12	89.61	82.28	54.07	84.11
	†SEG (Yan et al., 2020)	85.66	54.43	86.2	86.57	76.12	86.67	84.78	62.3	86.28
	‡DeepUnk (Lin and Xu, 2019)	84.31	58.54	84.75	86.23	81.15	86.27	78.28	37.59	81
	‡ADB (Zhang et al., 2021)	85.96	66.47	86.29	88.53	83.92	88.58	85.99	73.86	86.8
	†SELFSUP (Zhan et al., 2021)	86.98	60.71	87.47	89.43	86.28	89.46	85.85	65.44	87.22
	⋆KNN-Contra (Zhou et al., 2022)	87.07	67.66	87.41	-	-	-	87.06	74.2	87.92
	∗DE-OOD (Zhou et al., 2023)	88.98	-	-	91.03	-	-	87.61	-	-
	SCOOS (Ours)	**88.4**	**73.2**	**88.67**	**90.17**	**88.2**	**90.18**	**88.01**	**74.85**	**88.83**

Table 2: Results of OOS intent classification with different proportions (25%, 50% and 75%) of ID classes on 3 benchmark datasets in terms of Macro-F1 (F1-Score), F1-score for ID (F1-ID) and OOS (F1-Open) samples separately. The results of the baseline methods with ‡ symbols are retrieved from Zhang et al. (2021), the methods with † symbol are from Zhan et al. (2021), the method with ⋆ symbol is from Zhou et al. (2022), and the method with ∗ symbol is from Zhou et al. (2023).



(a) 25% (b) 50% (c) 75%

Figure 4: t-SNE visualization of features of ID classes on Stackoverflow dataset. The circle (○) legend denotes ID classes, the cross (×) legend denotes to OOS class, and the square (□) legend denotes class names. Different colour of (○) represents distinct ID classes.

are available (i.e., 50% and 75%).

4.4 Analysis

This section provides the results of the ablation study conducted on the Banking dataset. Initially, we visualize the textual features and class names in the latent space. Subsequently, we examine the influence of various individual losses on the dataset. Furthermore, we conduct an ablation study to evaluate the impact of hyperparameters (λ_w, λ_{rs}, λ_{rc}, λ_{cls}, λ_s, and β). Finally, we present results to

underscore the effectiveness of the SVAE in the proposed method.

Textual Feature and Class Name Visualization in the Latent Space: To illustrate the trained latent space, we plot the latent space variables of textual features and class names in the 2-D plane using t-SNE. The plots are visualized in Figure 4 for Stackoverflow datasets on all three settings (i.e., 25%, 50%, and 75%). In this figure, $\times$ represents OOS samples, $\circ$ represents samples from ID classes with distinct colors for distinct classes, and $\square$ represents class names. As there is only one class name corresponding to each class, only one $\square$ with red colour is shown in the plot for each class. Here, each $\square$ is compactly surrounded by same the colours of $\circ$, which means intra-class distance is minimized well by forming a compact group for each class. For better visualization, we provide zoomed snaps from a few parts of the plots, where each zoomed snap represents different classes of the Stackoverflow dataset. Moreover, the group corresponding to each class is well-separated in the latent space, and all $\times$ (i.e., OOS samples) are erratically dispersed over the latent space.

Significance of Individual Losses: An Ablation study is conducted to study the significance of the individual loss functions on the proposed method SCOOS, and the results are presented in Table 3. Base-Model with $\mathcal{L}_{Re_s}$ is slightly better than Base-Model with $\mathcal{L}_{Re_c}$ for all three settings of the Banking dataset. Moreover, it is observed that the result further improves with the combination of both the losses to the Base-Model as shown in Table 2. Finally, the inclusion of the $\mathcal{L}_{CA}$ along with the $\mathcal{L}_{Re_s}$ and $\mathcal{L}_{Re_c}$ to the Base-Model improves the performance significantly and outperformed all mentioned state-of-the-art method by a significant margin (as discussed above).

Effect of Various Hyperparameters (λ_w, λ_{rs}, λ_{rc}, λ_{cls}, λ_s, and β): We conducted the ablation study on the coefficient of different losses and presented results in Table 4 and Table 8, where the first table provides F1-SCORE and the latter table provides ACC for the same, which is available in Table 8 of Appendix D due to space constraints. It can be analyzed from these tables, the proposed method is robust towards the hyperparameter values. Further, we also analyzed the coefficient of scaled cosine similarity (β) and presented results in Table 5. As shown in this table, the model is robust to $\beta > 5$, hence, we use 15 in all our experiments. Further, the effect of latent space dimension is given in Appendix C.

%	METHODS	ACC	F1	F1-OPEN	F1-ID
25	BASE-MODEL				
	+ $\mathcal{L}_{Re_s}$	91.13	83.13	94.2	82.55
	+ $\mathcal{L}_{Re_c}$	90.77	82.58	93.96	81.98
	+ $\mathcal{L}_{Re_s}$ + $\mathcal{L}_{Re_c}$	91.72	81.61	94.7	80.92
	+ $\mathcal{L}_{Re_s}$ + $\mathcal{L}_{Re_c}$ + $\mathcal{L}_{CA}$(OURS)	**92.65**	**84.75**	**95.29**	**84.2**
50	BASE-MODEL				
	+ $\mathcal{L}_{Re_s}$	84.74	82.1	86.7	81.97
	+ $\mathcal{L}_{Re_c}$	84.47	81.77	86.49	81.65
	+ $\mathcal{L}_{Re_s}$ + $\mathcal{L}_{Re_c}$	85.87	84.28	87.36	84.19
	+ $\mathcal{L}_{Re_s}$ + $\mathcal{L}_{Re_c}$ + $\mathcal{L}_{CA}$(OURS)	**87.04**	**86.68**	**88.11**	**86.64**
75	BASE-MODEL				
	+ $\mathcal{L}_{Re_s}$	80.68	85.85	69.73	86.12
	+ $\mathcal{L}_{Re_c}$	79.67	84.93	68.75	85.21
	+ $\mathcal{L}_{Re_s}$ + $\mathcal{L}_{Re_c}$	82.91	87.82	71.98	88.09
	+ $\mathcal{L}_{Re_s}$ + $\mathcal{L}_{Re_c}$ + $\mathcal{L}_{CA}$(OURS)	**83.79**	**88.4**	**73.2**	**88.67**

Table 3: Ablation study on the different kinds of losses. The results are conducted on Banking on all three settings, i.e., 25%, 50%, and 75%. Base-Model contains two basic losses (i.e., $\mathcal{L}_w$ and $\mathcal{L}_{cls}$), which are necessary for all other losses.

Value	λ_w	$\lambda_r s$	λ_{rc}	λ_{cls}	λ_s
0.1	**84.75**	84.15	84.76	84.41	84.32
0.5	84.73	84.79	84.75	84.54	84.48
1	84.86	**84.75**	**84.75**	**84.75**	**84.75**
10	82.39	82.59	83.87	84.56	84.5
20	70.28	81.48	83.6	84.78	85.01

Table 4: Ablation of Hyperparameters of different losses on Banking dataset for 25% ID classes setting. All presented results in the table are F1-Score.

β	ACC	F1-SCORE	F1-OPEN	F1-ID
5	81.81	78.87	87.29	71.05
10	91.13	82.58	94.25	81.96
15	**92.65**	**84.75**	**95.29**	**84.2**
30	92.75	84.42	95.35	83.85
60	92.49	83.59	95.2	82.98

Table 5: Ablation of coefficient of scaled cosine similarity (β)

Effectiveness of SVAE in the proposed method: We provide results to emphasize the effectiveness of SVAE in the proposed method through two studies: (i) performance of the proposed method with SVAE replaced by VAE in Table 6. (ii) performance of the proposed method without using SVAE (i.e., Dual-BERT with a classifier) in Table 9. In Table 6, the proposed method's performance significantly dropped when we used VAE instead of

%	Methods	ACC	F1-SCORE	F1-OPEN	F1-ID
25	with SVAE	**92.65**	**84.75**	**95.29**	**84.2**
	with VAE	91.03	78.97	94.34	78.17
50	with SVAE	**87.04**	**86.68**	**88.11**	**86.64**
	with VAE	85.93	83.66	87.64	83.56
75	with SVAE	**83.79**	**88.4**	**73.2**	**88.67**
	with VAE	81.16	85.95	71.04	86.21

Table 6: Ablation when SVAE is replaced by VAE

SVAE. Hence, it is evident that the SVAE is a suitable candidate. Further, the second analysis, i.e., the Dual-BERT with a classifier is provided in Table 9 of Appendix E due to space constraints.

5 Conclusion

In this paper, we propose an unsupervised out-of-scope intent detection method, namely, Semantics of Class Label-based Unsupervised out-of-scope intent detection (SCOOS). For the first time in literature, we explored leveraging the semantics of the In Domain (ID) class labels during training to aid better representations of the ID samples, so that the OOS intents can be accurately detected. Specifically, we align the representations of the ID class samples to the representations of the semantics of the class labels. Performance studies in comparison with SOTA methods for OOS detection show the stellar performance of the proposed method. Ablation studies showed that the alignment between sentence and class name is very critical for accurate OOS intent detection. Future work may involve classifying the OOS intents further $(m + n)$ and adapting the model with detected intents.

Limitations

Our proposed method outperformed all methods in all three settings, i.e., 25%, 50%, and 75%. However, the proposed method works only when a sufficient number of ID samples are available, it will not work in the case when only a few samples are available. Moreover, it also cannot handle data in real-world, which contains unlabelled and non-stationary data. It opens scope for the further improvement of the proposed method.

Ethical Concern

In this work, we have used AI assistance of ChatGPT only for paraphrasing and polishing our content. All our results are reproducible, and the implementation details are provided in the Appendix.

Acknowledgments

This work was supported by the National Research Foundation, Singapore, under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-027).

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, Sydney, NSW, Australia. PMLR, PMLR.

Abhijit Bendale and T. 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 *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, Dallas, Texas, USA. ACM.

Inigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. *Proceedings of the 2nd ACL Workshop on NLP for Conversational AI*, abs/2003.04807:38.

Paulo Cavalin, Victor Henrique Alves Ribeiro, Ana Appel, and Claudio Pinhanez. 2020. Improving out-of-scope detection in intent classification by using embeddings of the word graph space of the classes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3952–3961.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. 2018. Hyperspherical variational auto-encoders. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 856–865. Association For Uncertainty in Artificial Intelligence (AUAI).

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 506–514, San Diego California, USA. The Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, Toulon, France. OpenReview.net.

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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5496, Florence, Italy. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689.

Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2018. Generalized zero-shot learning with deep calibration network. In *Advances in neural information processing systems*, pages 2009–2019, Montréal, Canada. Curran Associates, Inc.

Sridhama Prakhya, Vinodini Venkataram, and Jugal Kalita. 2017. Open set text classification using CNNs. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 466–475, Kolkata, India. NLP Association of India.

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, Hu Xu, and Bing Liu. 2017. DOC: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916, Copenhagen, Denmark. Association for Computational Linguistics.

David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning*, 54:45–66.

Tsung-Hsien Wen, Pei-Hao Su, Paweł Budzianowski, Iñigo Casanueva, and Ivan Vulić. 2019. Data collection and end-to-end learning for conversational ai. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*, Hong Kong, China. Association for Computational Linguistics.

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 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 62–69, Denver, Colorado, USA. The Association for Computational Linguistics.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding*, pages 78–83, Olomouc, Czech Republic. IEEE.

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, Online. Association for Computational Linguistics.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert YS Lam. 2021. Out-of-scope intent detection with self-supervision and discriminative training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3521–3532, Online. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. In *AAAI*, pages 14374–14382, Online. AAAI Press.

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.

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, Dublin, Ireland. Association for Computational Linguistics.

Yunhua Zhou, Pengyu Wang, Peiju Liu, Yuxin Wang, and Xipeng Qiu. 2024. The open-world lottery ticket hypothesis for ood intent classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15988–15999.

Yunhua Zhou, Jianqiang Yang, Pengyu Wang, and Xipeng Qiu. 2023. Two birds one stone: Dynamic ensemble for ood intent classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10659–10673.

A Implementation Details

The proposed method consists of two BERT encoders and two SVAEs: one for sentence and the other for class name. In the experiment, we use the BERT as a pre-trained model with the learning rate 5e-5 and output dimension of 768 and implemented using Huggingface Transformers in PyTorch. Further, SAVE consists of an encoder and a decoder, where both consist of two fully connected layers. The encoder's hidden node size is 512 and is followed by ReLU, and the output layer is passed to two different fully connected layers to generate mean (dimension=64) and concentration parameter (dimension=1) for vMF distribution. The mean is further normalized to make it a unit hypersphere. In the decoder, the hidden node size is 512, which is followed by ReLU, and the output layer's dimension is 768. The learning rate for SVAE is 1e-4. We kept the same set of parameters for both SVAEs. Moreover, the latent space variable is passed to a linear Softmax classification layer. For training the overall model, we use the Adam optimizer and the batch size is 128. The hyper-parameters λ_s, λ_w, λ_{rs}, λ_{rc}, and λ_{cls} for different losses are set to 1.0, 0.1, 1.0, 1.0, and 1.0, respectively for all the experiments. Further, we set the coefficient of scaled cosine similarity (β) at 15 and the rejection ratio (Ω) is set to 95% for all cases. The threshold α is estimated from Eq. (8) by setting the rejection ratio ($\Omega = 95\%$) of the ID samples used in training. Here, the threshold α is estimated such that 95% of the samples used in the training must be classified as ID by the model. We performed all of our experiments on an AMD EPYC 7763 64-core Processor with an NVIDIA A40 graphics card.

B Results in terms of $(m+1)$-Accuracy (ACC)

Apart from the Macro-F1 (F1-Score) in the main results Table 2, we also present the results in term of $m+1$)-Accuracy (ACC) along with F1-Score for a fair comparison in Table 7. The proposed method still outperformed all existing methods when compared in terms of $m+1$)-Accuracy (ACC).

C Impact of Latent Space Dimensions

In this analysis, we study the robustness of the proposed method SCOOS with respect to latent space dimensions. We experiment over ranges of latent dimensions $[16, 32, 64, 128, 256]$ on the banking dataset for all three settings and provide the plot

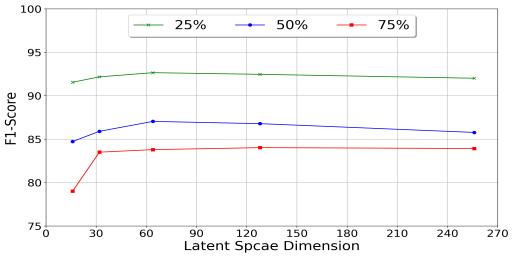


Figure 5: Impact of latent space dimensions

in Figure 5. While higher dimensions of the latent space allow for more freedom of representation at the cost of a large network, lower dimensions allow for a more compact feature representation at the cost of representational freedom. Therefore, care should be exercised in choosing the latent dimensions with a good balance of both. Similar observations can be made from Figure 5 in all three settings, where F-Score initially increased with increasing latent dimensions, then it either flattens or starts declining. It means $60-130$ is a good range for capturing the best features of text data and class names by the proposed method. Through these experiments, we select a latent dimension of 64 for all the experiments in the paper.

D Ablation of Hyperparameters of different losses

The first part of this analysis is provided in Section 4.4 of the main paper. More specifically, the same analysis is provided in terms of F1-Score (in Table 4 of the main paper) and Accuracy (in Table 8 of this appendix).

E Proposed method with and without using SVAE

First analysis is provided in Section 4.4 of the main paper and the second analysis is provided here due to space constraints. It is evident from Table 9, although the Dual-BERT with classifier model has a decent performance without the SVAE, its performance drops substantially (F1 score drops by 5%) in the 25% ID-class setting.

F How vMF distribution helps in OOS detection?

We first present the vMF distribution for identifying the oos label and then explain the mean direction and concentration, for clarity.

% of ID Classes	METHODS	BANKING		CLINC (OOS)		STACKOVERFLOW	
		ACC	F1-SCORE	ACC	F1-SCORE	ACC	F1-SCORE
25%	‡MSP (Hendrycks and Gimpel, 2017)	43.67	50.09	47.02	47.62	28.67	37.85
	‡DOC (Shu et al., 2017)	56.99	58.03	74.97	66.37	42.74	47.73
	‡OPENMAX (Bendale and Boult, 2016)	49.94	54.14	68.5	61.99	40.28	45.98
	†Softmax (Yan et al., 2020)	57.88	58.32	76.5	67.74	46.17	50.78
	†SEG (Yan et al., 2020)	51.11	55.68	72.86	65.44	47	52.83
	‡DeepUnk (Lin and Xu, 2019)	64.21	61.36	81.43	71.16	47.84	52.05
	‡ADB (Zhang et al., 2021)	78.85	71.62	87.59	77.19	86.72	80.83
	†SELFSUP (Zhan et al., 2021)	74.11	69.93	88.44	80.73	68.74	65.64
	⋆KNN-Contra (Zhou et al., 2022)	85.62	77.13	-	-	89.04	81.61
	∗DE-OOD (Zhou et al., 2023)	89.0	80.35	92.38	82.82	93.09	87.04
	∗OLT (Zhou et al., 2024)	-	-	-	-	-	-
	SCOOS (Ours)	**92.65**	**84.75**	**93.98**	**84.43**	**93.76**	**87.99**
50%	‡MSP (Hendrycks and Gimpel, 2017)	59.73	71.18	62.96	70.41	52.42	63.01
	‡DOC (Shu et al., 2017)	64.81	73.12	77.16	78.26	52.53	62.84
	‡OPENMAX (Bendale and Boult, 2016)	65.31	74.24	80.11	80.56	60.35	68.18
	†Softmax (Yan et al., 2020)	67.44	74.19	82.47	82.86	65.96	71.94
	†SEG (Yan et al., 2020)	68.44	76.48	77.05	79.42	68.5	74.18
	‡DeepUnk (Lin and Xu, 2019)	72.73	77.53	83.35	82.16	58.98	68.01
	‡ADB (Zhang et al., 2021)	78.86	80.9	86.54	85.05	86.4	85.83
	†SELFSUP (Zhan et al., 2021)	72.69	79.21	88.33	86.67	75.08	78.55
	⋆KNN-Contra (Zhou et al., 2022)	83.14	83.87	-	-	87.62	87.18
	∗DE-OOD (Zhou et al., 2023)	-	-	-	-	-	-
	∗OLT (Zhou et al., 2024)	-	-	-	-	-	-
	SCOOS (Ours)	**87.04**	**86.68**	**90.13**	**88.01**	**88.33**	**88.17**
75%	‡MSP (Hendrycks and Gimpel, 2017)	75.89	83.6	74.07	82.38	72.17	77.95
	‡DOC (Shu et al., 2017)	76.77	83.34	78.73	83.59	68.91	75.06
	‡OPENMAX (Bendale and Boult, 2016)	77.45	84.07	76.8	73.16	74.42	79.78
	†Softmax (Yan et al., 2020)	78.2	84.31	86.26	89.01	77.41	82.28
	†SEG (Yan et al., 2020)	78.87	85.66	81.92	86.57	80.83	84.78
	‡DeepUnk (Lin and Xu, 2019)	78.52	84.31	83.71	86.23	72.33	78.28
	‡ADB (Zhang et al., 2021)	81.08	85.96	86.32	88.53	82.78	85.99
	†SELFSUP (Zhan et al., 2021)	81.07	86.98	88.08	89.43	81.71	85.85
	⋆KNN-Contra (Zhou et al., 2022)	81.77	87.07	-	-	83.85	87.06
	∗DE-OOD (Zhou et al., 2023)	85.2	88.98	89.32	91.03	84.76	87.61
	∗OLT (Zhou et al., 2024)	82.89	-	92.3	-	75.92	-
	SCOOS (Ours)	**83.79**	**88.4**	**89.34**	**90.17**	**84.93**	**88.01**

Table 7: Results of OOS intent classification with different proportions (25%, 50% and 75%) of ID classes on 3 benchmark datasets in terms of $(m+1)$-Accuracy (ACC) and Macro-F1 (F1-Score). The results of the baseline methods with ‡ symbols are retrieved from Zhang et al. (2021), the methods with † symbol are from Zhan et al. (2021), the method with ⋆ symbol is from Zhou et al. (2022), and the method with ∗ symbol is from their corresponding paper.

Value	λ_w	$\lambda_r s$	λ_{rc}	λ_{cls}	λ_s
0.1	**92.65**	92.3	92.65	92.49	92.78
0.5	92.69	92.69	92.65	92.59	92.49
1	92.78	**92.65**	**92.65**	**92.65**	**92.65**
10	91.65	91.16	92.07	92.53	92.78
20	88.95	90.32	91.97	92.72	92.95

Table 8: Ablation of Hyperparameters of different losses on Banking dataset for 25% ID classes setting. All presented results in the table are Accuracy.

%	Methods	ACC	F1-SCORE	F1-OPEN	F1-ID
25	SCOOS	**92.65**	**84.75**	**95.29**	**84.2**
	Dual-Bert + Classifier	88.95	79.76	92.87	79.07
50	SCOOS	**87.04**	**86.68**	**88.11**	**86.64**
	Dual-Bert + Classifier	85.48	84.52	86.97	84.45
75	SCOOS	**83.79**	**88.4**	**73.2**	**88.67**
	Dual-Bert + Classifier	81.77	86.54	70.66	86.82

Table 9: Performance comparison of the proposed method with and without using SVAE(i.e., only Dual-BERT with a classifier)

The von Mises-Fisher (vMF) distribution for identifying oos label: The proposed out-of-scope intent classifier learns a shared latent space for both input sequences (Sentences) and the seman-

tics (class label) in order to learn a bounded manifold for each ID class. The distributions of both characteristics are class-wise aligned in the latent space. The Hyper-Spherical Variational Auto-Encoder (SVAE) helps to construct the latent space on a unit hyper-sphere, which is different from the prior latent distribution technique of VAE. Particularly, the von Mises-Fisher (vMF) distribution, whose mean direction ($\mu(v)$ and $\mu(u)$) and concentration ($\kappa(v)$ and $\kappa(u)$) are connected to the relevant class label, is urged to be aligned with the estimated posterior of each input sentence. Since it is simple to determine the manifold border, each class can be represented by a vMF distribution on the unit hyper-sphere. Additionally, the class center may be thought of as the mean direction anticipated by the semantic property. If a sample is projected onto the manifold, we may tell by using the boundary and the class center whether that sample belongs to (m+1) ID classes or OOS class. These vMF distributions of the sentence and class embeddings are approximated by the S-Enc and C-Enc, respectively. As described in Appendix A, both encoders (S-Enc and C-Enc) consist of two fully connected layers. The encoder's hidden node size is 512 and is followed by ReLU, and the output layer is passed to two different fully connected layers to generate mean (dimension=64) and concentration parameter (dimension=1) for vMF distribution.

We would like to point out that the vMF distribution is not directly used for OOS detection. Rather, as shown in Fig 4 of the main paper, it is used as a posterior distribution of encoder to get better embeddings of the ID class samples in the latent space such that the ID class sentence embeddings (denoted by 'o') and the corresponding class name embeddings (denoted by squares) form compact clusters while the OOS samples are scattered around the clusters (denoted by yellow 'x'). OOS detection is performed based on how close a sentence is to its class embedding in the latent space based on the cosine similarity. If the sentence is close to its class embeddings, then it is considered as ID else it is categorized as OOS. How close a sample should be to the class embedding for it to be considered as ID is determined by the threshold (α). It is worth noting that if we use normal distribution instead of vMF distribution then the performance degraded by $3-6\%$ as shown in the table of the main paper.

Computation of mean direction (μ) and concentration (κ): The S-Enc approximates u as $q_{\theta_s}(z_s|u) = q(z_s|\mu(u), \kappa(u))$, where z_s denotes latent space representations of sentence embeddings, $\mu(u)$ and $\kappa(u)$ represents the mean direction and the concentration around $\mu(u)$, respectively. Similarly, the C-Enc approximates v as $q_{\theta_s}(z_c|v) = q(z_c|\mu(v), \kappa(v))$, where z_c denotes latent space representations of class embeddings, $\mu(v)$ and $\kappa(v)$ represent the mean direction and the concentration around $\mu(v)$, respectively. In the S-Enc and C-Enc, both prior and posterior distributions are based on vMF distributions in the hypersphere. We briefly explain the S-Enc used to obtain the $\mu(u)$ and $\kappa(u)$, and the explanation also applies to the C-enc used to obtain $\mu(v)$ and $\kappa(v)$ as follows:

$$q(z_s|\mu(u), \kappa(u)) = C_m(\kappa(u)) \exp(\kappa(u)\mu(u)^T z_s) \tag{10}$$

$$C_m(\kappa(u)) = \frac{\kappa(u)^m/2 - 1}{(2\pi)^{m/2} I_{m/2-1}(\kappa(u))}, \tag{11}$$

where $\mu(u) \in \mathbb{R}^m, ||\mu(u)||_2 = 1$ denotes the mean direction on the hypersphere and $\kappa(u) \in \mathbb{R}_{\geq 0}$ denotes the concentration around $\mu(u)$. $C_m(\kappa(u))$ is the normalizing constant, and I_m is the modified Bessel function of the first kind at order m. As our objective is to minimize the Wasserstein distance between the latent spaces of both encoders to align both representations, we compute this distance using the Sinkhorn iteration algorithm.

G Why to choose ablation over 25% case?

As the proportion of categories within the domain increases, the setting moves towards closed-world. In the near closed-world setting (75% ID classes), generally, all methods performed well as it entails a large number of ID samples which enables it for better classification. In such a setting (75% ID classes), even a simple threshold-based method (MSP) performs well. Moreover, as a large number of ID class labels are readily available in such a setting, any supervised learning model can also be used with a simple threshold for OOS detection, and they will yield decent performance in OOS detection. However, the setting with lower ID samples (For eg. 25% ID classes) is more challenging, where the proposed method outperforms the simple threshold-based MSP by 34.66% on the Banking dataset. It should be noted that a good OOS detection algorithm should be able to learn from a small set of ID samples, which our proposed method does as it performs much better than SOTA in the 25% ID class setting.