UniNL: Aligning Representation Learning with Scoring Function for OOD Detection via Unified Neighborhood Learning

Yutao Mou[1*], **Pei Wang**[1*], **Keqing He**[2*], **Yanan Wu**[1]
Jingang Wang[2], **Wei Wu**[2], **Weiran Xu**[1*]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan, Beijing, China
{myt,wangpei,yanan.wu,xuweiran}@bupt.edu.cn
{hekeqing,wangjingang,wuwei}@meituan.com

Abstract

Detecting out-of-domain (OOD) intents from user queries is essential for avoiding wrong operations in task-oriented dialogue systems. The key challenge is how to distinguish in-domain (IND) and OOD intents. Previous methods ignore the alignment between representation learning and scoring function, limiting the OOD detection performance. In this paper, we propose a unified neighborhood learning framework (UniNL) to detect OOD intents. Specifically, we design a K-nearest neighbor contrastive learning (KNCL) objective for representation learning and introduce a KNN-based scoring function for OOD detection. We aim to align representation learning with scoring function. Experiments and analysis on two benchmark datasets show the effectiveness of our method. [1]

1 Introduction

Out-of-domain (OOD) intent detection aims to know when a user query falls outside the range of pre-defined supported intents, which helps to avoid performing wrong operations and provide potential directions of future development in a task-oriented dialogue system (Akasaki and Kaji, 2017; Tulshan and Dhage, 2018; Shum et al., 2018; Lin and Xu, 2019; Xu et al., 2020; Zeng et al., 2021a,b). Compared with normal intent detection tasks, we don't know the exact number and lack labeled data for unknown intents, which makes it challenging to identify OOD samples in the task-oriented dialog.

Previous OOD detection works can be generally classified into two types: supervised (Fei and Liu, 2016; Kim and Kim, 2018; Larson et al., 2019; Zheng et al., 2020) and unsupervised (Bendale and Boult, 2016; Hendrycks and Gimpel, 2017; Shu et al., 2017; Lee et al., 2018; Ren et al., 2019; Lin and Xu, 2019; Xu et al., 2020; Zeng et al., 2021a) OOD detection. The former indicates that there are extensive labeled OOD samples in the training data. Fei and Liu (2016); Larson et al. (2019), form a *(N+1)*-class classification problem where the *(N+1)*-th class represents the OOD intents. Further, Zheng et al. (2020) uses labeled OOD data to generate an entropy regularization term. But these methods require numerous labeled OOD intents to get superior performance, which is unrealistic. We focus on the unsupervised OOD detection setting where labeled OOD samples are not available for training. Unsupervised OOD detection first learns discriminative representations only using labeled IND data and then employs scoring functions, such as Maximum Softmax Probability (MSP) (Hendrycks and Gimpel, 2017), Local Outlier Factor (LOF) (Lin and Xu, 2019), Gaussian Discriminant Analysis (GDA) (Xu et al., 2020) to estimate the confidence score of a test query.

All these unsupervised OOD detection methods only focus on the improvement of a single aspect of representation learning or scoring function, but none of them consider how to align representation learning with scoring functions. For example, Lin and Xu (2019) proposes a local outlier factor for OOD detection, which considers the local density of a test query to determine whether it belongs to an OOD intent, but the IND pre-training objective LMCL (Wang et al., 2018) cannot learn neighborhood discriminative representations. Xu et al. (2020); Zeng et al. (2021a) employ a gaussian discriminant analysis method for OOD detection, which assumes that the IND cluster distribution is a gaussian distribution, but they use a cross-entropy or supervised contrastive learning (Khosla et al., 2020) objective for representation learning, which cannot guarantee that such an assumption is satisfied. The gap between representation learning and scoring function limits the overall performance of these methods.

*The first three authors contribute equally. Weiran Xu is the corresponding author.

[1]We release our code at https://github.com/Yupei-Wang/UniNL

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7317 - 7325
December 7-11, 2022

To solve the conflict, in this paper, we propose a **Uni**fied **N**eighborhood **L**earning framework (**UniNL**) for OOD detection, which aims to align IND pre-training representation objectives with OOD scoring functions. Our intuition is to learn neighborhood knowledge (Breunig et al., 2000a) to detect OOD intents. For IND pre-training, we introduce a K-Nearest Neighbor Contrastive Learning Objective (KNCL) to learn neighborhood discriminative representations. Compared to SCL (Zeng et al., 2021a) which draws all samples of the same class closer, KNCL only pulls together similar samples in the neighbors. To align KNCL, we further propose a K-nearest neighbor scoring function, which estimates the test sample confidence score by computing the average distance between a test sample and its K-nearest neighbor samples. The KNCL objective learns neighborhood discriminative knowledge, which is more beneficial to promoting KNN-based scoring functions.

Our contributions are three-fold: (1) We propose a unified neighborhood learning framework (UniNL) for OOD detection, which aims to match IND pre-training objectives with OOD scoring functions. (2) We propose a K-nearest neighbor contrastive learning (KNCL) objective for IND pre-training to learn neighborhood discriminative knowledge, and a KNN-based scoring function to detect OOD intents. (3) Experiments and analysis demonstrate the effectiveness of our method for OOD detection.

2 Approach

Overall Architecture Fig 1 shows the overall architecture of our proposed UniNL, which includes K-nearest contrastive learning (KNCL) and KNN-based score function. We first train an in-domain intent classifier using our KNCL objective in the training stage, which aims to learn neighborhood discriminative representation. Then in the test stage, we extract the intent feature of a test query and employ our proposed KNN-based score function to estimate the confidence score. We aim to align representation learning and scoring functions.

KNN Contrastive Representation Learning Existing OOD detection methods generally adopt cross-entropy (CE) (Xu et al., 2020) and supervised contrastive learning (SCL) (Zeng et al., 2021a) objectives for representation learning. Both CE and SCL tend to bring all samples of the same classes closer, and samples of different classes are pushed

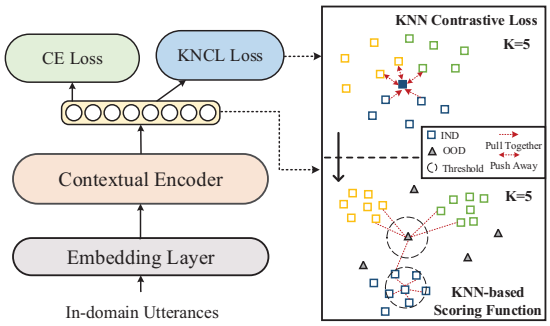


Figure 1: Overall architecture of UniNL.

away. They learn inter-class discriminative features in a global representation space. However, we find that when performing OOD detection, we care more about the data distribution within the neighborhood of a given sample. Inspired by Breunig et al. (2000b), we hope to learn neighborhood discriminative knowledge in the IND pre-training stage to facilitate OOD detection. We propose a K-nearest neighborhood contrastive learning (KNCL) objective to learn discriminative features in a local representation space. Given an IND sample x_i, we firstly obtain its intent representation z_i using a BiLSTM (Hochreiter and Schmidhuber, 1997) or BERT (Devlin et al., 2019) encoder. Next, we perform KNCL as follows:

$$\mathcal{L}_{KNCL} = \sum_{i=1}^{N} -\frac{1}{|N_k(i)|} \sum_{j=1}^{N_k(i)} \mathbf{1}_{i \neq j} \mathbf{1}_{y_i = y_j} \log \frac{\exp\left(z_i \cdot z_j / \tau\right)}{\sum_{k=1}^{N_k(i)} \mathbf{1}_{i \neq k} \exp\left(z_i \cdot z_k / \tau\right)} \quad (1)$$

where $N_k(i)$ is the KNN set of z_i in the representation space. KNCL only draws closer together samples of the same class in the neighborhood. Specifically, given an anchor, KNCL first finds its KNN set in a batch, and then selects samples of the same class as positives, and different classes as negatives. Similar to Zeng et al. (2021a), we use an adversarial augmentation strategy to generate augmented views of the original samples within a batch. In the implementation, we first pre-train the intent classifier using KNCL, then finetune the model using CE, both on the IND data. We leave the implementation details in the appendix. Section 3.3 proves that KNCL learns neighborhood discriminative knowledge and helps to distinguish IND from OOD.

KNN-based Score Function To align with the KNCL representation learning objective, we pro-

Detection	Training	CLINC-Full				CLINC-Small			
		IND		OOD		IND		OOD	
		ACC	F1	Recall	F1	ACC	F1	Recall	F1
MSP	CE	91.21	87.75	45.28	54.90	88.98	86.22	45.16	52.04
	SCL	91.10	87.69	47.98	59.64	89.62	85.04	34.64	47.33
	KNCL(ours)	91.09	87.05	**58.73**	**66.98**	90.15	86.63	**47.26**	**59.55**
LOF	CE	85.46	85.80	57.40	58.78	82.45	82.73	52.88	53.90
	SCL	86.52	86.80	60.72	61.80	83.13	83.39	56.88	57.48
	KNCL(ours)	85.77	85.05	**70.10**	**66.30**	86.13	85.6	**60.92**	**62.63**
GDA	CE	86.34	87.73	63.72	65.23	84.24	84.30	60.40	61.07
	SCL	87.01	88.28	66.80	67.68	87.07	86.54	61.46	63.83
	KNCL(ours)	88.45	87.08	**71.59**	**70.37**	87.15	87.53	**64.00**	**66.18**
KNN(ours)	CE	90.30	88.89	67.03	72.32	88.64	86.85	60.48	66.65
	SCL	89.42	88.69	71.45	74.17	89.45	87.35	62.23	70.42
	KNCL(ours)	91.24	88.28	**72.08**	**76.53**	89.32	87.61	**66.00**	**72.79**

Table 1: The performance of different OOD scoring functions and IND pre-training objectives on CLINC-Full and CLINC-Small datasets for the BiLSTM-based model (p<0.01 under t-test). The last line is our full UniNL model.

pose a KNN-based scoring function, which makes full use of the neighborhood data distribution to estimate confidence scores. Specifically, given a test query x_i, we first obtain its intent representation z_i through the pre-trained encoder, and then perform L2 normalization. For each sample in the test set, we find its KNN set from the training set, and then calculate the average Euclidean distance as the scoring function. The formula of the KNN-based scoring function is as follows:

$$\mathcal{G}_\lambda(x_i) = \begin{cases} IND & \text{if } \mathcal{S}(x_i) < \lambda \\ OOD & \text{if } \mathcal{S}(x_i) \geq \lambda \end{cases} \quad (2)$$

$$S(x_i) = \frac{1}{|N_k(x_i)|} \sum_{j=1}^{N_k(x_i)} ||z_i - z_j||_2 \quad (3)$$

where $N_k(x_i)$ is the KNN set of test query x_i from the training set, λ is the threshold, and we use the best IND F1 scores on the validation set to calculate the threshold adaptively. The KNN-based scoring function needs to consider the data distribution in the neighborhood of a test query to determine whether it is an OOD sample, and the KNCL objective function distinguishes samples of different classes in the neighborhood, so we believe that KNCL representation learning objective aligns with the KNN-based scoring function, which is beneficial to improve the OOD detection performance. We discuss it in section 3.3.

3 Experiments

3.1 Setup

Datasets We perform experiments on four public benchmark OOD datasets, CLINC-Full, CLINC-Small (Larson et al., 2019), Banking (Casanueva et al., 2020) and Stackoverflow (Xu et al., 2015). **Metrics** We use four common metrics for OOD detection to measure the performance, including IND metrics: Accuracy and macro F1, and OOD metrics: Recall and F1. OOD Recall and F1 are the main evaluation metrics. **Baselines** We compare UniNL with different pre-training objectives CE and SCL, and scoring functions including MSP, LOF and GDA. Besides, we also compare our model with the following state-of-the-art baselines, SCL(Zeng et al., 2021a), Energy(Ouyang et al., 2021), ADB(Zhang et al., 2021) and KNN-CL(Zhou et al., 2022). For a fair comparison, we use the same BiLSTM and BERT as backbone. We provide a more comprehensive comparison and implementation details of these methods in Appendix A.

3.2 Main Results

Table 1 show the main results on BiLSTM. Our proposed UniNL significantly outperforms all the baselines, which shows that aligning IND pre-training objectives with OOD scoring functions helps improve OOD detection. For example, for OOD metrics, our UniNL outperforms the previous state-of-the-art method SCL+GDA (Zeng et al., 2021a) by 5.28%(Recall) and 8.85%(F1) on CLINC-Full, 4.54%(Recall) and 8.96%(F1) on CLINC-Small. Compared to CE and SCL, KNCL shows significant improvements under all the scoring functions. And our proposed KNN-based scoring function also outperforms previous methods MSP, LOF and GDA. For IND metrics, we find there is no significant difference, which denotes UniNL improves

Models	CLINC-Full		Banking-25%		Banking-75%		Stackoverflow-25%		Stackoverflow-75%	
	IND F1	OOD F1	IND F1	OOD F1	IND F1	OOD F1	IND F1	OOD F1	IND F1	OOD F1
SCL(Zeng et al., 2021a)	90.03	68.21	63.32	75.82	86.56	67.51	82.45	94.09	85.76	57.46
Energy(Ouyang et al., 2021)	91.23	75.93	-	-	-	-	-	-	-	-
ADB(Zhang et al., 2021)	90.94	76.52	-	-	-	-	-	-	-	-
KNN-CL(Zhou et al., 2022)	92.61	76.36	76.44	90.19	87.41	67.66	79.39	92.70	87.92	74.20
UniNL(Ours)	**93.58**	**78.92**	**78.03**	**92.46**	87.13	**69.44**	**87.70**	**96.33**	**88.19**	**74.54**

Table 2: The performance of our UniNL compared with previous state-of-the-art baselines using BERT.

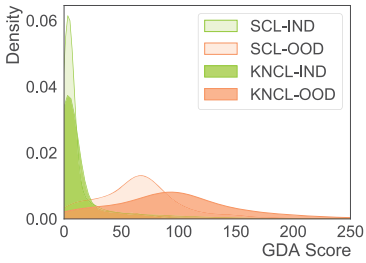


Figure 2: GDA score distribution curves of IND and OOD data using different IND pre-training losses SCL and KNCL.

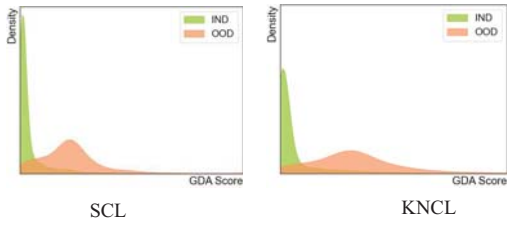


Figure 3: Confidence score distribution curves of IND and OOD data using different scoring functions GDA and KNN.

OOD performance without hurting IND classification. Table 2 also proves our UniNL achieves the state-of-the-art with the same BERT backbone as baselines. All the results show the effectiveness of our proposed KNCL pre-training loss and KNN-based scoring function. Learning unified neighborhood knowledge is beneficial to OOD detection.

3.3 Qualitative Analysis

Effect of KNCL To show the effect of KNCL, we adopt GDA as the score function and compare the GDA score distribution curves of IND and OOD data under different pre-training objectives in Fig 2. The smaller overlapping area of IND and OOD curves means better performance. We find that KNCL makes the aliasing area of IND and OOD smaller, and improves OOD F1 by 2.69% compared to SCL. This proves that neighborhood discriminative features help distinguish IND from OOD.

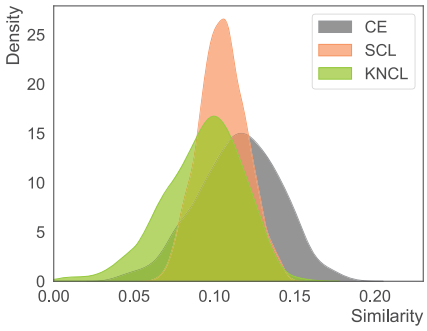
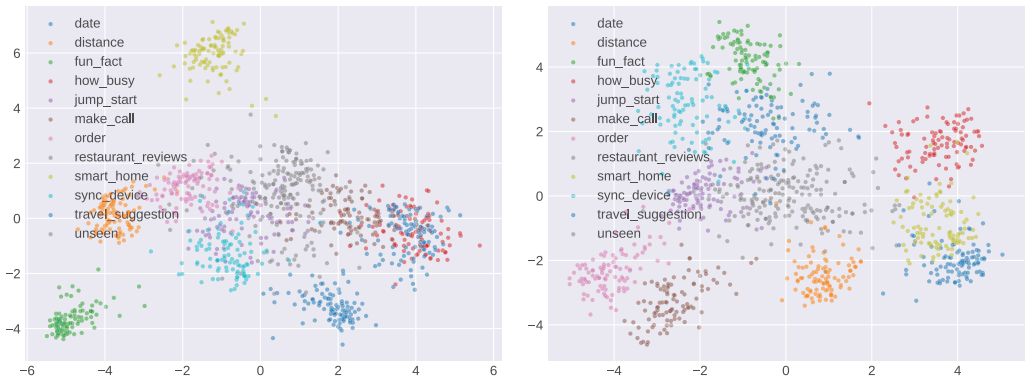


Figure 4: Similarity score distribution between OOD and IND (The smaller the similarity, the easier it is to distinguish between IND and OOD).

Alignment between KNCL and KNN With KNCL as the pre-training objective, we compare the GDA and KNN score distribution curves for IND and OOD data, as shown in Fig 3. The more separated the IND and OOD distribution curves are, the more favorable it is for OOD detection. It can be seen that the KNN scores have better effect on distinguishing IND and OOD, which also indicates that aligning the IND pre-training objective and the OOD scoring function helps to improve the performance of OOD detection.

Why Alignment Works Well To discuss why the KNCL representation learning objective matches the KNN-based scoring function, we compare the cosine similarity distance between OOD and IND under different representation learning objectives in Fig 4. For each OOD sample in the test set, we calculate the average of its cosine similarity scores with the K-nearest neighbor IND samples from training data, and obtain the cosine similarity score distribution curve. We find that KNCL can decrease the average similarity between OOD and IND, which has a boosting effect on the KNN-based scoring function.

The effect of the K value for KNCL The KNCL pre-training objective requires a reasonable choice of K value. We compare the OOD F1 scores under different batch sizes and K values, as shown in Fig 6. We found that the batch size will affect the choice of K value. When the batch size is larger, a



(a) SCL (b) KNCL

Figure 5: Visualization of IND and OOD intents (The "unseen" legend label means OOD intents).

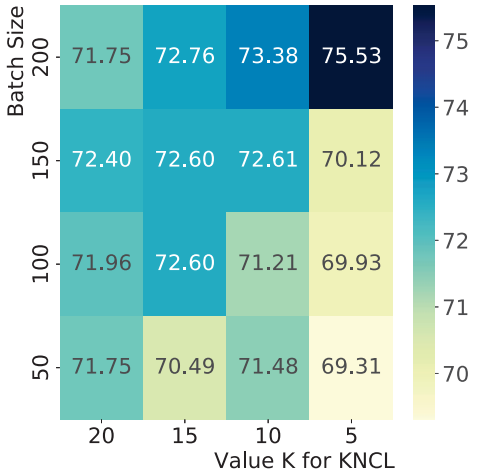


Figure 6: Effect of batch size and K value of KNCL on clustering performance. The larger the number is, the deeper the color is.

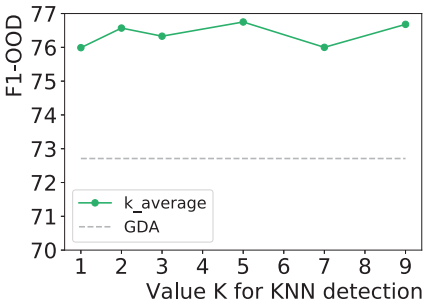


Figure 7: Effect of K value of KNN-based score function on clustering performance.

smaller K value can achieve better results; when the batch size is smaller, a larger K value is required to achieve good performance. We argue that this is because the larger the batch size is, the better it can approximate the real distribution of the entire dataset, and a smaller K value can simulate the real neighborhood distribution.

The effect of the K value for KNN score The K value of the KNN-based score function is also an important hyperparameter, and we compare the effect of different K values on the OOD detection performance, as shown in Fig 7. It can be seen that this K value has little effect on the OOD detection performance, which illustrates the robustness of our proposed KNN-based detection method.

Visualization Fig 5 displays IND and OOD intent visualization for different IND pre-training objectives SCL and KNCL. It shows that the SCL objective will confuse the OOD samples with the "jump_start" and "make_call" intent types, while KNCL can distinguish them well. This proves that KNCL can better distinguish IND and OOD by modeling neighborhood discriminative features, which is beneficial to improving the performance of the KNN-based scoring function.

4 Conclusion

In this paper, we focus on how to align representation learning with the scoring function to improve OOD detection performance. We propose a unified neighborhood learning framework (UniNL) to detect OOD intents, in which a KNCL objective is employed for IND pre-training and a KNN-based score function is used for OOD detection. Experiments and analyses confirm the effectiveness of UniNL for OOD detection. We hope to provide new guidance for future OOD detection work.

Limitations

This paper mainly focuses on how to align the representation learning and scoring functions to achieve better OOD detection performance. Thus we follow

similar experiment settings as previous work. However, similar to these works, we only experiment with datasets in the field of intent recognition. Actually, OOD detection has applications in a wider range of NLP topics, such as relation classification, entity recognition, text classification, etc. We will try our proposed method on more NLP topics in the future to verify the universality.

Acknowledgements

We thank all anonymous reviewers for their helpful comments and suggestions. This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

References

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

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

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

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

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

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

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.

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

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, A. Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *ArXiv*, abs/2004.11362.

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

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

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

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

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.

Yawen Ouyang, Jiasheng Ye, Yu Chen, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2021. Energy-based unknown intent detection with data manipulation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2852–2861, Online. Association for Computational Linguistics.

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

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

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.

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

Amrita S Tulshan and Sudhir Namdeorao Dhage. 2018. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *International symposium on signal*

processing and intelligent recognition systems, pages 190–201.

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.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1452–1460, Barcelona, Spain (Online). International Committee on 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 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021a. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 870–878, Online. Association for Computational Linguistics.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Hong Xu, and Weiran Xu. 2021b. Adversarial self-supervised learning for out-of-domain detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5631–5639, Online. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. In *AAAI*.

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.

A Experiment Setups

A.1 Datasets

We perform experiments on four public benchmark OOD datasets, CLINC-Full, CLINC-Small (Larson et al., 2019), Banking (Casanueva et al., 2020) and Stackoverflow (Xu et al., 2015). We show the detailed statistics of the datasets in Table 3 .

Statistic	CLINC-Full	CLINC-Small	Banking	Stackoverflow
Avg utterance length	9	9	12	10
Intents	150	150	77	20
Training set size	15100	7600	9003	12000
Training samples per class	100	50	-	-
Training OOD samples amount	100	100	-	-
Development set size	3100	3100	1000	2000
Development samples per class	20	20	-	-
Development OOD samples amount	100	100	-	-
Testing Set Size	5500	5500	3080	6000
Testing samples per class	30	30	-	-
Development OOD samples amount	1000	1000	-	-

Table 3: Statistics of the CLINC datasets.

A.2 Baselines

We compare UniNL with different pre-training objectives and different scoring functions. For the feature extractor, we use the same BiLSTM(Hochreiter and Schmidhuber, 1997) or BERT (Devlin et al., 2019) as backbone. We compare our training objective KNCL with CE and SCL(Zeng et al., 2021a), and scoring function KNN with MSP(Hendrycks and Gimpel, 2017), LOF(Lin and Xu, 2019) and GDA(Xu et al., 2020). Besides, we also compare our model with the following state-of-the-art baselines, Energy(Ouyang et al., 2021) and ADB(Zhang et al., 2021). We supplement the relevant baseline details as follows:

MSP (Maximum Softmax Probability)(Hendrycks and Gimpel, 2017) uses maximum softmax probability as the confidence score. If the score is lower than a fixed threshold, the query is regarded as OOD. In this paper, we use the best IND macro F1 scores on the validation set to calculate the threshold adaptively.

LOF (Local Outlier Factor)(Lin and Xu, 2019) uses the local outlier factor to detect unknown intents. It detects OOD by comparing the local density of a test query with its k-nearest neighbor's. If a query's local density is significantly lower than its k-nearest neighbor's, it is more likely to be regarded as OOD.

GDA (Gaussian Discriminant Analysis)(Xu et al., 2020) is a generative distance-based classifier to detect OOD samples. It estimates the class-conditional distribution on feature spaces of DNNs via Gaussian discriminant analysis, and then applies Mahalanobis distance to measure the confidence score. When estimating the class conditional

distribution with labeled IND data, we assume that it follows a Gaussian distribution. However, the representation space modeled by the cross-entropy objective cannot actually satisfy the Gaussian distribution assumption.

SCL (Zeng et al., 2021a) uses a supervised contrastive learning objective to minimize intra-class variance by pulling together in-domain intents belonging to the same class and maximize inter-class variance by pushing apart samples from different classes. To keep fair comparison, we follow the original paper using GDA as score function.

Energy (Ouyang et al., 2021) maps a sample x to a single scalar called the energy. It uses the threshold on the energy score to consider whether a test query belongs to OOD.

ADB (Zhang et al., 2021) learns adaptive decision boundary using a loss function to balance both the empirical risk and the open space risk.

KNN-CL (Zhou et al., 2022) is concurrent work with our UniNL. It proposes a KNN-based contrastive loss for IND pre-training, which is conceptually similar to our KNCL. But our implementations are different: KNN-CL selects k-nearest neighbors from samples of the same class as positives and uses samples of the different classes as negatives. Our KNCL only uses the k-nearest neighbors of an anchor as the positive and negative set, which is more efficient and doesn't require a large momentum queue as KNN-CL. Moreover, we aim to align IND pre-training representation objectives with OOD scoring functions instead of proposing a better IND pre-training loss.

A.3 Implementation Details

To conduct a fair comparison, we follow a similar evaluation setting as Zeng et al. (2021a). We use the public pre-trained GloVe 300 dimensions embeddings(Pennington et al., 2014) and BERT-uncased model to embed tokens. We set the learning rate to 1e-03 for LSTM and 1e-04 for BERT(Devlin et al., 2019). We use Adam optimizer(Kingma and Ba, 2014) to train our model and set the dropout rate to 0.5. In the training stage, we firstly conduct 100 epochs of K-nearest neighbor contrastive training, and then 10 epochs with CE. The K value of KNCL objective is set to 5. When performing KNCL, we first select the KNN set on the original view, and then extend the KNN set to its augmented view to participate in the calculation of the contrast loss. In the test stage, we set the K value of KNN scoring function to 5. And we use the best IND macro F1 scores on the validation set to calculate the threshold adaptively. To avoid randomness, we average results over 5 random runs. Table 4 shows the comparison of the computational efficiency for different pre-training objectives and scoring functions. For training efficiency, we report the running time of each epoch; for inference efficiency, we report the total running time on the entire test set of CLINC-Full. It can be seen that the inference efficiency of our proposed KNN-based scoring function has been greatly improved. Compared with GDA, the efficiency of KNN score is increased by 14.13 times. The training efficiency for KNCL objective function has a slight decrease due to the need for K-nearest neighbor search, but it gives about 2%-4% OOD F1 performance improvement for all scoring functions. All experiments use a single Tesla T4 GPU(16 GB of memory).

Method		training time (seconds / epoch)	inference time (seconds)
Training objectives	CE	6	-
	SCL	10	-
	KNCL(ours)	20	-
Score functions	LOF	-	104.60
	GDA	-	6.50
	KNN(ours)	-	0.46

Table 4: The comparison of the computational efficiency for different pre-training objectives and scoring functions.

Training	CLINC-Full			
	IND ACC	IND F1	OOD Recall	OOD F1
only KNCL	-	-	62.41	60.31
only CE	90.30	**88.89**	67.03	73.32
CE+KNCL	85.30	83.12	71.55	72.57
multitask	90.38	87.47	69.25	73.16
KNCL+CE	**91.24**	88.28	**72.08**	**76.53**

Table 5: Comparison of different training strategies in the IND pre-training stage.

B Ablation Study

In order to verify which training strategy is the most effective in the IND pre-training stage, we compared the combination of different training objectives, and the results are shown in Table 5. We conduct experiments on the CLINC-Full dataset, using BiLSTM as encoder and KNN as scoring function. Only CE is the baseline that only uses the cross-entropy loss function to train the feature extractor. Only KNCL means that we use KNCL objective for representation learning. KNCL+CE means that we first train the feature extractor us-

ing KNCL, and then fine-fune it using CE loss. CE+KNCL means that we first train the network by minimizing cross-entropy loss, and then conduct K-nearest neighbor contrastive learning. Besides, we also compare the simple multitask paradigm, which simply adds the CE and KNCL objective functions for joint optimization. The results show that the best performance can be achieved by first learning the neighborhood discriminative knowledge with KNCL and then fine-tuning the model with CE [2].

[2]When only KNCL is used to train the feature extractor, softmax classifier cannot be used for IND classification, so we do not report relevant IND results here.