Class Lifelong Learning for Intent Detection via Structure Consolidation Networks

Qingbin Liu[1], **Yanchao Hao**[1], **Xiaolong Liu**[1], **Bo Li**[1], **Dianbo Sui**[2], **Shizhu He**[3,4], **Kang Liu**[3,4], **Jun Zhao**[3,4], **Xi Chen**[1]*, **Ningyu Zhang**[5], **Jiaoyan Chen**[6]
[1] Platform and Content Group, Tencent, China
[2] Harbin Institute of Technology, Weihai, China
[3] The Lab of Cognition and Decision Intelligence for Complex Systems, CASIA, China
[4] School of Artificial Intelligence, University of Chinese Academy of Sciences, China
[5] Zhejiang University & AZFT Joint Lab for Knowledge Engine, Zhejiang, China
[6] Department of Computer Science, The University of Manchester, UK
{qingbinliu, marshao, loongliu, ryanbli}@tencent.com, suidianbo@hit.edu.cn,
{shizhu.he, kliu, jzhao}@nlpr.ia.ac.cn, jasonxchen@tencent.com,
zhangningyu@zju.edu.cn, jiaoyan.chen@manchester.ac.uk

Abstract

Intent detection, which estimates diverse intents behind user utterances, is an essential component of task-oriented dialogue systems. Previous intent detection models are usually trained offline, which can only handle predefined intent classes. In the real world, new intents may keep challenging deployed models. For example, with the prevalence of the COVID-19 pandemic, users may pose various issues related to the pandemic to conversational systems, which brings many new intents. A general intent detection model should be intelligent enough to continually learn new data and recognize new arriving intent classes. Therefore, this work explores Class Lifelong Learning for Intent Detection (CLL-ID), where the model continually learns new intent classes from new data while avoiding catastrophic performance degradation on old data. To this end, we propose a novel lifelong learning method, called Structure Consolidation Networks (SCN), which consists of structure-based retrospection and contrastive knowledge distillation to handle the problems of expression diversity and class imbalance in the CLL-ID task. In addition to formulating the new task, we construct 3 benchmarks based on 8 intent detection datasets. Experimental results demonstrate the effectiveness of SCN, which significantly outperforms previous lifelong learning methods on the three benchmarks.

1 Introduction

Task-oriented dialogue systems provide a natural interface to help users accomplish a wide range of tasks, such as playing music, handling money transfer business, and providing information about the COVID-19 pandemic. Intent detection is an essential component of task-oriented dialogue systems, which aims to accurately estimate diverse user intents for downstream modules (Hemphill et al., 1990; Coucke et al., 2018). For example, given the user utterance "*Tell me some ways to avoid coronavirus*", an intent detection model should classify it into the intent class "*how to protect yourself*".

Existing intent detection models usually perform once-and-for-all training on a fixed dataset and can only handle predefined intent classes. However, this setting may not be practical enough in the real world, as new intent classes continually emerge after the model is deployed. For example, with the prevalence of the COVID-19 pandemic, users may pose various issues related to the pandemic to conversational systems, which brings many new intents, such as "*how to protect yourself*" and "*the latest number of infections*". A general intent detection model should be able to flexibly and efficiently learn new intents round by round. Therefore, this work proposes a realistic and challenging task, Class Lifelong Learning for Intent Detection (CLL-ID). This task continually trains an intent detection model using new data to learn new intents. At any time, the updated model should be able to perform accurate classification for all intents observed so far.

In the CLL-ID task, it is often infeasible to retrain the model from scratch with the data of all seen classes due to computational cost and data privacy (McMahan et al., 2017; Li et al., 2021). For example, the time to train a model with all data of the CLINC benchmark (Larson et al., 2019) is approximately 9.8 times longer than the time to train the same model with only new data. In practice,

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 293–306
July 9-14, 2023

Intent: *How to protect yourself?*
User 1: How can I protect against the virus?
User 2: Do any medications protect against the virus?
User 3: Will wearing gloves help me avoid COVID-19?
User 4: What kind of face mask helps?

Figure 1: An example of expression diversity. Different users have different expressions for the same intent.

virtual assistants, such as Alexa and Siri, typically provide a large number of services, which makes the time overhead of continual retraining extremely high (Rastogi et al., 2020). Moreover, the CLL-ID task allows flexible and scalable applications on embedded devices that have limited computing power and storage capacity, such as smartphones, to learn user-specific intents without privacy risks (Kemker and Kanan, 2018).

A plain lifelong learning method is to fine-tune a model pre-trained on old data directly on new data. However, this method usually suffers from catastrophic performance degradation on old data, also known as catastrophic forgetting (McCloskey and Cohen, 1989; French, 1999). To cope with this issue, current mainstream lifelong learning methods usually maintain a memory to store a small number of representative old data (Wang et al., 2019; Han et al., 2020; Cui et al., 2021).

However, when directly applying existing lifelong learning methods to the CLL-ID task, we find two severe problems: expression diversity and class imbalance. **Expression Diversity**: In the intent detection task, there are various expression types for the same intent class, as shown in Figure 1. Previous methods usually preserve similar old samples that involve only a few expression types and are inconsistent with the original data distribution. These samples are not conducive to maintaining the performance of the old intent classes. **Class Imbalance**: At each step of the lifelong learning process, there is generally a large amount of new data, yet only a small amount of old data is preserved due to the memory capacity limitation, leading to a severe imbalance between the new and old intent classes. In this case, the model will be significantly biased towards learning new data, leading to catastrophic forgetting on old data.

To address the above two problems, we propose Structure Consolidation Networks (SCN), which contains two core components: (1) to handle the problem of expression diversity, we propose structure-based retrospection, which selects and preserves diverse and informative old data based on the spatial structure of features; (2) to cope with the class imbalance problem, we propose contrastive knowledge distillation, which preserves the knowledge of the model trained at the previous step and improves the generalization between the old and new intent classes through contrastive learning. For the CLL-ID task, we constructed 3 benchmarks based on 8 widely used intent detection datasets. Experimental results show that SCN significantly outperforms previous lifelong learning methods. In summary, the contributions of this work are as follows:

- We formally introduce class lifelong learning into intent detection and we construct 3 benchmarks through 8 intent detection datasets.
- We propose structure consolidation networks, which can effectively handle expression diversity and class imbalance in the CLL-ID task through structure-based retrospection and contrastive knowledge distillation.
- Experimental results show that SCN significantly outperforms previous lifelong learning methods on the three benchmarks. The source code and benchmarks will be released for further research (`https://github.com/liuqingbin2022/CLL4ID`).

2 Task Formulation

The traditional intent detection task is usually formulated as a text classification task, which predicts an intent class for each input utterance (Hemphill et al., 1990; Coucke et al., 2018). The CLL-ID task adopts a realistic setting where the intent detection model is continually trained on new data to learn new intents. That is, new data arrives in a stream form, denoted as $(\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K)$. Each data $\mathcal{D}_i$ has its own training/validation/test set ($\mathcal{D}_i^{\text{train}}$, $\mathcal{D}_i^{\text{valid}}$, $\mathcal{D}_i^{\text{test}}$), as well as its own label set $\mathcal{C}_i$. The label set $\mathcal{C}_i$ contains one or multiple new classes that do not appear in the previous steps. When new data arrives, the intent detection model is updated using the new training set $\mathcal{D}_i^{\text{train}}$, and uniformly classifies each sample according to all observed intents (i.e., $\tilde{\mathcal{C}}_i = \bigcup_{n=1}^{i} \mathcal{C}_n$). The updated model should perform well on all seen classes. Therefore, in the testing stage of the i-th step, we evaluate the updated model on the test data of all observed classes (i.e., $\tilde{\mathcal{D}}_i^{\text{test}} = \bigcup_{n=1}^{i} \mathcal{D}_n^{\text{test}}$).

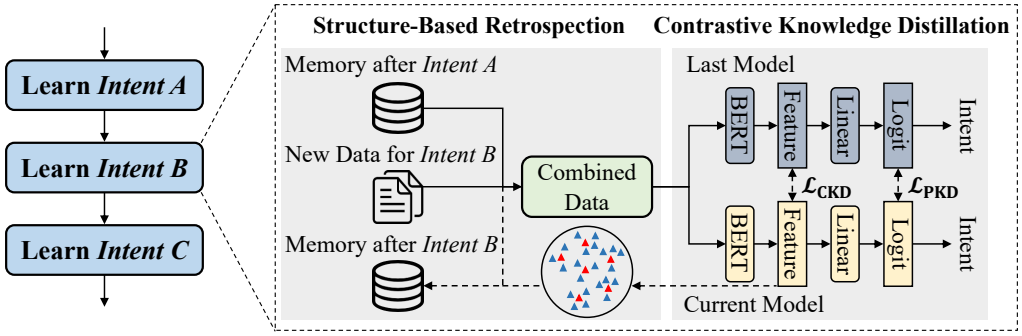


Figure 2: Illustrations of SCN. When learning the intent set *B*, the model is updated with the combination of the new training data for the *Intent B* and the old data in memory. SCN first adopts contrastive knowledge distillation to retain previous knowledge. Then, the method stores new representative samples through structure-based retrospection.

The arrival of new data round by round will constantly change the original data distribution, which makes it increasingly difficult for intent detection models to achieve high performance on old data. We experimentally demonstrate this claim in Section 4.4. Thus, how to alleviate catastrophic performance degradation on old data is a central research point of the CLL-ID task.

3 Method

In this work, we propose Structure Consolidation Networks to handle the CLL-ID task. The overall framework of SCN is shown in Figure 2. SCN consists of two core components, i.e., structure-based retrospection and contrastive knowledge distillation. Structure-based retrospection preserves diverse and informative samples to deal with the problem of expression diversity. Contrastive knowledge distillation alleviates the negative effects of class imbalance through knowledge distillation and contrastive learning.

3.1 Background

SCN is a model-agnostic lifelong learning method. The intent detection model is only a basic component and is not the focus of our research. We employ a BERT-based classifier as the base model because it proved to be a powerful model for intent detection (Devlin et al., 2019; Zhan et al., 2021). BERT is a pre-trained language model based on the Transformer architecture (Vaswani et al., 2017).

To match the input form of BERT, we add two tokens [CLS] and [SEP] at the beginning and end of each input sequence. The BERT encoder outputs the contextual representation for each sequence. We use the hidden state of the [CLS] token as the feature vector and feed it into a linear layer to calculate the probability. The cross-entropy loss is used to train the intent detection model:

$$\mathcal{L}_{\text{CE}} = -\sum_{n=1}^{|\mathcal{N}|} \boldsymbol{y}_n \log(\boldsymbol{p}_n), \tag{1}$$

where $\boldsymbol{y}_n$ is the ground-truth label and $\boldsymbol{p}_n$ is the predicted probability. $\mathcal{N}$ is the training samples.

3.2 Structure-Based Retrospection

To learn new intent classes, we study class lifelong learning for intent detection, which aims to train a unified model to handle all observed classes so far. Given a model trained on old data, we continually train the model based on a new combined dataset $\mathcal{N} = \mathcal{D}_i^{\text{train}} \bigcup \mathcal{M}$. $\mathcal{D}_i^{\text{train}}$ is the training data of the new intent classes at step i. $\mathcal{M}$ is a bounded memory that stores a small number of representative old samples to retain the performance on old classes (Han et al., 2020; Cui et al., 2021). $\mathcal{M}$ is denoted as $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_k)$, where $\mathcal{M}_k$ is the set of preserved samples of the k-th old class.

To select and store representative samples from diverse utterances, we propose structure-based retrospection. In each step, this approach performs two operations: (1) structure-based sample selection chooses informative and diverse samples based on the spatial structure of the feature vectors; (2) structure-preserved sample removal maintains a constant memory size by deleting some of the stored old samples while not affecting their original distribution as much as possible. In this way, important information about the data distribution of the previous classes enters the subsequent training process.

3.2.1 Structure-Based Sample Selection

After learning the new data, we select $|\mathcal{M}|/l$ samples for each new class, where $|\mathcal{M}|$ is the memory

size and l is the number of all observed classes. Specifically, for each new class, we transform all its training samples into feature vectors via the trained model. Then, we apply the K-means algorithm to these feature vectors and the number of clusters is $|\mathcal{M}|/l$. In each cluster, we select the sample closest to the centroid and store it in the memory. This operation tends to select diverse and informative samples. As shown in Figure 2, these selected samples are located at the center of different regions of the feature space. In this way, the distribution of the stored data is consistent with the distribution of the original data.

3.2.2 Structure-Preserved Sample Removal

Since the memory size is constant, we need to delete some of the stored old samples to allocate space for the representative samples of the new classes. Specifically, we need to delete $|\mathcal{M}|/k - |\mathcal{M}|/l$ training samples for each old class, where k is the number of old classes and l is the total number of observed classes.

In our method, we remove samples that are far from the center of the entire feature space because these samples usually have less impact on the overall data distribution (Snell et al., 2017; Yang et al., 2018). For the c-th new intent class, we first average the feature vectors of all its samples to serve as the center of the feature space:

$$\boldsymbol{\eta}_c = \frac{1}{|\mathcal{N}_c|} \sum_{n=1}^{|\mathcal{N}_c|} f(x_{c,n}), \tag{2}$$

where $\mathcal{N}_c$ is the training samples of the c-th class and $f(x_{c,n})$ is the feature vector of the sample $x_{c,n}$. Then, for the selected representative samples of the new class, we sort them according to their distances from the central vector $\boldsymbol{\eta}_c$. In the subsequent lifelong learning steps, we remove the samples that are far from the central vector based on the sorted list. In this way, the distribution of the original data is preserved as much as possible.

SCN utilizes the spatial structure of features in both sample selection and sample removal, which shows remarkable improvements in our experiments. Previous lifelong learning methods tend to select similar samples or ignore the importance of structure-preserved sample removal (Rebuffi et al., 2017; Han et al., 2020).

3.3 Contrastive Knowledge Distillation

Although preserving a small amount of old data can alleviate catastrophic forgetting, it introduces another problem, class imbalance. Due to the memory capacity limitation, the preserved old data is relatively small, while the new data is usually large. The imbalanced data makes the model significantly biased towards learning new data, affecting the performance on old data. In contrast, the model in the last step is trained on old data. It performs well in the old classes and is less biased towards the new classes. Therefore, to mitigate the negative effects of class imbalance, we propose contrastive knowledge distillation to learn the knowledge of the last model.

Specifically, for each sample x, we represent the feature vectors extracted by the current model and the last model by $f(x)$ and $g(x)$, respectively. The contrastive knowledge distillation is calculated as:

$$\mathcal{L}_{\text{SIM}} = \sum_{n=1}^{|\mathcal{N}|} 1 - \langle f(x_n), g(x_n) \rangle, \tag{3}$$

$$\begin{aligned}\mathcal{L}_{\text{MGN}} = \sum_{n=1}^{|\mathcal{N}|} \sum_{t=1}^{|\mathcal{N}|} \mathbb{1}_{\delta(n)\neq\delta(t)} [\\ \max(\langle f(x_n), f(x_t) \rangle - \alpha, 0) + \\ \max(\langle f(x_n), g(x_t) \rangle - \alpha, 0)],\end{aligned} \tag{4}$$

$$\mathcal{L}_{\text{CKD}} = \gamma_1 \mathcal{L}_{\text{SIM}} + \gamma_2 \mathcal{L}_{\text{MGN}}, \tag{5}$$

where $\langle f(x_n), g(x_n) \rangle$ denotes the cosine similarity between the two feature vectors. $\mathbb{1}_{\delta(n)\neq\delta(t)}$ is an indicator function that is 1 if the label of the sample x_n is not equal to the label of the sample x_t, otherwise it is 0. α is a scalar that represents the margin of separation between features. γ_1 and γ_2 are two adjustment coefficients that are used to control the proportion of different losses.

As shown above, the contrastive knowledge distillation loss $\mathcal{L}_{\text{CKD}}$ contains two elements, i.e., $\mathcal{L}_{\text{SIM}}$ and $\mathcal{L}_{\text{MGN}}$. The similarity loss $\mathcal{L}_{\text{SIM}}$ encourages the features extracted by the current model to be close to the features extracted by the last model so that the feature distribution of the last model can be effectively retained. However, since the last model did not learn the new data, it has difficulty distinguishing new classes. Thus, just adopting the similarity loss may weaken the generalization between the new and old classes. Contrastive learning can improve the generalization of the model by increasing the distance between each positive sample and multiple negative samples (Ke et al., 2021; Gao et al., 2021). Inspired by contrastive learning, we employ the margin loss $\mathcal{L}_{\text{MGN}}$ to ensure that the separation between each feature and multiple negative features is greater than the margin α. For each feature, we adopt other features in the same batch

that have different labels from the current feature as negative features. Contrastive knowledge distillation ultimately preserves the feature distribution of the last model and improves the generalization between the new and old classes.

In addition, we adopt the vanilla knowledge distillation method (Hinton et al., 2015) as an auxiliary loss. It encourages the current model to retain the probability distribution of the last model as:

$$\mathcal{L}_{\text{PKD}} = -\sum_{n=1}^{|\mathcal{N}|}\sum_{t=1}^{|\tilde{\mathcal{C}}^{\text{o}}|}\tau_t(\boldsymbol{u})\log(\tau_t(\boldsymbol{v})), \quad (6)$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the logits predicted by the last model and the current model for the sample x_n. $\tilde{\mathcal{C}}^{\text{o}}$ is the set of old classes. $\tau_t(\boldsymbol{u}) = \mathrm{e}^{\boldsymbol{u}_t/T}/\sum_{s=1}^{|\tilde{\mathcal{C}}^{\text{o}}|}\mathrm{e}^{\boldsymbol{u}_s/T}$. T is a scalar that is used to increase the weight of small probability values.

3.4 Optimization

When new data arrives, SCN optimizes the intent detection model with the above losses:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \gamma_1\mathcal{L}_{\text{SIM}} + \gamma_2\mathcal{L}_{\text{MGN}} + \gamma_3\mathcal{L}_{\text{PKD}}, \quad (7)$$

where γ_1, γ_2, and γ_3 are the adjustment coefficients. These coefficients are used to balance the performance of the old classes and the new classes. These losses are calculated for both the new data and the stored old data. After training the model, structure-based retrospection updates the memory with new representative samples. In this way, our method can continually learn new classes while avoiding catastrophic forgetting. Besides, at the end of each step, we can further fine-tune the model using the balanced sample set in the memory, which can moderately improve performance.

4 Experiments

4.1 Benchmarks for the CLL-ID Task

We construct three CLL-ID benchmarks based on the following method: for each benchmark, we arrange the classes of one or multiple datasets in a fixed random order. Each class has its own training/validation/test data. In a class incremental manner, the lifelong learning methods continually train an intent detection model on new data. To the best of our ability, we collected 8 intent detection datasets to construct the 3 benchmarks:

The **CLINC** benchmark is constructed based on the CLINC150 dataset (Larson et al., 2019). We use all the 150 classes provided by the CLINC150 dataset. The data splitting of each class follows the official CLINC150 dataset. 15 new classes are learned at each step.

The **Banking-ML** benchmark is constructed on the basis of three datasets, including Banking (Casanueva et al., 2020), M-CID-EN (Arora et al., 2020a), and Liu57 (Liu et al., 2019). The Banking and M-CID-EN datasets provide 77 and 16 classes, respectively. The data splitting of these classes follows the official datasets. Since the classes in Liu57 suffer from a severe long-tail data distribution, we only use the top 57 frequent classes. Since Liu57 does not provide data splitting, we split the data of each class of Liu57 in a 3:1:1 ratio into the training/validation/test set. Finally, the Banking-ML benchmark contains 150 classes. 15 new classes are learned at each step.

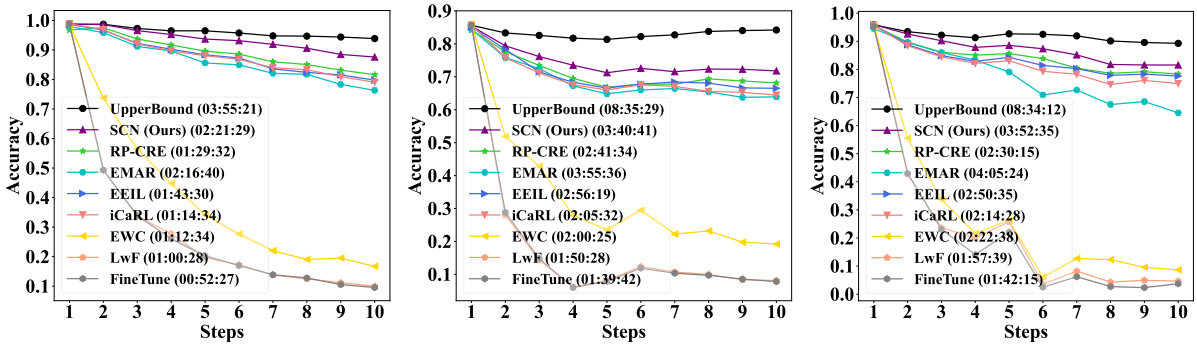
The **Stack-SHA** benchmark is constructed based on four datasets, including StackOverflow (Xu et al., 2015), SNIPS (Coucke et al., 2018), HINT3 (Arora et al., 2020b), and ATIS (Hemphill et al., 1990). We use all 20 and 7 classes provided by StackOverflow and SNIPS, as well as the official data split. We use the top 8 and 15 frequent classes of the ATIS and HINT3 datasets due to the long-tail data distribution. Similar to Liu57, the data of each class of ATIS and HINT3 is split into training, validation, and test sets in a 3:1:1 ratio. The total number of classes for the Stack-SHA benchmark is 50. At each step, 5 new classes are learned.

4.2 Implementation Details

Our BERT-based model is implemented with the HuggingFace's Transformer library[1]. The learning rate is 5e-5. The margin α is 0.3. The adjustment coefficients γ_1,γ_2, and γ_3 are 0.1, 0.9, and 0.005, respectively. The scalar T is 2. The batch size is 24. All hyper-parameters are obtained by a grid search on the validation set. The memory size is 500. For all experiments, we run each model with 5 different seeds on a single NVIDIA Tesla P40 GPU and report the average performance.

After each incremental step, we evaluate the model on the test data of all observed classes so far. Therefore, the test accuracy of the whole process can be plotted as a curve. After the last step, we report the average accuracy of all steps and the whole accuracy on the test data of all classes.

[1] https://github.com/huggingface



(a) CLINC Benchmark (b) Banking-ML Benchmark (c) Stack-SHA Benchmark

Figure 3: Test accuracy of the entire class lifelong learning process on the CLINC, Banking-ML, and Stack-SHA benchmarks. The training time of the entire process is shown in the brackets.

Method	CLINC		Banking-ML		Stack-SHA	
	Average Acc.	Whole Acc.	Average Acc.	Whole Acc.	Average Acc.	Whole Acc.
FineTune	29.20	9.54	19.15	7.81	21.51	3.73
UpperBound	96.10	93.86	83.17	84.22	91.87	89.26
LwF	29.38	9.97	19.19	8.14	23.34	4.66
EWC	41.25	16.70	34.63	19.25	28.17	8.66
iCaRL	87.97	79.08	69.63	64.45	81.70	74.97
EEIL	88.09	79.78	70.73	66.51	83.21	77.63
EMAR	86.37	76.32	69.04	63.93	77.66	64.51
RP-CRE	89.37	81.63	71.34	68.14	84.25	78.33
SCN (Ours)	**93.46**	**87.61**	**74.63**	**71.79**	**87.25**	**81.56**

Table 1: The average accuracy (%) of all steps ("Average Acc." column) and the whole accuracy (%) on the whole test data ("Whole Acc." column) after the last step.

4.3 Baselines

To provide a comprehensive comparison, we compare SCN with a variety of previous lifelong learning methods.

LwF (Li and Hoiem, 2016) utilizes knowledge distillation to preserve the probability distribution of the last model. **EWC** (Kirkpatrick et al., 2017) retains parameters that are important to old classes through L_2 regularization. **iCaRL** (Rebuffi et al., 2017) selects representative samples based on class prototypes and trains the model with knowledge distillation. **EEIL** (Castro et al., 2018) fine-tunes the model on the balanced data in the memory to cope with class imbalance. **EMAR** (Han et al., 2020) uses K-Means to select samples and consolidates the model by class prototypes. **RP-CRE** (Cui et al., 2021) utilizes class prototypes as external features and selects samples by K-Means. **FineTune** fine-tunes the model pre-trained on old data directly on new data. **UpperBound** uses training data of all observed classes to train the model, which is regarded as the upper bound.

4.4 Main Results

Figure 3 shows the test accuracy during the entire lifelong learning process. We present the average and whole accuracy after the last step in Table 1. From the results, we can see that:

(1) The proposed method SCN achieves state-of-the-art performance on all benchmarks. Compared to RP-CRE, SCN achieves 5.98%, 3.65%, and 3.23% improvements in terms of the whole accuracy on the CLINC, Banking-ML, and Stack-SHA benchmarks, respectively. It verifies the effectiveness of our method on the CLL-ID task.

(2) At each step of the entire process, there is a significant performance gap between RP-CRE and our method SCN. The reason is that RP-CRE ignores the problems of expression diversity and class imbalance in the CLL-ID task. Due to the lack of structure-preserved sample removal, RP-CRE may delete important samples and corrupt the data distribution. In addition, RP-CRE suffers from class imbalance, which eventually leads to performance degradation.

Method	CLINC		Banking-ML		Stack-SHA	
	Average Acc.	Whole Acc.	Average Acc.	Whole Acc.	Average Acc.	Whole Acc.
SCN (Ours)	**93.46**	**87.61**	**74.63**	**71.79**	**87.25**	**81.56**
- SBSS	92.14	84.97	74.06	70.41	85.41	79.91
- SPSR	93.15	87.11	74.15	71.13	86.38	79.24
- SBR	92.09	84.73	73.82	70.14	85.32	79.17
+ CPBR	90.29	83.21	71.63	67.47	83.79	78.18

Table 2: Ablation studies of structure-based retrospection. We describe these variants in detail below.

Method	CLINC		Banking-ML		Stack-SHA	
	Average Acc.	Whole Acc.	Average Acc.	Whole Acc.	Average Acc.	Whole Acc.
SCN (Ours)	**93.46**	**87.61**	**74.63**	**71.79**	**87.25**	**81.56**
- SIM	92.21	84.64	73.92	70.07	86.63	79.87
- MGN	92.75	85.86	74.10	70.76	86.98	80.38
- CKD	92.08	84.11	73.65	69.73	86.39	79.32
- PKD	93.18	86.45	74.32	71.40	87.11	80.59
- CKD and PKD	91.54	83.82	73.26	69.45	86.34	79.24

Table 3: Ablation studies of contrastive knowledge distillation. We describe these variants in detail below.

(3) FineTune always achieves the worst performance on all benchmarks. It proves that catastrophic forgetting is indeed a core challenge in the CLL-ID task. Besides, there is still a performance gap between SCN and the upper bound. It indicates that although SCN is very effective in the CLL-ID task, there is still room for further improvement.

4.5 Ablation Study

To verify the effectiveness of the structure-based retrospection and contrastive knowledge distillation, we conduct ablation studies.

4.5.1 Effect of Structure-Based Retrospection

To gain more insights into structure-based retrospection, we compare our method with different data preservation methods. The results are shown in Table 2. From the results, we can see that:

(1) For "- SBSS", we remove the structure-based sample selection and randomly add samples to the memory. For "- SPSR", the model randomly removes samples without using structure-preserved sample removal. For "- SBR", this variant employs a random strategy in both sample selection and sample removal. SCN significantly outperforms these variants on all benchmarks. The results indicate that structure-based retrospection is effective in selecting and storing the representative samples from diverse user utterances.

(2) For "+ CPBR" (Rebuffi et al., 2017; Castro et al., 2018), the model computes a prototype for each class and selects samples based on this prototype. In the CLL-ID task, "+ CPBR" is even worse than the random strategy "- SBR" because it usually selects similar samples. In contrast, our method utilizes the spatial structure of features to effectively select diverse and informative samples.

SCN	**1** How can I protect against the virus? **2** Do any medications protect against the virus? **3** Will wearing gloves help me avoid COVID-19? **4** Do kids need to wear face masks?
+ CPBR	**1** How should I protect myself? **2** How can I protect myself from coronavirus? **3** How can I stay safe from COVID-19? **4** Tell me some ways to avoid coronavirus.

Figure 4: Case study. We show some preserved samples.

(3) To give a visual comparison, we show some samples preserved by SCN and "+ CPBR" for the class "*how to protect yourself*" in Figure 4. "+ CPBR" tends to preserve similar samples, such as sample 1 and sample 2. In contrast, the samples preserved by our method tend to be diverse, covering a wide range of typical expressions. It qualitatively demonstrates the effectiveness of our method.

4.5.2 Effect of Contrastive Knowledge Distillation

To verify the effectiveness of the proposed contrastive knowledge distillation, we conduct ablation experiments and show the results in Table 3. From the results, we can see that:

(1) Removing any part of the contrastive knowl-

Number	SCN (Ours)		RP-CRE	
	Average Acc.	Whole Acc.	Average Acc.	Whole Acc.
500	**93.46**	**87.61**	**89.37**	**81.63**
450	93.25	86.55	88.16	80.17
400	92.53	84.61	87.49	78.34
350	92.03	84.14	87.13	78.03
300	91.71	83.38	86.63	77.55

Table 4: Comparison of the robustness of models to memory size on the CLINC benchmark.

edge distillation, i.e., the similarity loss ("- SIM") or the margin loss ("- MGN"), brings significant performance degradation. When we remove the contrastive knowledge distillation ("- CKD"), the performance degrades further. It demonstrates that contrastive knowledge distillation can effectively improve performance by preserving the knowledge of the original model. In addition, the results show that utilizing contrastive learning in our method to increase the generalization between the new and old classes can improve performance.

(2) When we remove the vanilla knowledge distillation ("- PKD"), the performance drops. When we remove both contrastive knowledge distillation and vanilla knowledge distillation, the performance decreases significantly. It indicates that simultaneously exploiting both methods is effective.

4.6 Discussion: Memory Size

In replay-based lifelong learning methods (Cao et al., 2020; Cui et al., 2021), the memory size is a key factor affecting performance. Therefore, we conduct experiments to verify whether our method can stably outperform the baselines under different memory sizes. As shown in Table 4, our method significantly outperforms RP-CRE in each case. Furthermore, as the memory size decreases, the performance improvement of our method usually becomes larger. Our method using only 300 samples surpasses RP-CRE using 500 samples. These results demonstrate the effectiveness of our method.

5 Related Work

5.1 Intent Detection

Recently, there are many research works on intent detection (Larson et al., 2019; Qin et al., 2019; Yan et al., 2020; Gerz et al., 2021). Zhang et al. (2019) utilize capsule networks to model the relations between intent detection and slot filling. Zhang et al. (2021b) propose a contrastive pre-training method to handle few-shot intent detection. Besides, unknown intent detection is a hot research task that aims to detect samples belonging to the unknown intent class (Brychcín and Král, 2017; Kim and Kim, 2018; Lin and Xu, 2019; Gangal et al., 2020). Cavalin et al. (2020) utilize the word graph information of classes to detect the unknown intent. Zhang et al. (2021a) propose an adaptive method to learn decision boundaries of the unknown intent.

Despite the great progress in intent detection tasks, these existing methods usually cannot flexibly and efficiently learn new intents, which limits their application in the real world. In this paper, we address the realistic and challenging task, i.e., class lifelong learning for intent detection.

5.2 Lifelong Learning

Lifelong learning is a key research topic in machine learning, which enables models to learn new data online (Cauwenberghs and Poggio, 2000; Kuzborskij et al., 2013; Wang et al., 2019; Cui et al., 2021). Existing lifelong learning methods can be roughly divided into three categories: architecture-based methods (Fernando et al., 2017; Shen et al., 2019), regularization-based methods (Zenke et al., 2017; Aljundi et al., 2018), and replay-based methods (Rebuffi et al., 2017; Hou et al., 2019). Architecture-based methods dynamically change the model architecture in response to new data (Geng et al., 2021; Madotto et al., 2021). Regularization-based methods slow down the update of the parameters that are important to old data (Kirkpatrick et al., 2017; Li and Hoiem, 2016). Replay-based methods alleviate catastrophic forgetting by preserving a small number of old samples (Han et al., 2020; Cui et al., 2021). In addition, generative replay-based methods generate old samples via generative models (Shin et al., 2017; Kemker and Kanan, 2018; Ostapenko et al., 2019). Replay-based methods have proven to be the most effective solutions for many lifelong learning tasks in NLP (Han et al., 2020; Cui et al., 2021).

In recent years, researchers have gradually begun to investigate lifelong learning in NLP scenarios (Kirkpatrick et al., 2017; Cao et al., 2020; Liu et al., 2021). Lee (2017) adopts a one-step incremental setting, which fine-tunes the model pre-trained on open-domain dialogues on task-oriented dialogues. Xia et al. (2021) study incremental few-shot learning in text classification tasks, which aims to continually learn new classes with only a small number of training samples. Madotto et al. (2021) study

domain lifelong learning in task-oriented dialogues. However, they mainly focus on the dialogue state tracking task. In addition, they adopt a generic architecture-based method, which does not address the main challenges of the intent detection task.

6 Conclusion

In this paper, we introduce class lifelong learning into intent detection and further propose structure consolidation networks to overcome catastrophic forgetting. To cope with expression diversity, we propose structure-based retrospection to select diverse and informative samples. To alleviate the negative effects of class imbalance, we propose contrastive knowledge distillation to preserve the knowledge of the original model. Experimental results on three benchmarks demonstrate the effectiveness of our method.

Limitations

Although our method SCN achieves state-of-the-art performance in the CLL-ID task, there is still a performance gap between SCN and the upper bound. This result is inconsistent with human behaviors because humans usually do not forget old skills when learning new skills. Therefore, in future work, we hope to introduce findings from the brain science domain into the model design to overcome the problem of catastrophic forgetting.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106400) and the National Natural Science Foundation of China (No.U1936207, No.61922085, No.61976211). This research work was supported by the Youth Innovation Promotion Association CAS, Yunnan Provincial Major Science and Technology Special Plan Projects (No.202202AD080004).

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *15th European Conference on Computer Vision, ECCV 2018*, volume 11207 of *Lecture Notes in Computer Science*, pages 144–161.

Abhinav Arora, Akshat Shrivastava, Mrinal Mohit, Lorena Sainz-Maza Lecanda, and Ahmed Aly. 2020a. Cross-lingual transfer learning for intent detection of covid-19 utterances.

Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020b. HINT3: Raising the bar for intent detection in the wild. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 100–105.

Tomáš Brychcín and Pavel Král. 2017. Unsupervised dialogue act induction using Gaussian mixtures. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 485–490, Valencia, Spain. Association for Computational Linguistics.

Pengfei Cao, Yubo Chen, Jun Zhao, and Taifeng Wang. 2020. Incremental event detection via knowledge consolidation networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 707–717.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. 2018. End-to-end incremental learning. In *15th European Conference on Computer Vision, ECCV 2018*, volume 11216 of *Lecture Notes in Computer Science*, pages 241–257.

Gert Cauwenberghs and Tomaso A. Poggio. 2000. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000*, pages 409–415.

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, Online. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi, and Yanghua Xiao. 2021. Refining sample embeddings with relation prototypes to enhance continual relation extraction. 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 232–243.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734.

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135.

Varun Gangal, Abhinav Arora, Arash Einolghozati, and Sonal Gupta. 2020. Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 7764–7771.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Binzong Geng, Fajie Yuan, Qiancheng Xu, Ying Shen, Ruifeng Xu, and Min Yang. 2021. Continual learning for task-oriented dialogue system with iterative network pruning, expanding and masking. 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 517–523, Online. Association for Computational Linguistics.

Daniela Gerz, Pei-Hao Su, Razvan Kusztos, Avishek Mondal, Michał Lis, Eshan Singhal, Nikola Mrkšić, Tsung-Hsien Wen, and Ivan Vulić. 2021. Multilingual and cross-lingual intent detection from spoken data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7468–7475, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6440.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2019. Learning a unified classifier incrementally via rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 831–839.

Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021. CLASSIC: Continual and contrastive learning of aspect sentiment classification tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6871–6883.

Ronald Kemker and Christopher Kanan. 2018. Fearnet: Brain-inspired model for incremental learning. In *6th International Conference on Learning Representations*.

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. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association*, pages 556–560.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. 2013. From N to N+1: multiclass transfer incremental learning. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3358–3365.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 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*, pages 1311–1316.

Sungjin Lee. 2017. Toward continual learning for conversational agents. *CoRR*, abs/1712.09943.

Zhizhong Li and Derek Hoiem. 2016. Learning without forgetting. In *14th European Conference on Computer Vision, ECCV 2016*, pages 614–629.

Zhuang Li, Lizhen Qu, and Gholamreza Haffari. 2021. Total recall: a customized continual learning method for neural semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3816–3831.

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.

Qingbin Liu, Pengfei Cao, Cao Liu, Jiansong Chen, Xunliang Cai, Fan Yang, Shizhu He, Kang Liu, and Jun Zhao. 2021. Domain-lifelong learning for dialogue state tracking via knowledge preservation networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2301–2311.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction - 10th International Workshop on Spoken Dialogue Systems, IWSDS 2019*, volume 714 of *Lecture Notes in Electrical Engineering*, pages 165–183.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282.

Oleksiy Ostapenko, Mihai Marian Puscas, Tassilo Klein, Patrick Jähnichen, and Moin Nabi. 2019. Learning to remember: A synaptic plasticity driven framework for continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 11321–11329.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. 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 2078–2087, Hong Kong, China. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8689–8696.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542. IEEE Computer Society.

Yilin Shen, Xiangyu Zeng, and Hongxia Jin. 2019. A progressive model to enable continual learning for semantic slot filling. 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 1279–1284, Hong Kong, China. Association for Computational Linguistics.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 2990–2999.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 4077–4087.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 796–806.

Congying Xia, Wenpeng Yin, Yihao Feng, and Philip Yu. 2021. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1360, Online. 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 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 Y.S. 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.

Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2018. Robust classification with convolutional prototype learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 3474–3482.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert Y.S. 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.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy. Association for Computational Linguistics.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021a. Deep open intent classification with adaptive decision boundary. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 14374–14382.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021b. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1906–1912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section Abstract, Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.