Prompt Conditioned VAE: Enhancing Generative Replay for Lifelong Learning in Task-Oriented Dialogue

Yingxiu Zhao[1]*, **Yinhe Zheng**[2]†, **Zhiliang Tian**[1], **Chang Gao**[3], **Jian Sun**[2], **Nevin L. Zhang**[1]

[1] The Hong Kong University of Science and Technology, Hong Kong

[2] Alibaba Group, China, [3] The Chinese University of Hong Kong, Hong Kong

{yzhaocx,ztianac,lzhang}@connect.ust.hk, zhengyinhe1@163.com,

gaochang@se.cuhk.edu.hk, jiansun_china@hotmail.com

Abstract

Lifelong learning (LL) is vital for advanced task-oriented dialogue (ToD) systems. To address the catastrophic forgetting issue of LL, generative replay methods are widely employed to consolidate past knowledge with generated pseudo samples. However, most existing generative replay methods use only a single task-specific token to control their models. This scheme is usually not strong enough to constrain the generative model due to insufficient information involved. In this paper, we propose a novel method, *prompt conditioned VAE for lifelong learning* (PCLL), to enhance generative replay by incorporating tasks' statistics. PCLL captures task-specific distributions with a conditional variational autoencoder, conditioned on natural language prompts to guide the pseudo-sample generation. Moreover, it leverages a distillation process to further consolidate past knowledge by alleviating the noise in pseudo samples. Experiments on natural language understanding tasks of ToD systems demonstrate that PCLL significantly outperforms competitive baselines in building lifelong learning models. We release the code and data at GitHub.

1 Introduction

Task-oriented dialogue (ToD) systems are of great importance in advanced AI applications (Zhang et al., 2020b; Dai et al., 2020, 2021; He et al., 2022a,b,c). However, most existing ToD systems are developed under the assumption that the data distribution remains unchanged (Zhu et al., 2022). Unless the entire system is retrained, this setup may not be realistic when the ToD system deployed in practice needs to support new features and provides more services over time based on user demands. Without incurring the high cost of retraining, Lifelong Learning (LL) is able to acquire new knowledge continuously while preserving previously learned knowledge (Delange et al., 2021). Hence, it's crucial to equip natural language understanding (NLU) modules, the vital components of ToD systems, with the lifelong learning ability.

The main issue for lifelong learning is *catastrophic forgetting* (McClelland et al., 1995; Parisi et al., 2019), which refers to the phenomenon that a model forgets previously learned tasks when learning new tasks. Various approaches have been proposed to alleviate this issue (Schwarz et al., 2018; Aljundi et al., 2018; Rusu et al., 2016; Aljundi et al., 2017). The replay-based methods are among the most effective and widely used ones (Rebuffi et al., 2017; Shin et al., 2017; Dai et al., 2022). The main idea of replay-based methods is to retrain samples or representations from already seen tasks when learning new tasks (Mundt et al., 2020). Some methods explicitly store previously seen real samples for replaying (*experience replay*) (Rebuffi et al., 2017; Chaudhry et al., 2019). However, this setting will be infeasible when data from previous tasks is unavailable due to data security concerns. Other methods try to generate pseudo samples using a generative model (*generative replay*). This variant relieves the burden of storing previously seen data and has been widely adopted in previous studies (Delange et al., 2021; Shin et al., 2017; Kemker and Kanan, 2018).

The key to generative replay is to produce pseudo samples to approximate the real data distribution of previous tasks. Intuitively, higher quality pseudo samples can better preserve learned tasks and lead to less forgetting in LL. However, the generation of pseudo samples for each seen task in previous studies (Sun et al., 2020; Chuang et al., 2020) is usually controlled by a single task-specific token. It has been observed that this scheme is usually insufficient to constrain the PLM (Sun et al., 2020), due to limited information involved. Consequently, the generated pseudo samples suffer from

* Work done while the author was interning at Alibaba.

† Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11153 - 11169

problems such as not being fluent or not corresponding well to the designated task. Moreover, those special tokens are only introduced in the fine-tuning stage of the PLM. This enlarges the gap between pre-training and fine-tuning of the PLM (Gu et al., 2022) and harms the quality of the generated pseudo samples. In addition, generated noisy pseudo samples may degenerate the LL performance.

To address the above issues, we propose a novel method, Prompt Conditioned VAE for Lifelong Learning (PCLL), to enhance generative replay on NLU tasks of ToD systems. To impose strong control over the pseudo-sample generation, PCLL explicitly models latent task-specific distributions using a conditional variational autoencoder (CVAE) (Kingma and Welling, 2014; Zhao et al., 2017). Then it incorporates the corresponding task statistics to guide the generation of pseudo samples. To reduce the gap between pretraining and finetuning, we construct natural language prompts to unify different NLU tasks while being specific to each task. These prompts not only contain meaningful semantics compared to special tokens, but also serve as conditions to assist CVAE in capturing task distributions. Moreover, PCLL employs a knowledge distillation scheme to alleviate the impact of noisy pseudo samples during the replay process. Leveraging the above strategies, PCLL can generate high-quality pseudo samples that better approximate the real distributions of previous tasks while tackling the aforementioned issues.

We validate our method on NLU tasks of ToD systems including both intent detection and slot filling. The results indicate that our approach generates high-quality pseudo samples and significantly outperforms competitive baselines. Our main contributions are as follows,

(1) We propose a novel method, PCLL, to enhance generative replay for building lifelong NLU modules of ToD systems.

(2) Conditioned on prompts, PCLL models latent task distributions with CVAE to guide the pseudo-sample generation and leverages knowledge distillation to further avoid forgetting.

(3) Our extensive experiments and comprehensive analyses demonstrate the superior performance of PCLL and the high quality of its generated samples.

2 Related Work

2.1 Lifelong Learning

There are generally three categories of LL methods:

Regularization-based Methods aim to strike a balance between protecting already learned tasks while granting sufficient flexibility for a new task (Mundt et al., 2020). Some methods (Schwarz et al., 2018; Aljundi et al., 2018; Zenke et al., 2017; Ebrahimi et al., 2019) impose constraints on the modification of important weights. Other methods introduce a distillation loss to constrain predicted features of the LL model. (Li and Hoiem, 2017; Dhar et al., 2019; Rannen et al., 2017). However, these additional regularization terms may downgrade the model performance (Parisi et al., 2019).

Architecture-based Methods dedicate model parameters for each task to prevent forgetting (Delange et al., 2021). Some studies (Fernando et al., 2017; Serrà et al., 2018; Hu et al., 2018) use static architectures and rely on task specific information to route through the architecture (Mundt et al., 2020), while other studies (Rusu et al., 2016; Aljundi et al., 2017; Zhai et al., 2020; Madotto et al., 2021; Ke et al., 2021; Geng et al., 2021; Zhao et al., 2022b) dynamically grow the architecture in the LL training process. However, these methods either require capacity allocation for tasks at the beginning or are not feasible when model expansion is prohibited with limited resources (Sun et al., 2020).

Replay-based Methods aim to preserve previous knowledge by replaying data from learned tasks. One line of studies (Rebuffi et al., 2017; Chaudhry et al., 2019; Lopez-Paz and Ranzato, 2017; Mi et al., 2020; Han et al., 2020; Liu et al., 2021b) keeps a small number of real samples from old tasks for replaying. However, these methods are unpractical when data from old tasks are unavailable. Another line of studies (Shin et al., 2017; Kemker and Kanan, 2018; Xiang et al., 2019) utilizes a generative model to reproduce pseudo samples or representations from old tasks.

In this paper, we focus on improving generative replay, as it does not require allocating extra parameters or model capacity and can be used with any LL model. Specifically, Sun et al. (2020) propose a general framework LAMOL for lifelong language learning to replay pseudo samples of previous tasks. Chuang et al. (2020) improve LAMOL by training an extra teacher model before learning each new task, however, this increases the burden of the LL

process. Kanwatchara et al. (2021) freeze critical parameters in LAMOL based on rationales, but those rationales are not always available for NLP tasks. All these previous works do not take task statistics into consideration, whereas our PCLL method incorporates the information of tasks' distributions to enhance generative replay.

2.2 Prompt-based Learning in NLP

Prompt-based learning has been found to be more effective than typical finetuning to use PLM (Schick and Schütze, 2021). With prompts, we can convert various downstream tasks to a unified language modeling task (Brown et al., 2020; Schick and Schütze, 2021). Prompts can be either manually designed (Petroni et al., 2019; Yu et al., 2019) or generated automatically (Shin et al., 2020; Jiang et al., 2020; Gao et al., 2021). Some recent studies employ prompt tuning on continual learning for dialogue state tracking (Zhu et al., 2022) and few-shot learning (Qin and Joty, 2022).

3 Methodology

3.1 Problem Definition

We aim to build an LL model to learn a stream of NLU tasks sequentially $\mathcal{T}^T = \{t\}_{t=1}^{T}$ in dialogue systems, where T can be infinite potentially. For each task t, a set of samples $\mathcal{D}_t = \{(x_k, y_k)\}_{k=1}^{N_t}$ are drawn from its underlying data distribution. Here, x_k denotes the input utterance, and y_k denotes the output label of NLU. In intent detection tasks, y_k is the intent label of x_k; in slot filling tasks, y_k is the slot-value pairs contained in x_k. Our objective is to learn a model that can perform well on all seen tasks and forget as little as possible.

3.2 Overview

We start with a brief overview of our proposed PCLL method for generative replay (See Fig. 1). PCLL consists of two components: an LM-based task solver to solve NLU tasks (Fig. 3) and a CVAE-based generator (Fig. 2) to generate pseudo samples with the help of task-specific latent distributions. For the first task, PCLL is initialized with PLMs along with other parameters randomly initialized. Before learning a new task t, we first use the PCLL model trained on previous tasks to generate pseudo samples for each of the learned tasks $\mathcal{T}^{t-1}$. Then we interleave these pseudo samples with the training data in $\mathcal{D}_t$ and continue to train PCLL. In this way, the model can learn the new task t while consolidating the knowledge of past tasks.

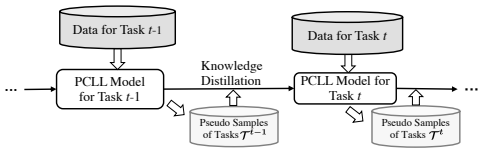

Figure 1: The training process of our model PCLL.

In the following sections, we first illustrate how PCLL learns the current task (Sec. 3.3, 3.4). Then we describe the pseudo-sample generation process (Sec. 3.5), and finally, we introduce a knowledge distillation process to further improve the LL performance (Sec. 3.6).

3.3 LM-based Task Solver

Following recent studies (Sun et al., 2020; Chuang et al., 2020), PCLL unifies different NLU tasks into a language modeling (LM) task and implements a task solver based on a PLM. Different from previous studies that introduce randomly initialized special tokens in the fine-tuning stage (Sun et al., 2020), we construct task-specific natural language prompts for the solver. These prompts carry rich semantic information to alleviate the mismatch between fine-tuning and pre-training of PLM.

For each input-output pair (x, y) from task t, our task solver is a LM that takes a prompt $g_t(x)$ as an input and predicts y. Specifically, $g_t(x)$ is constructed as $g_t(x) = g_t^{pre} \oplus x \oplus g_t^{post}$, where g_t^{pre} and g_t^{post} are prompt prefix and postfix designed for task t, respectively, and $\oplus$ means the concatenation of word tokens. For instance, if the task t is an intent detection task, we design $g_t(x)$ as: "`For an utterance from the ID task, x has the following intent` ", where "`ID`" represents the task name of t. After serializing the output y into a token sequence, we can obtain a natural language sentence by simply concatenating $g_t(x)$ with y. We list detailed examples in Appendix B.1. Then the PLM f_{θ_t} for the current task t is optimized on the concatenated sentence

$$g_t(x, y) = g_t^{pre} \oplus x \oplus g_t^{post} \oplus y, \quad (1)$$

by maximizing the following objective (see Fig. 3):

$$\mathcal{L}_{LM} = \log p_\theta(g_t(x, y)) + \lambda \log p_\theta(y|g_t(x)),$$

in which the first term learns to decode the constructed sentence given the start token [BOS], and

the second term learns to predict the output y after reading the prompt $g_t(x)$. λ is a scalar used to balance these two terms.

3.4 Prompt Conditioned VAE Generator

To construct high-quality pseudo-samples, PCLL leverages a CVAE module to build a pseudo-sample generator so that it can incorporate tasks' statistics to guide the generation of pseudo samples. The CVAE module captures task-specific latent distributions by taking utterances as the input, conditioned on prefix prompts, and reconstructing the input during training.

Specifically, given an input utterance x in task t, we assume a random variable z captures the latent distribution over x. We define a conditional distribution as $p(x, z|t) = p(x|z, t)p(z|t)$, where we approximate $p(z|t)$ and $p(x|z, t)$ using deep neural networks with parameters ϕ and θ, respectively. We refer to $p_\phi(z|t)$ as the *prior network* and $p_\theta(x|z, t)$ as the *decoder*. To reconstruct x, a latent variable z is first sampled from $p_\phi(z|t)$ and then x is decoded through $p_\theta(x|z, t)$.

In this study, we assume the prior of z to be a multivariate Gaussian distribution with a diagonal covariance matrix, and introduce a *recognition network* $q_\psi(z|x, t)$ to approximate the intractable true posterior $p(z|x, t)$. The goal of CVAE is to maximize the conditional log-likelihood $\log p(x|t) = \int p(x|z, t)p(z|t)dz$. Employing variational inference, we can get the following evidence lower bound (ELBO) (Zhao et al., 2017) to maximize:

$$\begin{aligned}\mathcal{L}_{\text{CVAE}} &= \underbrace{\mathbb{E}_{q_\psi(z|x,t)} \log p_\theta(x|z, t)}_{\mathcal{L}_{\text{REC}}} \\ &- \beta \underbrace{\text{KL}\left(q_\psi(z|x, t)\|p_\phi(z|t)\right)}_{\mathcal{L}_{\text{KL}}} \leq \log p(x|t),\end{aligned} \tag{2}$$

where β is a scalar to balance the reconstruction term $\mathcal{L}_{\text{REC}}$ and the Kullback–Leibler (KL) divergence term $\mathcal{L}_{\text{KL}}$ and is adjusted by a cyclic annealing schedule (Fu et al., 2019) to alleviate the vanishing latent variable issue (Bowman et al., 2016).

CVAE Implementation. When implementing each network in Eq.2, we use the prompt prefix g_t^{pre} to represent the task t because g_t^{pre} involves the task name that can exclusively identify t. Fig. 2 shows the overall architecture of our PCLL model, in which we use an unidirectional transformer (Vaswani et al., 2017) to encode the concatenated sentence $g_t^{pre} \oplus x$ into hidden representations. Then an attention-average block (Fang et al., 2021) is introduced to pool the hidden representations of g_t^{pre} and $g_t^{pre} \oplus x$ to single vectors, which are further fed into a prior network $p_\phi(z|t)$ and recognition network $q_\psi(z|x, t)$ respectively. Next, the reparametrization trick (Kingma and Welling, 2014) is used to obtain latent variables z from the prior and posterior distributions. Then z is injected to the decoder $p_\theta(x|z, t)$ by adding to each token embedding (word embedding and position embedding, elementwisely) of the prompt (Fang et al., 2021; Li et al., 2020).

In PCLL, the decoder $p_\theta(x|z, t)$ shares the same parameters with the PLM-based task solver f_θ. This allows us to inherit the advantage of PLM and leverage a unified model to solve each task and generate pseudo samples simultaneously.

3.5 Pseudo Sample Generation

Generating pseudo samples for learned tasks involves two steps: (1) PCLL generates a pseudo input utterance x guided by a latent task distribution using the CVAE-based generator. Specifically, for each seen task t', $(t' < t)$, the model samples a latent variable $z_{t'}$ from the prior network $p_\phi(z_{t'}|t')$ with the constructed prompt prefix $g_{t'}^{pre}$ as the input. Then the decoder takes $z_{t'}$ and $g_{t'}^{pre}$, and decodes them into the pseudo input x using top-k sampling[1] (Holtzman et al., 2019). (2) PCLL generates the output y associated with x using the solver (i.e., following Fig. 3).

3.6 Knowledge Distillation

Previous generative replay approaches indistinguishably interleave pseudo data with the current task's training data. However, this naive approach hurts the model performance since these pseudo data may contain noise and may drift from the real data distribution. In this study, we utilize a knowledge distillation (KD) (Hinton et al., 2015) process to prevent our model from being affected by these noisy pseudo data.

When training on a new task t, we treat the model obtained on previous tasks $\mathcal{T}^{t-1}$ as a fixed teacher model $f_{\theta_{\text{Tch}}}$. For each input-output pair (x, y) in the pseudo data, $f_{\theta_{\text{Tch}}}$ is distilled on the generated pseudo data to the current model f_θ (i.e., serves as the student model) by maximizing the

[1]Using other diversity enhanced decoding scheme may help produce more diverse pseudo samples (Wang et al., 2021). We leave it for future works.

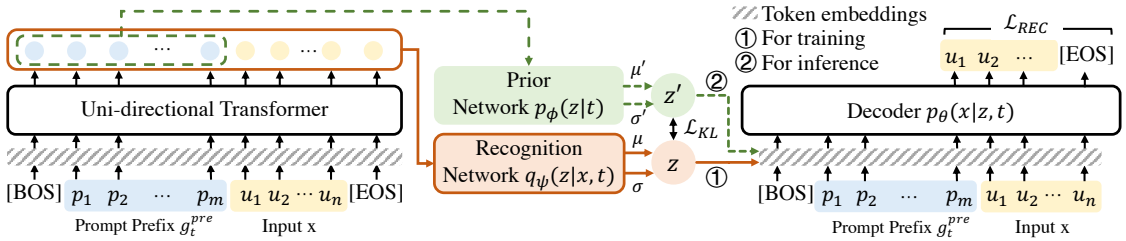

Figure 2: The architecture of the prompt conditioned VAE generator in PCLL. It captures the task distribution conditioned on prompts and incorporates the latent variable z (or z') into tokens' embeddings to guide the decoding.

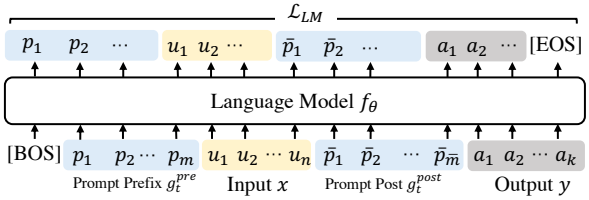

Figure 3: The LM-based solver for NLU tasks. The input-output pair (x, y) is converted into a natural language prompts with g_t^{pre} and g_t^{post}.

token-level distillation objective:

$$\mathcal{L}_{\text{LM}}^{\text{KD}} = \sum_{l=1}^{|g_t(x,y)|} \sum_{v\in\mathcal{V}} p_{\theta_{\text{Tch}}}(v|g_t(x,y)_{<l}) \log p_\theta(v|g_t(x,y)_{<l})$$
$$+\sum_{l=1}^{|y|} \sum_{v\in\mathcal{V}} p_{\theta_{\text{Tch}}}(v|g_t(x),y_{<l}) \log p_\theta(v|g_t(x),y_{<l}),$$

where $g_t(x, y) < l$ and $y < l$ refers to the token sequence before the l-th token in $g_t(x, y)$ and y, respectively. $\mathcal{V}$ represents the vocabulary set.

Similarly, when training the CVAE module, we replace the reconstruction term $\mathcal{L}_{REC}$ of in Eq. 2 with a distillation objective:

$$\mathcal{L}_{\text{REC}}^{\text{KD}} = \mathop{\mathbb{E}}_{q_\psi(z|x,t)} \sum_{l=1}^{|x|} \sum_{v\in\mathcal{V}} p_{\theta_{\text{Tch}}}(v|z,t,x_{<l}) \times \log p_\theta(v|z,t,x_{<l}),$$

and thus we maximize the following objective over the pseudo data $\mathcal{L}_{\text{CVAE}}^{\text{KD}} = \mathcal{L}_{\text{REC}}^{\text{KD}} - \beta\mathcal{L}_{\text{KL}}$.

Using the above KD strategy, the distributions produced by the teacher model contain richer knowledge compared to one-hot labels (Hinton et al., 2015). These distributions constrain the student model (i.e., f_θ) by preventing its weights from drifting too far when learning new tasks, thereby mitigating forgetting in lifelong learning.

Fig.1 illustrates the training process of PCLL. Specifically, when learning a new task t, we optimize PCLL on training samples of t with the following objective: $\mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{CVAE}}$. For pseudo samples of previous tasks t', $(t' < t)$, we optimize the loss

$$\mathcal{L} = \alpha(\mathcal{L}_{\text{LM}}^{\text{KD}} + \mathcal{L}_{\text{CVAE}}^{\text{KD}}) + (1-\alpha)(\mathcal{L}_{\text{LM}} + \mathcal{L}_{\text{CVAE}}),$$

where $\alpha \in [0, 1]$ is a scalar used to adjust knowledge distillation terms.

4 Experiments

4.1 Datasets

We evaluate the PCLL method on intent detection and slot filling based on public NLU benchmarks:

For intent detection, we collect six datasets that carry intent annotations: HWU (Liu et al., 2019), BANKING (Casanueva et al., 2020), CLINC (Larson et al., 2019), SNIPS (Coucke et al., 2018), AITS (Hemphill et al., 1990), and TOP (Gupta et al., 2018). The dataset TOP is divided into three disjoint subsets TOP-S1, TOP-S2, and TOP-S3, and these three subsets along with the other five datasets are regarded as separate LL tasks to increase the total number of tasks for sequential training. Finally, we have eight tasks to be learned sequentially for this intent detection experiment.

For slot filling, we adopt five datasets that provide slot labels: SNIPS, AITS, DSTC (Rastogi et al., 2020), MIT-MOVIE, and MIT-RESTAURANT [2]. Each dataset above is regarded as a separate LL task, and thus five tasks are learned in lifelong slot filling experiments. More descriptions about datasets are in Appendix A.

4.2 Implementation Details

We use the pretrained 12-layer GPT2 model (Radford et al., 2019) to initialize the encoder and decoder of our CVAE model. The prior network and the recognition network are both set to be a 2-layer MLP with hidden size of 128. When learning a new task t, PCLL balances the training data of t and pseudo samples by generating γN_t pseudo samples for previously learned tasks. γ is the sampling ratio

[2] groups.csail.mit.edu/sls/downloads

and γ is set to 0.2 in our experiment following Sun et al. (2020). Each task for intent detection and slot filling is trained for 5 and 10 epochs, respectively. We train PCLL on six random permutations of the task order. See Appendix B.2 and B.3 for more details.

4.3 Baselines

We compare PCLL with the following baselines: **Fine-tune** directly fine-tunes the model on the task stream without preventing catastrophic forgetting; **EWC** (Schwarz et al., 2018) and **MAS** (Aljundi et al., 2018) are two regularization methods that mitigate forgetting by penalizing changes of important parameters for learned tasks; **LAMOL-g** and **LAMOL-t** (Sun et al., 2020) are two variants of the generative replay method LAMOL that control the generation of pseudo samples either using a global special token (LAMOL-g) or task-specific special tokens (LAMOL-t); **L2KD** (Chuang et al., 2020) improves LAMOL by assigning an extra teacher for each new task to perform knowledge distillation; **ER** (Rolnick et al., 2019) preserves previously seen real samples for replay to prevent forgetting. We also consider some architecture-based baselines: **HAT** (Serrà et al., 2018) creates a task-based hard attention during training; **CTR** (Ke et al., 2021) inserts continual learning plug-ins into BERT to mitigate forgetting and encourage knowledge transfer; **Adapter** (Madotto et al., 2021) builds residual adapter for each task independently. Since works in Liu et al. (2021b) and Qin and Joty (2022) are specially designed for dialogue state tracking and few-shot learning, respectively, we do not consider them as our baselines.

Besides the above baselines, we further evaluate the model performance when all tasks are trained simultaneously in a multitask learning setting (**Multi**), which is often seen as an upper bound of LL. For fair comparisons, all baselines are implemented following either the settings of Sun et al. (2020), or their own reported settings. For ER, we store 1% of previously seen samples in memory following the setting of Madotto et al. (2021).

4.4 Evaluation Metrics

We use the accuracy score, and macro-averaged F1 score (Coope et al., 2020) to evaluate the performance of intent detection and slot filling tasks, respectively. Moreover, we consider access to a test set for each of the T tasks to learn in the LL process, and define $R_{i,j}$ as the test score of the task j after finishing learning the task i. We follow previous studies Lopez-Paz and Ranzato (2017); Chaudhry et al. (2018a) to use the following two metrics to evaluate the performance of LL: (1) **Average Score** (***Score***) is defined as the average test score of all T tasks after the LL process: $\text{Score} = \frac{1}{T}\sum_{j=1}^{T} R_{T,j}$. (2) **Learning Curve Area** (***LCA***) is the area under the Z_b curve, which captures the model's performance on all T tasks (Chaudhry et al., 2018b). Specifically, Z_b is the average score for all seen tasks at the training step b. Here, high *Score* and high *LCA* are preferred for a good LL model.

Methods	Intent Detection		Slot Filling	
	Score	LCA	Score	LCA
Finetune	14.09	28.76	15.38	19.55
EWC	14.16	28.34	15.67	19.51
MAS	14.15	28.61	15.59	19.37
L2KD	35.22	61.78	44.16	39.94
LAMOL-g	50.30	60.67	45.12	38.03
LAMOL-t	51.81	67.97	44.83	37.58
ER	78.19	71.36	44.95	39.32
HAT	73.92	73.03	61.99	67.33
CTR	67.44	71.11	63.84	67.28
Adapter	81.15	75.60	58.21	48.47
PCLL	**90.25**	**88.82**	**74.48**	**68.41**
Multi (Upper Bound)	96.25	N/A	80.80	N/A

Table 1: Experiment results on both intent detection and slot filling tasks. Each result is an average of six random task orders. The best results among LL models are bold. Our model PCLL is significantly better than other LL baselines with p-value < 0.05 under t-test.

4.5 Main Results

Table 1 shows the performances of our model PCLL and all the baselines. Our method PCLL significantly outperforms all baselines by a large margin on both intent detection and slot filling tasks. To better understand the LL process, we also plot the curve of the average score for all the models when trained using the same task order (see Fig. 4). From those results, we can observe that:
(1) Regularization-based methods (EWC and MAS) suffer from serious catastrophic forgetting, consistent with the observation of Madotto et al. (2021).
(2) Generative replay methods LAMOL-g, LAMOL-t, and L2KD alleviate the forgetting issue to some extent. However, replaying real samples (i.e., ER) performs much better. This indicates that the quality of samples used for replaying is critical to addressing catastrophic forgetting, which matches our motivation to improve generative

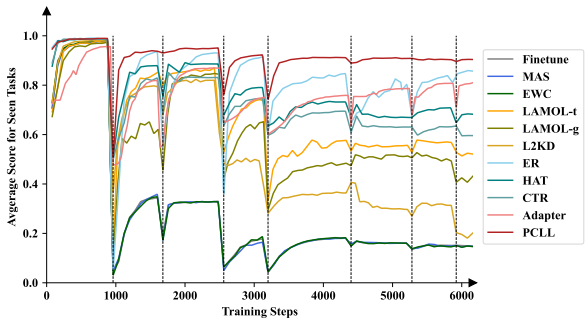

Figure 4: Learning curves of different methods on intent detection tasks. The dotted lines mean task switching.

replay by generating high-quality pseudo samples. Our method PCLL achieves higher performance than ER, indicating that PCLL can generate high-quality pseudo samples under the guidance of task distributions. Our analyses in Sec. 5.3 further prove this claim.

(3) Architecture-based methods HAT, CTR, and Adapter achieve good performance. However, PCLL still outperforms these baselines. This further validates the effectiveness of PCLL. Note that replay-based methods such as PCLL can be used together with these architecture-based methods to further improve the LL performance.

(4) From Fig 4, we can notice that when switching to new tasks, PCLL retains more knowledge about previous tasks (less performance degradation) compared to the baselines. This suggests that PCLL has a better ability to consolidate knowledge and mitigate catastrophic forgetting for LL.

4.6 Ablation Studies

We conduct ablation studies to verify the effectiveness of each proposed component in PCLL. (1) **w/o Latent** means no latent distribution is modeled for each task, i.e., the CVAE model in Section 3.4 is removed, and pseudo samples are generated by directly feeding the prompt prefix into the LM f_θ without incorporating task-specific statistics. (2) **w/o Task ID** means no task indicators are involved in the prompts. In other words, we design a task-independent prompt prefix by replacing the task ID with a general description "current task" (see Appendix B.1 for more details). In this way, the CVAE model degenerates to a VAE model that captures a global latent space for all tasks. (3) **w/o KD** means that the knowledge distillation process in Section 3.6 is not applied.

From Table 2, we can see that: (1) Capturing task-specific latent distributions and incorporating them in the pseudo-sample generation process is crucial for building better LL models (**w/o Latent**). (2) Using task-specific prompts helps to generate high-quality pseudo samples, thereby improving the LL performance (**w/o Task ID**). (3) The proposed knowledge distillation process does mitigate the effects of noisy pseudo-samples and is beneficial for consolidating previously learned knowledge to prevent forgetting (**w/o KD**).

5 Analyses

5.1 Soft Prompts vs. Manual Prompts

We conduct analyses on soft prompts by replacing manually designed prompts with soft tokens in PCLL. Specifically, the prompt prefix g_t^{pre} and postfix g_t^{post} in Eq. 1 are replaced by several randomly initialized task-specific soft (learnable) tokens (Liu et al., 2021a). We also vary the lengths of these soft prompts to analyze their behaviors.

Results in Table 3 show that: (1) Longer prefix prompts (i.e. more parameters guiding the pseudo-sample generation) generally lead to better LL performance; (2) Longer postfix prompts may not always lead to better LL performance. This may be because the postfix prompts are less important than prefix prompts since they do not participate in the pseudo-sample generation. Longer postfix prompts may bring in more noise, degenerating the performance; (3) Using manual prompts in PCLL outperforms all its soft-prompt variants even though some soft prompts are much longer than manual prompts. This justifies our claim that manual prompts carrying rich semantic information help to alleviate the mismatch between fine-tuning and pre-training of PLM and capture tasks' distributions, and thus mitigate catastrophic forgetting in lifelong learning.

5.2 Manual Prompts

Different Designs. We validate different designs of manual prompts in PCLL. Specifically, we implement five different prompt templates with dif-

	Intent Detection		Slot Filling	
	Score	LCA	Score	LCA
PCLL	**90.25**	**88.82**	**74.48**	**68.41**
w/o Latent	86.09	54.59	74.11	66.62
w/o Task ID	72.37	87.17	66.40	65.76
w/o KD	81.63	87.46	32.90	47.91

Table 2: Ablation studies on two NLU tasks. Each result is an average of 6 random task orders.

	Different Lengths		Score	LCA
PCLL-Soft	#postfix=1	#prefix=1	51.30	57.77
		#prefix=25	83.41	75.99
		#prefix=100	89.47	82.66
	#prefix=25	#postfix=1	83.41	75.99
		#postfix=25	74.04	75.29
		#postfix=50	79.76	79.79
PCLL	#prefix=15	#postfix=1	**90.25**	**88.82**

Table 3: Applying soft prompts on lifelong intent detection tasks. #prefix and #postfix indicate the lengths of prefix and postfix prompts, respectively. Each result is an average of 6 random task orders.

ferent lengths (Appendix B.4). We observe that different manual prompts yield almost the same performance. This indicates that our method is robust to the design of manual prompts. (See Table 8 in the Appendix).

Visualization of Attentions. We provide the visualization of the attention scores over several manual prompts employed by PCLL. High attention scores of task names in Fig. 6 indicate that the task indicators play an important role in our manually designed prompts (see Appendix B.5).

5.3 Qualities of Pseudo Samples

We validate the quality of pseudo samples generated by PCLL and all our generative replay baselines on intent detection tasks. We use the distinct score **Dist-n** (Li et al., 2016) to measure the proportion of unique n-grams in the generated pseudo samples' inputs (n=1,2,3,4). Higher Dist-n indicates more diverse generated pseudo samples, which is usually preferred because diverse samples help to approximate task distributions. As shown in Table 4, PCLL can generate more diverse pseudo samples compared to other generative replay methods. This demonstrates that pseudo samples constructed by our method are closer to real samples.

Further, we measure whether the generated pseudo samples can restore the distribution of real samples by visualizing samples' feature space with t-SNE (Van der Maaten and Hinton, 2008). As shown in Fig. 7, pseudo samples generated by PCLL are clustered in a similar pattern compared to real samples, while those of LAMOL-t are scattered in the feature space. It shows that the pseudo samples generated by PCLL share closer distribution with the real samples compared to our baselines (see Appendix B.6 for more details).

	Dist-1	Dist-2	Dist-3	Dist-4
LAMOL-g	0.0602	0.2466	0.4489	0.6178
LAMOL-t	0.1758	0.4733	0.6837	0.8090
PCLL	0.2836	0.6566	0.8369	0.9221
Real Sample	0.4000	0.7972	0.9255	0.9717

Table 4: Distinct scores for generated pseudo samples.

z dimension	Score	LCA
32	90.00	88.27
128	90.25	88.82
256	90.10	88.30
512	90.04	88.26

Table 5: Analysis of different dimensions of the latent variable z of PCLL on lifelong intent detection tasks. Each result is an average of six random task orders.

5.4 Analyses of Latent Variables

To further analyze the behavior of the pseudo sample generator, we visualize the latent space captured by the recognition network on slot filling tasks. Specifically, for each sample in the test dataset, we extract a latent variable z from its posterior distribution and use the t-SNE algorithm (Van der Maaten and Hinton, 2008) to visualize these variables in 2D space. It can be seen from Figure 5 that the latent spaces of different tasks are well clustered and clearly separated. This indicates that the latent variable z is able to capture task-specific knowledge among learned tasks.

We also analyze the influence of dimensions for latent variable z. The results are listed in Table 5. We can notice that when we select the dimension of z as 128, it can reach the best performance. This phenomenon is reasonable, when the dimension of z is small, it may not catch enough information to model the task distribution; when the dimension is large, it may contain some noisy information, leading to poorer performance.

5.5 Influence of Sampling Ratio γ

We analyze the influence of the sampling ratio γ (ranging from 0.01 to 1.0) on the performance of PCLL. The results in Table 11 indicate that PCLL is more effective in improving the LL performance when considering a small number of pseudo samples (See more details in Appendix B.7).

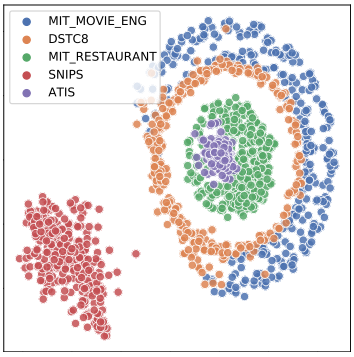

Figure 5: t-SNE visualization of latent variables.

5.6 Case Study

We present several pseudo samples generated by PCLL and LAMOL in Table 6 on the BANKING task for intent detection (see more cases in Appendix C). We can observe that: (1) Compared to LAMOL, pseudo samples produced by PCLL are closer to real samples from the BANKING dataset; (2) Some samples generated by LAMOL are inconsistent with the task: LAMOL generates samples for the weather domain, which is not related to the BANKING task; (3) LAMOL may also generate unmatched inputs and outputs in pseudo samples (last line in Table 6). These observations verify our claim that a single task-specific token is too weak to constrain the PLM, and our method PCLL helps to generate high-quality pseudo samples that are consistent with each task.

	Input x	Output y
Real	What exchange rate is it?	exchange rate
	My card never arrived.	card arrival
	I would like to reactivate my card.	card linking
PCLL	What is my exchange rate?	exchange rate
	My card hasn't come in yet.	card arrival
	How do I activate my card?	card linking
LAMOL	the weather forecast	GetWeather
	is it going to be on my card	card arrival
	I bought a used car	card linking

Table 6: Real samples and generated pseudo samples for the BANKING task.

5.7 Analyses of Forgetting for PCLL

We provide more fine-grained analyses for the forgetting issue based on findings when learning with our proposed method PCLL. In Appendix D, we carry out the analyses from the following four aspects: (1) unbalanced classes in some tasks, (2) conflicted label spaces for different tasks, (3) noisy pseudo labels for generated samples and (4) the diversity of pseudo samples created by PCLL.

6 Conclusion

In this paper, we propose PCLL to enhance generative replay for addressing catastrophic forgetting of lifelong learning in building NLU modules of ToD systems. To construct high-quality pseudo samples, PCLL captures task-specific distributions with a prompt conditioned VAE to guide the generation of pseudo samples. Empirical results on two NLU tasks and extensive analyses demonstrate the superior performance of PCLL and the high quality of its generated pseudo samples. Currently, we do not consider lifelong learning in the low-resource setting where only limited labeled data are available. In the future, we will extend our framework to lifelong few-shot learning.

Limitations

Here are some limitations of our work:

- We have not investigated lifelong learning in the low-resource setting where only limited labeled data are available. In future works, we will consider combining PCLL with meta-learning (Zhao et al., 2022a) to extend our framework to a lifelong few-shot learning setting. We will also extend previous studies by using unlabeled data (Zhang et al., 2020a; Zhao et al., 2022b) to build lifelong learning dialogue models.
- We have not considered architecture-based methods for lifelong learning. However, our method PCLL can be readily combined with the architecture-based approach by leveraging parameter-efficient modules (e.g., Adapter (Houlsby et al., 2019; Zhang et al., 2021), LoRA (Hu et al., 2021)) into the model architecture to further mitigate the catastrophic forgetting issue. We will explore this direction in the future.

Ethical Considerations

All our experiments are conducted on public available datasets. All metrics used in our paper are automatic and do not need manual labor. There are no direct ethical concerns in our study.

Acknowledgement

Research on this paper was supported by Alibaba Group through Alibaba Research Intern Program and Hong Kong Research Grants Council (Grant No. 16204920).

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv e-prints*, pages arXiv–2003.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018a. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2018b. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Yung-Sung Chuang, Shang-Yu Su, and Yun-Nung Chen. 2020. Lifelong language knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2914–2924.

Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-convert: Few-shot span extraction for dialog with pretrained conversational representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 107–121.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, Luo Si, and Yongbin Li. 2022. Lifelong learning for question answering with hierarchical prompts. *arXiv preprint arXiv:2208.14602*.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. 2021. Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking. 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 879–885.

Yinpei Dai, Hangyu Li, Chengguang Tang, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Learning low-resource end-to-end goal-oriented dialog for fast and reliable system deployment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 609–618.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. 2019. Learning without memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5138–5146.

Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. 2019. Uncertainty-guided continual learning with bayesian neural networks. In *International Conference on Learning Representations*.

Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformer-based conditional variational autoencoder for controllable story generation. *arXiv e-prints*, pages arXiv–2101.

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. *arXiv preprint arXiv:1701.08734*.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. 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 240–250.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. 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.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. Ppt: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *EMNLP*.

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.

Wanwei He, Yinpei Dai, Binyuan Hui, Min Yang, Zheng Cao, Jianbo Dong, Fei Huang, Luo Si, and Yongbin Li. 2022a. Space-2: Tree-structured semi-supervised contrastive pre-training for task-oriented dialog understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 553–569.

Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022b. Unified dialog model pre-training for task-oriented dialog understanding and generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 187–200.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022c. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10749–10757.

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 Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *stat*, 1050:9.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2018. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International Conference on Learning Representations*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*.

Kasidis Kanwatchara, Thanapapas Horsuwan, Piyawat Lertvittayakumjorn, Boonserm Kijsirikul, and Peerapon Vateekul. 2021. Rational lamol: A rationale-based lifelong learning framework. 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 2942–2953.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021. Achieving forgetting prevention and knowledge transfer in continual learning. In *Advances in Neural Information Processing Systems*.

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

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. *stat*, 1050:10.

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.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Qingbin Liu, Pengfei Cao, Cao Liu, Jiansong Chen, Xunliang Cai, Fan Yang, Shizhu He, Kang Liu, and Jun Zhao. 2021b. 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.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul A 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.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual learning for natural language generation in task-oriented dialog systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3461–3474.

Martin Mundt, Yong Won Hong, Iuliia Pliushch, and Visvanathan Ramesh. 2020. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *arXiv e-prints*, pages arXiv–2009.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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)*, Hong Kong, China. Association for Computational Linguistics.

Chengwei Qin and Shafiq Joty. 2022. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. *ICLR 2022*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1320–1328.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P Lillicrap, and Greg Wayne. 2019. Experience replay for continual learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 350–360.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537. PMLR.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *NIPS*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yida Wang, Yinhe Zheng, Yong Jiang, and Minlie Huang. 2021. Diversifying dialog generation via adaptive label smoothing. 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 3507–3520.

Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. 2019. Incremental learning using conditional adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6619–6628.

Bowen Yu, Zhenyu Zhang, Jianlin Su, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2019. Joint extraction of entities and relations based on a novel decomposition strategy. *CoRR*, abs/1909.04273.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR.

Mengyao Zhai, Lei Chen, Jiawei He, Megha Nawhal, Frederick Tung, and Greg Mori. 2020. Piggyback gan: Efficient lifelong learning for image conditioned generation. In *European Conference on Computer Vision*, pages 397–413. Springer.

Rongsheng Zhang, Yinhe Zheng, Xiaoxi Mao, and Minlie Huang. 2021. Unsupervised domain adaptation with adapter. *arXiv preprint arXiv:2111.00667*.

Rongsheng Zhang, Yinhe Zheng, Jianzhi Shao, Xiaoxi Mao, Yadong Xi, and Minlie Huang. 2020a. Dialogue distillation: Open-domain dialogue augmentation using unpaired data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3449–3460.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020b. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664.

Yingxiu Zhao, Zhiliang Tian, Huaxiu Yao, Yinhe Zheng, Dongkyu Lee, Yiping Song, Jian Sun, and Nevin Zhang. 2022a. Improving meta-learning for low-resource text classification and generation via memory imitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 583–595, Dublin, Ireland. Association for Computational Linguistics.

Yingxiu Zhao, Yinhe Zheng, Bowen Yu, Zhiliang Tian, Dongkyu Lee, Jian Sun, Yongbin Li, and Nevin L. Zhang. 2022b. Semi-supervised lifelong language learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Findings*.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual prompt tuning for dialog state tracking.

A Details of Datasets

We list the statistics of datasets for the intent detection and slot filling in Table 7 and give detailed descriptions as follows.

- **ATIS** consists of audio recordings and corresponding manual transcripts about humans asking for flight information on automated airline travel inquiry systems. The data consists of 17 unique intent categories.
- **BANKING** contains 13,083 utterances related to banking domain with 77 different fine-grained intents.
- **CLINC** contains 10 domains (e.g., travel, kitchen, utility, etc.) and 150 different intent classes.
- **DSTC** consists of slot annotations spanning 4 domains (buses, events, homes, rental cars).
- **HWU** includes 64 intents spanning 21 domains (e.g., alarm, music, IoT, news, calendar, etc.)
- **MIT_RESTAURANT** is a semantically tagged training and test corpus in BIO format.
- **MIT_MOVIE** is a semantically tagged training and test corpus in BIO format. We choose "eng" corpus for implementation which consists of simple queries.
- **TOP** is a dataset of 44K utterances where each utterance is annotated with a hierarchical semantic representation.
- **SNIPS** contains crowdsourced queries distributed among 7 user intents of various complexity.

Intent Detection Tasks				
Task	Train	Valid	Test	# Intent
ATIS	4.5K	0.5K	0.9K	17
BANKING	8.6K	1.5K	3.1K	77
SNIPS	11.0K	1.3K	1.3K	7
CLINC	15.0K	3.0K	4.5K	150
HWU	8.9K	1.1K	1.1K	64
TOP-S1	11.9K	1.7K	3.4K	6
TOP-S2	11.9K	1.7K	3.4K	6
TOP-S3	7.4K	1.0K	2.2K	7
Slot Filling Tasks				
Task	Train	Valid	Test	# Slot
ATIS	4.5K	0.5K	0.8K	79
SNIPS	11.0K	1.3K	1.3K	39
DSTC	3.7K	1.8K	1.8K	13
MIT-MOVIE	8.5K	1.2K	2.4K	12
MIT-RESTAURANT	6.1K	1.5K	1.5K	8

Table 7: Statistics of datasets for intent detection and slot filling.

B Experiment Details

B.1 Prompt Examples of NLU Tasks

We provide some detailed examples for inputs and outputs of the model with the designed prompts in PCLL. For intent detection, when we train on "BANKING" task, an input utterance x of the language model (LM) for a sample is modified as "`For an utterance from the BANKING task, "I already have one of your cards, how do I link them?" has the following intent `", the output of LM y is its corresponding intent annotation: "`Card linking`". For the ablation study of **w/o Task ID**, the prompt of the above sample becomes "`For an utterance from the current task, "I already have one of your cards, how do I link them?"`".

For slot filling, when we train on the "MIT-RESTAURANT" task, an input utterance `x` is "`Does the Casanova restaurant at Kendall Square offer a fixed price menu?`" of LM is modified as "`In the MIT-RESTAURANT task, if there are any slots and values, what are they in this sentence: "Does the Casanova restaurant at Kendall Square offer a fixed price menu?"? Answer: `", the output y locating the contained slot-value pairs is modified as "`Restaurant name: Casanova; Location: Kendall Square.`". Here, different slot-value pairs are formatted as "`slot: value`" separated with ";". If the input x does not contain any slot-value pairs, we use the sentence "`No slot in this sentence.`" as the output y.

B.2 Different Task Orders

We list the six random permutations of tasks that we use to implement all competing methods in Table 10.

B.3 Model Implementation Details

We use a pre-trained GPT2 model (Radford et al., 2019) as the initialization for the encoder and decoder of CVAE in PCLL. We set the maximum context length as 256. Our model contains a total number of 240M parameters. We train all competing methods on 1 Tesla-V100 GPU and it takes around 6 to 10 hours to train all the tasks. Moreover, the training and testing batch sizes are set to 64. The maximum learning rate is $5e-5$, the Adam optimizer is used with parameters $\beta_1 = 0.9$,

$\beta_2 = 0.98$ and $\epsilon = 1e-8$. The number of cycles for the cyclic annealing schedule is set to 4 in each epoch. When generating pseudo samples, the maximum decoded sequence length is set to 96. For baselines implementations, we use BERT to implement HAT and CTR, and choose GPT-2 as the backbone model for other baselines (LAMOL, L2KD, ER, Adapter, EWC, MAS, Finetune).

	Score	LCA
Prompt1 (12 tokens)	90.25	88.82
Prompt2 (13 tokens)	89.94	88.78
Prompt3 (4 tokens)	90.34	88.75
Prompt4 (11 tokens)	90.05	88.50
Prompt5 (28 tokens)	89.20	88.22

Table 8: Applying different manual prompts on lifelong intent detection tasks. Each result is an average of 6 random task orders.

B.4 Analysis of Manual Prompts Designs

We list five different manual templates as the designed prompts of intent detection in Table 9, where `Prompt1` is the one we use in Table 1. Let `ID` refers the task name, `x` refers the input utterance and `y` means the intent of `x`.

B.5 Analysis of Prompt Attention

We provide the visualization of the attention scores over several samples employed with our designed prompts for intent detection tasks. Specifically, the attention score on each prompt token is calculated using the averaged attention it receives when generating the output prediction. From the following Fig 6, we can notice that the task names do contain meaningful information to be attended to when generating predictions.

B.6 Analysis of Pseudo-sample Quality

We analyze the quality of generated pseudo samples with PCLL and other generative replay-based baselines. Specifically, we first fine-tune a pre-trained BERT (Devlin et al., 2019) model using these observed real samples to construct a task classifier. This classifier can determine the task identity of a given sample, and it reaches an accuracy of 98.67% on a hold-out test set. The fine-tuned BERT is used to extract the representation vector of each sample, and the t-SNE algorithm (Van der Maaten and Hinton, 2008) is used to map these vectors into 2-dimensions. For a specific task order [3] in LL, we gather pseudo samples generated when learning the last task and visualize the feature space of these samples. Note that the last task, ATIS, is not shown in Fig. 7 since there is no need to replay the last task.

Ratio γ	Score	LCA
0.01	73.61	84.35
0.05	84.09	89.54
0.20	90.25	88.82
0.50	91.02	91.44
1.00	91.31	91.77

Table 11: The LL performance on various sampling ratio γ. Each result is an average of 6 random task orders.

B.7 Analysis of Sampling Ratio

Table 11 shows the results on intent detection tasks. It can be seen that generating more pseudo samples helps to improve the LL performance. Besides, the performance gain slows down as the sampling ratio γ exceeds 0.2, i.e., generating 5 times more pseudo samples from $\gamma = 0.01$ to $\gamma = 0.05$ yields 10.48 absolute improvement on the *Score* metric, while increasing γ from 0.2 to 1.0 only yields 1.63 absolute improvement.

C Case Study

We present more generated pseudo samples from PCLL and LAMOL along with real samples in Table 12. For intent detection, we list real and pseudo samples from HWU tasks; for slot filling, we list those samples from MIT-RESTAURANT and DSTC tasks in Table 12.

D Analyses of Forgetting

We provide more fine-grained analyses for the forgetting issue.

- Classes with fewer samples are easier to be forgotten. Some tasks (e.g., ATIS, TOP, MIT-MOVIE) have unbalanced classes. These minor classes that only occupy a small portion of training samples are less likely to appear in pseudo samples used for replay. For example, the intent "meal" only takes 0.13% of the training samples for ATIS, and there are barely any pseudo samples generated for this intent when replaying.

[3] "TOP-S1, HWU, SNIPS, BANKDING, CLINC, TOP-S2, TOP-S3, ATIS"

	Different Manual Prompts for Intent Detection
Prompt1	For an utterance from the ID task, x has the following intent y
Prompt2	In the ID task, what intent best describes: x? Answer: y
Prompt3	Task ID utterance x intent y
Prompt4	In the task ID, this utterance x has the intent of y
Prompt5	If we consider the intent detection task, for a sample in the ID task, what's the intent of the utterance x? The intent is: y

Table 9: Different manual prompts are designed for intent detection module of a ToD system.

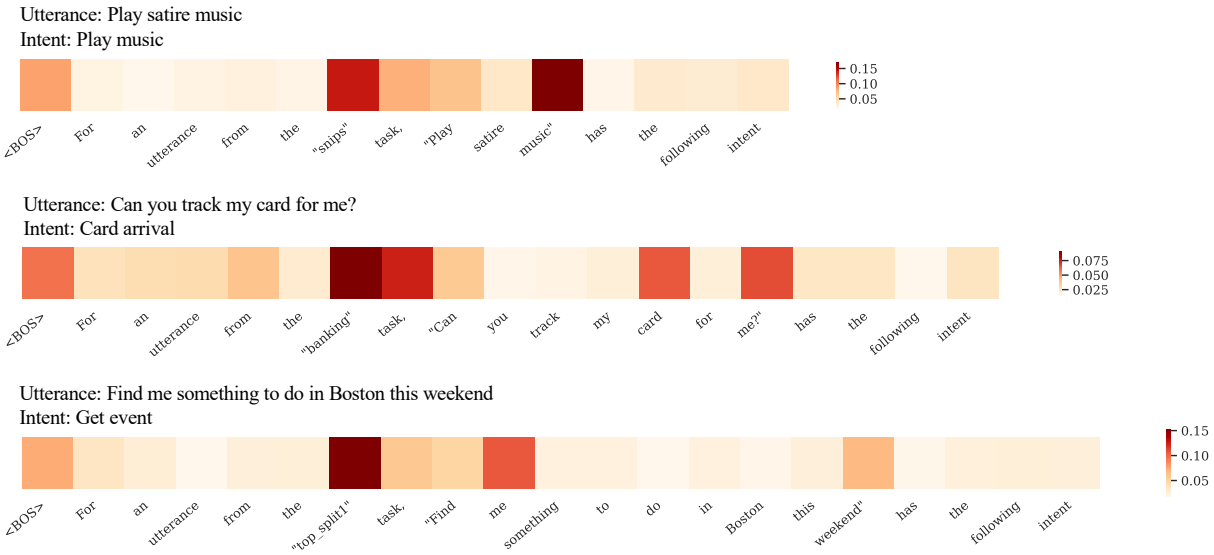

Figure 6: Visualization of attention scores for the natural language prompts of PCLL.

Without these pseudo samples, the model is more likely to forget these minor classes.

- Different tasks may have partially overlapping data distributions and conflicted label spaces, i.e., some tasks may assign different labels to the same set of utterances. For example, in the CLINC dataset, the utterance "transfer funds to the other account" is assigned with a label of "transfer"; however, in the BANKING dataset, the same utterance is assigned with a label of "transfer into account". These conflicted label spaces may confuse the model, resulting in incorrectly labeled pseudo samples.
- Noisy pseudo labels created by generative replay may lead to error accumulation, which will downgrade the performances of previously learned tasks.
- The diversity of generated pseudo samples for previous tasks tends to decrease as more replay times are performed, and these less diversified pseudo samples lead to more forgetting.

Specifically, we conduct analyses on lifelong intent detection tasks with the following task order (CLINC, SNIPS, TOP_S3, BANKING, TOP_S2, HWU, TOP_S1, ATIS). We compare the diversity of pseudo-samples for the first task (i.e., CLINC) generated at different replay moments: (1) after learning the first task, (2) after learning three subsequent tasks, and (3) after learning eight subsequent tasks (i.e., after the last task's learning). In Table 13, we use the distinct scores (Li et al., 2016) to measure the diversity of pseudo samples. We can notice that as we learn more tasks, the diversity of pseudo samples for the first learned task decreases. Therefore, replaying less diverse pseudo samples leads to performance degradation on previous tasks (i.e., forgetting of previous tasks).

After N Tasks	Dist-1	Dist-2	Dist-3	Dist-4
1	0.3593	0.7985	0.9439	0.9838
4	0.3193	0.7308	0.8951	0.9526
8	0.3091	0.6927	0.8593	0.9301

Table 13: Diversity scores of generated pseudo samples after learning N tasks.

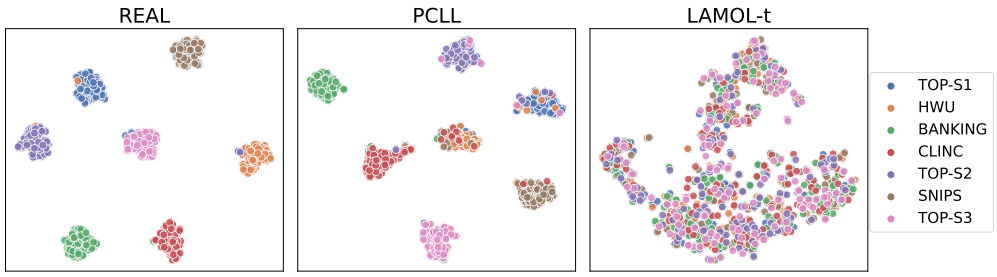

Figure 7: t-SNE visualization of the feature spaces associated with the generated pseudo samples.

	Intent Detection Tasks
Order 1	TOP_S1, HWU, SNIPS, BANKING, CLINC, TOP_S2, TOP_S3, ATIS
Order 2	BANKING, HWU, TOP_S1, TOP_S3, CLINC, TOP_S2, SNIPS, ATIS
Order 3	SNIPS, ATIS, TOP_S2, TOP_S3, CLINC, BANKING, HWU, TOP_S1
Order 4	CLINC, SNIPS, TOP_S3, BANKING, TOP_S2, HWU, TOP_S1, ATIS
Order 5	BANKING, TOP_S2, TOP_S1, ATIS, TOP_S3, HWU, CLINC, SNIPS
Order 6	CLINC, TOP_S1, TOP_S2, ATIS, SNIPS, HWU, BANKING, TOP_S3
	Slot Filling Tasks
Order 1	MIT_MOVIE, DSTC, MIT_RESTAURANT, SNIPS, ATIS
Order 2	MIT_MOVIE, SNIPS, DSTC, MIT_RESTAURANT, ATIS
Order 3	ATIS, MIT_MOVIE, DSTC, MIT_RESTAURANT, SNIPS
Order 4	DSTC, MIT_RESTAURANT, MIT_MOVIE, ATIS, SNIPS
Order 5	MIT_MOVIE, ATIS, SNIPS, MIT_RESTAURANT, DSTC
Order 6	SNIPS, ATIS, MIT_RESTAURANT, MIT_MOVIE, DSTC

Table 10: Six random permutations of tasks for intent detection and slot filling.

	Utterance	Labels
	Intent Detection (HWU)	
REAL	What is the most current exchange rate in china. Book a train ticket to Canada this evening. Set up each day an alarm at ten time for my meal.	qa_currency transport_ticket alarm set
PCLL	How many currencies can I exchange for a one pound dollar? Book a train ticket to make it home by five pm. Please set a new alarm for nine am on Friday.	qa_currency transport_ticket alarm set
LAMOL	How much money is going on for the election. I want to stay in the next week and get a decent. I am home from work tomorrow evening.	qa_currency GET_EVENT alarm set
	Slot Filling (MIT-RESTAURANT)	
REAL	Does the caranova restaurant at kendall square offer a fixed price menu. Could you find me a place that's open every day. I'm in the mood for chinese food.	Restaurant name: caranova; Location: kendall square Hours: open every day Cuisine: Chinese
PCLL	What pizza place that serves pizza in Manhattan has a great prices and is open at 2 am. What restaurant has good beer that is open before 9 am. Find me a vegetarian restaurant with outdoor seating.	Cuisine: pizza; Location: manhattan Hours: open before 9 am Cuisine: vegetarian; Amenity: outdoor seating
LAMOL	I would like to rent a room in this area. Does this place offer an open menu. The reason: It has been a while.	area: this area Hours: open menu No slot in this sentence.
	Slot Filling (DSTC)	
REAL	Would I would be able to get another car for the 10th of this month? I plan to go to Hayward. I need to pick it up from SD next Thursday.	dropoff_date: 10th of the month area: Hayward Pickup_city: SD; Pickup_data: next Thursday
PCLL	I need the car until 11th of March, can you find me the available one. The one in San Mateo is not fully furnished. Thanks for not leaving on 2nd of March and I would be in Long Beach, CA.	dropoff_date: 11th of March area: San Mateo pickup_city: Long Beach, CA
LAMOL	I am looking for an apartment in San Francisco, CA. I want to purchase the car, and it's parked on the street. I need to get to the city.	area: San Francisco, CA dropoff_date: on the street pickup_city: Seattle

Table 12: Real samples and generated pseudo samples by PCLL and LAMOL-t.