Prompt Augmented Generative Replay via Supervised Contrastive Learning for Lifelong Intent Detection

Vaibhav Varshney, Mayur Patidar, Rajat Kumar, Lovekesh Vig, Gautam Shroff
TCS Research, New Delhi, India
{varshney.v, patidar.mayur, k.rajat2,
lovekesh.vig, gautam.shroff}@tcs.com

Abstract

Identifying all possible user intents for a dialog system at design time is challenging even for skilled domain experts. For practical applications, novel intents may have to be inferred incrementally on the fly. This typically entails repeated retraining of the intent detector on both the existing and novel intents which can be expensive and would require storage of all past data corresponding to prior intents. In this paper, the objective is to continually train an intent detector on new intents while maintaining performance on prior intents without mandating access to prior intent data. Several data replay-based approaches have been introduced to avoid catastrophic forgetting during continual learning, including exemplar and generative replay. Current generative replay approaches struggle to generate representative samples because the generation is conditioned solely on the class/task label. Motivated by the recent work around prompt-based generation via pre-trained language models (PLMs), we employ generative replay using PLMs for incremental intent detection. Unlike exemplar replay, we only store the relevant contexts per intent in memory and use these stored contexts (with the class label) as prompts for generating intent-specific utterances. We use a common model for both generation and classification to promote optimal sharing of knowledge across both tasks. To further improve generation, we employ supervised contrastive fine-tuning of the PLM. Our proposed approach achieves state-of-the-art (SOTA) for lifelong intent detection on four public datasets and even outperforms exemplar replay-based approaches. The technique also achieves SOTA on a lifelong relation extraction task, suggesting that the approach is extendable to other continual learning tasks beyond intent detection.

1 Introduction

Modern dialog systems are reliant on user intent detection wherein the system is required to classify a user utterance into one of multiple pre-defined intents. A static intent detection model is often insufficient for real-world applications because (i) User intents evolve over time (ii) Additional functionality may be added to the system and (iii) Some intents may become obsolete.

Consider a conversational Information Technology (IT) helpdesk in an enterprise, where a ticket raised by an employee is classified into one of the multiple pre-defined intents (corresponding to different IT complaints) and accordingly routed by the system to the appropriate IT support staff. Each ticket comprises a textual IT problem description manually entered by an employee in natural language. In a typical IT enterprise, the helpdesk team provides support for every IT application from installation to maintenance. As applications get upgraded, and newer applications are introduced, the set of intents can change and the intent detection model must be updated.

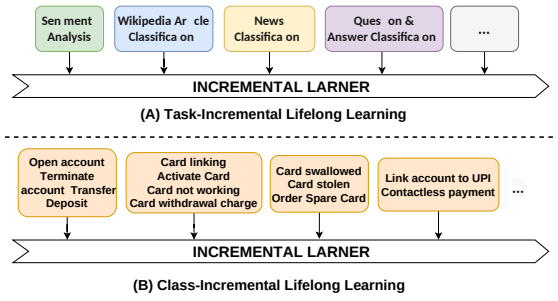

Figure 1: Task-Incremental (*Task-IL*) Vs Class-Incremental Lifelong Learning (*Class-IL*).

In the above example, the intent detection model is required to classify both old and newly introduced intents after updation. However, privacy concerns may limit access to prior intent data, and even if prior data is available, retraining the intent detection model from scratch becomes computationally expensive due to the ever-increasing data volume. An additional problem is that the data is skewed towards prior intents. Recently, there has

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1113 - 1127

been a lot of emphasis on continual learning to address these issues for computer vision (Parisi et al., 2018) and natural language processing (Biesialska et al., 2020), wherein the objective is to train a model on a sequence of novel tasks while maintaining performance on prior learned tasks, i.e., to prevent catastrophic forgetting (McCloskey and Cohen, 1989) with little or no data from old tasks.

Sun et al. (2020a); Wang et al. (2021); Madotto et al. (2021) formulate multiclass-text classification as a task-incremental learning (*Task-IL*) problem where the objective is to train an "incremental learner" on a sequence of tasks. As shown in Fig. 1 (A), they treat each classification task from a novel domain as a separate task introduced at each incremental step. They further assume that each task definition is complete, i.e., for each classification task they have access to all possible class labels and the set of class labels for that task remains immutable. However, in real-world applications, a task itself may evolve over time as shown in Fig. 1 (B), where, for an intent detection task in the banking domain a new set of intents are added at each incremental step based on new user needs and new functionality introduced by the service provider. Also in *Task-IL*, the task identity is provided at inference time, i.e., "Which task has to be performed for a given utterance?". This restricts the job of the "task incremental learner" to choosing one of the task-specific classes as a response. This is in contrast to lifelong intent detection, where the "incremental learner" has to consider the complete set of intents that have been added so far, independent of the number of incremental steps.

To overcome catastrophic forgetting in *Task-IL*, LAMOL (Sun et al., 2020a) uses a PLM as the task incremental learner and uses the same learner for generative replay, i.e., for generating samples of old tasks in *Task-IL*. Madotto et al. (2021) propose a residual adapter (Houlsby et al., 2019) based architecture for lifelong intent detection in task-oriented dialogue systems. Unlike existing Task-IL approaches, the authors also predict task-id, i.e., which adapter to use at test time by using an entropy-based classifier. Similar to LAMOL, they also model lifelong intent detection as Task-IL where dialogues from new domains are added to the system over a period of time. However, the task-specific adapters are trained in isolation which often results in erroneous task-id predictions due to an overlap between the intents introduced over different incremental steps. Liu et al. (2021b) propose a novel strategy based on exemplar-replay (MSR) which tries to perform knowledge distillation at prediction level and feature level and uses inter-class margin loss to minimize distance between new and old class embeddings.

In the current work, we model "lifelong intent detection" as a class-incremental learning (*Class-IL*) (Rebuffi et al., 2017) problem where a new set of intents/classes are added over a period of time, i.e., at each incremental step. To address the issue of catastrophic forgetting during "lifelong intent detection", we extend LAMOL to the *Class-IL* scenario and propose the "**P**rompt **A**ugmented **Ge**nerative **R**eplay" (**PAGeR**) model where, similar to existing approaches (de Masson d'Autume et al., 2019; Wang et al., 2020; Qian et al., 2021; Xia et al., 2021; Wang et al., 2019; Obamuyide and Vlachos, 2019; Han et al., 2020) we do not store real samples in replay memory but instead store concept words for each intent. In successive incremental steps, we generate samples for each prior intent based on the natural language intent definition and the stored intent specific concept words via PAGeR. To further improve the quality of generative replay and overall classification performance, we perform supervised contrastive fine-tuning of *PAGeR* and distill knowledge from the previous version of the "lifelong intent detector" while adapting it to the new set of intents.

Our key contributions are as follows:
(i) We propose a novel incremental learning approach using prompt based classification and generative replay via a common pre-trained language model.
(ii) Our generative replay approach uses prompt based generation guided by intent specific concept words and intent definition, yielding high quality samples.
(iii) To boost the quality of generative replay, we propose an approach for supervised contrastive fine-tuning of pre-trained Language Models (PLMs).
(iv) Our approach, *PAGeR*, outperforms exemplar replay based approaches for lifelong intent detection on three public intent classification datasets, one text classification dataset and two public multi-domain dialog intent detection datasets.
(v) We further demonstrate that *PAGeR* also outperforms the state-of-the-art on a lifelong relation extraction task by a wide margin suggesting generalizability of the proposed technique across tasks.

2 Related Work

2.1 Lifelong Learning

van de Ven and Tolias (2018) proposed three problem scenarios of increasing difficulty for evaluating continual learning algorithms based on whether the task identity is known or not and if not known whether it has to be predicted at inference time or not. The three scenarios correspond to (i) Task Incremental Learning (*Task-IL*) (ii) Domain-incremental learning (*Domain-IL*) and (iii) Class-incremental learning (*Class-IL*) respectively. Among these three scenarios, *Class-IL* is the most challenging whereas *Task-IL is* comparatively the easiest. Different approaches have been proposed to alleviate catastrophic forgetting for continual learning including *Rehearsal-based methods* where a subset of labelled training data from prior tasks is stored in a memory buffer and later replayed while training the model on the current task. Chaudhry et al. (2019b) proposed Experience Replay (*ER*) where the data from prior tasks is interleaved with data from the current task to retain performance on old tasks. *GEM* (Lopez-Paz and Ranzato, 2017) ensures that at every training step the loss of each previous task, approximated by the samples in episodic memory, does not increase. whereas *A-GEM* (Chaudhry et al., 2019a) attempt to ensure that at every training step the average episodic memory loss over previous tasks does not increase. *EWC* (Kirkpatrick et al., 2017) is an example of a *Regularization based approach* which remembers old tasks by selectively slowing down learning on the weights important for old tasks. *iCaRL* (Rebuffi et al., 2017) uses prototype rehearsal along with *Knowledge distillation* to alleviate catastrophic forgetting in a *Class-IL* scenario along with a herding (Welling, 2009) based approach for prioritized exemplar selection. *DER* (Buzzega et al., 2020) combines *rehearsal* with *knowledge distillation* and *regularization* to alleviate catastrophic forgetting in all three scenarios by matching the network's logits sampled throughout the optimization trajectory. In *architecture based approaches* the model is expanded for each new task with task-specific components (Rusu et al., 2016) or the same model is used for different tasks by identifying appropriate subnetworks (Wortsman et al., 2020). Shin et al. (2017) provide a *Generative Replay based approach* where pseudo-labelled data corresponding to the prior task is generated and interleaved with data from the current task to train a solver whereas van de Ven et al. (2020) use the same model as both a solver and generator and instead of replaying actual samples, replay latent representations.

2.2 Rehersal-based Methods

Monaikul et al. (2021); Qian et al. (2021); Cao et al. (2021) use different variants of knowledge distillation and representative sample selection methods to alleviate catastrophic forgetting for lifelong learning of NER (Monaikul et al., 2021), NMT (Qian et al., 2021) and hate speech classification on social media (Cao et al., 2021). For lifelong language learning on stream of text examples, de Masson d'Autume et al. (2019) use sparse experience replay, local adaptation during training, and inference respectively. Wang et al. (2020) extend the work of de Masson d'Autume et al. (2019) and propose a sparse experience replay augmented meta-learning based approach to learn better initialization for local adaptation. *EA-EMR* (Wang et al., 2019), *EMAR* (Han et al., 2020) are different methods for the lifelong relation extraction task described in section 6. Xia et al. (2021) propose a novel few-shot class incremental text classification task and model text classification as textual entailment.

2.3 Generative Replay based Methods

Sun et al. (2020a) propose an approach for *TASK-IL* where they generate pseudo-labelled samples corresponding to old tasks for replay and use the same model for generating a label given a sample and task identifier. Sun et al. (2020b); Chuang et al. (2020) use a common classifier-generator model LAMOL with distillation (Sun et al., 2020a) for *TASK-IL*.

2.4 Prompting PLMs

(Radford et al., 2019) demonstrate the possibility of solving multiple NLP tasks in the zero-shot setting by expressing each of them via a task specific natural language prompt where the corresponding output is generated in an autoregressive fashion. For a given task, there was still a significant performance gap between zero-shot and supervised learning. To reduce this gap, Schick and Schütze (2021) propose an approach where they express text classification and natural language inference in a cloze-style format and fine-tune the PLM in a few-shot setting. The same apprach has since been attempted for various NLP tasks (Liu et al., 2021a).

In the context of *Task-IL*, LAMOL expresses different NLP tasks in the format proposed by (McCann et al., 2018) and fine-tune GPT-2 on the entire task-specific data.

In *PAGeR*, we propose three prompts, two for expressing intent detection and one for labelled data generation for *Class-IL*. Pseudo-labelled utterance generation is guided by intent labels along with concepts words which are stored in memory instead of actual samples. We jointly fine-tune GPT-2 on an intent detection and labelled utterance generation task. To minimize the likelihood of incorrect (Utterance, label) pairs, unlike (Chen et al., 2020; Wu et al., 2020; Khosla et al., 2020; Gunel et al., 2021), we fine-tune GPT-2 on a supervised contrastive learning objective. To further alleviate catastrophic forgetting, we also use *Knowledge Distillation* (Hinton et al., 2015).

3 Preliminaries

Recent work has demonstrated how downstream tasks can be performed with PLMs using prompts (Liu et al., 2021a) in a zero shot setting. Typically, a prompting function $x' = f_{prompt}(x)$ is applied to an input x to obtain the corresponding prompt x' containing x, an (intermediate) answer z and task specific discrete or continuous tokens as task descriptors. For instance, for the sentiment analysis task, $\langle x' = [x]$ The movie is $[z]\rangle$ is an example of a prompt where $[x]$="I like this movie." and "The movie is" are task specific discrete tokens. Given a prompt, an answer z can be generated using a PLM model $p(z \mid [x]$ The movie is; $\varphi)$. Prompts with a random or true value of z are referred to as *Filled Prompts* and *Answered Prompts* respectively. Schick and Schütze (2021) propose prompt-based fine-tuning of PLMs to minimize the performance gap between supervised and zero-shot prompt approaches on downstream tasks.

Our approach is inspired by LAMOL (Sun et al., 2020a) which fine-tunes a GPT-2 PLM over a sequence of tasks. In LAMOL, each task is represented as a prompt generated via a prompt function $f_{prompt}(\textit{Context}) = \textit{Context}, \textit{Question}, ANS, \textit{Answer}, EOS))$, where *Context* corresponds to the input, *Question* acts as a task descriptor, *Answer* refers to the output and ANS, EOS refers to special tokens. LAMOL was initially used for training a single multi-task NLP model without any task-specific parameters (McCann et al., 2018). For the Sentiment Analysis task (*Context*: "I like this restaurant.", *Question*: "Is this sentence positive or negative?", *Answer*: "positive"), *Context* represents the input sentence/paragraph, *Question* includes all possible class labels and *Answer* represents the true label. Similar to generative pre-training (Radford and Narasimhan, 2018) of a transformer based decoder only model, LAMOL performs prompt based fine-tuning of GPT-2 by maximizing $P(\textit{Answer} \mid \textit{Context}, \textit{Question}, ANS)$ for answer generation and $P(\textit{Context}, \textit{Question}, ANS, \textit{Answer}, EOS \mid \textit{Task})$ for generative replay. At inference time, LAMOL generates a class label given a *Context* and task-identifier, i.e., $P(\textit{Answer} \mid \textit{Context}, \textit{Question}, ANS)$. In LAMOL, generative replay is conditioned solely on task-specific tokens. Similar to LAMOL, we use GPT-2 as PLM for both lifelong intent detection and generative replay.

4 Problem Description

In this paper, we model the "lifelong intent detection" problem as a class-incremental learning problem where a set of new intents are introduced at each incremental step. Consider $\mathcal{D}^i = \{\mathcal{I}^i_1, \mathcal{I}^i_2, ..., \mathcal{I}^i_m\}$ and $\mathcal{I}^i_j = \{\bigcup_{k=1}^{r}(x^i_{jk}, y^i_j)\}$ where $\mathcal{D}^i = \mathcal{D}^i_{train} \cup \mathcal{D}^i_{val} \cup \mathcal{D}^i_{test}$ represents the set of new intents and $\mathcal{I}^i_j$ represents labelled data corresponding to the j^{th} intent introduced at the i^{th} incremental step respectively. x^i_{jk} represents the k^{th} natural language utterance corresponding to the j^{th} intent introduced in the i^{th} incremental step and the corresponding intent label is represented by y^i_j. Unique intent labels introduced at the i^{th} incremental step are represented by $Y^i = \{y^i_1, y^i_2, ..., y^i_m\}$ and there will not be any overlap between the intents introduced at different incremental steps, i.e., $Y^i \cap Y^t = \emptyset$ where $t \in \{1, 2, ..., i-1\}$ and Y^i_{all} represent unique intent labels introduced so far where $Y^i_{all} = Y^i \cup Y^{i-1} \cup ... \cup Y^1$. At the i^{th} incremental step the whole labelled data corresponding to previous intents, i.e., $\mathcal{D}^i_{old} = \mathcal{D}^{i-1}_{train} \cup \mathcal{D}^{i-2}_{train} \cup ... \cup \mathcal{D}^1_{train}$ is not available for training but a fixed size memory $\mathcal{M} \ll |\mathcal{D}_{old}|$ is accessible by the incremental learner. We represent the incremental learner trained at the i^{th} incremental step by $\mathcal{IL}_i$. And the $\mathcal{IL}_i$ is evaluated on $\mathcal{D}^i_{test-full} = \mathcal{D}^i_{test} \cup \mathcal{D}^{i-1}_{test} \cup ..., \mathcal{D}^1_{test}$ which includes utterances corresponding to all the intents seen so far in addition to new intents.

$\mathcal{I}_{dist} = \{\mathcal{N}^1, \mathcal{N}^2, ..., \mathcal{N}^{\mathcal{T}}\}$ represents the num-

ber of intents introduced over each incremental step where $\mathcal{N}^i$ represents number of new intents introduced at i^{th} incremental step and $\mathcal{T}$ represents the total number of incremental steps for a dataset $\mathcal{D} = \mathcal{D}^1 \cup \mathcal{D}^2 \cup, ..., \cup \mathcal{D}^{\mathcal{T}}$. Thus at the $i+1^{th}$ incremental step the objective is to train a learner $\mathcal{IL}_{i+1}$ given $\mathcal{D}^{train}_{i+1}$, $\mathcal{M}$ and evaluate it on $\mathcal{D}^{i+1}_{test-full}$.

5 Proposed Approach

In *PAGeR*, we model "Lifelong Intent Detection" as a text generation problem, as shown in Fig. 2 (A), where the objective is to generate the correct label y^{i+1}_j for intents given a user utterance x^{i+1}_{jk}, while also utilizing generative replay to maintain performance on old intents. At the $i+1^{th}$ incremental step, we jointly fine-tune $\mathcal{IL}_{i+1}$ on the "Lifelong Intent Detection" *LID* and "Labelled Utterance Generation" (*LUG*) task given $\mathcal{D}^{i+1}_{train}$, $\mathcal{S}^i$ which consists of labelled ($\mathcal{D}^{i+1}_{train}$) and pseudo-labelled ($S_i$) utterances respectively. Pseudo-labelled utterances corresponding to intents present in $\mathcal{D}^i_{old}$ are generated from $\mathcal{IL}_i$ based on prompts stored in a fixed size memory, $\mathcal{M}$ as shown in Fig. 2 (B). To minimize the likelihood of incorrect (x^{i+1}_{jk},y^{i+1}_{j+1}) pairs, we do supervised contrastive fine-tuning of $\mathcal{IL}_{i+1}$, as shown in Fig. 2 (C). To alleviate catastrophic forgetting, we use knowledge distillation (Hinton et al., 2015) by fine-tuning $\mathcal{IL}_{i+1}$ on $\mathcal{S}^i$ with soft targets on *LID* and *LUG* whereas we use hard targets while fine-tuning $\mathcal{IL}_{i+1}$ on novel intents in $\mathcal{D}^{i+1}_{train}$.

For the remainder of this section, we drop the superscript $i+1$ for simplicity which indicates the $i+1^{th}$ incremental step. We use (x,y) for (x^{i+1}_{jk},y^{i+1}_j) and refer to the $i+1^{th}$ step as the current incremental step.

5.1 Lifelong Intent Detection (*LID*)

In *Task-IL* (van de Ven and Tolias, 2018), a task-identifier is provided at inference time, and the model performs the corresponding task. In *Class-IL*, no class/task information is provided apriori which makes the problem more challenging and not amenable to existing models such as LAMOL. We propose two ways of creating a prompt so that we can extend LAMOL to *Class-IL*.

(i) *Prompt without question* (*PWQ*): In *PWQ*, we convert each user utterance x in a prompt by applying a pre-defined template as mentioned in Eq 1. An *Answered Prompt* corresponding to *PWQ* is shown in Fig. 3 (A).

(ii) *Prompt with incremental question* (*PWIQ*): In *PWIQ*, we transform each user utterance x into a prompt by application of a pre-defined template as mentioned in Eq 2. where *IncQ* at the current incremental step includes all the intent labels Y^{i+1}_{all} seen so far. The *Answered Prompt* corresponding to *PWIQ* is shown in Fig. 3 (B).

$$f^{PWQ}_{prompt}(x) = (x, ANS, y, EOS) \quad (1)$$

$$f^{PWIQ}_{prompt}(x) = (x, \textit{IncQ}, ANS, y, EOS)) \quad (2)$$

ANS, EOS refers to special tokens used during prompt creation. We perform *PWQ* based fine-tuning of $\mathcal{IL}_i$ on $\mathcal{D}^{i+1}_{train} \cup \mathcal{S}^i$ to arrive at the $\mathcal{IL}_{i+1}$ by maximizing $log(p(y, EOS \mid x, ANS; \varphi))$ or by minimizing the loss function as mentioned in Eq. 3. $(w_1, ..., w_n)$, $(y_1, ..., y_N)$ are the set of tokens present in x and y respectively. n and N represent the number of tokens in the utterance and intent label respectively.

$$L_{ID} = -log(p(y, EOS \mid x, ANS; \varphi)) \quad (3)$$

5.2 Labelled Utterance Generation (*LUG*)

To generate high quality pseudo-labelled utterances for generative replay in subsequent incremental steps, we condition our intent specific generative model on an "intent label" and the corresponding intent *Concepts*. For each intent I_j, we obtain the top-K tf-idf words $tf - idf_j$ from the utterances corresponding to intent I_j to create a prompt. We use a maximum of $K = 50$ tf-idf words per intent. Words which are part of both x and $tf - idf_j$ are referred to as intent *Concepts* and $C_j = \{c_{j1}, c_{j2}, ..., c_{jr}\}$ represents a set of *Concept* sequences one per labelled utterance present in I_j. We use the template outlined in Eq. 4 to create a prompt for *LUG* and the corresponding *Answered Prompt* as shown in Fig. 3 (C). We train $\mathcal{IL}_{i+1}$ for prompt based generation by minimizing the loss function in Eq. 5.

$$f^{LUG}_{prompt}(y, c) = (y, c, SEP, x, ANS, y, EOS) \quad (4)$$

We use $\mathcal{IL}_i$ along with $\mathcal{M}$ to generate the pseudo-labelled data $\mathcal{S}^i$ corresponding to intents present in $\mathcal{D}^i_{old}$ for training of $\mathcal{IL}_{i+1}$. Existing replay based approaches either store a few labelled utterances per intent in $\mathcal{M}$ or generate an equal number of pseudo-labelled utterances. PAGeR, instead uses the same memory to store tf-idf concept

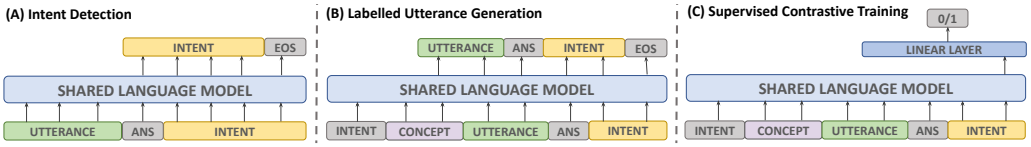

Figure 2: Proposed approach for Lifelong Intent Detection, where a PLM is jointly fine-tuned on (A) Intent detection, for generating the "intent label" corresponding to a user utterance (B) Generative replay, i.e., given a prompt consisting of the "intent" and a representative set of words from that intent, generate a user utterance along with the corresponding "intent label"(C) Supervised contrastive learning, i.e., given a valid utterance intent pair (U, I_1) and another randomly chosen intent I_2, predict whether I_1 and I_2 correspond to the same intent. For E.g., ("How do I link my new card ?","Card Linking") is an example of an (utterance,intent) pair and "Cash Withdrawal Charge" refers to a (possibly different) intent.

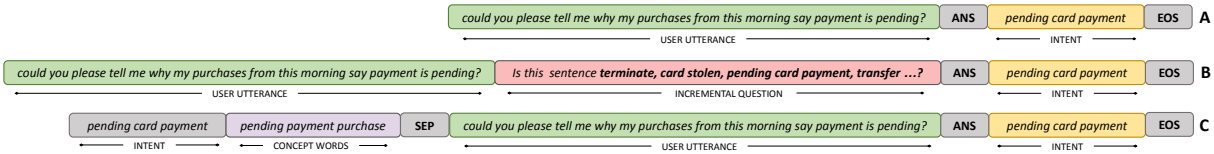

Figure 3: *Answered Prompts*: (A) Prompt without a question (*PWQ*), (B) Prompt with an incremental question (*PWIQ*), (C) Prompt for Labelled Utterance Generation (*LUG*)

words from C_j for every intent along with y_j and generates pseudo-labelled utterances corresponding to each entry in memory.

$$L_R = -(log(p(x, ANS, y, EOS \mid y, c_{jk}, SEP; \varphi))) \tag{5}$$

5.3 Supervised Contrastive Training (*SCT*)

To minimize the likelihood of incorrect utterance-label pairs during *LID* and *LUG*, we explicitly fine-tune $\mathcal{IL}_{i+1}$ on positive and negative utterance-label pairs via contrastive loss as shown in Eq. 6. A pair $((x, y), y')$ where $y' \in Y_{all}$ is positive if $y = y'$ or negative otherwise. $p_{ul} = sigmoid(W_c h_{EOS} + b_c)$, represents the probability that $((x, y), y')$ is positive and h_{EOS} is a representation of EOS token from the PLM. W_c and b_c represents the weight and bias of a linear layer respectively.

$$L_{SCT} = p_{ul} * log(p_{ul}) + (1 - p_{ul}) * log(1 - p_{ul}) \tag{6}$$

5.4 Knowledge Distillation (*KD*)

To alleviate catastrophic forgetting, we also use *KD* (Hinton et al., 2015) in the context of *LID* where instead of hard targets we use soft targets to train $\mathcal{IL}_{i+1}$ on $\mathcal{S}^i$. For intent detection, we distill the knowledge from $\mathcal{IL}_i$ while training $\mathcal{IL}_{i+1}$ by minimizing the loss function in Eq. 7 where $p'(y_{s+1})$ is obtained from $\mathcal{IL}_i$ and $y_0 = ANS$.

$$L_{KD} = -\frac{1}{N}\sum_{s=0}^{N} p'(y_{s+1}) * log(p(y_{s+1} \mid x, y_0, ..., y_s; \varphi)) \tag{7}$$

At the current incremental step, we incrementally update $\mathcal{IL}_i$ based on $\mathcal{D}_{train}^{i+1} \cup \mathcal{S}^i$ by minimizing L_{total} as shown in Eq. 8 to get $\mathcal{IL}_{i+1}$.

$$L_{total} = \lambda_1 * L_{ID} + \lambda_1 * L_{KD} + \lambda_2 * L_R + \lambda_3 * L_{SCT} \tag{8}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the set of hyperparameters. At inference, we use $\mathcal{IL}_{i+1}$ to generate intent labels for all utterances in $\mathcal{D}_{test-full}^{i+1}$.

We define *PAGeR-Base* as a baseline for *PAGeR*, which extends LAMOL for *Class-IL*. We train *PAGeR-Base* on $\lambda_1 * L_{ID} + \lambda_2 * L_R$ as our loss function where replay samples are generated based on the *PWQ* prompt.

6 Experimental Setup

In this section, we describe the various datasets, baseline approaches, evaluation metrics and training details. We subsequently present the results and discuss our key observations. For further training details, please refer to A.2 in appendix.

6.1 Dataset Description

Lifelong Intent Detection: **CLINC150** (Larson et al., 2019) is a crowdsourced multi-domain (10

domains such as utility, travel etc.) intent detection dataset. **HWU64** (Liu et al., 2019) is a crowdsourced multi-domain (21 domains such as alarm, cooking etc.) intent detection dataset. **BANKING77** (Casanueva et al., 2020) is an imbalanced, fine-grained intent detection dataset from the banking domain. **SGD** (Rastogi et al., 2020) and **MWOZ** (Budzianowski et al., 2018) are multi-domain dialogue datasets which consist of dialogues from 19 and 8 domains respectively.

Text Classification: **Stackoverflow** (*S20*) (Xu et al., 2017) is a multi-class classification dataset comprising of 20,000 question titles from Stackoverflow each tagged with one of 20 different tags. *Lifelong Relation Extraction*: **FewRel** (Han et al., 2018): is a relation extraction dataset where the objective is to predict a relation type between a pair of entities in a given sentence/paragraph.
For all datasets, we use standard train-test split provided by respective authors (shown in Table 5). For further dataset details, please refer to A.1 in appendix.

6.2 Baseline Approaches

We use *Upper Bound* (**UB**) where we assume that data corresponding to all incremental steps is available apriori and fine-tune $\mathcal{IL}$ on all $\mathcal{D}^i$'s together and in *Lower Bound* (**LB**), we incrementally fine-tune $\mathcal{IL}$ on $\mathcal{D}^i$'s with $|\mathcal{M}| = 0$.

6.2.1 Lifelong Intent Detection

(i) **EWC** (Kirkpatrick et al., 2017): Remembers old tasks by selectively slowing down learning on the weights important for these tasks.
(ii) **A-GEM** (Chaudhry et al., 2019a): A-GEM tries to ensure that at every training step the average episodic memory loss over the previous tasks does not increase.
(iii) **ER** (Rolnick et al., 2019): Interleaving old samples with current data in training batches.
(iv) **DER** (Buzzega et al., 2020): Rehearsal with knowledge distillation and regularization.
(v) **AdapterCL** (Madotto et al., 2021): Train a task-specific residual adapters (Houlsby et al., 2019) and uses an entropy-based classifier to select which adapter to use at test time.

6.2.2 Lifelong Relation Extraction

(i) **GEM** (Lopez-Paz and Ranzato, 2017): GEM ensures that at every training step the loss for each of the previous tasks, approximated by the samples in episodic memory, does not increase.
(ii) **EMR** (Parisi et al., 2019): Jointly train a model on current and old labelled samples stored in memory.
(iii) **EA-EMR** (Wang et al., 2019): They utilize an explicit alignment model to mitigate the sentence embedding distortion of the learned model when training on new data and new relations.
(iv) **EMAR** (Han et al., 2020): Every time neural models are activated to learn both new and memorized data, EMAR utilizes relation prototypes as a memory reconsolidation exercise to keep a stable understanding of old relations.

6.3 Evaluation Metrics

We use the following two metrics for evaluating performance on the lifelong intent detection task.
Average Accuracy (Chaudhry et al., 2018) We use $Avg^{k'}_{acc}$ to evaluate the performance after k' incremental steps where $Acc^{k'}_{avg} = \frac{1}{k'}\sum_{i=1}^{k'} A(\mathcal{IL}_{k'}(\mathcal{D}^i_{test}))$ and $A(\mathcal{IL}_{k'}(\mathcal{D}^i_{test}))$ represents classification accuracy obtained by $IL_{k'}$ after k' incremental steps on $\mathcal{D}^i_{test}$.
Average Forgetting (**F**) (Chaudhry et al., 2018) We use $F^{k'}_{avg}$ to measure the average drop in intent classification accuracy for intents introduced after k' incremental steps where $F^{k'}_{avg} = \frac{1}{k'}\sum_{i=1}^{k'-1} \max_{1\le t<k'} A(\mathcal{IL}_t(\mathcal{D}^i_{test})) - A(\mathcal{IL}_{k'}(\mathcal{D}^i_{test}))$ and $F^{k'}_{avg} \in [-1, 1]$ (lower value of $F^{k'}_{avg}$ implies less forgetting).

7 Results and Discussion

For lifelong intent detection task, we compare *PAGeR* with SOTA approaches for *Class-IL* from the NLP and vision community on three imbalanced and one balanced intent detection dataset. The comparison was across three different sizes for $\mathcal{M}$. For a given $\mathcal{M}$, we run all baselines, including *PAGeR-Base* and *PAGeR* over four different intent permutations and report $Acc^{\mathcal{T}}_{avg}$ and $F^{\mathcal{T}}_{avg}$. As shown in Table 1, *PAGeR* outperforms the baselines for all possible dataset-$\mathcal{M}$ combinations. In low memory scenarios, i.e., $\mathcal{M} = 1\%$, *PAGeR* outperforms exemplar replay based baselines by high margins which indicates that while it's not possible to store all representative samples in a given memory, these may be generated based on concept words. Similar to (Buzzega et al., 2020), we also find *EWC* and *A-GEM* less effective in the *Class-IL* setting. With $\mathcal{M} = 1\%$, *PAGeR* also outperforms *AdapterCL* on all datasets except $C150^I$. We also compare *PAGeR* with MSR (Liu et al., 2021b) and

$\mathcal{M}$	Approach	Intent Detection				TC
		C150^B	**C150^I**	**B77**	**HWU64**	**S20**
	UB	*95.7*	*94.6*	*91.9*	*90.3*	*90.8*
	LB	36.9/0.66	30.1/0.75	23.3/0.88	24.4/0.85	21.6/0.94
	EWC	36.9/0.66	31.6/0.74	21.1/0.89	23.8/0.86	20.8/0.94
	AdapterCL	88.1/0.02	79.9/0.07	77.7/0.07	77.6/0.08	82.5/0.06
10%	A-GEM	30.8/0.71	13.6/0.92	17.3/0.93	18.8/0.87	67.6/0.3
	ER	88.9/0.06	81.7/0.15	84.7/0.13	78.3/0.17	85.2/0.11
	DER	90.8/0.03	83.9/0.1	80.7/0.15	81.3/0.12	85.6/0.11
	PAGeR-Base (LAMOL)	92.7/0.04	89.7/0.07	82.9/0.13	84.0/0.123	72.5/0.28
	PAGeR	**94.9/0.01**	**93.4/0.02**	**90.2/0.02**	**89.9/0.04**	**88.7/0.05**
5%	A-GEM	30.8/0.7	13.4/0.92	19.2/0.91	18.4/0.89	68.2/0.30
	ER	85.2/0.1	71.1/0.28	76.5/0.21	72.0/0.25	82.6/0.15
	DER	88.1/0.07	78.1/0.18	73.9/0.19	76.5/0.19	83.9/0.13
	PAGeR-Base (LAMOL)	92.0/0.04	87.5/0.09	81.1/0.16	82.2/0.15	71.3/0.30
	PAGeR	**94.4/0.02**	**92.9/0.03**	**89.2/0.04**	**88.0/0.06**	**88.0/0.07**
1%	A-GEM	28.7/0.73	13.9/0.91	25.3/0.83	17.9/0.89	62.4/0.4
	ER	64.8/0.34	41.3/0.62	51.6/0.53	46.2/0.56	76.3/0.22
	DER	73.0/0.24	45.9/0.56	53.0/0.5	49.4/0.51	77.4/0.22
	PAGeR-Base (LAMOL)	86.9/0.10	63.6/0.37	68.1/0.32	70.5/0.30	68.3/0.34
	PAGeR	**92.4/0.04**	**76.4/0.22**	**80.1/0.17**	**79.2/0.19**	**84.3/0.13**

Table 1: *Average Accuracy (↑) / Average Forgetting (↓)* on four intent detection dataset, **C150^B**, **C150^I** : Balanced and Imbalanced version of **CLINC150**, **BANKING77 (B77)**, **HWU64** and on one Text Classification (**TC**) dataset.

show the results in table 8.

PAGeR also outperforms other baselines for incremental text classification on the *S20* dataset. Even with $\mathcal{M} = 1\%$, *PAGeR* gets an average accuracy, $Acc^{\mathcal{T}}_{avg}$ of 84.3% which is only 6.5% below the *UB*. For *S20*, we use numeric label ids as class labels to demonstrate that *PAGeR* also works when labels are not informative.

7.1 *PWQ* Vs *PWIQ*

As shown in Fig. 4, the length of *PWIQ* prompt depends on the number of intent labels which increases at every incremental step and leads to large sequence length whereas the length of *PWQ* prompt is independent of the number of intent labels. And large sequence length leads to more running time. Hence, *PWQ* prompt can be extended to operate over a large number of incremental steps.

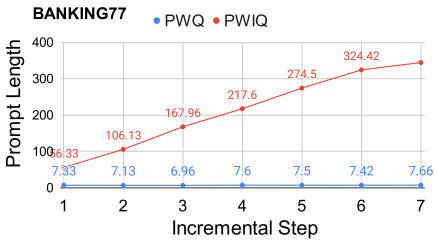

Figure 4: PWQ Vs PWIQ

7.2 How good are pseudo-labelled utterances?

To alleviate catastrophic forgetting in the *LID* and *LUG* tasks, we use pseudo-labelled utterances for generative replay. To empirically assess the quality of pseudo-labelled utterances over incremental steps, we perform the following experiment: We only consider intents that have been introduced at the first incremental step. On these intents, we train an intent detection model at each incremental step where we use pseudo labelled utterances generated by $\mathcal{IL}_i$ at i^{th} incremental step and compare it with the model trained on real labelled-utterances (Upper Bound). As shown in Fig. 5, on the B77 dataset, the drop in average accuracy is not significant for these intents over incremental steps. This suggests that pseudo-labelled utterances generated by the model are close to original utterances, i.e., generative replay plays an important role towards alleviating catastrophic forgetting in both *LID* and *LUG* tasks.

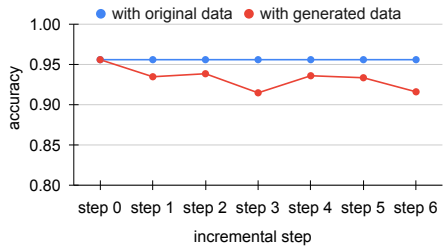

Figure 5: Accuracy for intents introduced during first incremental step at subsequent incremental steps

8 Ablation Study

We conduct ablation experiments on the B77 dataset across all memory sizes to demonstrate the efficacy of *PAGeR*'s sub-components as shown in Table 2. Individually, all three components when applied with *PAGeR-Base*, show significant improvement in performance ($Acc^{\mathcal{T}}_{avg}$). **LUG**, when applied with *PAGeR-Base* to generate better representative samples for replay, boosted the $Acc^{\mathcal{T}}_{avg}$ of *PAGeR-Base* by 4.7%, 4.9% and 5.7% in experiments where $\mathcal{M}$ is set to 10%, 5% and 1% respectively. Applying **SCT** with *PAGeR-Base* also shows significant improvement in experiments

across different sizes for $\mathcal{M}$. Results of *PAGeR-Base* with **KD** only, affects the $Acc_{avg}^{\mathcal{T}}$ of *PAGeR-Base* by -0.5%, -0.4% and +1.0% in experiments when $\mathcal{M}$ is 10%, 5% and 1% respectively suggesting that usage of **KD** by itself is not effective in alleviating catastrophic forgetting.

In row *PAGeR w/o LUG*, removing LUG from the combined approach highlights the importance of LUG and how the other two components, SCT and KD, are complementary to each other. This is reflected in the improved performance of *PAGeR-Base* by 6.8%, 7.6% and 10.4% in experiments with $\mathcal{M}$ set to 10%, 5% and 1% respectively. A similar trend is proven by rows *PAGeR w/o SCT*, *PAGeR w/o KD*, suggesting how all of these components are complementary to each other. In case of small memory size ($\mathcal{M}$=1%), all three components are required to achieve good intent detection accuracy, i.e., there is a drop of 1.5%, 3.4%, and 5.3% if we remove LUG, SCT and KD from *PAGeR*.

When all components are applied together with *PAGeR-Base* to give **PAGeR**, the $Acc_{avg}^{\mathcal{T}}$ improves from 82.9% to 90.2%, 81.1% to 89.2%, and 68.1% to 80.1% with $\mathcal{M}$ set to 10%, 5% and 1% respectively.

Approach	$\mathcal{M}$=10%	$\mathcal{M}$=5%	$\mathcal{M}$=1%
PAGeR-Base (LAMOL) (A)	82.9	81.1	68.1
A w/ *LUG only*	87.6	86.0	73.8
A w/ *SCT only*	87.9	86.2	72.1
A w/ *KD only*	82.4	80.7	69.1
B w/o *LUG*	89.7	88.7	78.5
B w/o *SCT*	88.8	88.0	76.7
B w/o *KD*	90.0	87.7	74.8
PAGeR (B)	**90.2**	**89.2**	**80.1**

Table 2: Ablation on components of **PAGeR** on **BANKING77** dataset. In *PAGeR-Base* (LAMOL), generative replay is based on only intent labels whereas in *PAGeR-Base* (LAMOL) + *LUG*, we use intent specific concept words along with intent labels for generative replay.

9 Applications of PAGeR

$\mathcal{M}$	Approach	FewRel
50 labelled samples per incremental step	**UB**	92.7
	LB	66.2
	EWC	30.2
	GEM	59.8
	A-GEM	47.5
	EMR	65.1
	EA-EMR	69.9
	EMAR	77.9
	PAGeR	**91.3**

Table 3: Average Accuracy on Lifelong relation extraction task. Except **UB** and **LB**, we have taken all baseline results from (Han et al., 2020)

9.1 Lifelong Relation Extraction

We applied *PAGeR* on a lifelong relation extraction task with the same experimental setup as described in Han et al. (2020), where the objective is to identify the correct relation mentioned in a given sentence from a set of candidate relations. To utilize candidate relations in *PAGeR*, we use *PWID* instead of *PWD* and replace *User Utterance*, *Incremental Question* and *Intent* with sentence, candidate relations and correct relation respectively as outlined in Fig. 3 (B). *PAGeR* outperforms the state-of-the-art baseline *EMAR* by 13.4% in terms of $Acc_{avg}^{\mathcal{T}}$ and is only 1.4% below the *UB* with $\mathcal{M} = 50$ per incremental step, as shown in Table 3. These results suggest that *PAGeR* can be extended to other class incremental NLP problems.

9.2 Multi-domain Dialog

We also evaluate *PAGeR* in a scenario where new intents are added as part of new domain at each incremental step and we use *K* previous utterances along with current utterance for intent detection. We use *SGD*, *MWOZ* multi-domain dialogue datasets and compare *PAGeR* with *AdapterCL*. As shown in Table 4, in this scenario, *PAGeR* outperforms *AdapterCL*.

	SGD		MWOZ	
Approach	**K=0**	**K=1**	**K=0**	**K=1**
AdapterCL	65.0	77.0	94.5	**96.51**
PAGeR	**77.12**	**92.6**	**98.10**	96.0

Table 4: Average Accuracy on incremental Multi-domain intent detection task

10 Conclusion

The current paper addresses the problem of lifelong intent detection and proposes *prompt augmented generative replay* (*PAGeR*) to alleviate catastrophic forgetting. We model lifelong intent detection as a *Class-IL* problem and use a common PLM for intent detection and pseudo-labelled data generation which is jointly trained for both tasks yielding SOTA performance. Future work would involve demonstrating the versatility of PAGeR on a variety of different NLP tasks, and to apply it to real world *Class-IL* scenarios wherein the initial number of classes is large and relatively few intents are added or removed in each iteration with significant class imbalance.

References

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and SIMONE CALDERARA. 2020. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930. Curran Associates, Inc.

Yue Cao, Hao-Ran Wei, Boxing Chen, and Xiaojun Wan. 2021. Continual learning for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3964–3974, Online. Association for Computational Linguistics.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *The European Conference on Computer Vision (ECCV)*.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019a. Efficient lifelong learning with a-gem. In *ICLR*.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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, Online. Association for Computational Linguistics.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

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, Online. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. 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 *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

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

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.

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 (EMNLP-IJCNLP)*.

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*, abs/2107.13586.

Qingbin Liu, Xiaoyan Yu, Shizhu He, Kang Liu, and Jun Zhao. 2021b. Lifelong intent detection via multi-strategy rebalancing.

X. Liu, Arash Eshghi, P. Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *IWSDS*.

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

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. Academic Press.

Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. 2021. Continual learning for named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13570–13577.

Abiola Obamuyide and Andreas Vlachos. 2019. Meta-learning improves lifelong relation extraction. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 224–229, Florence, Italy. Association for Computational Linguistics.

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.

German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2018. Continual lifelong learning with neural networks: A review. *CoRR*, abs/1802.07569.

Jing Qian, Hong Wang, Mai ElSherief, and Xifeng Yan. 2021. Lifelong learning of hate speech classification on social media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2304–2314, Online. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. 34:8689–8696.

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 (CVPR)*.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, J. Kirkpatrick, K. Kavukcuoglu, Razvan Pascanu, and R. Hadsell. 2016. Progressive neural networks. *ArXiv*, abs/1606.04671.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. 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*, volume 30. Curran Associates, Inc.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020a. {LAMAL}: {LA}nguage modeling is all you need for lifelong language learning. In *International Conference on Learning Representations*.

Jingyuan Sun, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2020b. Distill and replay for continual language learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3569–3579, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. 2020. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11:4069.

Gido M van de Ven and Andreas S Tolias. 2018. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*.

Chengyu Wang, Haojie Pan, Yuan Liu, Kehan Chen, Minghui Qiu, Wei Zhou, Jun Huang, Haiqing Chen, Wei Lin, and Deng Cai. 2021. *MeLL: Large-Scale Extensible User Intent Classification for Dialogue Systems with Meta Lifelong Learning*, page 3649–3659. Association for Computing Machinery, New York, NY, USA.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Zirui Wang, Sanket Vaibhav Mehta, Barnabas Poczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 535–548, Online. Association for Computational Linguistics.

Max Welling. 2009. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1121–1128, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. 2020. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, volume 33, pages 15173–15184. Curran Associates, Inc.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *ArXiv*, abs/2012.15466.

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, Bo Xu, P. Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. 2017. Self-taught convolutional neural networks for short text clustering. *Neural networks : the official journal of the International Neural Network Society*, 88:22–31.

A Appendix

A.1 Dataset Description

Lifelong Intent Detection: **CLINC150** (Larson et al., 2019) is a crowdsourced multi-domain (10 domains such as utility, travel etc.) intent detection dataset comprising of 23,700 queries with 22,500 in-scope queries labelled with 150 intents and 1,200 out-of-scope queries. We only use in-scope queries, with both the balanced and imbalanced versions of the provided dataset. **HWU64** (Liu et al., 2019) is a crowdsourced multi-domain (21 domains such as alarm, cooking etc.) intent detection dataset comprising of 25,716 queries labelled with 64 intents and is an imbalanced dataset. **BANKING77** (Casanueva et al., 2020) is an imbalanced, fine-grained intent detection dataset from the banking domain comprising of 13,083 customer queries labelled with 77 intents. **SGD** (Rastogi et al., 2020) and **MWOZ** (Budzianowski et al., 2018) are multi-domain dialogue datasets which consist of dialogues from 19 and 8 domains respectively. Total number of intents in **SGD** and **MWOZ** are 39 and 5 respectively.

Text Classification: **Stackoverflow** (*S20*) (Xu et al., 2017) is a multi-class classification dataset comprising of 20,000 question titles from Stackoverflow each tagged with one of 20 different tags.

Lifelong Relation Extraction: **FewRel** (Han et al.,

2018): is a relation extraction dataset where the objective is to predict a relation type between a pair of entities in a given sentence/paragraph. It consists of 56,000 examples tagged with 80 relations introduced incrementally at each step.
For all datasets, we use standard train-test split provided by respective authors (shown in table 5).

Table 5 contains details about different intent detection datasets $\mathcal{D}$ such as *CLINC150* with balanced ($C150^B$) and imbalanced version ($C150^I$), *BANKING77*, *HWU64 (H64)*, and one text classification dataset, i.e., *Stackoverflow (S20)*. It also contains details about *SGD*, *MWOZ* two multi-domain dialogue datasets which is used for intent detection task only. In these dialog datasets, the intent of an utterance can be determined on the basis of that utterance along with K previous history utterances from the dialog where K is hyperparameter for preprocessing during dataset creation. In our experiments, we have taken K=0 and K=1 to create the dataset for intent detection from SGD and MWOZ datasets.
Table 6 contains details about total number of incremental steps ($\mathcal{T}$), total number of intents ($Y^{\mathcal{T}}_{all}$) in an intent detection dataset ($\mathcal{D}$) and how these intents have been introduced over incremental steps, i.e., $\mathcal{I}_{dist}$. And all of these datasets are available in english language and released under creative Commons licences.

	$\mathcal{D}_{train}$	$\mathcal{D}_{val}$	$\mathcal{D}_{test}$
CLINC150 ($C150^B$)	15000	3000	4500
CLINC150 ($C150^I$)	10525	3000	4500
BANKING77 (B77)	8463	1540	3080
HWU64	8954	1076	1076
Stackoverflow (S20)	16000	-	4000
SGD	8843	1291	2421
MWOZ	2494	208	227

Table 5: Dataset Details

Dataset	Intents	$\mathcal{T}$	$\mathcal{I}_{dist}$
$C150^B$	150	10	{15, 15, 15, 15, 15, 15, 15, 15, 15, 15}
$C150^I$	150	10	{15, 15, 15, 15, 15, 15, 15, 15, 15, 15}
B77	77	7	{20, 10, 10, 10, 10, 10, 7}
H64	64	7	{10, 10, 10, 10, 10, 10, 4}
S20	20	5	{4, 4, 4, 4, 4}
SGD	39	19	{1,2,2,2,4,1,2,2,3,1,2,3,3,2,2,4,2,1,1}
MWOZ	5	5	{1,1,2,1,1}

Table 6: Dataset Details, where "Intents" ($Y^{\mathcal{T}}_{all}$) represents total number of intents introduced after $\mathcal{T}$ incremental steps.

Different shuffled versions of a dataset Same set of intents are introduced over different incremental steps to create different shuffled versions of a dataset $\mathcal{D}$, as shown in Fig. 6. We run all baselines, *PAGeR* and it's variant on four different shuffled versions of $\mathcal{I}_{dist}$ for a given $\mathcal{D}$.

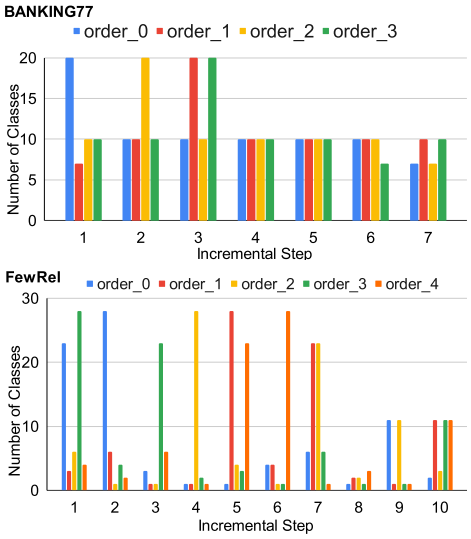

Figure 6: Different shuffled versions of *BANKING77* and *FewRel* dataset.

A.2 Training Details

For all the experiments reported in this paper we employ the GPT-2 PLM. To ensure a fair comparison across baselines we use GPT-2 as a common backbone for all baseline models. Training for all experiments is run for a fixed number of eight epochs. During training, the batch size was constrained to lie within the range $[8-32]$ and the maximum utterance length was fixed to 200 words. We used the Adam (Kingma and Ba, 2014) optimizer (with default values) for all the models and the learning rate was set to 5e-5. In all variants of PAGeR, we use top-k sampling (Holtzman et al., 2020) with $k = 20$ and $\lambda_1 = 1, \lambda_2 = 0.25, \lambda_3 = 0.25$. While synthesizing samples for contrastive learning, four negative samples were created for each positive sample. The distillation loss was optimized with a temperature value of 2. Baselines *EWC*, *A-GEM*, *ER*, *DER* are derived from the *DER*[1] code repository. Code for PAGeR experiments is derived from LAMOL and "Hugging Face" (Wolf et al., 2020). All the experiments were run on Titan P100, V100 and A100 GPUs. For exemplar replay based approaches, $\mathcal{M}$ is defined as the percentage of total training samples that can be stored for replay. $\mathcal{M}$ is uniformly distributed across existing intents. Upon arrival of new intents, $\mathcal{M}$ is uniformly redistributed across the total intents, both

[1]https://github.com/aimagelab/mammoth

old and new. In *PAGeR*, instead of real samples, we only store the corresponding concept words. For a fair comparison, we calculate the number of tokens that can be stored in the same amount of memory $\mathcal{M}$ and we generate one pseudo-labelled utterance corresponding to each entry in $\mathcal{M}$. We run every approach four times with shuffled versions of $\mathcal{D}$, i.e., set of intents introduced at $\mathcal{T}^{th}$ incremental step can be introduced at 1^{st} incremental step in another run. Please refer to A.3 for further details about memory management over incremental steps for different datasets.

A.3 Memory ($\mathcal{M}$)

To alleviate catastrophic forgetting, $\mathcal{M}$ contains labelled utterances corresponding to old intents Y^i_{all} while training the model at $i+1^{th}$ incremental step. The size of $\mathcal{M}$ is defined in terms of the percentage of whole labelled training utterances which can be stored in it. For *PAGeR*, we define the memory in terms of number of tokens where the number of tokens are obtained by tokenizing the labelled utterances present in the memory at a given incremental step. Instead of storing labelled utterances in *PAGeR*, we only store corresponding *Concepts* which take less memory in terms of tokens. So in *PAGeR* we can store more number of *Concepts* in the same amount of memory as shown in Table 7.

And after $i+1^{th}$ incremental step, $\mathcal{M}$ is equally divided among the intents seen so far, i.e., Y^{i+1}_{all} and excess existing labelled utterances corresponding to old intents Y^i_{all} are removed randomly.

		Intent Detection				TC
$\mathcal{M}$	**Approach**	**C150^B**	**C150^I**	**B77**	**H64**	**S20**
10%	Baselines	1500	1040	1000	900	1600
	PAGeR	6574	4025	3673	2480	4042
5%	Baselines	750	520	500	450	800
	PAGeR	2964	1751	1892	1238	2041
1%	Baselines	150	104	100	90	160
	PAGeR	512	516	232	121	401

Table 7: No. of *Labelled Utterances*/*Concepts* stored in $\mathcal{M}$.

A.4 Results

A.4.1 MSR Vs PAGeR

Liu et al. (2021b) (MSR) shows the experiments in lifelong intent detection with a fixed memory of 200 only. We performed same experiment with *PAGeR* and compiled the average accuracy at the last incremental step in table 8.

Dataset	**MSR**	**PAGeR**
CLINC150 (C150^B)	78.00	**92.16**
HWU64	52.14	**82.38**
SNIPS	93.57	**94.27**

Table 8: PAGeR experiments with fixed memory size (200)

A.4.2 Catastrophic forgetting over incremental steps

We evaluate all baselines, PAGeR-Base and *PAGeR* on four shuffled versions of a dataset and report *Average Accuracy* along with the Standard Deviation as shown in Table 9. We observed that *PAGeR* gets similar results on different shuffled versions of a dataset.

It's also evident from Fig. 7, forgetting is less in case of *PAGeR* as compared to other baselines as number of intents increases over incremental steps.

		Intent Detection				TC
$\mathcal{M}$	**Approach**	**C150^B**	**C150^I**	**B77**	**HWU64**	**S20**
	UB	*95.7*	*94.6*	*91.9*	*90.3*	*90.8*
	LB	36.9±3.9	30.1±1.4	23.3±2.2	24.4±2.9	21.6±2.9
	EWC	36.9±4.9	31.6±4.0	21.1±2.0	23.8±2.4	20.8±1.2
	AdapterCL	88.1±0.06	79.9±0.32	77.7±0.97	77.6±0.15	82.5±0.55
10%	A-GEM	30.8±10.7	13.6±0.8	17.3±3.5	18.8±2.9	67.6±5.1
	ER	88.9±0.8	81.7±0.4	84.7±1.6	78.3±1.4	85.2±0.6
	DER	90.8±0.9	83.9±0.7	80.7±3.7	81.3±2.1	85.6±1.4
	PAGeR-Base (LAMOL)	92.7±0.29	89.7±2.14	82.9±2.9	84.0±1.5	72.5±3.5
	PAGeR	**94.9**±0.22	**93.4**±0.37	**90.2**±0.26	**89.9**±0.67	**88.7**±0.3
5%	A-GEM	30.8±9.1	13.4±1.8	19.2±2.9	18.4±3.1	68.2±4.4
	ER	85.2±1.7	71.1±2.1	76.5±1.8	72.0±1.2	82.6±0.6
	DER	88.1±0.8	78.1±3.4	73.9±2.7	76.5±1.2	83.9±0.9
	PAGeR-Base (LAMOL)	92.0 ±0.27	87.5±1.45	81.1±3.8	82.2±2.3	71.3±2.2
	PAGeR	**94.4**±0.35	**92.9**±0.29	**89.2**±0.48	**88.0**±0.21	**88.0**±0.6
1%	A-GEM	28.7±10.5	13.9±1.0	25.3±15.0	17.9±1.4	62.4±6.4
	ER	64.8±3.2	41.3±1.8	51.6±3.7	46.2±0.8	76.3±0.8
	DER	73.0±1.8	45.9±2.7	53.0±2.7	49.4±1.5	77.4±1.3
	PAGeR-Base (LAMOL)	86.9±0.34	63.6 ±2.37	68.1±1.4	70.5 ±2.8	68.3±2.1
	PAGeR	**92.4**±0.71	**76.4**±1.53	**80.1**±1.46	**79.2**±0.84	**84.3**±1.0

Table 9: *Average Accuracy (↑)* with standard deviation on different datasets.

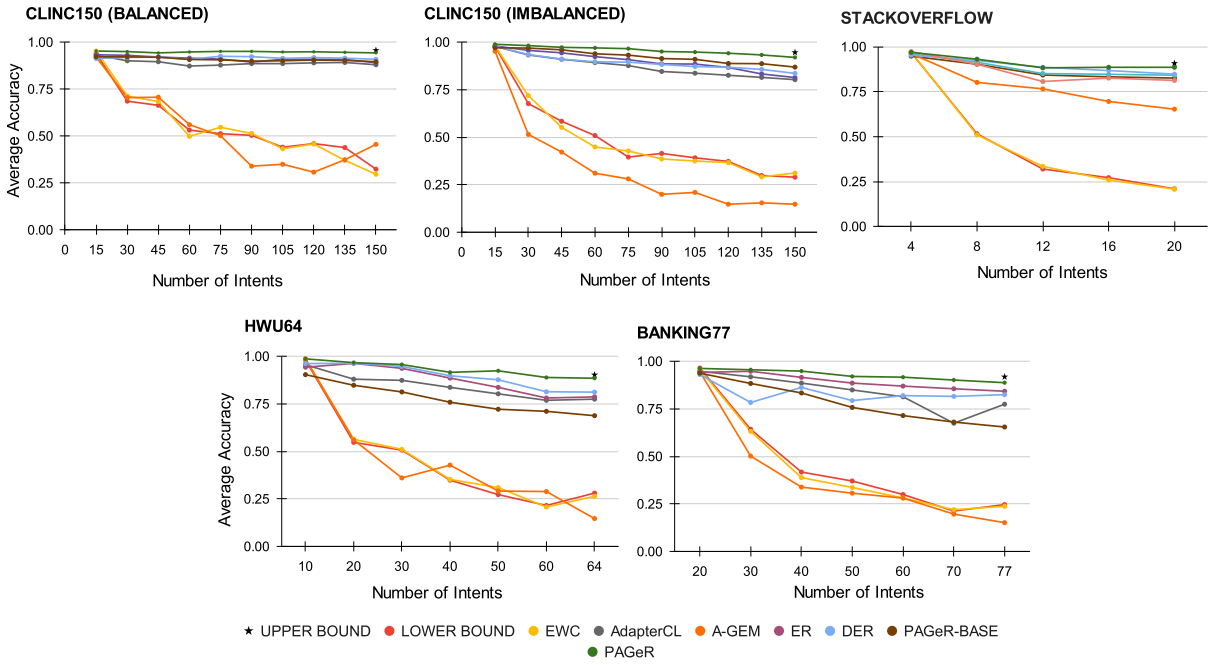

Figure 7: Average accuracy over incremental steps for different datasets. Here, average accuracy is shown on one shuffled version of each dataset.