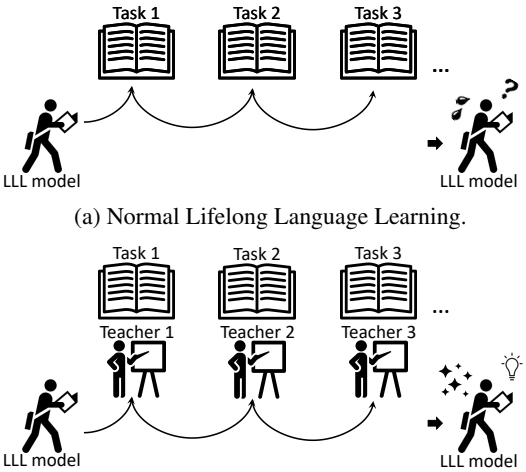
Lifelong Language Knowledge Distillation

Yung-Sung Chuang Shang-Yu Su Yun-Nung Chen
National Taiwan University, Taipei, Taiwan
{b05901033,f05921117}@ntu.edu.tw y.v.chen@ieee.org

Abstract

It is challenging to perform lifelong language learning (LLL) on a stream of different tasks without any performance degradation comparing to the multi-task counterparts. To address this issue, we present Lifelong Language Knowledge Distillation (L2KD), a simple but efficient method that can be easily applied to existing LLL architectures in order to mitigate the degradation. Specifically, when the LLL model is trained on a new task, we assign a teacher model to first learn the new task, and pass the knowledge to the LLL model via knowledge distillation. Therefore, the LLL model can better adapt to the new task while keeping the previously learned knowledge. Experiments show that the proposed L2KD consistently improves previous state-of-the-art models, and the degradation comparing to multi-task models in LLL tasks is well mitigated for both sequence generation and text classification tasks.[1]



(a) Normal Lifelong Language Learning.

(b) Lifelong Language Knowledge Distillation.

Figure 1: The difference between LLL and L2KD.

1 Introduction

Training a single model to learn a stream of different tasks sequentially usually faces the catastrophic forgetting problem (McCloskey and Cohen, 1989): after learning a new task, the model forgets how to handle the samples from previous tasks. Lifelong learning manages to accumulate the knowledge and retain the performance of previously learned tasks. It is important especially for real-world natural language processing (NLP) applications, because these applications need to interact with many users from different domains everyday, and the language usage also evolves from time to time. Hence, various NLP tasks have been studied for lifelong learning in the previous work, including sentiment analysis (Chen et al., 2015; Xia et al., 2017), conversational agents (Lee, 2017), word and sentence representation learning (Xu et al., 2018; Liu et al., 2019), text classification, and question answering (d'Autume et al., 2019).

In recent, LAMOL (Sun et al., 2020) improved the performance of LLL by a general framework: 1) it followed the idea about considering many NLP tasks as question answering (QA) (McCann et al., 2018) and adapted all tasks into the language modeling (LM) form. In the unified framework, it can perform LLL on many NLP tasks by generating answers based on the contexts and the questions using a single language model, and 2) it outperformed the previous methods by a considerable margin and is only 2%-3% worse than the multi-tasking upper bound, which jointly learns all tasks in a mixed dataset.

This paper further improves LLL by introducing Lifelong Language Knowledge Distillation (L2KD), which can be flexibly applied upon the LAMOL architecture or other LLL methods for sequence generation learning.

[1]The source code and data are available at https://github.com/voidism/L2KD.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2914–2924,
November 16–20, 2020.

The motivation of our work mainly comes from how to efficiently compress the knowledge under a lifelong language learning framework. If the model can learn a new task in an efficient way, the previously learned knowledge may not be affected and thus the problem of catastrophic forgetting can be mitigated.

Inspired by knowledge distillation (Bucila et al., 2006; Hinton et al., 2015; Kim and Rush, 2016), in which a student (smaller) model is trained to imitate the behavior of a teacher (larger) model in order to reach the performance closer to the teacher model, the LLL model in L2KD can be seen as a weak learner that needs to compress knowledge from different tasks into a compact single model. Thus LLL can benefit from the similar procedure of knowledge distillation, although the model size is equal to its teacher model. The similar idea about distilling knowledge from equal-size models has also been studied in born-again neural network (Furlanello et al., 2018), multitask learning (Clark et al., 2019) and lifelong computer vision learning (Hou et al., 2018), but never been explored in lifelong language learning research.

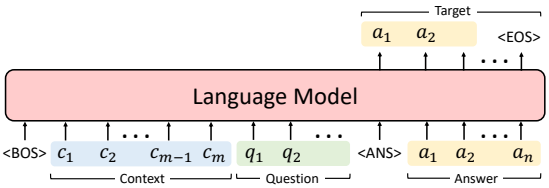
In L2KD, we train a new teacher model when facing a new task, and the LLL model imitates the behavior of its teacher at each training stage, as illustrated in Figure 1. This method only needs a little extra time to train a disposable teacher model for each new task, and the teacher model can be discarded when learning the next task; therefore, there is no extra memory or model capacity required for the target LLL model, making the proposed model more memory-efficient for real-world usage.

2 Proposed Approach

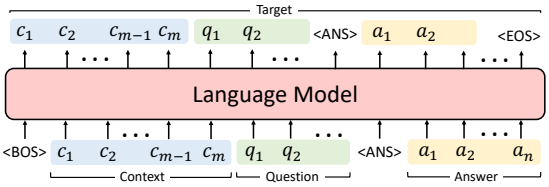
Before describing how L2KD works, in Section 2.1 we briefly introduce the architecture of LAMOL (Sun et al., 2020), which L2KD is built upon. Then we introduce different knowledge distillation strategies in Section 2.2, and how to apply them to L2KD in Section 2.3.

2.1 LAMOL: Language Modeling for Lifelong Language Learning

In the setting of LAMOL, all samples in language datasets have three parts: *context*, *question* and *answer*. We can simply concatenate these three parts into a single sentence and train the model to generate the answer based on the context and question prior to it, as illustrated in Figure 2a.



(a) Learning to solve target tasks (QA).



(b) Learning to generate pseudo-data (LM).

Figure 2: Illustration of learning QA and LM in LAMOL.

Besides generating answers for the given questions, the model simultaneously learns to model the whole training sample, as illustrated in Figure 2b. By doing that, when training on the next task, the model can generate training samples for the previous tasks and train on both data from the new task and the generated pseudo-data for the prior tasks. Thus the model would forget less when adapting to the new tasks.

LAMOL can outperform previous regularization-based (Schwarz et al., 2018; Aljundi et al., 2018) or memory-based (Lopez-Paz et al., 2017; Yogatama et al., 2019) LLL methods by a large margin. While most of previous methods usually get results slightly better than the finetuning baseline (doing nothing to prevent forgetting), LAMOL already get significant results that are very close to the multitasking upper bound and only 2%-3% worse (Sun et al., 2020) than it. Thus, in this paper, we focus on how to apply L2KD based on LAMOL.

2.2 Knowledge Distillation

Language Modeling The training objective for normal language modeling is to minimize the negative log-likelihood (NLL) in predicting the next word (hard target):

$$\mathcal{L}_{\text{NLL}}(x;\theta) = \sum_{t=t_0}^{T} -\log P(x_t \mid x_{<t};\theta),$$

where x_t denotes the t-th word in the sentence, $x_{<t}$ denotes all words prior to x_t, and θ is the parameters of the language model.

In knowledge distillation, instead, we minimize the prediction errors between student and teacher

models. The target unit for considering the errors can be done in the word level or the sequence level.

Word-Level (Word-KD) We minimize the cross-entropy between the output distributions from student and teacher models when predicting the next word:

$$\mathcal{L}_{\text{Word-KD}}(x;\theta_S;\theta_T) = \sum_{t=t_0}^{T}\sum_{k=1}^{|\mathcal{V}|} -P(\mathcal{V}_k \mid x_{<t};\theta_T)\log P(\mathcal{V}_k \mid x_{<t};\theta_S),$$

where the input $x_{<t}$ is from the ground truth sequence. $\mathcal{V}$ denotes the vocabulary set and $\mathcal{V}_k$ is the k-th word in $\mathcal{V}$. θ_S and θ_T are parameters of the student model and teacher model respectively.

Sequence-Level (Seq-KD) Similar to Kim and Rush (2016), we minimize the negative log-likelihood directly on the greedy decode or beam search output sequence $\hat{x}$ from the teacher model as the hard target, just like normal language modeling:

$$\mathcal{L}_{\text{Seq-KD}}(\hat{x};\theta_S) = \sum_{t=t_0}^{T} -\log P(\hat{x}_t \mid \hat{x}_{<t};\theta_S).$$

Seq-KD is usually applied for improving weak non-autoregressive translation (NAT) models (Zhou et al., 2020) by reducing the multi-modality problem in machine translation datasets (Gu et al., 2018).

Soft Sequence-Level (Seq-KD$_{\text{soft}}$) We further investigate whether the soft target plus the teacher decoded sequence can help the model more, so we conduct Seq-KD$_{\text{soft}}$, in which we perform Word-KD on the greedy decode or beam search outputs from the teacher model. The only difference between Seq-KD$_{\text{soft}}$ and Word-KD is that the input $x_{<t}$ of Word-KD is now replaced with $\hat{x}_{<t}$, the output sequence from the teacher model:

$$\mathcal{L}_{\text{Seq-KD}_{\text{soft}}}(\hat{x};\theta_S;\theta_T) = \sum_{t=t_0}^{T}\sum_{k=1}^{|\mathcal{V}|} -P(\mathcal{V}_k \mid \hat{x}_{<t};\theta_T)\log P(\mathcal{V}_k \mid \hat{x}_{<t};\theta_S).$$

Note that no matter what kind of loss we use in knowledge distillation, the teacher model is always fixed. Hence, the optimization process of finding parameters θ_S^* of the LLL model can be written as follows:

$$\theta_S^* = \arg\min_{\theta_S} \mathcal{L}_{\text{KD}}.$$

Algorithm 1 L2KD: Lifelong Language Knowledge Distillation

Input: current task dataset D_m, teacher model with parameters θ_T, knowledge distillation loss function $\mathcal{L}_{\text{KD}}$, pseudo-data sample rate γ.
Output: LLL model parameters θ_S.
Optimize teacher model on D_m to get parameters θ_T.
Sample $\gamma \cdot |D_m|$ pseudo-data from θ_S to form D_{prev}.
for all training samples $\{X_i^m\}_{i=1}^n \in D_m$ **do**
  for $i = 1$ **to** n **do**
    update θ_S to minimize $\mathcal{L}_{\text{KD}}(X_i^m;\theta_S;\theta_T)$
  end for
  Sample $n' = \gamma n$ samples $\{X_j^{\text{prev}}\}_{j=1}^{n'}$ from D_{prev}
  for $j = 1$ **to** n' **do**
    update θ_S to minimize $\mathcal{L}_{\text{NLL}}(X_j^{\text{prev}};\theta_S)$
  end for
end for

2.3 L2KD: Lifelong Language Knowledge Distillation

Knowledge distillation can be applied to minimizing both LM and QA loss in LAMOL. Assuming that there is a stream of tasks with datasets $\{D_1, D_2, ...\}$, our LLL model has learned from D_1 to D_{m-1} and now was adapted to D_m. First we train a teacher model on D_m by minimizing the negative log-likelihood loss both for LM and QA in LAMOL and obtain the model parameters θ_T^m.

Now our LLL model (with parameters θ_S) can be trained on D_m by knowledge distillation from the teacher model. Given a training sample $X_i^m = \{x_1, x_2, ..., x_T\} \in D_m$ (including the context, question and answer), we minimize

$$\mathcal{L}_{\text{new}}(X_i^m;\theta_S;\theta_T^m) = \mathcal{L}_{\text{new}}^{\text{QA}} + \mathcal{L}_{\text{new}}^{\text{LM}}$$
$$\mathcal{L}_{\text{new}}^{\text{QA}} = \mathcal{L}_{\text{Word-KD}}(X_i^m;\theta_S;\theta_T^m;t_0 = a_1)$$
$$\mathcal{L}_{\text{new}}^{\text{LM}} = \mathcal{L}_{\text{Word-KD}}(X_i^m;\theta_S;\theta_T^m;t_0 = 0),$$

where a_1 denotes the start position of the answer. Here we take Word-KD for illustration, but we can also replace the text in the answer part with the teacher-generated answers, so as to conduct Seq-KD$_{\text{soft}}$ or Seq-KD.

Besides training on samples from D_m, the LLL model also generates pseudo-data D_{prev} for previous tasks. For samples in D_{prev}, however, we cannot perform knowledge distillation here, because in our setting the teacher models of previous tasks will be discarded after adapting to the next task. Therefore, given the generated data $X_i^{\text{prev}} \in D_{\text{prev}}$,

we only minimize NLL loss here:

$$\mathcal{L}_{\text{prev}}(X_i^{\text{prev}};\theta_S) = \mathcal{L}_{\text{prev}}^{\text{QA}} + \mathcal{L}_{\text{prev}}^{\text{LM}}$$
$$\mathcal{L}_{\text{prev}}^{\text{QA}} = \mathcal{L}_{\text{NLL}}(X_i^{\text{prev}};\theta_S;t_0 = a_1)$$
$$\mathcal{L}_{\text{prev}}^{\text{LM}} = \mathcal{L}_{\text{NLL}}(X_i^{\text{prev}};\theta_S;t_0 = 0).$$

Finally we jointly optimize two loss and obtain the parameters θ_S^* for the LLL model:

$$\theta_S^* = \arg\min_{\theta_S}(\sum_{X_i^m \in D_m} \mathcal{L}_{\text{new}} + \sum_{X_i^{\text{prev}} \in D_{\text{prev}}} \mathcal{L}_{\text{prev}})$$

The training procedure is detailed in Algorithm 1.

3 Experimental Setup

To evaluate the proposed method, we conduct a set of experiments detailed below.

3.1 Model and Training Details

We build our proposed approach based on the implementation of LAMOL[2] to make the results comparable. We use the same pre-trained small GPT-2 (Radford et al., 2019) for all single-task teacher, multitask and LLL models, and train the GPT-2 nine epochs for each dataset. We use the best setting in LAMOL: using task-specific tokens as begin-of-sentence tokens, and the pseudo-data sample rate is 0.2. During inference, we use greedy decoding to generate sequence. More details can be found in Appendix A.

3.2 Datasets

To evaluate the capability of L2KD on diverse sequence generation tasks, we pick the following three tasks from DecaNLP (McCann et al., 2018):

- **WikiSQL** (Zhong et al., 2017): a dataset for developing natural language interfaces for relational databases, in which the model needs to generate structured queries from natural language.
- **CNN/DailyMail** (See et al., 2017): a text summarization dataset collected from online news articles.
- **MultiWOZ** (Budzianowski et al., 2018): a multi-domain wizard-of-oz dataset for task-oriented dialogue modeling, in which the model needs to generate the semantic state sequences based on the given partial dialogues.

Note that we skip machine translation dataset (IWSLT) in DecaNLP here, because GPT-2 does not contain a multilingual vocabulary. These three datasets focus on *different tasks*, representing the most general case in LLL.

[2] https://github.com/jojotenya/LAMOL

Dataset	Metric	# Train	# Test
Sequence Generation for Different Tasks			
WikiSQL	lfEM	6,525	15,878
CNN/DailyMail	ROUGE	6,604	2,250
MultiWOZ	dsEM	2,536	1,646
Sequence Generation for Different Domains			
E2E NLG		6,000	2,000
RNNLG (rest.)		6,228	1,039
RNNLG (hotel)	ROUGE	6,446	1,075
RNNLG (tv)		8,442	1,407
RNNLG (laptop)		7,944	2,649
Text Classification for Different Tasks			
AGNews		115,000	7,600
Yelp		115,000	7,600
Amazon	Exact Match	115,000	7,600
DBPedia		115,000	7,600
Yahoo		115,000	7,600

Table 1: Dataset sizes and the evaluation metrics.

However, in real-world scenarios, it is more common that the LLL model is trained to solve the same task, but in *different domains* that change through time. Thus we conduct the experiments on the following natural language generation (NLG) datasets with five different domains:

- **E2E NLG** (Novikova et al., 2017): a dataset for training end-to-end natural language generation systems in the *restaurant domain*.
- **RNNLG** (Wen et al., 2015): a dataset for NLG in spoken dialogue system application domains. It contains four domains: *San Francisco restaurant search* (rest.), *San Francisco hotel search* (hotel), *Television sale/search* (tv), *Laptop sale/search* (laptop). We use the full dataset for the first three domains and the reduced set for the laptop domain for keeping them balance.

Although our method is mainly designed for sequence generation tasks, we also use five different text classification datasets to evaluate whether the proposed method also benefits text classification tasks. We use the random sampled subsets released by Sun et al. (2020), each of which has 115,000 training and 7,600 testing instances.

- **AGNews**: News articles, including 4 classes for their topics.

	Method	WOZ	CNN	SQL	Avg	WOZ	CNN	SQL	Avg	WOZ	CNN	SQL	Avg
		WOZ → CNN → SQL				CNN → SQL → WOZ				SQL → WOZ → CNN			
(a)	Finetune	0.0	26.3	64.3	30.2	84.6	6.8	2.1	31.2	0.1	26.0	0.0	8.7
(b)	LAMOL	67.6	27.3	62.5	52.4	83.0	27.8	60.8	57.2	76.1	26.0	55.0	52.4
(c)	(b) + Word-KD	**82.4**	27.6	**65.0**	**58.3**	**86.1**	27.5	63.2	**59.0**	79.5	26.2	**59.6**	55.1
(d)	(b) + Seq-KD$_{soft}$	81.0	26.9	64.7	57.5	84.1	27.6	**63.4**	58.4	**81.7**	25.9	58.4	**55.3**
(e)	(b) + Seq-KD	76.4	**28.0**	63.7	56.1	83.0	**28.3**	61.5	57.6	81.0	**27.5**	57.3	**55.3**
		WOZ → SQL → CNN				CNN → WOZ → SQL				SQL → CNN → WOZ			
(a)	Finetune	0.0	25.8	0.0	8.6	3.6	24.5	64.0	30.7	85.0	7.3	0.0	30.8
(b)	LAMOL	76.1	26.3	59.3	53.9	79.8	27.3	64.1	57.0	84.0	27.2	58.7	**56.6**
(c)	(b) + Word-KD	**81.4**	26.7	59.6	**55.9**	83.5	27.8	**65.0**	58.8	78.7	26.4	**59.0**	54.7
(d)	(b) + Seq-KD$_{soft}$	80.4	26.1	**59.9**	55.5	**83.7**	28.6	64.8	**59.0**	84.7	26.2	58.8	**56.6**
(e)	(b) + Seq-KD	77.2	**27.0**	59.5	54.5	82.8	**29.5**	64.4	58.9	**84.9**	**27.8**	57.3	**56.6**

Table 2: Detailed experimental results on MultiWOZ (WOZ), CNN/DailyMail (CNN), WikiSQL (SQL), with six different lifelong learning orders.

- **Yelp**: Customer reviews on Yelp, including 5 classes for their rating scores.
- **Amazon**: Customer reviews on Amazon, including 5 classes for their rating scores.
- **DBPedia**: Articles on Wikipedia, including 14 classes for their categories.
- **Yahoo**: QA pairs on the Yahoo! platform, including 10 classes for their categories.

Due to the limitation of computational resources and the data imbalance, we reduce the big datasets (WikiSQL, CNN/DailyMail, E2E NLG, RNNLG (laptop)) to a smaller size by random sampling. The reduced data size and other data statistics in the experiments are detailed in Table 1.

4 Results and Discussion

We discuss the results for three settings: 1) different sequence generation tasks, 2) same tasks in different domains, and 3) different text classification tasks in order to validate the effectiveness of the proposed approach.

4.1 Different Sequence Generation Tasks

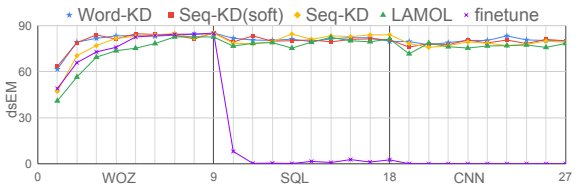
In the experiments, we perform lifelong learning on the WikiSQL (SQL), CNN/DailyMail (CNN) and MultiWOZ (WOZ) datasets with six different permutation orders, and test the performance at the end of the training streams. The detailed results are shown in Table 2, where the average scores indicate the average of three tasks for overall comparison. Note that the evaluation metrics of these three tasks are all ranging from 0 to 100. The overall results of six orders compared with single-task methods and multitask upper bounds are shown in Table 3.

In Table 2, the first baseline is **(a) Finetune**, in which we directly train three tasks one after another without preventing catastrophic forgetting. It is obvious that the Finetune model would forget one or two tasks learned before the final one. **(b) LAMOL** is the current state-of-the-art approach that significantly reduce the catastrophic forgetting for comparison. In the rows (c)-(e), it is shown that applying L2KD upon LAMOL significantly outperforms LAMOL for almost all cases, no matter which knowledge distillation strategy is used: **(c) Word-KD**, **(d) Seq-KD$_{soft}$**, **(e) Seq-KD**. We also observe that among three different knowledge distillation strategies, **(e) Seq-KD** consistently improves the most on the CNN/DailyMail dataset, which is probably caused by the noisy nature of this summarization dataset. Therefore, sequence-level knowledge distillation produces a easy-to-learn answer comparing to the original complex answer, so that the LLL model can learn better on it.

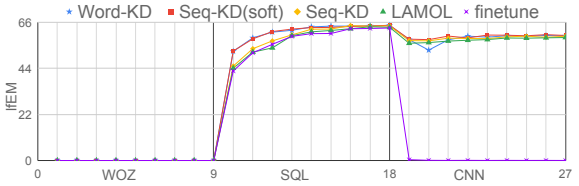
On the other hand, for other two tasks (MultiWOZ, WikiSQL), **(c) Word-KD** and **(d) Seq-KD$_{soft}$** improve more for most cases. Because the target sequences of these two tasks are relatively simple, where MultiWOZ focuses on producing semantic state sequences from dialogues, and WikiSQL produces the structured query sequences from the given natural language sentences, the target sequences usually contain the patterns less complex than natural language. So, in these

Non-Lifelong Methods		WOZ	CNN	SQL	Avg
(1)	Single QA	84.8	25.5	63.1	**57.8**
(2)	Single QA+LM	82.2	25.9	**63.7**	57.3
(3)	$\text{Multi}_{\text{same}}$ QA	66.2	25.6	53.0	48.3
(4)	$\text{Multi}_{\text{same}}$ QA+LM	59.0	26.3	53.6	46.3
(5)	$\text{Multi}_{\text{long}}$ QA	82.7	26.1	61.1	56.6
(6)	$\text{Multi}_{\text{long}}$ QA+LM	**85.4**	**26.7**	61.3	**57.8**
(7)	(6) + Seq-KD	84.4	27.6	61.8	58.0
Lifelong Methods (averaged over six orders)					
(a)	Finetune	28.9	19.5	21.7	23.4
(b)	LAMOL	77.7	27.0	60.0	54.9
(c)	(b) + Word-KD	81.9	27.0	**61.9**	57.0
(d)	(b) + $\text{Seq-KD}_{\text{soft}}$	**82.6**	26.9	61.7	**57.1**
(e)	(b) + Seq-KD	80.9	**28.0**	60.6	56.5
STD of Lifelong Methods					
(f)	Finetune	43.3	9.6	32.9	28.6
(g)	LAMOL	6.0	**0.7**	3.2	3.3
(h)	(g) + Word-KD	2.7	**0.7**	**2.8**	2.1
(i)	(g) + $\text{Seq-KD}_{\text{soft}}$	**1.8**	1.0	3.0	**1.9**
(j)	(g) + Seq-KD	3.4	0.9	3.1	2.5

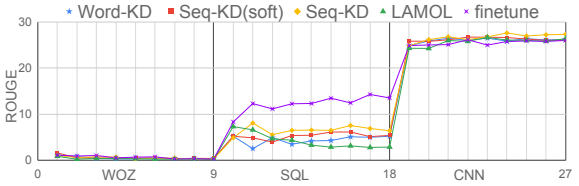
Table 3: Averaged results at the final task of the lifelong learning procedures over six orders, comparing to single task and multitask upper bound. The bold numbers are the best in the group.



(a) Learning curve of WOZ.



(b) Learning curve of SQL.



(c) Learning curve of CNN.

Figure 3: The learning curves of different LLL methods in the order of WOZ → SQL → CNN.

cases, the soft targets may bring more advantages than teacher decoding sequences for the LLL model to learn from.

In Table 3, the overall performance (averaged over six permutation orders) is compared with single-task methods and multi-task upper bounds. There are two training methods here: optimizing QA loss only (in rows (1)(3)(5)) or optimizing both QA and LM loss (in rows (2)(4)(6)), as illustrated in Figure 2. For multi-task models, we find that the same training steps (9 epochs on the mixed set) may not lead the models to converge (in row (3)(4)), so we additionally train multi-task models for three times longer (27 epochs on the mixed set) in rows (5)(6).

The second part of Table 3 shows the average performance in lifelong learning of six permutation orders. It is clear that L2KD significantly improves the average score from 54.9 in **(b) LAMOL** to 57.1 in **(d) $\text{Seq-KD}_{\text{soft}}$**. The performance of $\text{Seq-KD}_{\text{soft}}$ is only 0.7% worse than the multi-task upper bound, 57.8 in **(6) $\text{Multi}_{\text{long}}$ QA+LM**. Hence, the results show that L2KD can bridge the gap between lifelong learning and multi-task learning.

Note that we can also apply similar distillation strategy on multitask learning to obtain a stronger upper bound, which might be a more fair comparison. Thus, we add Seq-KD to **(6) $\text{Multi}_{\text{long}}$ QA+LM** by making the model learn from five single-task teachers and the results are shown in row (7). We observe that the improvement on multitask learning is only 0.2%, while L2KD can improve LAMOL by 2.2%. This result indicates that the benefits brought by knowledge distillation may be saturated for multitask learning, but is not saturated for L2KD. The gap between lifelong learning and multi-task learning is still reduced even if we apply similar strategy on both of the models.

The third part of Table 3 shows the standard deviations of six permutation orders. As mentioned in Sun et al. (2020), if an algorithm has smaller standard deviation over different training orders, it means that the algorithm is more robust and not susceptible to learning orders. It can be found that the average standard deviation of LAMOL is reduced from 3.3 to 1.9 with **$\text{Seq-KD}_{\text{soft}}$**. Therefore, both soft target training and teacher decode sequence can stabilize the training process of LLL and make it more order-agnostic.

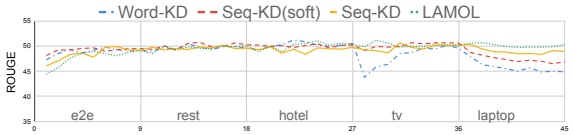
To further analyze the performance change when

Method	e2e	rest	hotel	tv	laptop	Avg
Single$_{(QA)}$	48.8	64.0	65.4	70.8	73.0	64.4
Single$_{(QA+LM)}$	48.8	64.2	65.5	71.0	72.8	64.5
Multi$_{(QA)}$	49.2	**65.6**	**67.2**	72.7	**74.8**	**65.9**
Multi$_{(QA+LM)}$	**49.5**	65.2	66.7	**73.4**	74.6	**65.9**
Left-to-right (e2e → rest → hotel → tv → laptop)						
LAMOL	**50.1**	58.7	61.5	73.7	72.0	63.2
+ Word-KD	44.9	60.0	62.8	**76.7**	73.3	63.5
+ Seq-KD$_{soft}$	46.9	58.4	63.2	76.4	73.6	63.7
+ Seq-KD	48.6	**62.2**	**66.4**	74.7	**75.5**	**65.5**
Right-to-left (laptop → tv → hotel → rest → e2e)						
LAMOL	**49.8**	65.0	65.9	75.8	77.0	66.7
+ Word-KD	49.3	**67.6**	**68.7**	76.8	77.7	**68.0**
+ Seq-KD$_{soft}$	49.4	66.6	68.0	76.7	77.4	67.6
+ Seq-KD	49.7	65.9	66.7	**77.4**	**78.8**	67.7

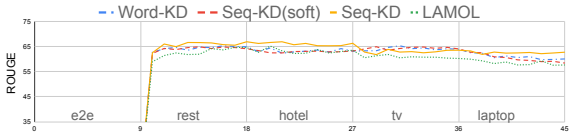
Table 4: Experimental results on NLG datasets from different domains.

training on different tasks, we plot the testing results during whole lifelong learning stages with the order of WOZ (1-9 epoch) → SQL (10-18 epoch) → CNN (19-27 epoch) in Figure 3. In Figure 3a, the performance of WOZ for all methods is illustrated. The finetune baseline (purple line) significantly degrades when moving to the next task (SQL) in the second training stage, while other methods can keep the performance. We observe that applying soft-target Word-KD (blue) or Seq-KD$_{soft}$ (red) can increase the scores faster than hard-target Seq-KD (yellow) and LAMOL baseline (green) at the initial epochs, indicating the effectiveness of the proposed L2KD. In terms of other two tasks, all distillation methods (Word-KD, Seq-KD$_{soft}$, Seq-KD) are capable of maintaining the performance of WOZ slightly better than LAMOL, and finally converge to better points in the third training stage. A similar trend can be observed in Figure 3b, where soft-target Word-KD and Seq-KD$_{soft}$ rise faster in the second training stage and finally drop less than original LAMOL in the third training stage, demonstrating the great property of our proposed methods as LLL models.

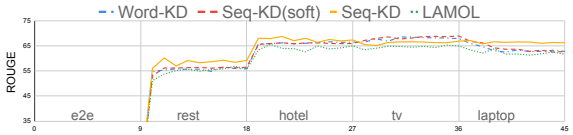
In Figure 3c, in the third stage, the yellow line (Seq-KD) converges to a better point than all other methods, because it reduces the complexity of the noisy summarization dataset. However, although Seq-KD$_{soft}$ also reduces the complexity, it does not achieve the same performance as Seq-KD. The probable reason is that the teacher decoding sentences may be easy enough for the LLL model to learn from, and the soft target here makes the model not completely converge on these easy sentences.



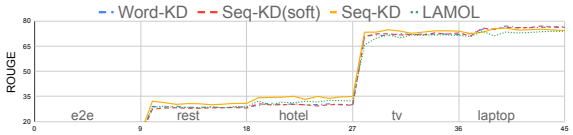
(a) Learning curve of E2ENLG.



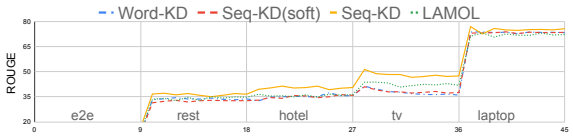
(b) Learning curve of RNNLG (restaurant).



(c) Learning curve of RNNLG (hotel).



(d) Learning curve of RNNLG (tv).



(e) Learning curve of RNNLG (laptop).

Figure 4: The learning curves on NLG tasks with the hardest-to-easiest (left-to-right) order.

4.2 Same Task in Different Domains

We perform L2KD on the same NLG task with five different domains: *restaurant* from E2ENLG, *restaurant/hotel/tv/laptop* from RNNLG. Note that although both E2ENLG and RNNLG has the *restaurant* domain, their input formats and label types are totally different. The results are shown in Table 4, where we only show two orders in the experiments: from the hardest task to the easiest one (left-to-right) and its reverse order (right-to-left)[3]. The results show that L2KD outperforms original LAMOL for most cases and improves the averaged ROUGE score by nearly 2 points.

We find that different training orders bring slightly different results. In the right-to-left order, the baseline LAMOL can easily outperform multi-task models due to its easiest-to-hardest order, which helps the model to better transfer the knowledge gradually in these NLG tasks, similar to curriculum learning (Bengio et al., 2009). There-

[3]The shown results are representative among all others.

fore, it does not mean that lifelong learning can beat multi-task model in all the experiment.

We plot the learning curves of these five tasks in left-to-right order in Figure 4 for further analysis. Except for E2ENLG, the Seq-KD in yellow lines usually gain more performance at the end of the training stream. Also, we observe that when forward transfer exists, Seq-KD usually benefits more. For example, in Figure. 4c, when training on RNNLG (restaurant) in 10-18th epochs, the ROUGE score on RNNLG (hotel) has risen even before the model first sees RNNLG (hotel) data at the 19th epoch, indicating that the forward transfer exists in this order. The rising trend is more obvious in Seq-KD (yellow), and the same trend can also be obversed in Figure 4d and 4e.

4.3 Text Classification Tasks

Although our method is mainly designed for sequence generation tasks, we investigate whether this idea also benefits text classification (TC) tasks. Thus we perform L2KD on five TC tasks, where the answers are always very short sequences representing the class labels of the given documents, such as `World`, `Sports`, `Business`, `or Sci/Tech` in the AGNews dataset. Hence, generating such short answers is not complex for the proposed model, and the performance mainly reflects the text understanding performance instead of the generating capability.

We also conduct the experiments from the hardest task to the easiest task, and its reverse order shown in Table 5. To our surprise, L2KD also improves LAMOL to get better results on TC tasks. The results of these two orders are only 0.1% worse than the multi-task upper bounds. The Word-KD improves the most on these TC tasks in most cases, and the improvements are more obvious especially for the earlier learned tasks. The details of the learning curves in TC tasks are also shown in Appendix B for reference.

Because the answers in TC tasks are not as complex as other sequence generation tasks, we investigate where the improvement mainly comes from during the distillation process. Therefore, we split each testing set into two groups: (A) *questions correctly answered by the teacher model*; (B) *questions incorrectly answered by the teacher model*. We suspect that the LLL model trained by L2KD may totally copy the behavior from the teacher models and get improvement mainly from the group (A),

Method	amazon	yelp	yahoo	ag	dbpedia	Avg
Single$_{(QA)}$	55.9	63.3	70.6	93.6	99.0	76.5
Single$_{(QA+LM)}$	56.9	**64.5**	70.1	93.7	**99.1**	76.9
Multi$_{(QA)}$	56.6	63.3	69.2	93.7	99.0	76.4
Multi$_{(QA+LM)}$	**57.8**	64.4	**70.9**	**94.0**	**99.1**	**77.2**
Left-to-right (amazon → yelp → yahoo → ag → dbpedia)						
LAMOL	52.7	61.6	70.3	93.6	99.1	75.5
+ Word-KD	**57.5**	**63.6**	**71.3**	**93.9**	**99.2**	**77.1**
+ Seq-KD$_{soft}$	55.7	62.0	**71.3**	**93.9**	**99.2**	76.4
+ Seq-KD	56.8	62.3	71.1	93.4	99.1	76.6
Right-to-left (dbpedia → ag → yahoo → yelp → amazon)						
LAMOL	57.9	63.5	70.7	91.7	98.3	76.4
+ Word-KD	57.0	64.1	**73.2**	**92.7**	**98.8**	**77.1**
+ Seq-KD$_{soft}$	57.0	64.1	71.9	92.4	**98.8**	76.8
+ Seq-KD	**58.4**	**64.4**	71.7	91.5	**98.8**	76.9

Table 5: Experimental results on five text classification datasets.

	Acc	**Acc in (A)**	**Acc in (B)**
Teacher	76.73	100.00	0.00
LAMOL	75.48	88.15	33.69
+ Word-KD	77.11	90.26 (+2.11)	33.75 (+0.06)
+ Seq-KD$_{soft}$	76.42	89.42 (+1.27)	33.52 (-0.17)
+ Seq-KD	76.56	89.56 (+1.41)	33.69 (+0.00)

Table 6: The accuracy in the group (A) and (B) averaged over five classification datasets. The teacher scores are from five single-task models.

while it fails to answer the questions in the group (B). To figure it out, we compute the accuracy of each LLL model (in left-to-right experiment) for the groups (A) and (B) respectively, and the difference between original LAMOL and three distillation strategies on five tasks. The averaged results are shown in Table 6, and the more detailed results for each task can be found in Appendix C. Surprisingly, applying L2KD does not largely degrade the accuracy in the group (B) comparing to the original LAMOL, and even improves for Word-KD, showing that the LLL model does not fully copy the behavior from its teacher models. On the other hand, the total improvement indeed mainly comes from the group (A), and Word-KD also can improve the most. The double improvement both on group (A) and (B) for Word-KD indicates that on these TC tasks, the LLL model trained by Word-KD can better reach the balance between the *teacher knowledge* and the *transfer ability*. Therefore, it can get the advantages from the teacher knowledge while avoid some false knowledge taught from its teacher by integrating the knowledge from other tasks.

5 Related Work

Knowledge distillation has been introduced to the field of lifelong learning; for example, Learning without Forgetting (LwF) (Li and Hoiem, 2017), Generative Replay with Distillation (DGR+distill), Replay-through-Feedback (RtF) (van de Ven and Tolias, 2018), and Lifelong GAN (Zhai et al., 2019), a lot of prior studies have also used knowledge distillation in lifelong learning, but all in computer vision tasks. Different from the prior work, this paper is the first attempt that adopts knowledge distillation for NLP tasks in the lifelong learning framework. Moreover, the prior work used the old model as a teacher to help the current model retain the knowledge about previous tasks. In contrast, our method trains a new teacher model on the incoming new task. Thus, these two directions of applying knowledge distillation are complementary to each other, showing the potential of applying the proposed method to the fields other than NLP.

6 Conclusion

This paper presents Lifelong Language Knowledge Distillation (L2KD), a simple method that effectively help lifelong language learning models to maintain good performance comparable to its multi-task upper bounds. The experiments show the consistent improvement achieved by L2KD for three different settings, indicating the effectiveness of the proposed method to train robust LLL models. In addition, the proposed approach only requires a little extra time for training the teacher without extra memory or capacity needed, showing the potential of being applied to the practical scenarios.

Acknowledgments

We thank reviewers for their insightful comments. We are grateful to Cheng-Hao Ho and Fan-Keng Sun for their source code of LAMOL and valuable suggestions. This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grant 109-2636-E-002-026.

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.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.

Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015. Lifelong learning for sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc Le. 2019. Bam! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *arXiv preprint arXiv:1906.01076*.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. *International Conference on Learning Representations*.

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

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2018. Lifelong learning via progressive distillation and retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 437–452.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Sungjin Lee. 2017. Toward continual learning for conversational agents. In *arXiv*.

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

Tianlin Liu, Lyle Ungar, and João Sedoc. 2019. Continual learning for sentence representations using conceptors. In *NAACL-HLT*.

David Lopez-Paz et al. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476.

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. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Jonathan Schwarz, Jelena Luketina, Wojciech M Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. *arXiv preprint arXiv:1805.06370*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

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

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*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

R. Xia, J. Jiang, and H. He. 2017. Distantly supervised lifelong learning for large-scale social media sentiment analysis. *IEEE Transactions on Affective Computing*, 8(4):480–491.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Lifelong domain word embedding via meta-learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. 2019. Lifelong gan: Continual learning for conditional image generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2759–2768.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Chunting Zhou, Graham Neubig, and Jiatao Gu. 2020. Understanding knowledge distillation in non-autoregressive machine translation. *International Conference on Learning Representations*.

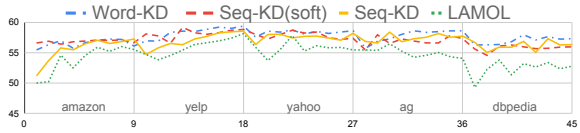
A Training Details

We use a single NVIDIA TESLA V100 (32G) for each experiment. The average runtime of experiments in Section 4.1 and 4.2 are 3-8 hours. The experiments in Section 4.3 need about 3 days for a single experiment.

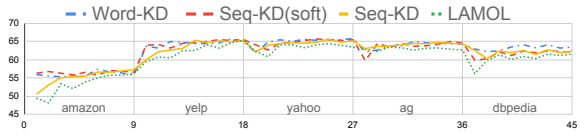
We did not conduct hyperparameter search, but follow all best settings in the official implementation of LAMOL [4] to keep the results comparable. The main hyperparameters are listed in Table 7. More details can be found in our released code.

B Learning Curves for Text Classification Tasks

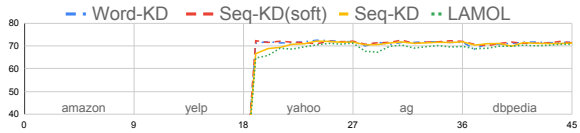
The learning curves for five text classification tasks are shown in Figure 5.



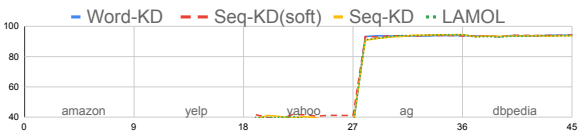
(a) Learning curve of Amazon.



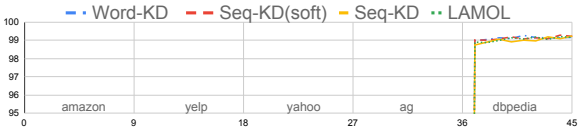
(b) Learning curve of Yelp.



(c) Learning curve of Yahoo.



(d) Learning curve of AGNews.



(e) Learning curve of DBpedia.

Figure 5: The learning curves on the five text classification tasks. X-axis represents epochs, Y-axis represents accuracy.

C Detailed Accuracy Analysis for Text Classification Tasks

The detailed accuracy in group (A) and (B) for five text classification tasks is shown in Table 8.

[4] https://github.com/jojotenya/LAMOL

hyperparameter	value
optimizer	Adam
adam epsilon	1.0×10^{-4}
learning rate	6.25×10^{-5}
training epochs / task	9
max gradient norm	1.0
learning rate schedule	warmup linear
warmup ratio	0.005
temperature for KD	2.0
top-k sampling	k=20
weight decay	0.01

Table 7: The main hyperparameters in the experiment.

	Acc	Acc in (A)	Acc in (B)
Amazon			
Teacher	55.50	100.00	0.00
LAMOL	52.74	66.22	35.93
+ Word-KD	57.54	73.33 (+7.11)	37.85 (+1.92)
+ Seq-KD$_{soft}$	55.74	70.41 (+4.20)	37.43 (+1.51)
+ Seq-KD	56.78	71.98 (+5.76)	37.82 (+1.89)
Yelp			
Teacher	64.11	100.00	0.00
LAMOL	61.61	75.82	36.22
+ Word-KD	63.59	79.92 (+4.10)	34.43 (-1.80)
+ Seq-KD$_{soft}$	62.00	77.50 (+1.68)	34.32 (-1.91)
+ Seq-KD	62.32	77.79 (+1.97)	34.68 (-1.54)
Yahoo			
Teacher	71.20	100.00	0.00
LAMOL	70.29	88.28	25.81
+ Word-KD	71.28	89.63 (+1.35)	25.90 (+0.09)
+ Seq-KD$_{soft}$	71.26	89.39 (+1.11)	26.45 (+0.64)
+ Seq-KD	71.13	89.52 (+1.24)	25.68 (-0.14)
AGNews			
Teacher	93.76	100.00	0.00
LAMOL	93.63	97.15	40.70
+ Word-KD	93.91	97.67 (+0.52)	37.32 (-3.37)
+ Seq-KD$_{soft}$	93.89	97.81 (+0.66)	35.00 (-5.69)
+ Seq-KD	93.45	97.15 (+0.00)	37.74 (-2.95)
DBPedia			
Teacher	99.11	100.00	0.00
LAMOL	99.13	99.78	26.61
+ Word-KD	99.24	99.85 (+0.07)	31.05 (+4.44)
+ Seq-KD$_{soft}$	99.18	99.85 (+0.07)	25.13 (-1.48)
+ Seq-KD	99.11	99.82 (+0.04)	19.22 (-7.39)

Table 8: The accuracy in the group (A) and (B) detailed in five classification datasets. The teacher scores are from five single-task models.