IEICE TRANS. INF. & SYST., VOL.E106–D, NO.1 JANUARY 2023

LETTER

Metacognitive Adaptation to Enhance Lifelong Language Learning

Han WANG[†,††a)], ***Student Member***, **Ruiliu FU**[†,††], **Xuejun ZHANG**[†,††b)], **Jun ZHOU**[†,††], ***Nonmembers***, ***and*** **Qingwei ZHAO**[†,††], ***Member***

SUMMARY Lifelong language learning (LLL) aims at learning new tasks and retaining old tasks in the field of NLP. LAMOL is a recent LLL framework following data-free constraints. Previous works have been researched based on LAMOL with additional computing with more time costs or new parameters. However, they still have a gap between multi-task learning (MTL), which is regarded as the upper bound of LLL. In this paper, we propose Metacognitive Adaptation (Metac-Adapt) almost without adding additional time cost and computational resources to make the model generate better pseudo samples and then replay them. Experimental results demonstrate that Metac-Adapt is on par with MTL or better.

key words: *lifelong learning, metacognition, adaptation, pseudo data, catastrophic forgetting*

1. Introduction

Lifelong learning [1], which is one of the cornerstones of the continuous development of human civilization, is the ability of humans to learn new knowledge while strengthening old knowledge [2]. We hope that a machine can learn and update itself like a human over a long period of time. However, traditional machine learning paradigms forget what they have learned before while learning a new task because of data shift, which is referred to as catastrophic forgetting [3].

In the field of NLP, lifelong learning is also known as lifelong language learning (LLL) which learns NLP tasks in the stream. LAMOL [4], which is a LLL framework following data-free constraint [2], applies a single language model to learn various NLP tasks where data are formatted as QA-style. LAMOL alleviated catastrophic forgetting by generating and replaying pseudo data of previously learned tasks instead of replaying real data. However, LAMOL still has a gap between multi-task learning (MTL) which is regarded as the upper bound [1], [5] of LLL. Many works have been researched based on LAMOL. L2KD [6], DnR [7], DFSD [8] improved LAMOL by distilling the parts or all layer of the model. Rational-LAMOL [9] applied rationale information of samples to critical freezing parts of the module. [10] and Adaptive Compositional Modules (ACM) [11] applied the Adapter [12] for each task to improve LAMOL. Although these methods mentioned above shorten the gap between the MTL, there is still space for improvement, especially in conditions of insufficient pseudo samples. In addition, all of them need much more time-cost or computational resources.

The reason why there is a gap between the above LLL methods and the MTL we analyze is detailed below: pseudo samples of earlier learned tasks are harder to be generated, resulting in the quality of pseudo data being worse than real data. The pseudo samples can be judged to belong to which tasks based on their contexts and questions. By way of judgment, it can be found that, in pseudo samples, the number of each task is the long-tail distribution which means that the earlier the task is learned, the fewer pseudo samples are generated. The model is easy to forget earlier learned tasks when jointly learning the long-tailed [13] distributed pseudo data with the new task. Then, as the number of learned tasks increases, the cycle of long-tail distributed pseudo data generation and joint training without limits leads to worse catastrophic forgetting.

In this paper, we propose Metacognitive Adaptation (Metac-Adapt) for making the model generate higher quality pseudo data to improve LAMOL almost without additional time cost and computational resources. Metacognition [14]–[16], which aims to help the learner study better through conscious supervision, control and regulation, has been researched in the field of education and psychology. Inspired by metacognition, to bias the model towards better semantic space for generating pseudo samples, Metac-Adapt adapts the model with a mini subset of previous tasks' questions before generating.

The contributions of our paper are listed below: (1) We proposed Metac-Adapt to alleviate catastrophic forgetting almost without additional time cost and computational resources. (2) We analyzed which type of pseudo samples is better for LLL. (3) We validated the effectiveness of Metac-Adapt is origin from improving the quality of pseudo samples.

2. Methodology

2.1 LAMOL

LAMOL [4] is a lifelong language model that applies a single GPT-2 [17] to learn various tasks in a stream by the language modeling (LM) task and question-answering (QA) task. From the second task on, before learning a new task,

Manuscript received July 29, 2022.
Manuscript publicized October 6, 2022.

[†]The authors are with the Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China.
[††]The authors are also with the University of Chinese Academy of Sciences, Beijing, China.
a) E-mail: wanghan2018@mail.ioa.ac.cn
b) E-mail: zhangxuejun@hccl.ioa.ac.cn

DOI: 10.1587/transinf.2022EDL8062

pseudo samples of previous tasks are generated and then jointly trained with a new task. In LAMOL, all task data are formatted in QA-style. Each sample consists of a beginning token (B), context (C), question (Q), and answer (A). There are two choices for the B: the task-specific token and the task-independent token. The LM task is to input "B" and then output "CQA". The QA task is to input "CQ" and then output "A". Let $T = \{T_1, \ldots, T_i, \ldots, T_N\}$ denote a stream including N tasks. Before learning T_i ($i > 1$), the model generates $\gamma|T_i|$ pseudo samples $\mathcal{P}_i$ for previous tasks, where γ is sampling ratio. When applying the task-specific token, there are $\frac{\gamma}{i-1}|T_i|$ for each previous task. Therefore, the objective can be calculated as Eq. (1).

$$\mathcal{L}(\mathcal{D}_i, \mathcal{P}_i)_{base} = \mathcal{L}(\mathcal{D}_i, \mathcal{P}_i)_{QA} + \lambda \mathcal{L}(\mathcal{D}_i, \mathcal{P}_i)_{LM} \tag{1}$$

$$\mathcal{L}(\mathcal{D}_i, \mathcal{P}_i)_{QA} = \sum_{t=|CQ|}^{|CQA|} \log P(x_t | x_{<t}) \tag{2}$$

$$\mathcal{L}(\mathcal{D}_i, \mathcal{P}_i)_{LM} = \sum_{t=|B|}^{|BCQA|} \log P(x_t | x_{<t}) \tag{3}$$

where $\mathcal{D}_i$ denotes the dataset of T_i, λ denotes the weight of the LM task, x_t denotes the t-th words of the sample, and $x_{<t}$ denotes all words prior to x_t.

2.2 Metacognitive Adaptation

We propose Metacognitive Adaptation (Metac-Adapt) for generating pseudo samples that evenly cover learned tasks to alleviate catastrophic forgetting. Firstly, we analyze the types of pseudo samples and their effects during lifelong language learning. Let $X = \{s_1, \ldots, s_i \ldots, s_M\}$ denote a sample with M segments. In GPT-2, a sample can be factorized as the product of conditional probabilities [18]:

$$p(X) = \prod_{i=1}^{M} p(s_M | s_1, \ldots, s_{M-1}) \tag{4}$$

As described in Sect. 2.1, there are four segments {B, C, Q, A} in a sample. Therefore, the $p(BCQA)$ can also be calculated by Eq. (4). In $p(BCQA)$, it can be found that B and Q are bridged by C. Since GPT-2 is a uni-directional autoregressive language model, much more information about the B is diminished with the increase in length of the C when generating the Q. As a result, the Q is easy to not correspond to what the B represents the task. Therefore, all types of pseudo data are detailed below: (1) **Serious Non-conformity (SN)**: Both the C and the Q are not corresponding to the B. (2) **Question-based Non-conformity (QN)**: The C is corresponds to the B but the Q is not. (3) **Question-based Conformity (QC)**: The Q corresponds to the B but the C is not. (4) **Normal Conformity (NC)**: Both the C and the Q correspond to the B.

Based on the principle of question answering, the question is more important to judge which task a sample belongs to because a certain context can be asked various questions about tasks. Therefore, the model learning the SN and QN pseudo samples will suffer from catastrophic forgetting. Theoretically, the upper bound of learning NC pseudo samples is learning the real data of previous tasks. The QC pseudo samples can be regarded as data augmentation for what their B represents tasks. Therefore, learning both the NC and the QC pseudo data makes the model have the potential to outperform learning real data of previous tasks during LLL.

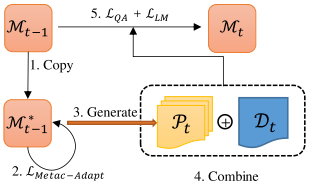

Fig. 1 The framework of lifelong language learning with Metac-Adapt.

However, during actual generation, the QC pseudo samples are hardly generated because both the contexts and questions are distinct for various tasks. For example, the contexts of WOZ are mainly about the restaurant search domain, but those of SST are mainly about movie reviews. There are significant differences between these two tasks in terms of words and speaking styles. Moreover, their contexts are strongly coupled to questions. Therefore, it is difficult to achieve forward transfer of knowledge between tasks with different styles to generate QC pseudo samples.

$$\mathcal{L}(\mathcal{D}_Q)_{Metac\text{-}Adapt} = \sum_{t=|B|}^{|BQ|} \log P(x_t | x_{<t}) \tag{5}$$

The objective of our Metac-Adapt is to make the model generate QC pseudo samples. Since a certain Q can be next to the ever-changing C, we just focus on generating the Q corresponding to what the B represents tasks. As shown in Eq. (5), Metac-Adapt is to learn how to generate the Q with the beginning of the B before generating all the pseudo samples of previous tasks. The process of Metac-Adapt is illustrated in Fig. 1 and detailed as follows:

(1) Judging the number of learned tasks $|T_{learned}| = n$. If $n > 0$, we sample m questions from the question database for each the learned task KB_Q to build a subset of learned task questions $\overline{KB}_Q$. $|\overline{KB}_Q| = n \times m$. Otherwise, train the model with the first task dataset.

(2) Concatenate the $\overline{KB}_Q$ with their corresponding B to obtain the question training set D_Q.

(3) Creating the model $\mathcal{M}^*_n$ by copying the learned n tasks model $\mathcal{M}_n$. Then training $\mathcal{M}^*_n$ with D_Q in k epochs.

(4) Applying $\mathcal{M}^*_n$ to generate pseudo samples $\mathcal{P}_{n+1}$ for n learned tasks.

(5) Training the $\mathcal{M}_n$ with the $n+1$-th task dataset $\mathcal{D}_{n+1}$ and $\mathcal{P}_{n+1}$ by the LM task and QA task.

3. Experimental Setting

3.1 Datasets and Implementation Details

For a fair comparison, we select three tasks from deca-

NLP [19] following previous LAMOL-based methods: (1) SST is a sentiment analysis dataset with two classes and is evaluated by the exact match (EM). (2) SRL is a semantic role labeling dataset and is evaluated by normalized F1 (nF1). (3) WOZ is a goal-oriented dialogue dataset and is evaluated by the turn-based dialogue state EM (dsEM).

All experiments are run on a single Tesla P100 (12 GB) five times and averaged. We implement our methods based on LAMOL[†]. We also select GPT-2 with 12 layers. The learning rate is 1e-4. All pseudo samples are generated with greedy decoding. Each task order is trained with 9 epochs. Other hyper-parameters are the same as LAMOL. All baselines are implemented based on the open-source code in their papers.

3.2 Baselines

LAMOL [4]: Apply a single language model trained with the LM task and QA task. LAMOL_T and LAMOL_G denotes the LAMOL applying task-specific token and task-independent token as the beginning token, respectively. LAMOL_R denotes that the LAMOL samples real data of previous tasks instead of generating pseudo data. **L2KD** [6]: Distill the output layer of GPT-2 in word-level or seq-level to improve LAMOL. L2KD applies the task-specific token as the beginning token. We use L2KD as the representative of the distillation method for improving LAMOL. **Adaptive Compositional Modules (ACM)** [11]: ACM adaptively adds new adapters and composes both old and new modules for new tasks. We use ACM as the representative of the adapter-based method for improving LAMOL. **Mulit-task**: Jointly train all tasks with the QA task.

$\text{Metac-Adapt}_T^{m\times k}$ and $\text{Metac-Adapt}_G^{m\times k}$ are used to represent Metac-Adapt applying task-specific token and task-independent token as the beginning token, respectively. The superscript denotes that Metac-Adapt samples m questions for each previous task and then adapt the model k epochs.

4. Experimental Results

4.1 Experiments on Three Tasks

To validate the effectiveness of our proposed Metac-Adapt, following LAMOL, we run experiments on all permutations of three decaNLP tasks. Each permutation is evaluated on three tasks after learning the third task and then gaining the average score of the three tasks. The final score of a method is the average and standard deviation (std.) of all permutations. As demonstrated in LAMOL, the performance is positively related to the value of sampling ratio γ but the gain disappears when $\gamma > 0.3$. We choose $\gamma \in \{0.2, 0.05\}$ as pseudo samples are sufficient and insufficient where $\gamma = 0.2$ is the best setting in previous LAMOL-based methods.

As shown in Table 1, no matter whether the pseudo samples are sufficient or insufficient, Metac-Adapt not only beats all previous LAMOL-based methods but also outperforms LAMOL_R which replays real data of previous tasks.

[†]https://github.com/jojotenya/LAMOL

Table 1 The summary results on [SST, SRL, and WOZ]. Average and Std. mean the average score and the standard deviation on six permutations of three tasks, respectively. The Time is the summary of training time on six permutations.

Methods	$\gamma = 0.2$			$\gamma = 0.05$	
	Average	Std.	Time (min.)	Average	Std.
LAMOL_T	79.5	0.5	802	76.0	1.6
LAMOL_G	79.7	0.8	801	74.9	3.7
LAMOL_R	81.0	0.5	797	79.4	1.3
L2KD	79.9	0.3	1632	77.0	1.9
ACM	78.9	1.3	1345	77.9	1.1
$\text{Metac-Adapt}_T^{8\times1}$	**81.6**	0.5	797	79.2	1.0
$\text{Metac-Adapt}_T^{8\times5}$	81.2	**0.3**	797	**79.6**	**0.6**
$\text{Metac-Adapt}_G^{8\times1}$	80.8	0.6	799	76.5	2.6
$\text{Metac-Adapt}_G^{8\times5}$	80.7	1.1	799	77.1	2.6
Multi-task	81.5				

When $\gamma = 0.2$, $\text{Metac-Adapt}_T^{8\times1}$ outperforms LAMOL_T by 2.1 percentage points. $\text{Metac-Adapt}_G^{8\times1}$ has an improvement of 1.1 percentage points over LAMOL_G while the std. decreases 0.4. $\text{Metac-Adapt}_T^{8\times1}$ is 1.7 percentage points higher than L2KD, indicating that Metac-Adapt is better than the distillation method. $\text{Metac-Adapt}_{T/G}^{8\times1}$ is better than $\text{Metac-Adapt}_{T/G}^{8\times5}$. It demonstrates that Metac-Adapt is a training-efficient method where the model can be biased to a better semantic space for generating pseudo data with only training one epoch. We counted the total training time of each method on all permutations. As shown in Table 1, the training time required by other LAMOL-based baselines is 1.68–2.04 times that of LAMOL, but Meta-Adapt is nearly the same as LAMOL_R. Metac-Adapt takes less time than LAMOL_T and LAMOL_G. We analyze this because Metac-Adapt can generate better pseudo-data from which models can learn faster and better.

When $\gamma = 0.05$, the improvement of Metac-Adapt is more significant. $\text{Metac-Adapt}_T^{8\times5}$ is 3.6 percentage points higher than LAMOL_T while the std. decreases by 1.0. $\text{Meta-Adapt}_G^{8\times5}$ is 2.2 percentage points higher than LAMOL_G while the std. decrease 1.1. $\text{Metac-Adapt}_G^{8\times5}$ outperforms L2KD and ACM by 2.6 and 1.7 percentage points, respectively, indicating that $\text{Meta-Adapt}_G^{8\times5}$ can alleviate catastrophic forgetting better than distillation and adapter-based methods when pseudo samples are insufficient.

The above are the results of each model with replaying pseudo data. Since Metac-Adapt aims to generate and replay better pseudo data instead of real data, we compare it with LAMOL_R. When $\gamma = 0.2$, $\text{Metac-Adapt}_T^{8\times1}$ is 0.6 percentage points higher than LAMOL_R. When $\gamma = 0.05$, $\text{Meta-Adapt}_T^{8\times5}$ outperforms LAMOL_R by 0.2 percentage points while the std. decreases by 0.7. It demonstrates that our proposed Meta-Adapt surpasses LAMOL_R in terms of performance and is more robust for learning order.

Notably, when $\gamma = 0.2$, $\text{Metac-Adapt}_T^{8\times1}$ is higher than MTL by 0.1 percentage points, indicating that Metac-Adapt has the potential to apply LLL to real-world scenarios.

4.2 The Quality of Pseudo Samples

In this section, we analyze whether the improvement of Metac-Adapt comes from generating better pseudo samples. We apply the CPS [8] to evaluate the quality of pseudo samples. The CPS is a BLEU-based method that calculates sample-averaged BLEU scores between the pseudo samples and the training dataset of each learned task to obtain the distribution of knowledge of learned tasks, and then calculates the Jensen-Shannon divergence between the BLEU distribution and the uniform distribution. As described in [8], the lower the value of CPS represents better pseudo samples, which are beneficial to prevent catastrophic forgetting. CPS-n represents the quality of pseudo-data generated after learning n tasks.

In order to correspond with the results in Table 1 and analyze the influence of different tasks, we calculate CPS-2 on three task orders where the first two tasks are different (SST-SRL-WOZ (Order I), SRL-SST-WOZ (Order III), WOZ-SST-SRL (Order IV)). As shown in Table 2, Metac-Adapt generates better pseudo data than other LAMOL-based baselines, which indicates that the gain of Metac-Adapt really comes from improving the quality of pseudo data. In addition, it can also be found that using the task-specific token is better than using the task-independent token, which is also in line with the performance of the corresponding method in Table 1.

When $\gamma = 0.05$, compared with LAMOL$_T$ and LAMOL$_G$, Metac-Adapt$_T$ and Metac-Adapt$_G$ can reduce CPS-2 by 30.4–95.8% and 0.8–12.8%, respectively. In the results of SST-SRL-WOZ, the CPS-2 of Metac-Adapt$_G$ is higher than LAMOL$_G$, but the performance is better. This opposite result is because both BLEU$_{SST}$ = 0.1704 and BLEU$_{SRL}$ = 0.264 in the pseudo samples generated by LAMOL$_G$ are greatly reduced, but in Metac-Adapt$_G$ only BLEU$_{SST}$ = 0.111 dropped significantly while BLEU$_{SRL}$ = 0.4358 remained high. When $\gamma = 0.2$, compared with LAMOL$_T$ and LAMOL$_G$, Metac-Adapt$_T$ and Metac-Adapt$_G$ can reduce CPS-2 by 41.9–63.6% and 38.8–85.5%, respectively.

Table 2 The CPS-2 results on task orders: SST-SRL-WOZ (Order I), SRL-SST-WOZ (Order III), WOZ-SST-SRL (Order IV).

γ	Methods	Order I	Order III	Order IV
0.05	LAMOL$_T$	0.0203	0.1303	0.1110
	L2KD	0.1110	0.0805	0.0080
	Metac-Adapt$_T^{8\times1}$	0.0043	0.0908	0.0067
	Metac-Adapt$_T^{8\times5}$	**0.0038**	**0.0587**	**0.0047**
	LAMOL$_G$	**0.0236**	0.6682	0.5258
	Metac-Adapt$_G^{8\times1}$	0.1253	0.8016	0.5218
	Metac-Adapt$_G^{8\times5}$	0.2029	**0.5828**	**0.4651**
0.2	LAMOL$_T$	0.0613	0.0647	**0.0109**
	L2KD	0.1317	0.0511	0.0223
	Metac-Adapt$_T^{8\times1}$	0.0330	**0.0355**	0.0172
	Metac-Adapt$_T^{8\times5}$	**0.0224**	0.0375	0.0112
	LAMOL$_G$	0.1163	0.2898	0.1641
	Metac-Adapt$_G^{8\times1}$	**0.0168**	**0.1557**	0.0841
	Metac-Adapt$_G^{8\times5}$	0.0293	0.1774	**0.0808**

5. Conclusion

In this paper, we proposed Metac-Adapt to alleviate catastrophic forgetting during LLL almost without additional time cost and computational resources. Before generating pseudo samples of previously learned tasks, we adapt the model with a mini-subset of previous tasks' questions towards better semantic space for generating. Experimental results demonstrate that Metac-Adapt is on par with MTL or even slightly higher, indicating that Metac-Adapt has the potential to be applied to real-world scenarios. In future work, we will explore longer task orders that cover more fields of NLP.

References

[1] G.I. Parisi, R. Kemker, J.L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," Neural Networks, vol.113, pp.54–71, 2019.
[2] R. Polikar, L. Upda, S.S. Upda, and V.G. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," IEEE Trans. Syst. Man Cybern. Part C, vol.31, no.4, pp.497–508, 2001.
[3] R.M. French, "Catastrophic forgetting in connectionist networks," Trends in Cognitive Sciences, vol.3, no.4, pp.128–135, 1999.
[4] F.K. Sun, C.H. Ho, and H.Y. Lee, "Lamol: Language modeling for lifelong language learning," Proc. ICLR 2020, 2019.
[5] Z. Chen and B. Liu, Lifelong Machine Learning, Second Edition, Morgan & Claypool Publishers, 2018.
[6] Y.-S. Chuang, S.-Y. Su, and Y.-N. Chen, "Lifelong language knowledge distillation," Proc. EMNLP, pp.2914–2924, 2020.
[7] J. Sun, S. Wang, J. Zhang, and C. Zong, "Distill and replay for continual language learning," Proc. COLING, pp.3569–3579, 2020.
[8] H. Wang, R. Fu, C. Li, X. Zhang, J. Zhou, X. Bai, Y. Yan, and Q. Zhao, "Reminding the incremental language model via data-free self-distillation," Applied Intelligence, 2022.
[9] K. Kanwatchara, T. Horsuwan, P. Lertvittayakumjorn, B. Kijsirikul, and P. Vateekul, "Rational LAMOL: A rationale-based lifelong learning framework," Proc. ACL, pp.2942–2953, 2021.
[10] A. Madotto, Z. Lin, Z. Zhou, S. Moon, P. Crook, B. Liu, Z. Yu, E. Cho, P. Fung, and Z. Wang, "Continual learning in task-oriented dialogue systems," Proc. EMNLP, pp.7452–7467, 2021.
[11] Y. Zhang, X. Wang, and D. Yang, "Continual sequence generation with adaptive compositional modules," Proc. ACL, pp.3653–3667, 2022.
[12] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," Proc. ICML, 2019.
[13] X. Wang, L. Lian, Z. Miao, Z. Liu, and S.X. Yu, "Long-tailed recognition by routing diverse distribution-aware experts," Proc. ICLR, 2021.
[14] J.H. Flavell, "Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry," American Psychologist, vol.34, no.10, pp.906–911, 1979.
[15] L. Bowler, "Talk as a metacognitive strategy during the information search process of adolescents," Inf. Res., vol.15, no.4, paper 449, 2010.
[16] D.W. Braithwaite and L. Sprague, "Conceptual knowledge, procedural knowledge, and metacognition in routine and nonroutine problem solving," Cogn. Sci., vol.45, no.10, e13048, 2021.
[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever,

"Language models are unsupervised multitask learners," OpenAI blog, 2019.

[18] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," J. Mach. Learn. Res., vol.3, pp.1137–1155, 2003.

[19] B. McCann, N.S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," arXiv preprint arXiv:1806.08730, 2018.