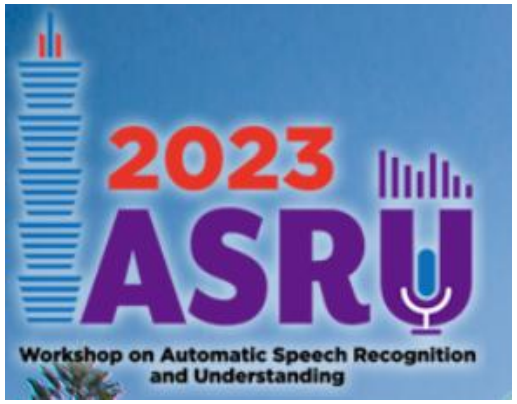


Prompt Pool Based Class-Incremental Continual Learning For Dialog State Tracking

Hong Liu[1,3,†], *Yucheng Cai*[1,3,†], *Yuan Zhou* [1,3,†], *Zhijian Ou*[*,1,3], *Yi Huang*[2,3], *Junlan Feng*[2,3]

[1]Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, Beijing, China
[2]China Mobile Research Institute, Beijing, China
[3]Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute, Beijing, China

Outline

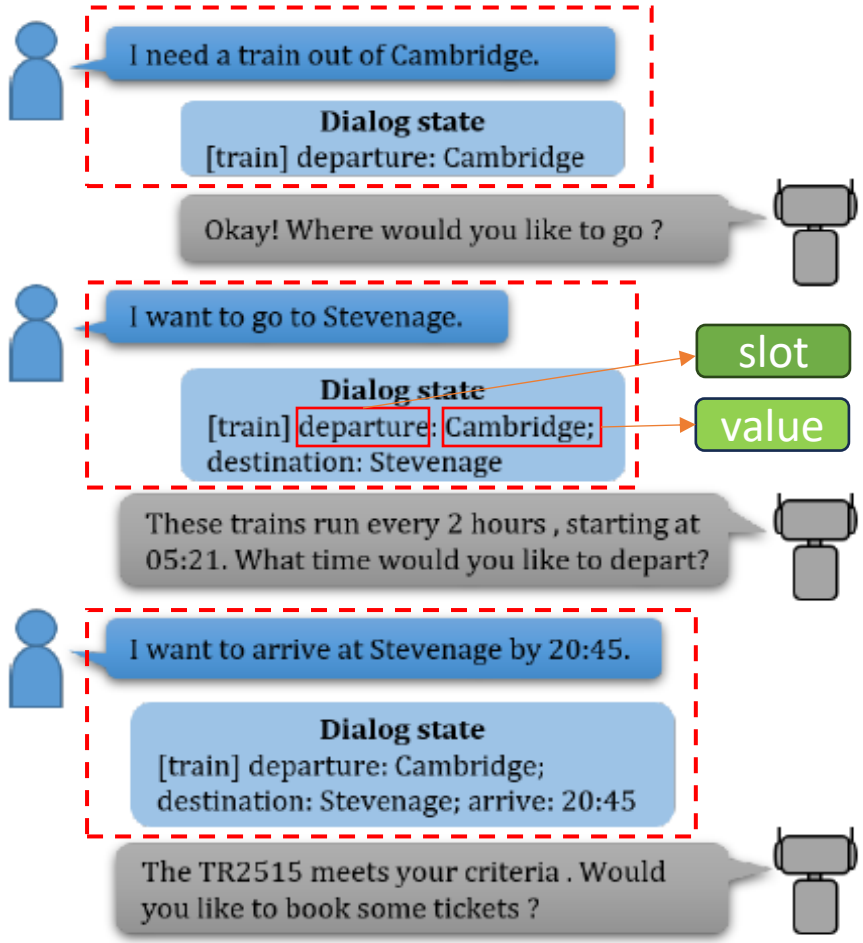
- **Introduction**
- **Methods**
- **Experiments**
- **Conclusions**

Dialog State Tracking and Continual Learning

- Dialog state tracking (DST) in task-oriented system

- The tasks are varying. It is costly if the DST model is re-trained when each new task is added.

- Continual learning is necessary. Continual learning refers to expanding a model to new tasks efficiently without catastrophic forgetting.

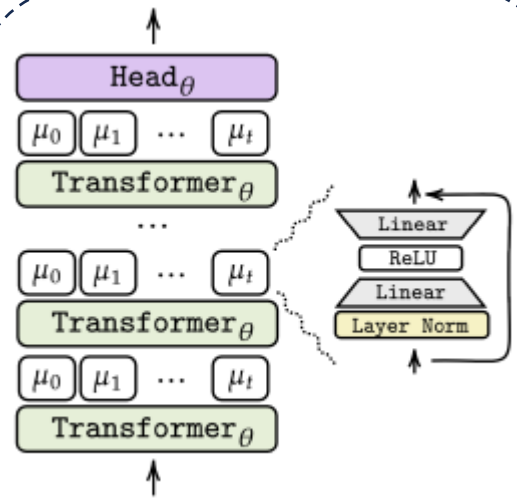


An example of a task oriented dialog

Categories of Continual Learning

➢ Continual learning can be divided into different categories according to training methods and test scenarios

Training Method	Details	Advantage	Deficiency
Rehearsal-based	Use a replay buffer to recall previous learned task	Easy to implement	The performance will decrease when the buffer size is reduced
Regularization-based	Adds regularization term to the loss to avoid forgetting	No need for additional memory or parameters	fail to achieve desirable performance
Architecture-based	Trains task-specific component for each task	More flexible and efficient	Most of these methods need to know the task identity during testing



AdapterCL [1], one of the architecture-based method, which train a separate adapter for each task

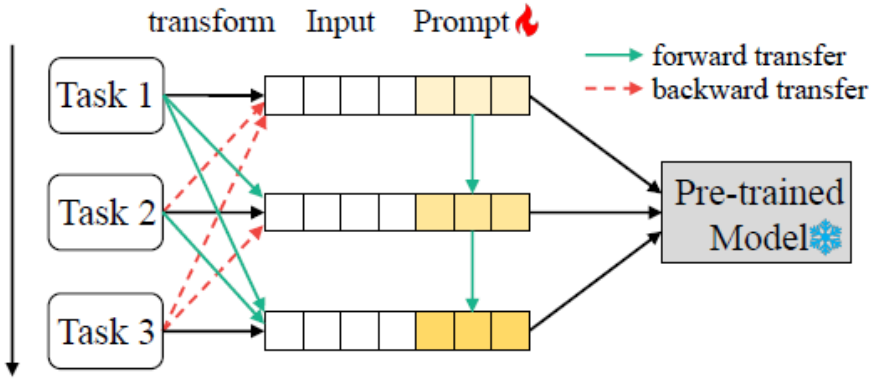
Test Scenarios	Explanation
Task-incremental	The task identity is known
Domain-incremental	The task identity is unknown, but the data labels remain consistent across tasks, e.g., all tasks are binary classification.
Class-incremental	The task identity is unknown (most challenge)

[1] Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang, "Continual learning in task-oriented dialogue systems," in EMNLP2021

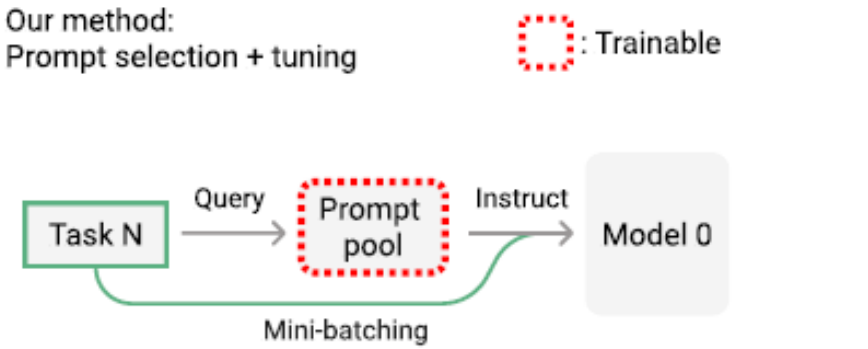
Our approach

- Architecture-based method for the most challenging class-incremental scenarios.
- Most prevalent method: prompt tuning (cannot work in the class-incremental)

- Prompt pool based method for DST



Continual prompt tuning [2]



Prompt pool based method [3]. It automatically selects prompts from the pool, without the need for task identity

[2] Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang, "Continual prompt tuning for dialog state tracking," in ACL2022
[3] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister, "Learning to prompt for continual learning," in CVPR2022.

Outline

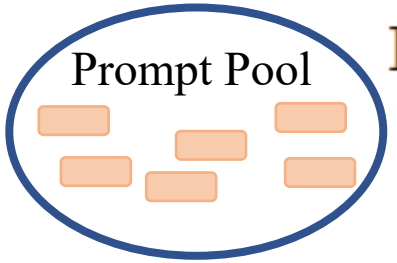
- **Introduction**
- **Methods**
- **Experiments**
- **Conclusions**

Notation

- A sequence of tasks $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$
- The corresponding datasets are $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$
- Data samples $(x_t, y_t)_{t=1,\ldots,T}$
  - x_t : the dialog history (the concatenation of dialog turns up to now)
  - y_t : the dialog state containing slot value pairs, in sequence form

seq2seq modeling using T5

- Prompt pool



$\mathbf{P} = \{P_1, P_2, \ldots, P_J\}$

J is the pool size

$P_j \in \mathbb{R}^{L_p \times D}$ is a single prompt with token length L_p and embedding size D

- The input vectors for the model:

Dialog history embedding | P_{t_1} | ... | P_{t_N}

Prompt Selection

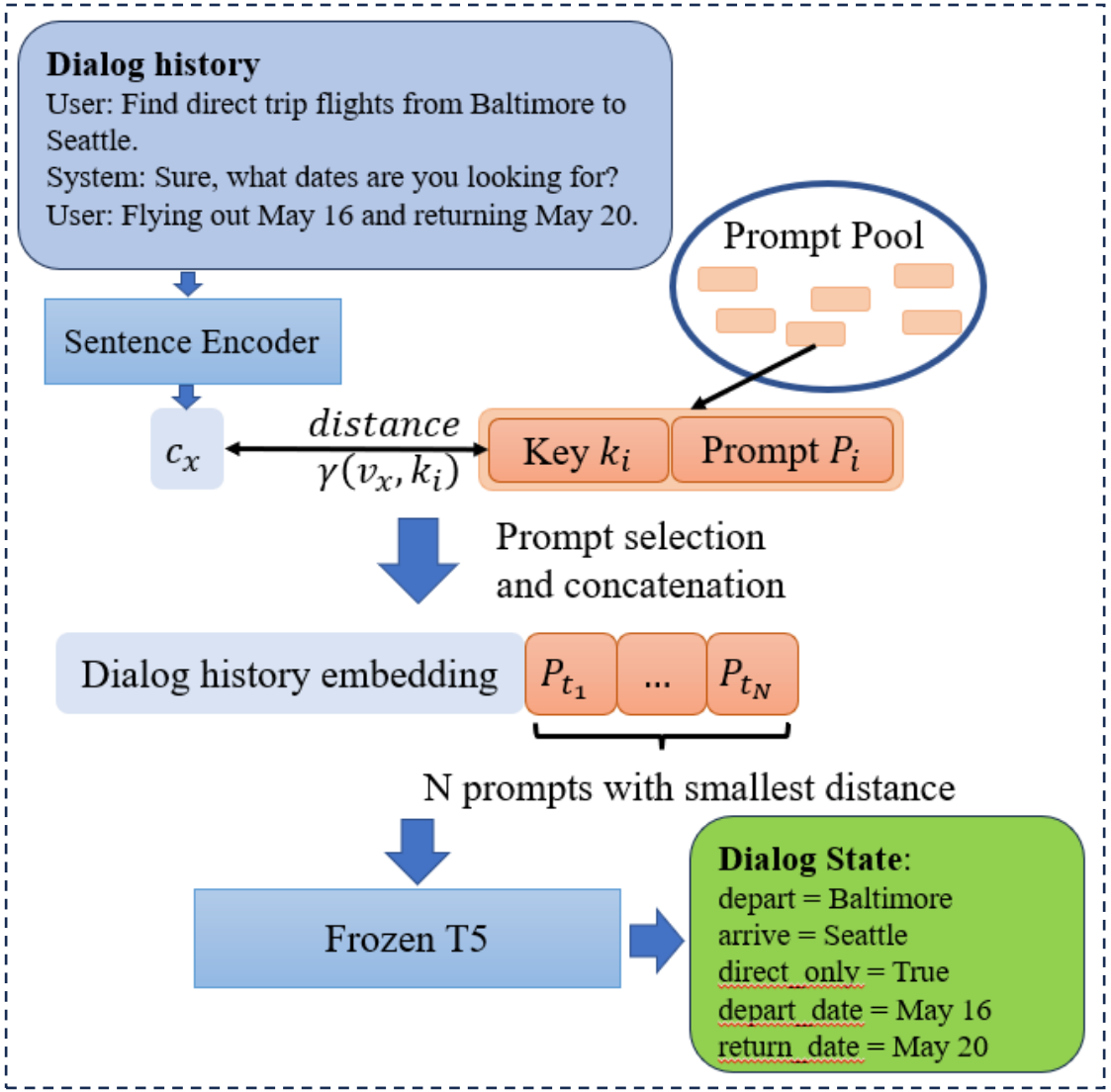
- How to select prompt from the pool if the task identity is unknown during testing?

➢Assign a key for each prompt

$$\{P_1, P_2, ..., P_J\} \longrightarrow \{(k_1, P_1), ..., (k_J, P_J)\}$$

➢Select N prompts with the smallest distance between the key and the context vector of the input sequence x_t

➢During training, we do not select. Instead, we directly use the $k_{Nt:N(t+1)-1}$ for task $\mathcal{T}_t$ to ensure that each key in the pool can be trained



Optimization

- The loss can be divided into two parts: the cross entropy loss between the model output and the label; the Euclid distance between the context vector and the selected prompt keys.

$$\mathcal{L} = CE(f_\theta(E_p(x)), y) + \lambda \sum_{k_i \in \mathbf{K}_x} \gamma(c_x, k_i)$$

- Utilize rehearsal buffer to improve model performance when past data are accessible.
  - The dataset of the t-th task $\mathcal{D}_t \rightarrow \mathcal{D}_t \cup M_{<t}$

The buffered data from previous task

  - The training loss need to be modified for data samples from previous tasks

$$\mathcal{L} = CE(f_\theta(E_p(x)), y) + \lambda \sum_{k_i \in \mathbf{K}_x} BCE(\gamma(c_x, k_i), \mathbf{I}(x \in \mathcal{D}_t))$$

Add an indicator to distinguish the data from current task and that from previous tasks

Outline

- **Introduction**
- **Methods**
- **Experiments**
- **Conclusions**

Settings

- Datasets
  - Schema-Guided Dialog (SGD) [4]. 44 services over 19 domains. We randomly select 15 tasks and split the dialogs of one service into train/val/test sets with the ratio of 7:1:2.
  - China Mobile Pickup dataset (CM-Pickup), collected from a real-world dialog application. The purpose is to automatically pickup the incoming call when the phone owner is not available. CM-Pickup has 39 domains and we retain 16 domains that have more than 100 dialog sessions.
- Metrics
  - Average Joint Goal Accuracy (JGA) for DST on all the tasks after continual learning
  - Key Selection Accuracy (Acc_{key}) during testing.
- Baselines
  - AdapterCL: trains a residual adapter for each task and select the adapter with lowest perplexity during testing
  - Upper bound 1: multitask prompt tuning (MPT), trains N prompts using all tasks' data simultaneously
  - Upper bound 2: oracle continual prompt tuning (OCPT), trains N prompts for each task sequentially but the task identity is provided during testing

[4] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan, "Towards scalable multi-domain conversational agents: The schemaguided dialogue dataset," in AAAI2020.

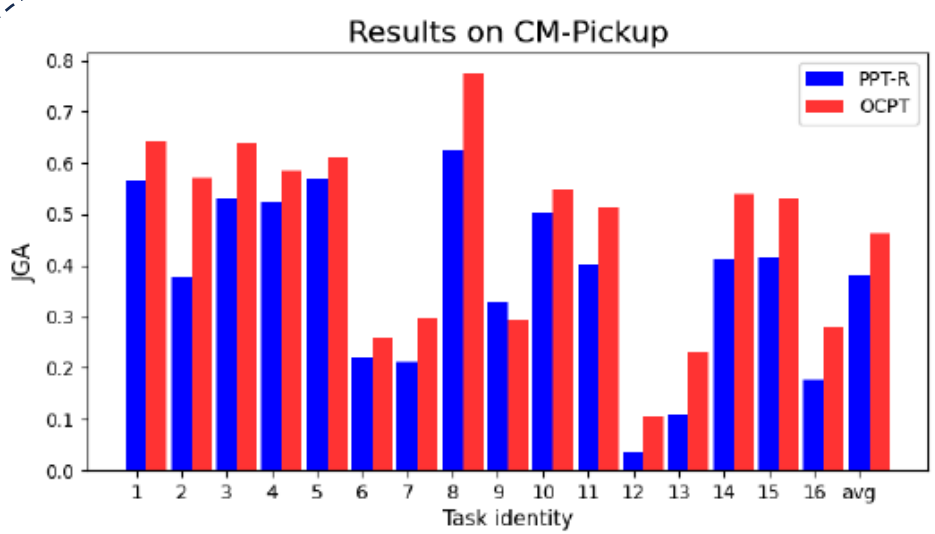
Results

- Results on SGD dataset
  - ➢The prompt pool training (PPT) method is much better than the AdapterCL baseline
  - ➢PPT with rehearsal buffer (PPT-R) outperforms PPT
  - ➢Both PPT and PPT-R are lower than the upper bound, indicating the challenge of class-incremental learning.

Method	JGA_{avg}	Acc_{key}
OCPT	0.481	-
MPT	0.614	-
AdapterCL	0.306	-
PPT	0.346	0.783
PPT-R	**0.363**	**0.811**

Average joint goal accuracy on 15 tasks over SGD

- Results on CM-pickup dataset
  - ➢The overall results are similar to those on SGD, where PPT-R achieves slightly lower JGA than the upper-bound OOCPT.



The joint goal accuracy of PPT-R and OCPT (upperbound) on each task of CM-Pickup.

Analysis

- The comparison of PPT and PPT-R: results on each task after continual learning
  - The JGA and Acc_{key} of PPT-R are higher than PPT on most tasks
  - There are some exceptions: flights_3、retalcars_3. The JGA and Acc_{key} decrease significantly after adding a rehearsal buffer, why?

The model cannot distinguish between two similar tasks such as flights_1 and flights_3, leading to errors in predicting the task identity of previous tasks

Task name	PPT		PPT-R	
	JGA	Acc_{key}	JGA	Acc_{key}
services_4	0.510	0.793	0.538	0.981
flights_1	0.537	0.741	0.563	0.802
services_3	0.534	1.000	0.555	0.990
flights_3	0.422	0.922	0.353	0.836
trains_1	0.368	0.983	0.419	1.000
homes_2	0.144	0.311	0.230	0.547
rentalcars_2	0.086	0.459	0.416	0.989
restaurants_1	0.497	1.000	0.473	1.000
music_1	0.324	1.000	0.211	0.951
hotels_4	0.014	0.014	0.255	0.355
media_2	0.254	1.000	0.338	0.972
hotels_3	0.264	0.808	0.228	0.829
rentalcars_3	0.263	0.828	0.020	0.263
hotels_1	0.352	0.880	0.256	0.720
homes_1	0.628	1.000	0.589	0.930
avg	0.346	0.783	0.363	0.811

The detailed metrics on 15 tasks of SGD after continual learning

- The comparison of different model size
  - The larger the better

Model	JGA_{avg}	Acc_{key}
T5-small (60M)	0.346	0.783
T5-base (200M)	0.420	0.800
T5-large (750M)	**0.464**	**0.822**

Outline

- **Introduction**
- **Methods**
- **Experiments**
- **Conclusions**

Conclusion

- We propose to address the class-incremental learning problem of dialog state tracking (DST) using the prompt pool method.
- We maintain a prompt pool and select the prompts that are close to the input sequence during continual learning.
- We conduct experiments on SGD and CM-Pickup. The results show that the prompt pool method outperforms the baseline.
- We also combine prompt pool with a rehearsal buffer, which further improves the joint goal accuracy.

Thanks!