Domain Incremental Learning and Continual Learning in NLP

Zixuan Ke

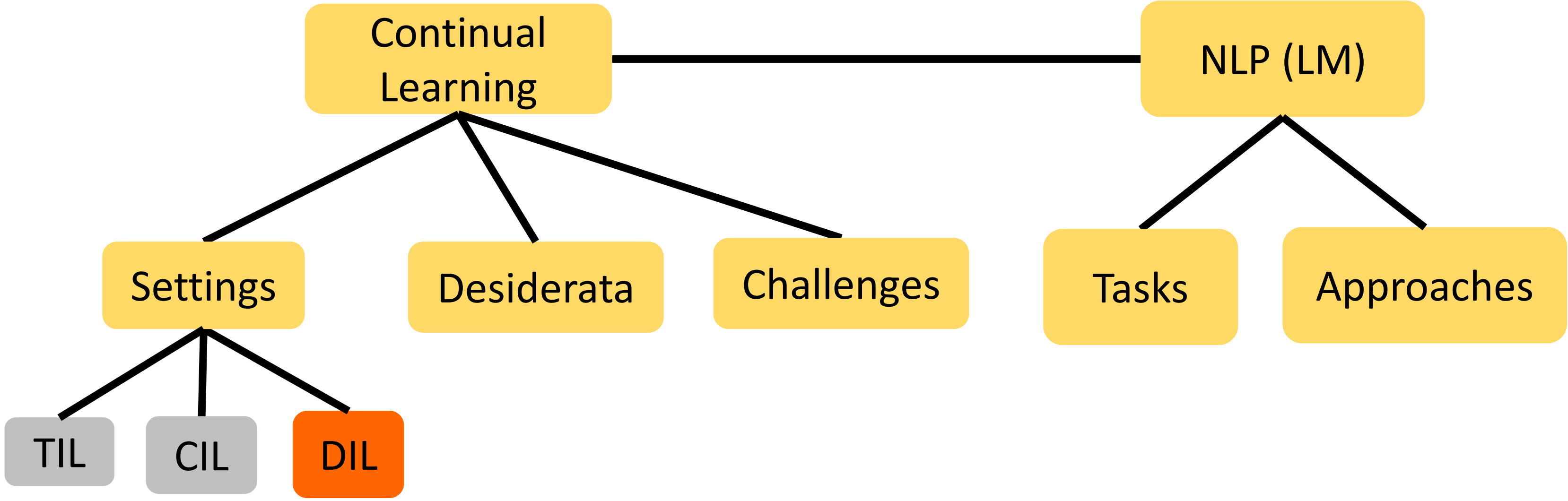
University of Illinois Chicago

https://vincent950129.github.io/

Plan

- A quick summary of TIL and CIL
- Another setting: Domain-incremental Learning
- What we have learned so far
- Continual learning of NLP Tasks
- Conclusion and Future work

Roadmap



*LM: Language Model

Settings

- We have known that

TIL

- **Assumption:** task-IDs are available in both training and testing

$$\mathcal{Y}_t \subseteq \mathcal{Y}$$

- **Goals:**

$$f : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{Y}$$

CIL

- **Assumption:** task-IDs are available only in training. In testing, a test instance from any class may be presented

$$\mathcal{Y}_t \cap \mathcal{Y}_{t'} = \varnothing, \forall\, t \neq t'$$

$$\mathcal{Y} = \bigcup_{t=1}^{T} \mathcal{Y}_t$$

- **Goals:** $f : \mathcal{X} \rightarrow \mathcal{Y}$

Notations:

- *T* is the last task learned and $\mathcal{T} = \{1...T\}$
- $\mathcal{Y}_t$ is the label space of task *t*
- For any given tasks, *t* and *t'*,

$$p(\mathcal{X}_t) \neq p(\mathcal{X}_{t'}), \forall\, t \neq t'$$

Plan

- A quick summary of TIL and CIL
- Another setting: Domain-incremental Learning
- What we have learned so far
- Continual learning of NLP Tasks
- Conclusion and Future work

Domain-incremental Learning

- Assumption
  - Class labels are assumed to be the same for all tasks
- E.g.,
  - A sequence of **sentiment classification tasks** (all tasks have the same labels {*positive*, *negative*, *neutral*})
  - **Generation tasks** in NLP (all tasks depend on the language model head which has the same number of vocabulary tokens)

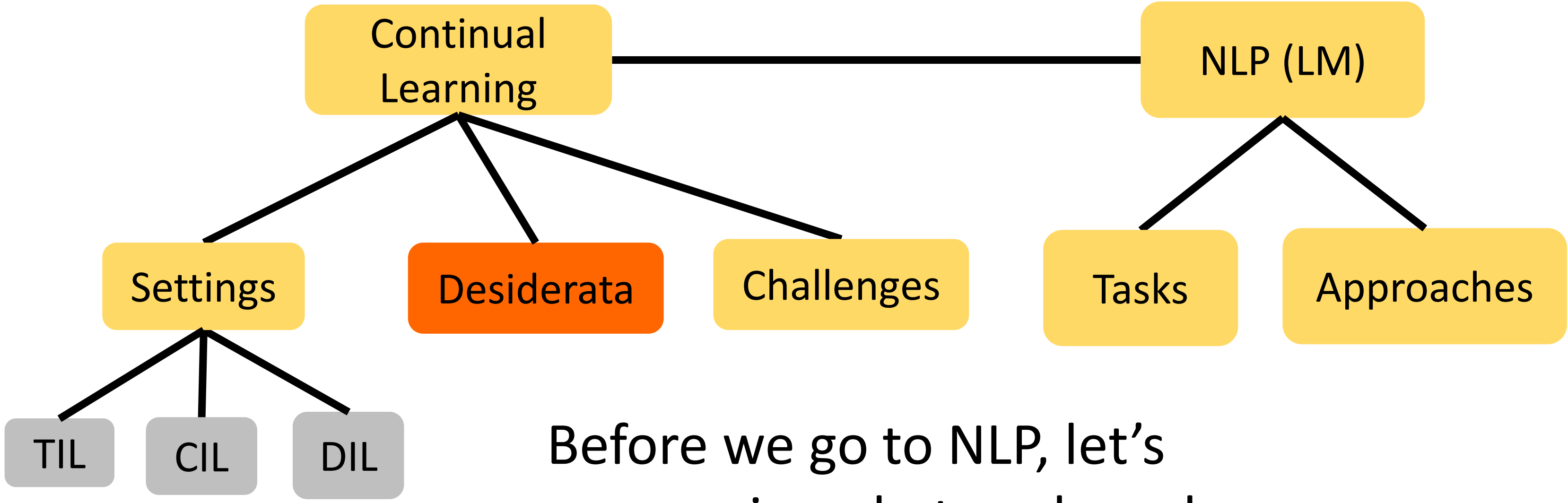
Domain-incremental Learning

- Goal
  - Learn a function $f : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{Y}$
- Yes, it is **a special case of TIL**
  - A DIL problem can always be solved with a TIL method
- However, you may see some existing DIL systems do not use task-ID in testing,
  - This is because the tasks are very similar (task-ID does not matter) or very dissimilar (task-ID is easily predicted using the test data)

Domain-incremental Learning

- Now we know all the 3 settings, why are they important?
  - Together with TIL and CIL, they constitute **the three fundamental settings** of continual learning.
  - When attempting to leverage continual learning, **the initial step** is to establish the appropriate setting.
  - Different settings are for different applications and lead to **different specific challenges and approaches**

Roadmap



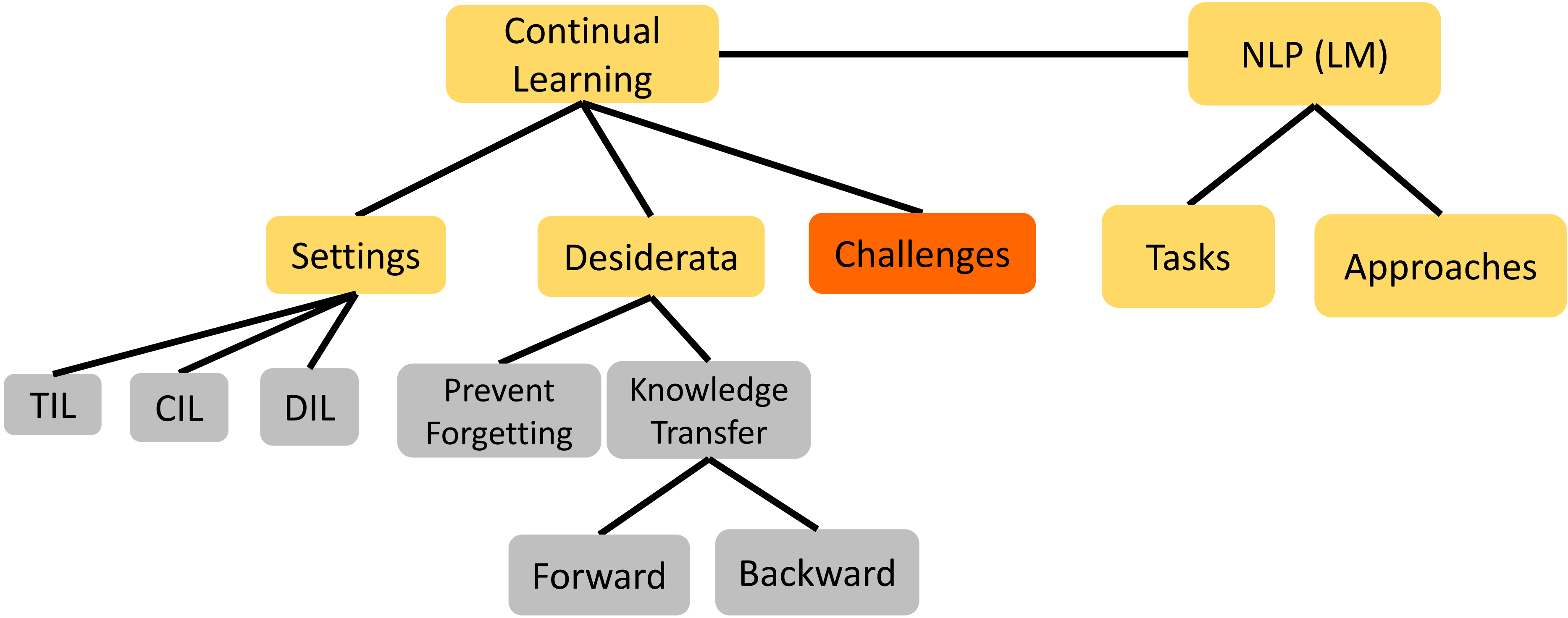
Before we go to NLP, let’s summarize what we have known about continual learning **so far**

*LM: Language Model

Continual learning

- General Desiderata (TIL, DIL, CIL)
  - Not suffer from **catastrophic forgetting** (CF)
    - i.e., perform reasonably well on what has been learned
  - Achieve **positive forward knowledge transfer** (forward KT)
    - i.e., old knowledge helps new task
  - Achieve **positive backward knowledge transfer** (backward KT)
    - i.e., relevant new task helps old tasks

Roadmap



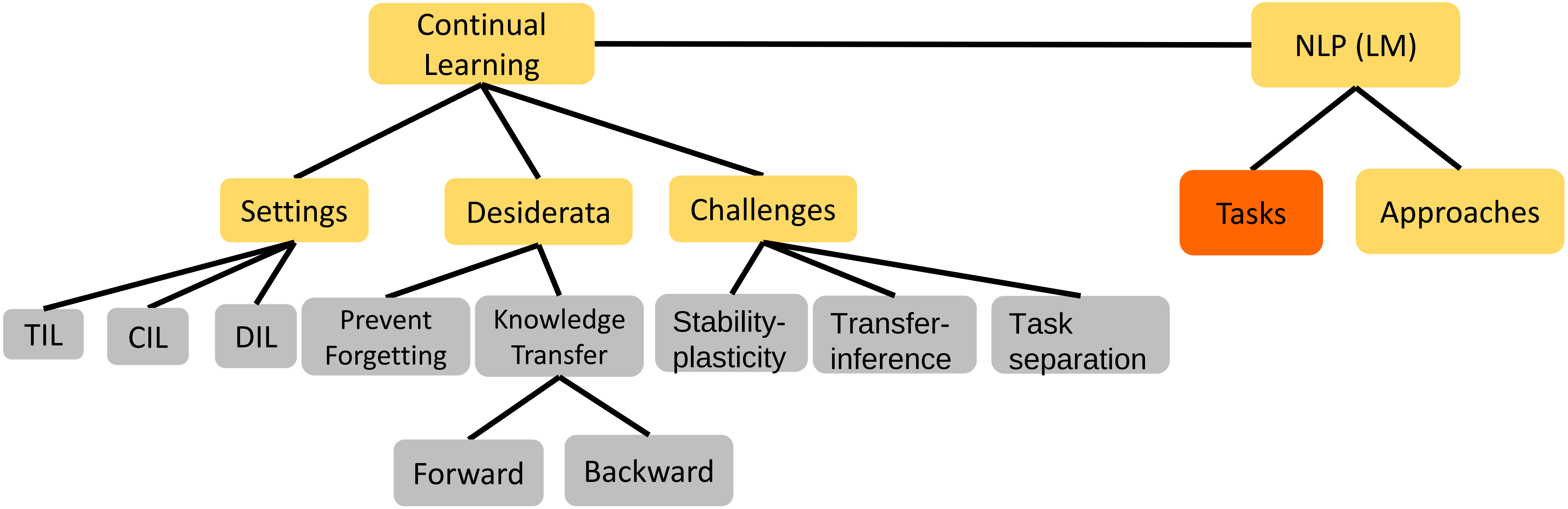
*LM: Language Model

Continual learning

- Challenges
  - **Stability-plasticity** (TIL, CIL, DIL)
    - Preserving the learned knowledge v.s. accurately learning new experiences
  - **Transfer-interference** (mostly in TIL and DIL)
    - Knowledge transfer vs. knowledge interference
    - (increase parameter-sharing vs. reduce parameter-sharing)
  - **Task separation** (mostly in CIL, and DIL w/o ID)
    - Establish decision boundaries between the current task and previous tasks
      - In learning the current task, the learner does not see previous or future task data

*Task separation a.k.a. inter-task class separation in CIL

Roadmap



*LM: Language Model

Plan

- A quick summary of TIL and CIL
- Another setting: Domain-incremental Learning
- What we have learned so far
- Continual learning of NLP Tasks
- Conclusion and Future work

Continual Learning in NLP

- **The world is dynamic**, and time is irreversible, new emerging *{classes/domains/tasks/topics/events}* keep coming

who will win the football world cup

I'm sorry, but I don't have the ability to browse the internet and don't have the information about the upcoming football world cup. My training only goes up until 2021, so I am not able to provide current information on any sports events. Is there something else I can help you with?

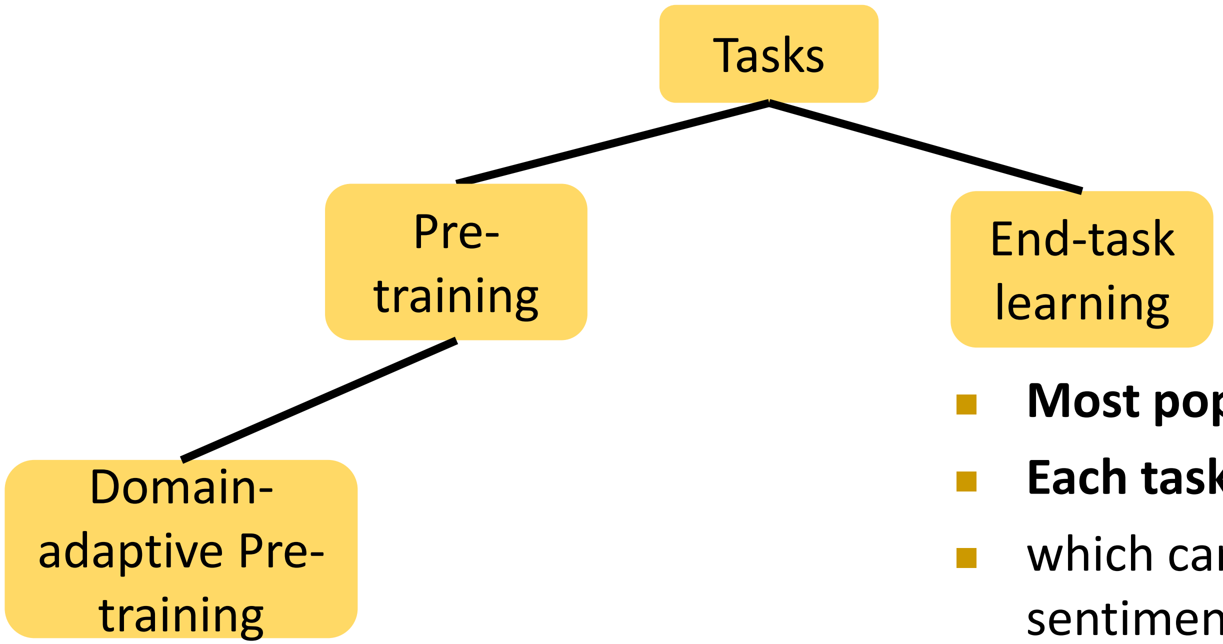
Continual Learning in NLP

- **The world is dynamic**, and time is irreversible, new emerging *{classes/domains/tasks}* keep coming
- We do not want to **re-train** for many reasons
- We do not want to train **separate** models for many reasons
- We want the model **accumulates knowledge** in its lifetime without forgetting what it has learned

Continual Learning in NLP

- “Task”
  - In terms of NLP, what is a task?
- Formulation
  - How are some NLP tasks learned continually?
- Approaches
  - What are some popular approaches in CL for NLP?

Continual Learning in NLP



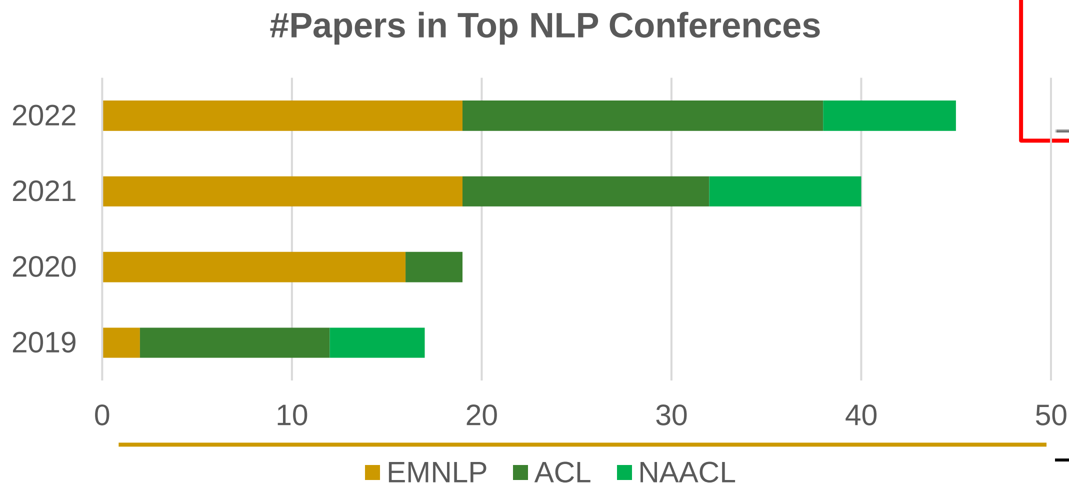
- **Each task uses a domain corpus** to adapt an LM to a domain/topic

- **Most popular**
- **Each task is an end-task**
- which can be any NLP problem (e.g., sentiment classification, NER, summarization)

*Domain-adaptive pre-training (DAPT) a.k.a. post-training or pre-finetuning

Formulation of NLP Problems

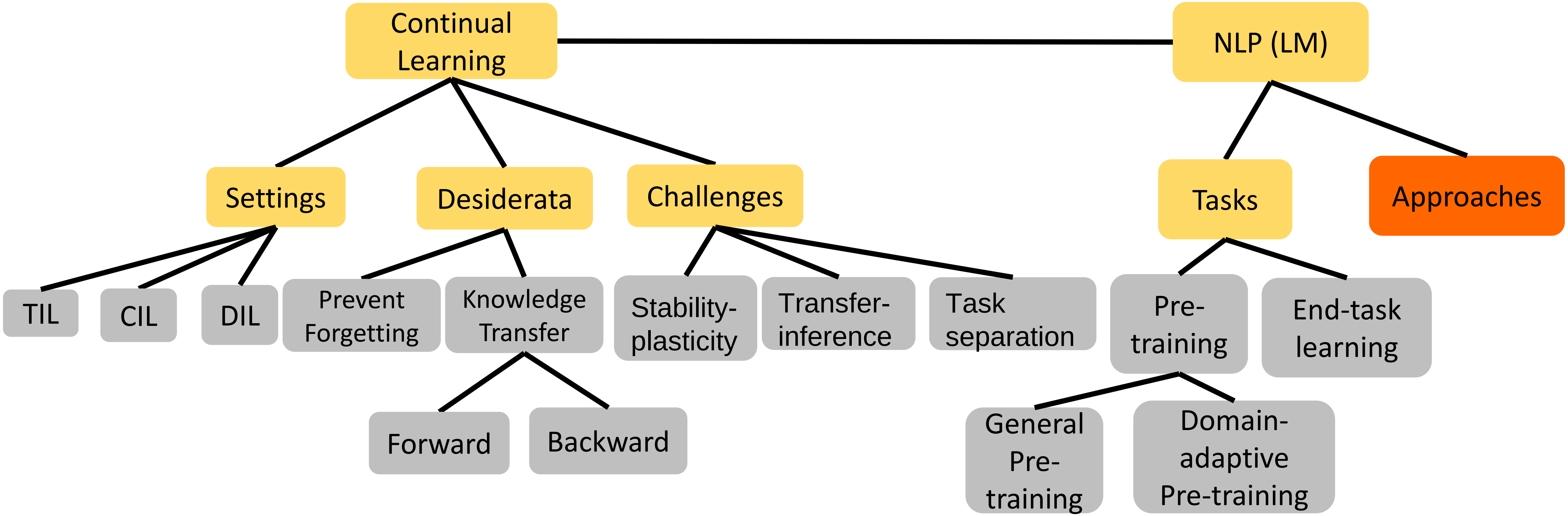
- Large #NLP problems belong to **DIL**
  - Many problems are converted to generation tasks in NLP via prompting
- Continual learning for NLP has been growing **rapidly**, this table keeps expanding.



CL Settings	Type	NLP Problems	CL Papers
TIL	End-task	Aspect sentiment classification	B-CL (Ke et al., 2021c) CTR (Ke et al., 2021a)
		Intent classification	MeLL (Wang et al., 2021a) PCLL (Zhao et al., 2022)
		Slot filling	PCLL (Zhao et al., 2022)
		Topic classification	CTR (Ke et al., 2021a)
		Mixed diverse tasks	CLIF (Jin et al., 2021)
DIL (w/o ID)	End-task	Mixed 5 classification tasks	LFPT5 (Qin and Joty, 2022)
		Named-entity recognition	LFPT5 (Qin and Joty, 2022)
		Summerization	LFPT5 (Qin and Joty, 2022)
		Paraphrase	RMR-DSE (Li et al., 2022a)
		Dialogue response generation	RMR-DSE (Li et al., 2022a) AdapterCL (Madotto et al., 2020)
		Dialogue state tracking	AdapterCL (Madotto et al., 2020)
		Dialogue end2end	AdapterCL (Madotto et al., 2020)
		Aspect sentiment classification	CLASSIC (Ke et al., 2021b)
		Question answering	MBPA++ (de Masson d'Autume et al., 2019) Meta-MBPA++ (Wang et al., 2020)
	DAPT	5 pre-training domains	ELLE (Qin et al., 2022)
		8 pre-training domains	DEMIX (Gururangan et al., 2021)
		10 pre-training domains	Continual-T0 (Scialom et al., 2022)
DIL (w/ ID)	End-task	Mixed 5 classification tasks	LAMOL (Sun et al., 2020)
		Mixed classification and labeling tasks	LAMOL (Sun et al., 2020)
		Dialogue state tracking	C-PT (Zhu et al., 2022)
		Dialogue response generation	TPEM (Geng et al., 2021)
		Mixed classification and generation	ConTinTin (Yin et al., 2022)
		Mixed 4 generation tasks	ACM (Zhang et al., 2022)
	DAPT	5 pre-training domains	CPT (Ke et al., 2022)
CIL	End-task	Named-entity recognition	ExtendNER (Monaikul et al., 2021)
		Intent classification	CID (Liu et al., 2021) PAGeR (Varshney et al., 2022)
		Mixed 5 classification tasks	IDBR (Huang et al., 2021)
		Slot filling	ProgM (Shen et al., 2019)
		Sentence representation	SRC (Liu et al., 2019a)
		Mixed 5 classification tasks	MBPA++ (de Masson d'Autume et al., 2019) Meta-MBPA++ (Wang et al., 2020)
	Few-shot	Intent classification	ENTAILMENT (Xia et al., 2021) CFID (Li et al., 2022b)

*DAPT: Domain-adaptive pre-training

Roadmap



*LM: Language Model

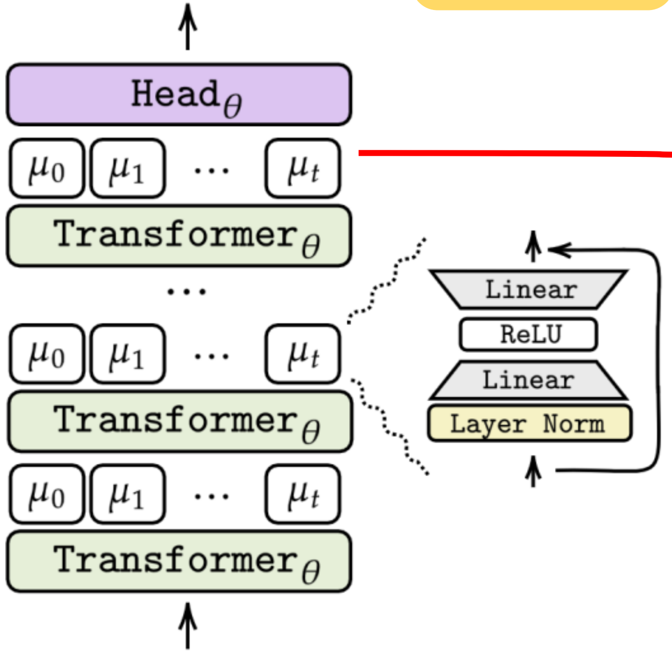
Approaches for Forgetting Prevention

- We have known the three popular families in continual learning to **prevent forgetting**
  - **Add penalty** when training new task — Regularization-based
  - **Memorize a subset** of old samples and train together with the new task — Replay-based
  - **Allocate specific parameters** to specific tasks — Parameter-isolation

Approaches for Forgetting Prevention

- A simple example in NLP: Adapter-CL

DIL | End-task learning | Parameter-isolation



$\boldsymbol{\mu}_t$ refers to the adapter for task t

Adapter-CL **uses separate adapters for different tasks,** which is a naïve way to prevent forgetting.

It can do well in **forgetting prevention**, but cannot achieve the other desideratum (**knowledge transfer**)

***adapter**: Adding a small number of parameters to the pre-trained LM. In fine-tuning, only the adapters are trainable.

Madotto et al., Continual learning in task-oriented dialogue systems, EMNLP 2020

Approaches for Knowledge Transfer

- Approaches for preventing forgetting mainly try to **reduce parameter-sharing**
- This is not enough for **knowledge transfer**, which needs to allow some used parameters to be updated or shared.

  - Compute task similarity and share the parameters for similar tasks — Similarity-based[1]
  - Learn the transferrable knowledge using replay data (e.g., meta-learning) — Replay-based[2]
  - Soft-masking the backward pass, but keep the forward pass untouched — Soft-mask-based

*Same name, but with different goals

[1]: Ke et al., Achieving forgetting prevention and knowledge transfer in continual learning, NeuIPS 2021

[2]: d'Autume et al., Episodic Memory in Lifelong Language Learning, NeurIPS 2019

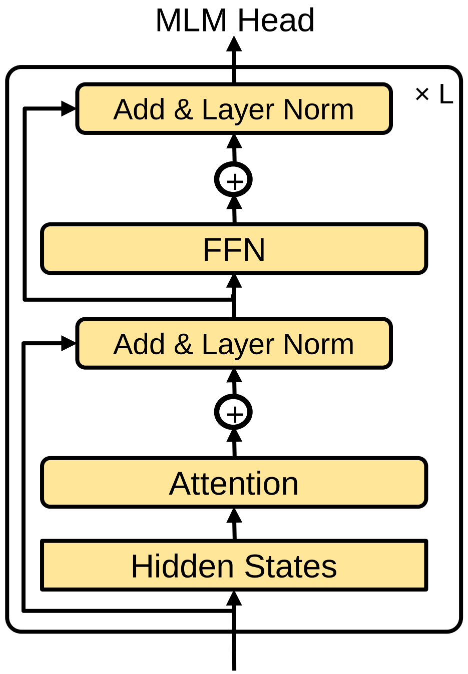
Continual Domain-adaptive Pre-training

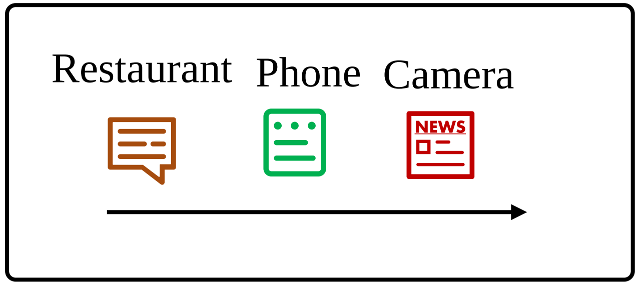
DIL | Domain-adaptive Pre-training | Soft-mask-based

- Setting
  - Continual domain-adaptive pre-training of a sequence of domains **without** accessing the data that was used in original **pre-training or previously learned domains**
  - End-task doesn't know its domain belonging
- Goals
  - CF prevention
  - KT (backward and forward)
- Approach
  - **DAS**

Ke et al., Continual learning of language models, ICLR 2023

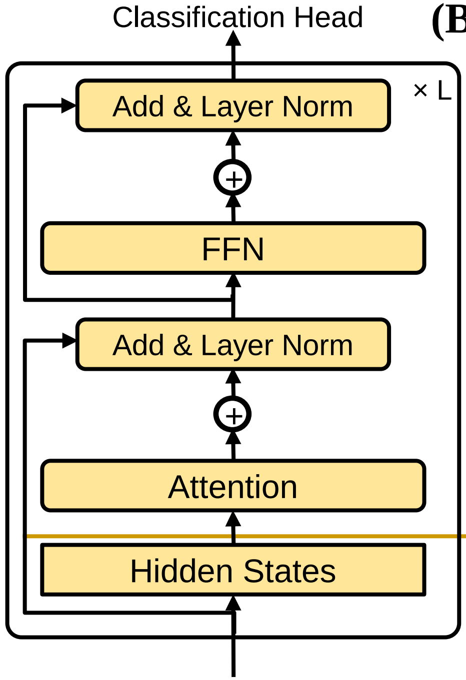
(A) Continual Domain-adaptive pre-training





Given a pre-trained LM, continually domain-adaptive pre-training **a sequence of domains**

(B) Individual Fine-tuning



End-tasks

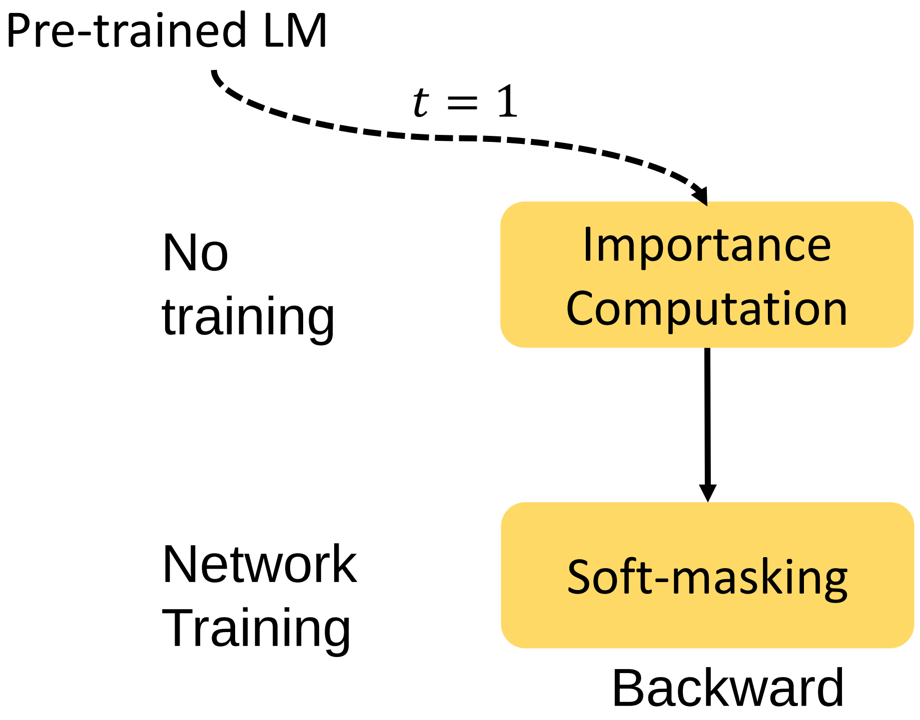
- ASC-Restaurant
- ASC-Phone
- ASC-Camera

After continual learning, the domain-adaptive pre-training performance is **evaluated** by end-tasks

Each end-task **corresponding** to one domain and has its **own** training and testing set. It is trained individually and **will not** affect the domain-adaptive pre-training

ASC: Aspect Sentiment Classification

Continual Domain-adaptive Pre-training



Key ideas:

1) Detect importance of units for general and domain knowledge

2) Soft-mask the important units when training new tasks/domains

3) This can prevent forgetting and allow knowledge transfer

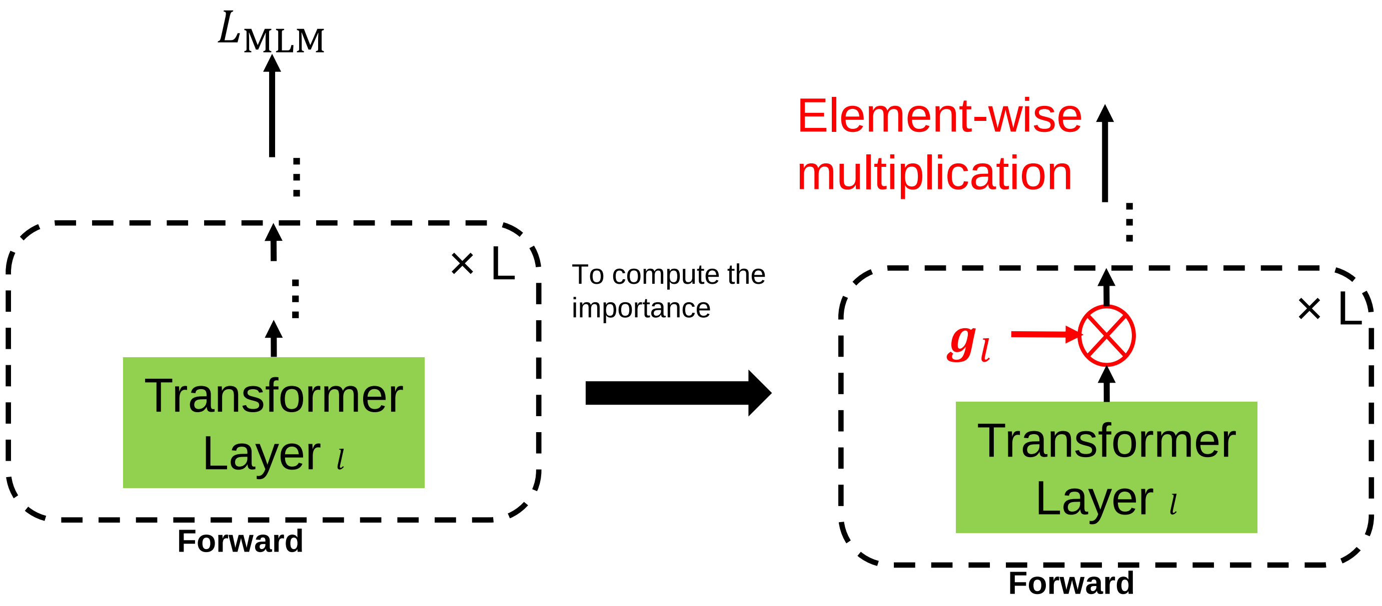
Key challenges:

1) How to detect importance for the two types of knowledge

2) How to soft-mask

Ke et al., Continual learning of language models, ICLR 2023

Importance Computation

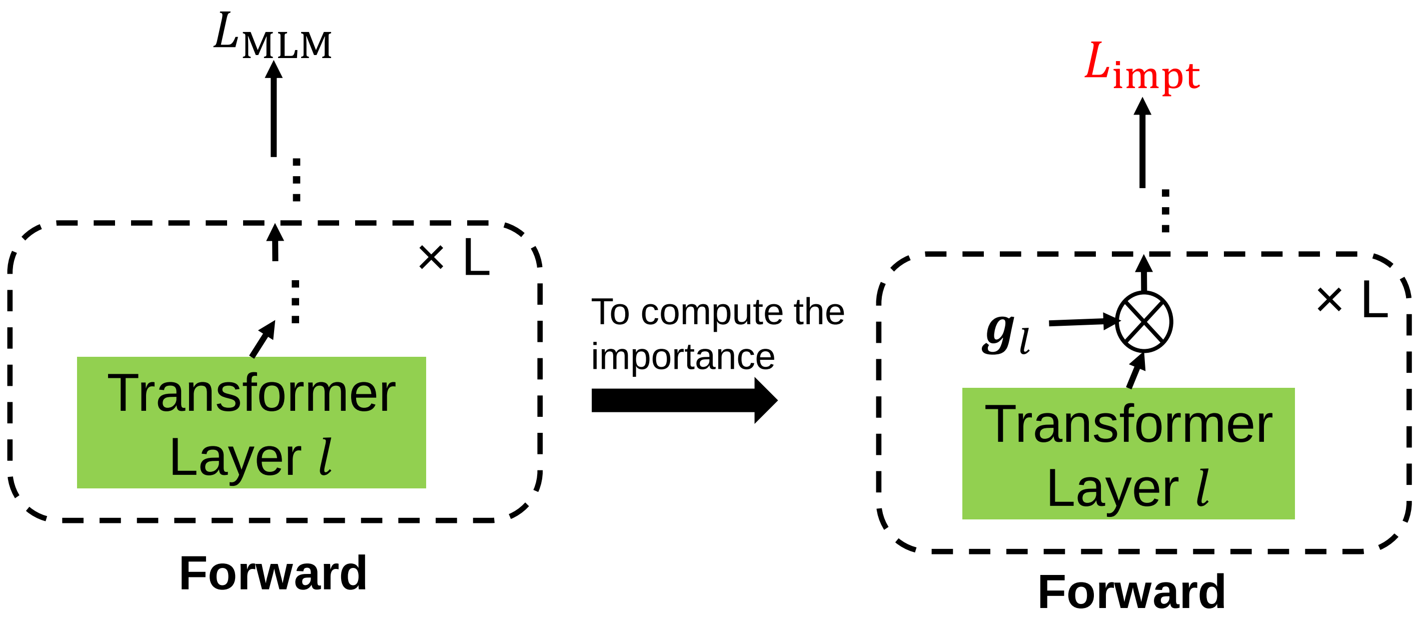


$\boldsymbol{g}_l$ is the **virtual parameters.** Each virtual parameter $g_{l,i}$ in $\boldsymbol{g}_l$ corresponding to an attention head or neurons (units)

It is **initialized as all 1's**, and has its gradient but will **never change**.

Why? We only use its gradient to compute importance

Importance Computation



For **domain knowledge,**

$L_{\text{impt}} = L_{\text{MLM}}$

$$\boldsymbol{\nabla}_{\boldsymbol{g}_l}^{m} = \frac{\partial L_{\text{impt}}(\boldsymbol{x}_m^{(t)}, \boldsymbol{y}_m^{(t)})}{\partial \boldsymbol{g}_l}$$

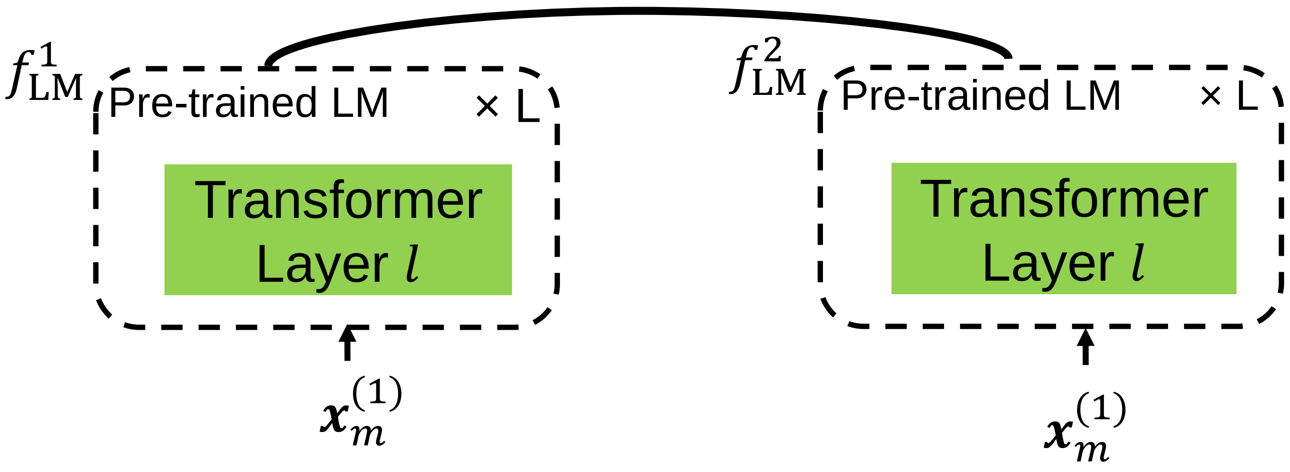
$$\boldsymbol{I}_l^{(t)} = \frac{1}{M} \sum_M |\boldsymbol{\nabla}_{\boldsymbol{g}_l}^{m}|$$

Use **absolute gradient** to indicate importance[1]

Michel et al. Are sixteen heads really better than one? NeurIPS, 2019.

Importance Computation

$$L_{\text{impt}} = \mathrm{KL}\left(f_{\text{LM}}^{1}(\boldsymbol{x}_m^{(1)}),\ f_{\text{LM}}^{2}(\boldsymbol{x}_m^{(1)})\right)$$



KL: How different are the two representations?

$\boldsymbol{f_{LM}^{1}}$/ $\boldsymbol{f_{LM}^{2}}$: Transformer with different dropouts

$\boldsymbol{x}_m^{(1)}$: We only use **the first domain** data because we want to keep the pre-trained general knowledge

With the new L_{impt}, we can use the absolute gradient to indicate the importance (same as in domain knowledge)

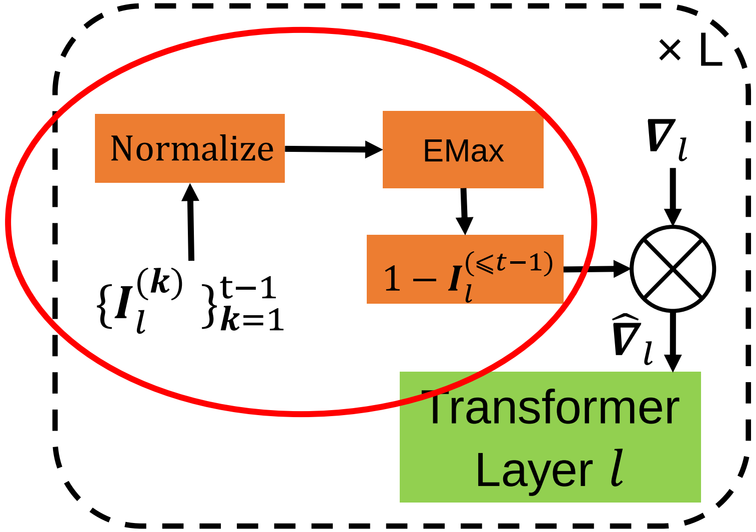
For **general knowledge**, we leverage the **random dropout** in standard Transformer

Random dropout introduces **random noise**. Given the **same input**, the difference between the representations with different random noise indicates the **robustness**.

The units that are important to the robustness is likely to be important to the **general/pre-trained knowledge** because its change will **cause the pre-trained LM** change a great deal

Ke et al., Continual learning of language models, ICLR 2023

Soft-masking



Backward

First, we normalize the importance so that they are comparable

$$\boldsymbol{I}_l^{(k)} = \text{Tanh}(\text{Norm}(\boldsymbol{I}_l^{(k)}))$$

Second, we accumulate the importance

$$\boldsymbol{I}_l^{(\leqslant t-1)} = \text{EMax}(\{\boldsymbol{I}_l^{(t-1)}, \boldsymbol{I}_l^{(t-2)}\})$$

Third, we soft-mask the gradient (only in backward pass)

$$\boldsymbol{\nabla}'_l = \left(1 - \boldsymbol{I}_l^{(\leqslant t-1)}\right) \otimes \boldsymbol{\nabla}_l$$

Ke et al., Continual learning of language models, ICLR 2023

Results

Overall end-task performance (final performance)

No DAPT

DAPT

NCL DAPT

SoTA DAPT

Category	Domain / Model	Restaurant MF1	Restaurant Acc	ACL MF1	ACL Acc	AI MF1	AI Acc	Phone MF1	Phone Acc	PubMed MF1	Camera MF1	Camera Acc	Average MF1	Average Acc	Forget R. MF1	Forget R. Acc
Non-CL	RoBERTa	79.81	87.00	66.11	71.26	60.98	71.85	83.75	86.08	72.38	78.82	87.03	73.64	79.27	—	
	DAPT (RoBERTa)	80.84	**87.68**	68.75	73.44	68.97	75.95	82.59	85.50	72.84	84.39	89.90	76.40	80.89	—	
	DAPT (Adapter)	80.19	87.14	68.87	72.92	60.55	71.38	82.71	85.35	71.68	83.62	89.23	74.60	79.62	—	
	DAPT (Prompt)	79.00	86.45	66.66	71.35	61.47	72.36	84.17	86.53	**73.09**	85.52	90.38	74.98	80.03	—	
CL Post-train	NCL	79.52	86.54	68.39	72.87	67.94	75.71	84.10	86.33	72.49	85.71	90.70	76.36	80.77	1.14	1.05
	NCL (Adapter)	80.13	87.05	67.39	72.30	57.71	69.87	83.32	85.86	72.07	83.70	89.71	74.05	79.48	0.15	-0.02
	DEMIX	79.99	87.12	68.46	72.73	63.35	72.86	78.07	82.42	71.73	86.59	91.12	74.70	79.66	0.74	0.36
	BCL	78.97	86.52	**70.71**	**74.58**	66.26	74.55	81.70	84.63	71.99	85.06	90.51	75.78	80.46	-0.06	-0.19
	CLASSIC	79.89	87.05	67.30	72.11	59.84	71.08	84.02	86.22	69.83	86.93	91.25	74.63	79.59	0.44	0.25
	KD	78.05	85.59	69.17	73.73	67.49	75.09	82.12	84.99	72.28	81.91	88.69	75.17	80.06	-0.07	0.01
	EWC	**80.98**	87.64	65.94	71.17	65.04	73.58	82.32	85.13	71.43	83.35	89.14	74.84	79.68	0.02	-0.01
	DER++	79.00	86.46	67.20	72.16	63.96	73.54	83.22	85.61	72.58	87.10	91.47	75.51	80.30	2.36	1.53
	HAT	76.42	85.16	60.70	68.79	47.37	65.69	72.33	79.13	69.97	74.04	85.14	66.80	75.65	-0.13	-0.29
	HAT-All	74.94	83.93	52.08	63.94	34.16	56.07	64.71	74.43	68.14	65.54	81.44	59.93	71.33	3.23	1.83
	HAT (Adapter)	79.29	86.70	68.25	72.87	64.84	73.67	81.44	84.56	71.61	82.37	89.27	74.63	79.78	-0.23	-0.18
	DAS	80.34	87.16	69.36	74.01	**70.93**	**77.46**	**85.99**	**87.70**	72.80	**88.16**	**92.30**	**77.93**	**81.91**	**-1.09**	**-0.60**

- w/o DAPT < DAPT < DAS
- +forgetting rate in NCL: it does suffer from forgetting
- Regularization-based methods (KD, EWC) and replay-based method (DER++) are all worse: focus on CF prevention is not enough
- Parameter-isolation method (HAT) preforms much worse: the full LM is needed for domain-adaptive pre-training
- Methods that tries to perform both KT and CF (DEMIX, BCL, CLASSIC): all weaker than DAS

Ke et al., Continual learning of language models, ICLR 2023

*Naïve continual learning (NCL): continual learning without any specific technique

Approaches for Task Separation

- Besides forgetting and knowledge transfer, another challenge in CIL and DIL w/o ID is the task separation

- Use heuristic methods like entropy/perplexity to detect task ID at test time

Post-prediction[1,2]

- Use replay data to establish the boundary

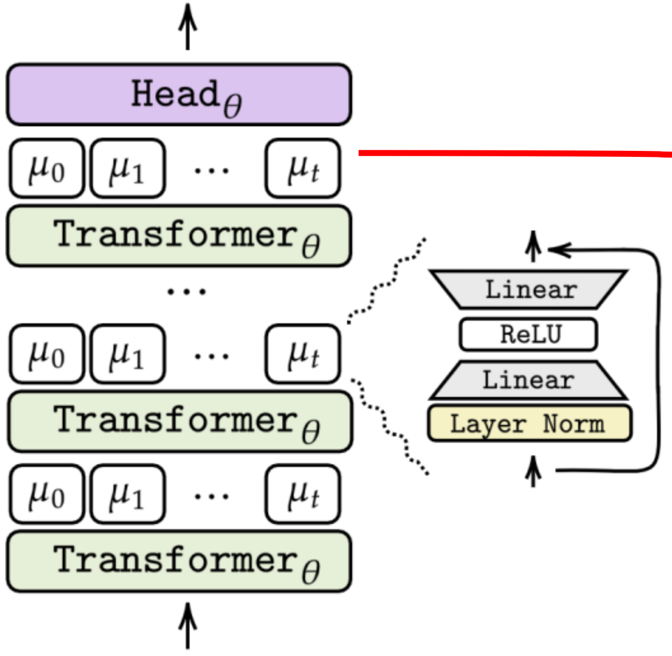
Replay-based

[1]: Gururangan et al., DEMix Layers: Disentangling Domains for Modular Language Modeling, NAACL 2022
[2]: Li et al., Overcoming Catastrophic Forgetting During Domain Adaptation of Seq2seq Language Generation, NAACL 2022

Approaches for Task Separation

DIL | End-task learning | Post-prediction

- A simple example in NLP: Adapter-CL



$\boldsymbol{\mu}_t$ refers to the adapter for task t

Adapter-CL uses separate adapters for different tasks, but how to know which one to use in testing?

Perplexity

$$\alpha_t = \mathrm{PPL}_{\mu_t}(X) \; \forall t \in 1, \cdots, N,$$

$$t^* = \operatorname{argmin} \alpha_0, \cdots, \alpha_N$$

Madotto et al., Continual learning in task-oriented dialogue systems, EMNLP 2020

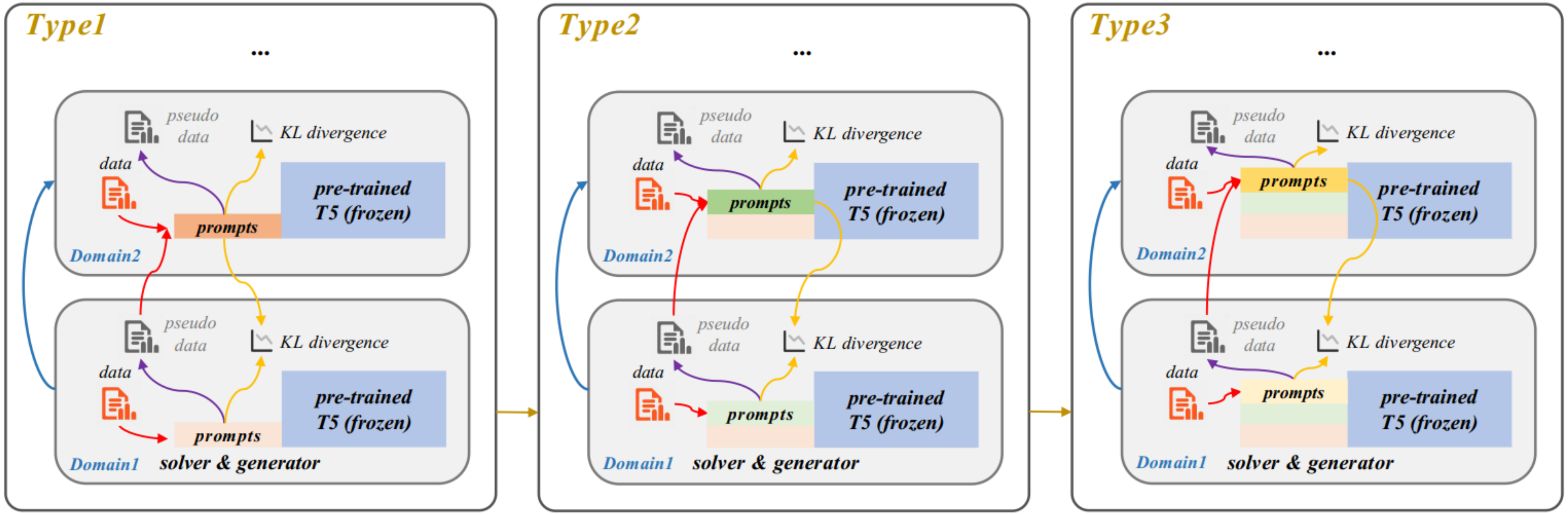
Approaches for Task Separation

DIL

End-task learning

Replay-based

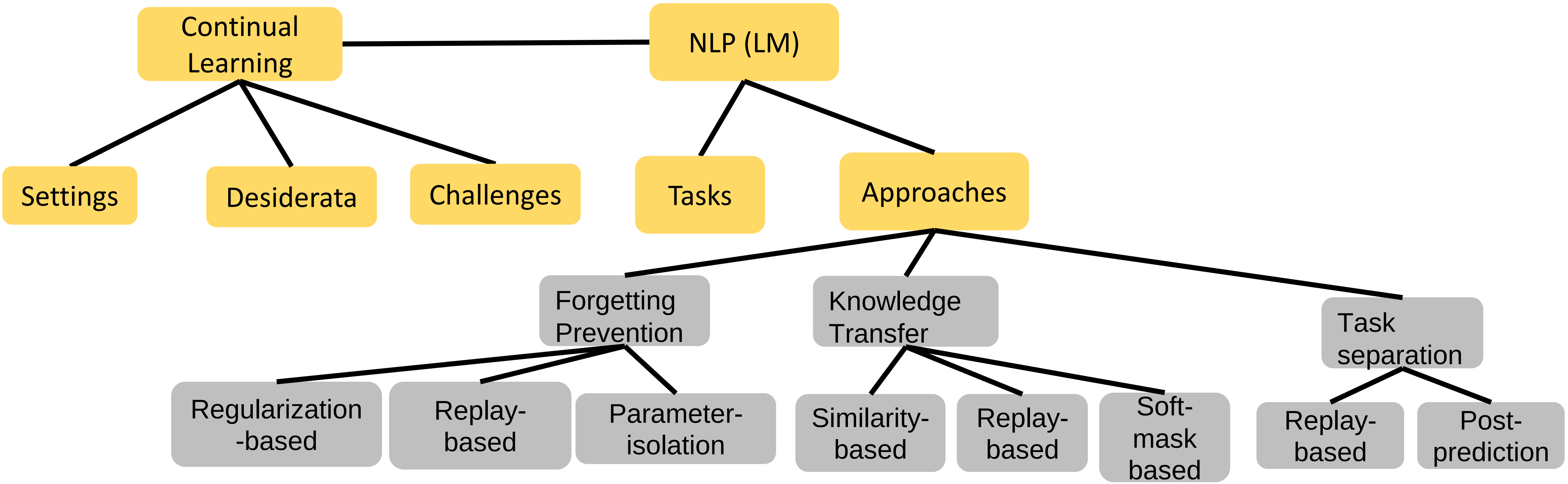
LFPT5



- Pre-trained LM (T5) serves as both the problem solver and generator
- When a new task comes, T5 first generates the old task data and then trains the pseudo data and new task altogether. Since some previous data is available, the decision boundary is **easier** to establish

Qin et al. A unified framework for lifelong few-shot language learning based on prompt tunning of T5. ICLR, 2022.

Roadmap



*LM: Language Model

Plan

- A quick review of we have talked
- Another setting: Domain-incremental Learning
- Continual learning of NLP Tasks
- Section summary and future work

KEY TAKE AWAY

- Continual Learning
  - Settings
    - TIL
    - CIL
    - **DIL**
  - Desiderata
    - Prevent Forgetting
    - Forward Knowledge Transfer
    - Backward Knowledge Transfer
  - Challenges
    - Stability-plasticity
    - Transfer-inference
    - Task separation
- NLP (LM)
  - Tasks
    - Pre-training
      - General Pre-training
      - **Domain-adaptive Pre-training**
    - End-task learning
  - Approaches
    - Forgetting Prevention
      - Regularization-based
      - Replay-based
      - **Parameter-isolation**
    - Knowledge Transfer
      - Similarity-based
      - Replay-based
      - **Soft-mask based**
    - Task separation
      - **Replay-based**
      - **Post-prediction**

- Knowledge transfer is the major issue in TIL and DIL
  - A task sequence can consist of a combination of similar and dissimilar tasks[1]
- Forgetting and task separation/discrimination are major issues for CIL and DIL w/o ID
  - Task separation has been proved to be related to OOD detection, but the accuracy is still far from the upper bound[2]

[1]: Ke et al., Continual learning of a mixed sequence of similar and dissimilar tasks, NeurIPS 2020
[2]: Kim et al., A Theoretical Study on Solving Continual Learning, NeurIPS 2022

- Continual learning of language models (LM) (e.g., domain-adaptive pre-training) is still in its infancy
  - How to better preserve the general knowledge? DAS is an initial attempt, but it is still limited.
- Scalability
  - How the network capacity issue affects the performance and how to alleviate the issue effectively are also unclear.
- Temporal continual learning
  - How to keep the LM up-to-date? Everything changes with time.

Thank you

- We have benchmarked many SoTA baselines
  - For continual end-task learning
    - https://github.com/ZixuanKe/PyContinual
  - For continual domain-adaptive pre-training
    - https://github.com/UIC-Liu-Lab/ContinualLM
- More details
  - **Survey:** Continual Learning of Natural Language Processing Tasks: A Survey *(Preprint)*
  - https://vincent950129.github.io/