UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Beyond Pre-training: Learning for Knowledge Updates in Language Models

by Zihan Zhang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Prof. Ling Chen and Dist. Prof. Jie Lu

University of Technology Sydney
Faculty of Engineering and Information Technology

August 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Zihan Zhang*, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science, Faculty of Engineering and Information Technology* at the University of Technology Sydney, Australia.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: Signature removed prior to publication.

Zihan Zhang

DATE: 10th August, 2024

PLACE: Sydney, Australia

ABSTRACT

Modern language models (LMs) have learned an enormous amount of knowledge during pre-training, making them versatile in solving various downstream natural language processing (NLP) tasks. These pre-trained models are capable of capturing rich semantic patterns within large-scale text corpora and learning high-quality representations of text. During inference, LMs can leverage the knowledge acquired during pre-training from their parameters to address various NLP tasks, demonstrating superior performance compared to traditional NLP approaches. However, there is a significant challenge remains unsolved: *LMs are static after pre-training, and there is no mechanism to update themselves or adapt to a changing environment*. Yet, our world is dynamic and constantly evolving. The static nature of trained LMs causes the memorized knowledge to become quickly obsolete, which often leads to hallucinations and renders them unreliable and impractical for evolving downstream applications.

In this thesis, we aim to address a central question: *how can new knowledge be incorporated efficiently into LMs beyond the pre-training stage?* Specifically, we introduce novel approaches from three aspects. First, we propose an efficient data annotation method for training new LMs. Our method significantly reduces the amount of annotation data required while achieving improved performance under a weakly-supervised setting, thereby efficiently integrating new knowledge into LMs beyond pre-training. Second, we introduce a continual adaptation approach for LMs to emerging knowledge. We formulated the problem of continual instruction tuning (CIT) to enable LMs to continuously learn from emerging tasks and established a benchmark suite that includes both learning and evaluation protocols. Lastly, we propose an adaptive retrieval augmentation approach for LMs at inference, which incorporates new knowledge efficiently without altering the original parameters. This method aims to integrate new information while mitigating negative effects.

Experiments conducted across various NLP tasks and benchmarks demonstrate the effectiveness of our approaches for incorporating new knowledge into LMs beyond the pre-training stage. Overall, our research findings address the central research question by presenting novel methods and analyses for enhancing LMs in diverse NLP applications.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisors, Professor Ling Chen and Distinguished Professor Jie Lu, for their unwavering guidance and support throughout my PhD journey. Their insightful feedback and encouragement have been invaluable in shaping my research.

Additionally, I would like to extend my thanks to Dr. Meng Fang from the University of Liverpool for his valuable guidance during times of uncertainty. Collaborating with him has broadened my understanding of the field and inspired me with numerous exciting and promising future directions.

I would also like to give special thanks to Dr. Mohammad-Reza Namazi-Rad, my research intern mentor, who offered me a unique perspective on academia and industry that enriched my research experience. Equally, my gratitude extends to my friends and colleagues, including Jiaxi Li, Mahati Suvvari, Dr. Parham Khojasteh, Dr. Getian Ye, and Dr. Weiwei Hou, for providing me with industrial guidance and life insights.

Furthermore, I am fortunate to have had a group of incredible colleagues at the UTS NLP group to support me along the way. I cherished the time we spent discussing research ideas, playing board games, and hiking together. I would also like to acknowledge Anthony Dang for developing this beautiful thesis template.

In addition, I am grateful to Qiwen Tian and Yi Yu, who have been my best friends for a long time. Despite the distance, our friendship has remained strong.

Finally, on a more personal note, I would like to express my deepest gratitude to my parents, Qing and Yinfang, my grandparents, and many other family members for their financial support and unconditional love. Studying abroad can be lonely, but with your support and love, I have persevered through the highs and lows.

I hope to have made you all proud.

Zihan Zhang
Sydney, Australia
August, 2024

TABLE OF CONTENTS

ABBREVIATIONS

AI Artificial Intelligence

AL Active Learning

CF Catastrophic Forgetting

CL Continual Learning

CoT Chain-of-Thoughts

DST Dialogue State Tracking

FFN Feed-Forward Networks

ICL In-context Learning

IT Instruction Tuning

KBs Knowledge Bases

KE Knowledge Editing

KGs Knowledge Graphs

LLMs Large Language Models

LMs Language Models

MHSA Multi-Head Self-Attention

MLP Multilayer Perceptron

NLP Natural Language Processing

PLMs Pre-trained Language Models

QA Question Answering

RAG Retrieval-Augmented Generation

LIST OF FIGURES

FIGURE **Page**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Recent years have witnessed the emergence of Transformer-based [246] Pre-trained Language Models (PLMs)[1], such as BERT [46], RoBERTa [156], T5 [206], GPT series [204, 205, 16, 185], and Llama series [242, 243, 51], which have been widely used in many NLP (Natural Language Processing) tasks and have shown impressive performance compared to previous approaches. For example, BERT [46] has significantly advanced the state-of-the-art in eleven NLP tasks since its release and has become a standard for language modelling. More recently, GPT-4 [185] and Llama-3 [51] have become the leading representatives of closed-source and open-source LLMs (Large Language Models), respectively. They demonstrate superior performance across various tasks and exhibit capabilities that go beyond mere language processing. The success of PLMs has made them the foundation of modern AI (Artificial Intelligence) applications used by millions of people today, including ChatGPT[2], Google's Gemini[3], and Microsoft's Copilot[4].

The majority of a PLM's intelligence primarily stems from large-scale pre-training [297], which has become the initial step in training a randomly initialized language model. PLMs trained on massive unlabeled text corpora collected from various high-quality sources such

[1] In this thesis, LMs (Language Models) and LLMs (Large Language Models) are used interchangeably with PLMs (Pre-trained Language Models), as almost all modern Transformer-based LMs or LLMs are pre-trained.

[2] `https://openai.com/index/chatgpt/`

[3] `https://blog.google/technology/ai/google-gemini-ai/`

[4] `https://www.microsoft.com/en-au/microsoft-copilot`

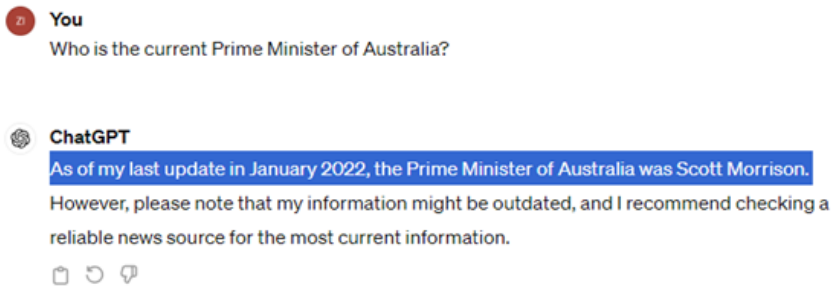


Figure 1.1: A pre-trained LM is static and can quickly be outdated, making factually incorrect predictions. (*e.g.*, ChatGPT; [184]).

as Wikipedia, academic papers, books, and GitHub, implicitly encode vast amounts of knowledge within their parameters [196, 213, 103]. This enables them to serve as versatile foundation models, capable of performing various tasks directly through ICL (In-context Learning) [154, 185, 17, 109], including QA (Question Answering) [195, 109], reasoning [198], summarization [67], and content generation [100], or further fine-tuning for domain-specific applications [233, 65, 155], including legal [22] and medical [113, 182] domains.

Despite the tremendous success of PLMs in NLP, several significant challenges remain unresolved. One major issue is that LMs become static after pre-training, meaning their embedded knowledge can quickly become outdated as the world continues to evolve [133, 1, 150]. During inference, LMs recall relevant knowledge stored in their parameters based on the given inputs and make predictions. However, due to the knowledge cut-off in LMs, this parametric memorization can lead to hallucinations, where the generated content is nonsensical or unfaithful to the provided source material, resulting in factual inaccuracies [91]. For instance, as shown in Fig. 1.1[5], the answer to `"Who is the current Prime Minister of Australia?"` has changed from `"Scott Morrison"` to `"Anthony Albanese"` since 2022, but ChatGPT (with no web browsing) fails to answer.

Another major limitation of PLMs is that they are restricted to the knowledge acquired during pre-training. This restriction means that PLMs often fall short on domain-specific tasks due to limited exposure to the relevant knowledge and vocabulary from the training corpus. As illustrated in Fig. 1.2, proprietary, commercial, and domain-specific data are often excluded from pre-training corpora, resulting in such knowledge being rarely learned by LMs. For instance, even powerful models like OpenAI's ChatGPT [184] perform poorly in specialized domains such as legal [22] and medical [113, 182] fields. Although increasing the model size can encode more knowledge in its parameters, it is impractical to encode the entirety of the world's knowledge [240]. Furthermore, there is always new

[5]The screenshot was taken in April 2024 for ChatGPT (3.5) without web browsing.

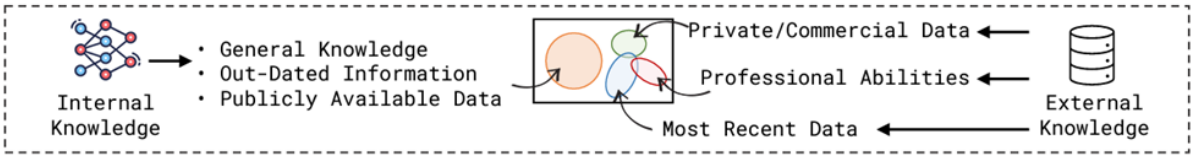


Figure 1.2: An illustration of knowledge that is outside of the scope of the pre-training corpora, such as private or commercial domain-specific data and new world knowledge [304].

data and emerging tasks that LMs have not encountered before, which can result in poor performance. The static nature of PLMs makes the memorized knowledge quickly obsolete, which often causes hallucinations, rendering them unreliable for knowledge-intensive tasks [132, 164, 100, 232]. Ensuring that PLMs remain aligned with the ever-changing world knowledge and emerging tasks is a pressing concern, especially after deployment, because many users and downstream applications rely on them, particularly in the era of LLMs.

Previous approaches have attempted to extend the knowledge of PLMs by fine-tuning them with supervised task-specific datasets [253, 158, 216, 250] or by incorporating new knowledge through retrieving relevant information at test time, thereby keeping the parameters of the models unchanged [138, 208, 229]. However, these approaches often overlook the high cost of collecting labelled supervised datasets or fail to consider the potential side effects of fine-tuning.

In this thesis, we are motivated to address a central question: *how can new knowledge be incorporated efficiently into LMs beyond the pre-training stage?* Specifically, we investigate this problem from three angles. First, we explore data-efficient methods for training new task-specific language models. Second, we focus on continually adapting language models to emerging tasks and domains while minimizing the side effects of frequent parameter updates. Lastly, for recent LLMs, we efficiently incorporate new knowledge to improve task performance with lower inference overhead. Through these research questions, we aim to advance the understanding of how to efficiently extend the capabilities of PLMs beyond the pre-training stage. Our goal is to minimize any negative effects on previously acquired skills while maximizing performance on tasks that require new knowledge, thereby enhancing various downstream NLP tasks.

1.2 Research Questions

In this thesis, we study methods for efficiently and effectively updating language models' existing knowledge and abilities gained during pre-training, aiming to minimize any adverse effects on those prior skills and maximize the performance of tasks that require new

knowledge. Therefore, we propose three research questions as follows.

Research Question 1

How can we efficiently and effectively label raw data to aid in training new task-specific language models?

Fine-tuning LMs on new data collected from different domains has been widely adopted in the literature to expand LMs' knowledge beyond the scope of pre-training [46, 156, 206, 167]. However, existing approaches typically assume that labelled data already exists for supervised training and can be used directly. Yet, practically, collecting high-quality labelled data for training is expensive and time-consuming. Moreover, not all data instances are equally important, so data selection is essential for training a high-performing LM [297]. Therefore, RQ1 focuses on task-specific LMs and explores methods from a data perspective, aiming to reduce the costs of annotating raw data for training while improving performance on downstream tasks.

Research Question 2

How can we adapt language models to emerging tasks while minimizing catastrophic forgetting and maximizing knowledge transfer?

After large-scale pre-training, LMs have shown impressive performance on various NLP tasks and can even generalize remarkably well to unseen tasks [253, 216, 250, 36, 158]. However, LMs still fall short on domain-specific tasks due to the limited exposure to relevant knowledge and vocabulary from the training corpus [163]. For example, a well-trained general LM may perform poorly in domains such as math, finance, and medicine, and it cannot solve tasks that it has not learned before. Fine-tuning LMs on task-specific data can solve this issue to some extent. However, frequently fine-tuning LMs on data with different distributions can lead to *catastrophic forgetting*, a phenomenon where previously learned knowledge or abilities degrade due to overwritten parameters [171]. Moreover, enabling knowledge transfer is also essential, as many tasks are similar and share common knowledge [121]. To address this issue, RQ2 explores methods to continually adapt LMs to new domains and tasks, aiming to alleviate catastrophic forgetting and maximize knowledge transfer.

Research Question 3

How can we efficiently incorporate new knowledge into language models without compromising their existing knowledge?

In RQ1 and RQ2, we explore how to adapt LMs to emerging skills and knowledge through fine-tuning. However, in the era of LLMs, model sizes have grown exponentially and often contain hundreds of billions of parameters [16, 186, 35, 285, 185, 243, 6], making fine-tuning prohibitively expensive. On the one hand, fine-tuning is both expensive and environmentally unfriendly [190], especially in the era of LLMs with billions of parameters. For instance, LLaMA-65B was trained for about one million GPU hours and emitted more than a hundred tons of carbon [243]. On the other hand, fine-tuning without constraints may have a "butterfly effect" and affect other knowledge or skills present in the model [128, 139, 3], causing degraded generalization [177], catastrophic forgetting [128, 300, 3], or knowledge conflicts [181]. While RAG (Retrieval-Augmented Generation) can effectively incorporate new information into language models, indiscriminate retrieval can lead to significant inference overheads. Therefore, RQ3 aims to provide a solution that maintains the general skills of LMs while being efficient and avoiding the need for expensive fine-tuning.

1.3 Contributions

This thesis studies efficient and effective knowledge updates in pre-trained language models. The core contributions of the thesis are summarized as follows:

- **Systematic Survey of Knowledge Updates in Language Models:** This work is one of the first to review the recent compelling advances in aligning pre-trained LLMs with the ever-changing world knowledge. We categorize research works systemically and highlight representative approaches in each category and provide an in-depth comparison with discussion for insights.

- **Learning New Knowledge via a Data Efficient Method**: This work is the first attempt to apply turn-level Active Learning to Dialogue State Tracking for training new task-specific LMs. In this work, we propose a novel model-agnostic turn-level Active Learning framework for dialogue state tracking, which provides a more efficient way to annotate new dialogue data. Our approach strategically selects the most valuable turn from each dialogue to label, which largely saves annotation costs. In addition, using

significantly reduced annotation data, our method achieves the same or better DST performance under the weakly-supervised setting.

- **Learning New Tasks Continually**: This work is the first step to exploring continual learning in instruction-tuned language models. First, we formulate the problem of Continual Instruction Tuning and establish a benchmark suite consisting of learning and evaluation protocols. Second, we curate two long task streams of various types to study different setups of Continual Instruction Tuning. Last, we implement various continual learning methods of different categories, conduct extensive experiments and ablation studies to analyze the lack of current practices and propose a future direction.

- **Learning New Knowledge Adaptively**: This work first points out the limitations of retrieval-augmented generation. To facilitate the research in this area, we create a new dataset to assess adaptive retrieval-augmented generation for short-form open-domain QA. Then, we benchmark existing methods and conduct extensive analysis, finding that vanilla prompting is insufficient in guiding LLMs to make reliable retrieval decisions. Finally, we propose a simple yet effective method to help LLMs assess the necessity of retrieval without calibration or additional training.

1.4 Thesis Organization

This thesis explores approaches to learning new knowledge and tasks of language models after the pre-training stage. The remaining of the thesis is organized as follows:

- **Chapter 2** describes how knowledge is acquired in language models and related work in various approaches to updating LMs' knowledge after the pre-training stage.

- **Chapter 3** presents our proposed approaches to address RQ1: *How can we efficiently and effectively label raw data to aid in training new task-specific language models?* We demonstrate the effectiveness of our methods on the task of supervised dialogue state tracking.

- **Chapter 4** outlines our answers to RQ2: *How can we adapt language models to emerging tasks while minimizing catastrophic forgetting and maximizing knowledge transfer?* We describe our proposed approaches to continually adapt instruction fine-tuned language models for new tasks without severely comprising their prior abilities. We

design the learning and evaluation protocols and validate the effectiveness of our approaches on two curated long task streams.

- **Chapter 5** tries to answer RQ3: *How can we efficiently incorporate new knowledge into language models without compromising their existing knowledge?* We first identify the limitations of existing RAG-based solutions and then describe our improved approach for incorporating knowledge into language models adaptively and efficiently. To validate our approach, we establish a new benchmark for the short-form question-answering task.

- **Chapter 6** summarises the thesis and provides the potential future directions and challenges on knowledge updates in pre-trained language models.

1.5 Publications

The main body of this thesis has been published in major artificial intelligence (NLP) conferences. Specifically, the primary content of Chapter 2 has been published in one paper, which is a comprehensive survey of recent advances in aligning language models with the ever-changing world knowledge:

- **Zihan Zhang**, Meng Fang, Ling Chen, Mohammad-Reza Namazi-Rad, and Jun Wang. 2023. How Do Large Language Models Capture the Ever-changing World Knowledge? A Review of Recent Advances. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8289–8311, Singapore. Association for Computational Linguistics. (**CORE A***)

The content of Chapter 3 has been published in one paper about efficient data annotation for learning new dialogue states:

- **Zihan Zhang**, Meng Fang, Fanghua Ye, Ling Chen, and Mohammad-Reza Namazi-Rad. 2023. Turn-Level Active Learning for Dialogue State Tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7705–7719, Singapore. Association for Computational Linguistics. (**CORE A***)

There is another paper on efficiently discovering topics from documents via contextualized clustering, but since it is an unsupervised method without model training, it is not included in the thesis:

- **Zihan Zhang**, Meng Fang, Ling Chen, and Mohammad Reza Namazi Rad. 2022. Is Neural Topic Modelling Better than Clustering? An Empirical Study on Clustering with Contextual Embeddings for Topics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3886–3893, Seattle, United States. Association for Computational Linguistics. (**CORE A**)

Chapter 4 primarily consists of a paper on learning new tasks and knowledge for language models via continual instruction tuning:

- **Zihan Zhang**, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. 2023. CITB: A Benchmark for Continual Instruction Tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9443–9455, Singapore. Association for Computational Linguistics. (**CORE A***)

The research described in Chapter 5 has been published in a paper focusing on learning new knowledge without altering language models' internal parameters via retrieval-augmented generation:

- **Zihan Zhang**, Meng Fang, and Ling Chen. 2024. RetrievalQA: Assessing Adaptive Retrieval-Augmented Generation for Short-form Open-Domain Question Answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand. Association for Computational Linguistics. (**CORE A***)

CHAPTER 2

LITERATURE REVIEW

In this chapter, we conduct a literature review to provide the necessary background on pre-trained language models (PLMs). Specifically, we first discuss how language models (LMs) acquire knowledge in §2.1, followed by an exploration of how knowledge can be updated in LMs in §2.2. The majority of this chapter is based on one published paper at EMNLP 2023 – How Do Large Language Models Capture the Ever-changing World Knowledge? A Review of Recent Advances [289].

2.1 Knowledge Acquisition in Language Models

Recent studies have demonstrated that language models (LMs) pre-trained on extensive datasets possess a broad spectrum of knowledge, including factual information, linguistic structures [238, 13], reasoning abilities [151, 268], mathematical skills [227], and coding proficiency [28, 254]. These abilities, collectively referred to as "*parametric knowledge*" have been shown to reliably reside within a subset of the trained parameters in pre-trained models [103, 200, 118]. For example, [196, 213] examine the knowledge encoded in the parameters of PLMs. [23] further explore how LMs, particularly LLMs, acquire factual knowledge during pre-training. Their findings indicate that factual knowledge acquisition in LLM pre-training occurs by progressively increasing the probability of factual knowledge presented in the pre-training data at each step. Other research, such as [2, 53, 141], investigates how LMs learn and capture factual knowledge from the training data. Additionally, [4] demonstrate that the knowledge used during pre-training should be as diverse as possi-

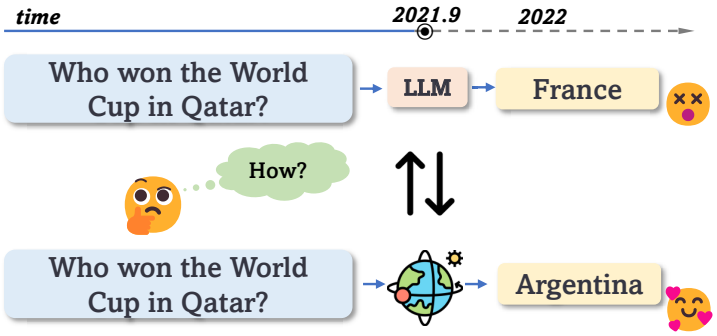


Figure 2.1: A trained LLM is static and can be outdated (*e.g.*, ChatGPT; [184]).

ble to be reliably extracted. On the other hand, recent investigations reveal that LMs struggle to acquire long-tail knowledge [110, 168] and cannot effectively leverage the knowledge learned during pre-training . [75, 162, 237] study the training dynamics of LMs, particularly how these dynamics evolve during training, while [245, 241] focus on the dynamics of memorization in LM pre-training.

2.2 Knowledge Updating in Language Models

In this section, we review various techniques for updating the knowledge in LMs (Language Models), with a particular emphasis on LLMs (Large Language Models). As the deployment of LLMs in various applications grows, the need for efficient and effective methods to keep their knowledge up-to-date becomes increasingly critical.

LLMs [16, 186, 35, 285, 185, 243, 6] trained on massive corpora from various sources (*e.g.*, Wikipedia, Books, Github) implicitly store enormous amounts of world knowledge in their parameters [196, 213, 103], enabling them to act as versatile foundation models for performing various natural language processing (NLP) tasks directly through in-context learning [154, 185, 17, 109] or for further fine-tuning for domain-specific uses [233, 65, 155].

Despite their impressive performance, LLMs are static after deployment, and there is no mechanism to update themselves or adapt to a changing environment [114, 17]. Our world, however, is dynamic and constantly evolving. As shown in Fig. 2.1, the static nature of trained LLMs makes the memorized knowledge quickly obsolete, which often causes hallucinations, rendering them unreliable for knowledge-intensive tasks [132, 164, 100, 232]. In the era of LLMs, ensuring their alignment with the ever-changing world knowledge and maintaining their up-to-date status after deployment is a pressing concern because many users and downstream applications rely on them. Unfortunately, simply re-training LLMs with the latest information is infeasible due to prohibitive costs [190].

Intuitively, to update an LLM, one can either replace the obsolete knowledge stored *implicitly* in the model with new ones by modifying its parameters, or override the outdated model outputs using new information *explicitly* retrieved from the world. Tremendous work has been proposed in the literature to implicitly or explicitly refresh deployed LLMs; however, these approaches, scattered among various tasks, have not been systematically reviewed and analyzed.

In this section, we survey the recent compelling advances in aligning deployed LLMs with the ever-changing world knowledge. We categorize research works systemically and highlight representative approaches in each category (§2.2.1) and provide an in-depth comparison with discussion for insights (§2.2.4).

To the best of our knowledge, reviews on this topic are scarce. Closest to our work, [3] review pre-trained language models (LMs) as KBs (Knowledge Bases) and review a set of aspects that a LM should have to fully act as a KB; [19] further divide the life cycle of knowledge in LLMs into five periods and survey how knowledge circulates; [269] conduct an empirical analysis of existing knowledge editing methods. Despite partially overlapping with our discussion of knowledge editing in §2.2.2.2, they only touch a subset of the scope that our survey studies and ignore other potentials in aligning LLMs with the world knowledge. [176, 252, 202] study augmented, interactive, and tool learning of LLMs respectively, which share different goals from ours. Previous knowledge-enhanced LMs surveys [301, 256, 278, 271, 294] focus on injecting knowledge into LMs, typically requiring modifying the model's architecture or re-training. Instead, we focus on the potential of how deployed LLMs capture the *ever-changing* world knowledge effectively and efficiently without re-training. [251] provide a comprehensive review of forgetting in deep learning that is not limited to continual learning. [188] review potential approaches that unify KGs (Knowledge Graphs) and LLMs. While structural knowledge, such as KGs, can broadly be categorised as explicit knowledge and augmented to LLMs for new knowledge, KG is static after creation and can still be outdated [99]. New information or discoveries not yet incorporated into KGs may lead to outdated knowledge. However, how to efficiently update KGs is out of the scope of this review.

2.2.1 Taxonomy of Methods

Based on whether the method tends to directly alter the knowledge stored implicitly in LLMs, or leverage external resources to override the outdated knowledge, we roughly categorize them as ***implicit*** (§2.2.2) or ***explicit*** (§2.2.3) approaches. We summarise representative works from each category in Fig. 2.4. Note that **Implicit** means the approaches seek to

directly alter the knowledge stored in LMs (*e.g.*, parameters) (§2.2.2), while **Explicit** means more often incorporating external resources to override internal knowledge (*e.g.*, search engine) (§2.2.3).

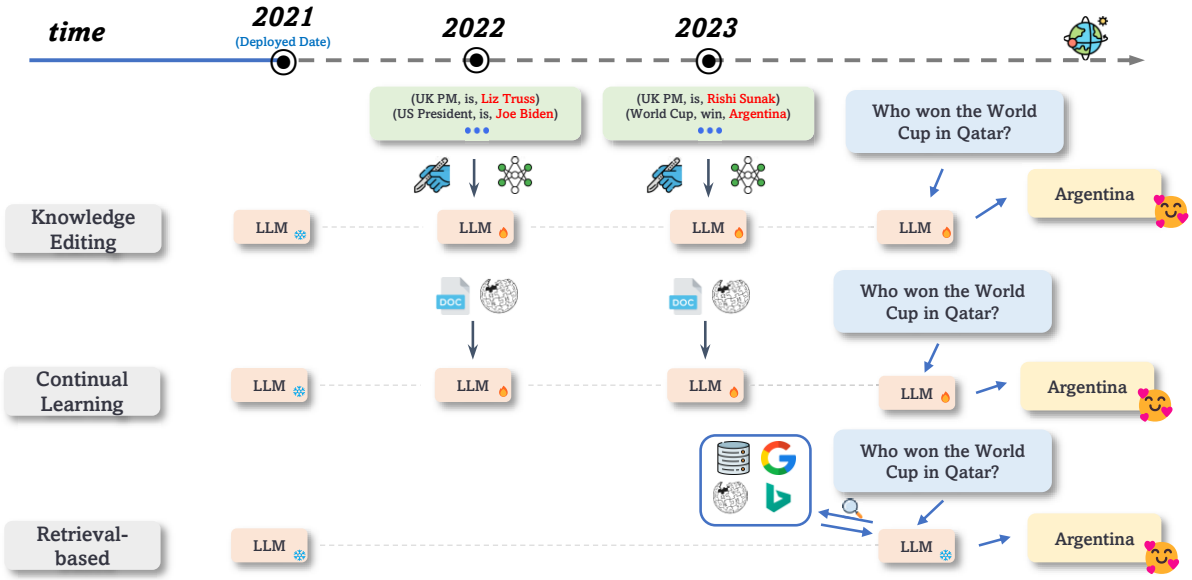


Figure 2.2: A high-level comparison of different approaches.

2.2.2 Implicitly Align LMs with World Knowledge

Previous studies have shown that LLMs can implicitly memorize knowledge in their large number of parameters after being pre-trained on massive corpora [196, 213, 103, 233]. To keep LLMs up-to-date and align with the current world knowledge, the straightforward way is to alter the model's behaviour from *themselves* to generate desired outputs. To cope with this issue, this line of work aims to design better strategies to modify the internal states of LLMs in a more controllable and efficient way, which can be categorized into *naive approaches* (§2.2.2.1), *knowledge editing* (§2.2.2.2), and *continual learning* (§2.2.2.3).

2.2.2.1 Naive Approaches

Naively, one can regularly *re-train* the model from scratch or *fine-tune* the model with the latest corpora to align with current world knowledge.

Re-training. Intuitively, one can regularly re-train the model from scratch with the latest corpora to align with current world knowledge. However, this naive solution has clear

downsides: (1) Re-training is both time and money expensive and environmentally unfriendly [190], especially in the era of LLMs with billions of parameters. For instance, LLaMA-65B was trained for about one million GPU-hours and emitted more than a hundred tons of carbon [243]; (2) It is unrealistic to frequently re-training an LLM in response to the constantly changing world.

Fine-tuning. Another simple approach is to periodically curate a small-scale dataset containing the desired knowledge we wish the model to add, update, or delete, then fine-tune the model on the dataset. Despite being computationally cheaper than re-training, it still falls short in that, without constraints, directly fine-tuning the model may have a "butterfly effect" and affect other knowledge or skills present in the model [139], causing degraded generalization [177], catastrophic forgetting [128, 300, 3], or knowledge conflicts [181].

Constrained Fine-tuning. To solve part of the above-mentioned issues, [300] propose to only fine-tune the model on the small-scale modified facts set and add explicit constraints on the model weights so that the model learns to answer the modified facts while keeping the remaining knowledge intact. Specifically, they use various norms ($\mathscr{L}_0$, $\mathscr{L}_2$, and $\mathscr{L}_\infty$) to prevent the parameters of the fine-tuned model θ' from drifting too far from the original model parameters θ. They further find that fine-tuning only the first and last layers of the Transformer model [246] results in better adaptation to the modified facts and better preservation of performance on the unmodified facts. However, the norm-based constraint on parameters ignores the highly non-linear nature of LMs and how parameters determine the outputs of the model, making their method potentially unreliable [43]. In addition, [177] confirm that constrained fine-tuning generally does not consistently provide edit generality.

2.2.2.2 Knowledge Editing

Since tuning LLMs to learn new knowledge can be prohibitively expensive [190], researchers seek efficient methods to directly update more specific, localized, or fine-grained knowledge that is preserved in LLMs [177]. **KE (Knowledge Editing)** is an arising and promising research area that aims to alter the parameters of some specific knowledge stored in pre-trained models so that the model can make new predictions on those revised instances while keeping other irrelevant knowledge unchanged [234, 43, 177, 172, 78, 173]. In this section, we categorize existing methods into *meta-learning*, *hypernetwork*, and *locate-and-edit* -based methods.

To facilitate the development of this area, [43] formulate three desiderata that an ideal editing method should follow: ① **Generality**: the method should be capable of altering the knowledge of any LM that is not specifically trained to be editable (*e.g.*, PaLM, GPT-4, LLaMA); ② **Reliability**: the method should only update the targeted knowledge without influencing the rest of the knowledge in the LM. For instance, the answer to `"Who is the current Prime Minister of Australia?"` has changed from `"Scott Morrison"` to `"Anthony Albanese"` since 2022, updating the knowledge from `"Scott Morrison"` to `"Anthony Albanese"` should not change the knowledge `"Argentina won the 2022 World Cup"`; ③ **Consistency (Generalization)**: after updates, the model predictions should be consistent across semantically equivalent inputs (*e.g.*, correctly predicts `"Anthony Albanese"` to `"Who is the AU PM?"`). Beyond updating outdated knowledge, knowledge editing can also delete sensitive information for privacy issues or eliminate biases in the pre-training corpora.

However, not until recently, [183, 296] show that, after performing knowledge editing, the LLM does not really "learn" the updated knowledge and thus cannot *propagate* the new knowledge and make further inferences based on them. For instance, after learning that `"the current PM of Australia is Anthony Albanese"`, the model might not able to make predictions of `"Who is the spouse of the current PM of Australia?"`.

Meta-learning. This line of work generally focuses on the *intrinsic* editability of the model itself, aiming to modify the model parameters so that they can be easily updated during inference [43, 177]. [234] propose a model-agnostic meta-learning-based [59] method that trains neural networks in a way that the trained parameters can be easily edited afterwards. [32] introduce a two-loop framework. In the inner training loop, they employ a few gradient updates to enable a pre-trained GPT-2 model [205] to efficiently memorize external knowledge. Subsequently, in the outer loop, the model parameters are dynamically adjusted through optimal meta-parameter learning to incorporate additional knowledge that aids reasoning tasks. [234], by constraining the training objective, encodes editability into the parameters of the model itself so that the model is "prepared" for incoming edits. While being effective and no new parameters are required, it does not conform to generality as it requires specialized training of the original model [43]. Moreover, to enforce the constraint that the editable model agrees with the original pre-trained model's predictions, [234]'s method needs to retain a copy of the original model, which significantly consumes computation memory [177]. [32] also requires training of the original LM, which could be computationally expensive for larger LMs. In addition, whether it will influence other irrel-

evant knowledge in the model remains unknown, making the method potentially unreliable.

Hypernetwork Editor. In contrast to pre-modifying the pre-trained language model, an alternative approach in the field involves training *extrinsic* editors that update knowledge during test time, thereby avoiding any modifications to the base model. [43] reframe editing the knowledge of a model as a *learning-to-update* problem. Specifically, given a single data instance that needs to be updated, their trained hypernetwork [74] predicts a shift $\Delta\theta$ such that $\theta' = \theta + \Delta\theta$, where θ is the original pre-trained LM weights and θ' is the updated weights. To keep editing effective while being easy to scale to larger LMs with billions of parameters, [177] decompose weight updates into low-rank components [87], thus making it possible to scale to LLMs. Orthogonal to [177], [78] introduce a new training objective considering sequential, local, and generalizing model updates. Although scaled beyond a single edit, their edit success rate significantly degrades when performing larger edits simultaneously. Unlike the above methods that operate on the model's weight, [83] perform edits on the representation level. [187] employ knowledge distillation to transfer knowledge generated from a teacher model to a student model. [43] can be more efficient than [234], as it does not retain the copy of the original model nor compute higher-order gradients. However, it can only update a single fact rather than multiple facts in a row and fail to edit large models, leading to poor scalability [177, 78]. [177] improve [43]'s work and is stable to edit LMs from BERT-base (110M) [46] to T5-XXL (11B) [206]. However, when editing multiple knowledge simultaneously, their edit success rate significantly degrades.

Locate and Edit. Generally, this line of work adopts the *locate and edit* pattern: they first identify the location of specific knowledge stored in the model via different assumptions, then directly modify the weights or representations to update knowledge. Inspired by the findings that FFN (Feed-Forward Networks) in Transformer [246] are key-value memories [64], [40] introduce the *knowledge neurons* concept and propose a gradient-based knowledge attribution method to identify these knowledge neurons in FFNs. Further, without fine-tuning, they directly modify the corresponding value slots (*e.g.*, embeddings) in the located knowledge neurons and successfully update or delete knowledge, demonstrating a preliminary potential to edit knowledge in LMs.

Different from [64]'s per-neuron view, [172] conduct casual tracing analysis on GPT-2 and hypothesize that the Transformer MLP (Multilayer Perceptron) can be viewed as a linear associative memory. They verify their hypothesis by directly updating the middle-layer

MLP weights with a rank-one update [10]. Following [172]'s work, [173] propose a scalable multi-layer method to update an LLM with thousands of facts simultaneously, significantly improving editing efficiency while maintaining generalization and specificity. [69] further adapt it to fix commonsense mistakes. [142] find that MHSA (Multi-Head Self-Attention) weights do not require updating when introducing new knowledge. Based on this, they propose precisely updating FFN weights by simultaneously optimizing the Transformer component hidden states of MHSA and FFN to memorize target knowledge. [31] propose an architecture-adapted multilingual integrated gradients method to localize knowledge neurons precisely across multiple architectures and languages. [63] analyze the internal recall process of factual associations in auto-regressive LMs, opening new research directions for knowledge localization and model editing.

While simple, [40] do not ensure reliability on other irrelevant knowledge and generalization on semantically equivalent inputs. Despite showing both generalization and specificity, [172] only edits a single fact at a time, making it impractical for large-scale knowledge updating in LLMs. Through casual tracing, [173] identify and update the critical MLP layers in one go. However, [77] argue that the relation between localization and editing may be misleading as they can edit factual knowledge in different locations that are not suggested by casual tracing.

Other. [260] propose an evaluation framework and dataset for measuring the effectiveness of knowledge editing of LLMs, as well as the ability to reason with the altered knowledge and cross-lingual knowledge transfer. Similarly, [37] evaluate the implications of an edit on related facts and show that existing methods fail to introduce consistent changes in the model's knowledge. [107] propose an evaluation benchmark for locate-and-edit-based methods, aiming to reassess the validity of the locality hypothesis of factual knowledge. [248] and [264] consider multilingual and extend existing knowledge editing methods into cross-lingual scenarios.

2.2.2.3 Continual Learning

While knowledge editing provides fine-grained control to update specific knowledge in LLMs, it often requires large amounts of supervised training data to make edits, which is non-trivial to create [76]. In addition, when an LLM needs to quickly acquire new domain knowledge (e.g., legal or medical), such small-scale model edits may not be efficient. Moreover, after multiple parameter patches to a deployed model, its internal knowledge may conflict, leading to unpredictable behaviours [177].

Sharing a related goal, **CL (Continual Learning)** aims to enable a model to learn from a continuous data stream across time while reducing CF (Catastrophic Forgetting) of previously acquired knowledge [12]. In contrast to knowledge editing, CL generally updates models on a larger scale and works in long learning sequences with minimal memory overheads [177]. Hence, CL can also be used for deployed models to update their knowledge. With CL, a deployed LLM has the potential to adapt to the changing world without costly re-training from scratch [17]. In this section, we introduce approaches that employ CL for aligning LLMs with the current world knowledge, including *continual pre-training* and *continual knowledge editing*.

Continual Pre-training. Unlike traditional continual learning, which sequentially fine-tunes a pre-trained LM on some specific downstream tasks (*e.g.*, QA, text classification), *continual pre-training* is used to further pre-train an LM to acquire new knowledge, where the data corpus is usually *unsupervised* [72, 120]. Since our target is the versatile foundation LLMs (*e.g.*, GPT-4) that can be applied to many different use cases rather than a fine-tuned model designed for a specific task, we focus on the literature on continual pre-training.

Early works [72, 214, 133, 47] empirically analyze continuing LM pre-training on emerging domain or temporal data, showing the potential to update the base LM with new knowledge. [98] explicitly categorize world knowledge as time-invariant, outdated, and new knowledge, which should be retained, acquired, and updated respectively by an LM when learning continually. [105, 97, 98] additionally implement traditional CL methods to alleviate *catastrophic forgetting*, a phenomenon in which previously learned knowledge or abilities are degraded due to overwritten parameters [128]. Among the literature, CL methods can be mainly categorized into ① **Regularization**, ② **Replay**, and ③ **Architectural** -based methods.

① **Regularization**. To mitigate forgetting, regularization-based methods apply regulations to penalize the changes in the critical parameters learned from previous data. [29] improve the traditional EWC [128] by recalling previously learned knowledge through the pre-trained parameters, and the method continually learns new information using a multi-task learning objective. [120] compute the importance of each unit (*i.e.*, attention head and neuron) to the general knowledge in the LM using a proxy based on model robustness to preserve learned knowledge. When continually learning new domains, the approach prevents catastrophic forgetting of the general and domain knowledge and encourages knowledge transfer via soft-masking and contrastive loss.

② **Replay**. These methods generally reduce forgetting by replaying previous training

data when learning new data. Assuming that the initial pre-training corpus is available, [81] use a gradual decay mix-ratio to adjust the quantity of the pre-training corpus mixed in the new data when learning sequentially. ELLE [203] and CT0 [219] also mix the old data while learning new data. However, ELLE starts the pre-training from a newly initialized and relatively small BERT [46] and GPT [204], while CT0 continues learning from T0-3B [216], a pre-trained and instruction-tuned model.

③ **Architectural**. These methods typically alleviate forgetting by using different subsets of parameters for distinct tasks or domains. [249, 87, 119] freeze the original parameters of the LM to preserve the learned knowledge and add lightweight tunable parameters for continually learning new knowledge. [249] add separate adapters [85] for each new task, while [119] let all domains share adapters and employ task masks to protect critical neurons from being updated. DEMix-DAPT [71] replaces every FFN layer in Transformer with a separate domain expert mixture layer, containing one expert per domain. When learning new knowledge, they only train the newly added expert in each DEMix layer while fixing all other experts. Similarly, Lifelong-MoE [30] progressively expands experts to increase model capacity for learning new knowledge, and mitigates forgetting by freezing previously trained experts and gatings with output-level regularization. [203] enlarge the model's width and depth to attain learning efficiency and employ memory replay to reduce forgetting.

④ **Other Methods**. [89] meta-trains an importance-weighting model to reweight the per-token loss of the continual data stream, intending to quickly adapt the base LM to new knowledge. [193] apply *k*NN-LM [122] to continual learning from streaming data and selectively store hard cases in a non-parametric memory, significantly improving the data-wise and model-wise scalability. [276] assess self-information-update in LLMs via CL and mitigate exposure bias by incorporating the selection of relevant facts into training losses.

Continual Knowledge Editing. [146, 135, 93] and [76] propose a more realistic setting that a deployed LM should be constantly corrected to fix its prediction errors, showing the potential to align the model with the latest world knowledge. [146] benchmark the continual model refinement problem by implementing traditional CL methods. [135] and [76] freeze the LM's original parameters and continually introduce trainable neurons to the FFN layer to rectify problematic model behaviours. In contrast, [76] learn to cache a chosen layer's activations in a key-value-based codebook and retrieve activations when previous similar edits have been performed. Without influencing unrelated inputs, it can efficiently edit the model thousands of times in a row while generalizing edits to previously unseen inputs.

Category	Representative Method	Base LM	LM Params	Augmentation	No Training	Black -box
Knowledge Editing	MEND [177]	T5 (11B)	❄	auxiliary model	✗	✗
	ROME [172]	GPT-J (6B)	🔥	–	✔	✗
	CaliNET [49]	T5 (0.7B)	❄	+params	✗	✗
	MEMIT [173]	GPT-NeoX (20B)	🔥	–	✔	✗
Continual Learning	K-Adapter [249]	RoBERTa (0.3B)	❄	+params	✗	✗
	CT0 [219]	T0 (3B)	🔥	memory	✗	✗
	DSA [120]	RoBERTa (0.1B)	🔥	–	✗	✗
Memory-enhanced	MemPrompt [166]	GPT-3 (175B)	❄	memory+retriever	✔	✔
	SERAC [178]	T5 (0.7B)	❄	memory +auxiliary model	✗	✔
	MeLLo [296]	GPT-3.5 (175B)	❄	memory+retriever	✔	✔
Retrieval-enhanced	IRCoT [244]	GPT-3.5 (175B)	❄	retriever	✔	✔
	RARR [60]	PaLM (540B)	❄	search engine +auxiliary model	✔	✔
	DecomP [124]	GPT-3 (175B)	❄	retriever	✔	✔
	ReAct [268]	PaLM (540B)	❄	search engine	✔	✔
	FLARE [104]	GPT-3.5 (175B)	❄	retriever/search engine	✔	✔
Internet-enhanced	[132]	Gopher (280B)	❄	search engine	✔	✔
	CRITIC [66]	GPT-3.5 (175B)	❄	various tools	✔	✔
	Chameleon [160]	GPT-4 (?B)	❄	various tools	✔	✔

Table 2.1: Comparison between representative methods.

2.2.3 Explicitly Align LMs with World Knowledge

Although altering the knowledge implicitly stored in LLMs has shown to be effective [98, 173], it remains unclear whether it will affect the models' general abilities due to the complexity of neural networks. In contrast, explicitly augmenting LLMs with the latest information retrieved from various sources can effectively adapt the models to new world knowledge without affecting the original LLMs [176]. However, previous retrieval-augmented methods [112, 73, 138, 96, 14, 101, 117] usually jointly train a retriever and an LM in an end-to-end fashion, making it challenging to apply to a deployed LLM (*e.g.*, GPT-3). Recently, researchers have focused on equipping a fixed LLM with external memory (*memory-enhanced*; §2.2.3.1), an off-the-shelf retriever (*retrieval-enhanced*; §2.2.3.2), or Internet (*Internet-enhanced*; §2.2.3.3) to cope with this issue.

2.2.3.1 Memory-enhanced

Pairing a static LLM with a growing non-parametric memory enables it to capture information beyond its memorized knowledge during inference [261]. The external memory can store a recent *corpus* or *feedback* that contains new information to guide the model generation.

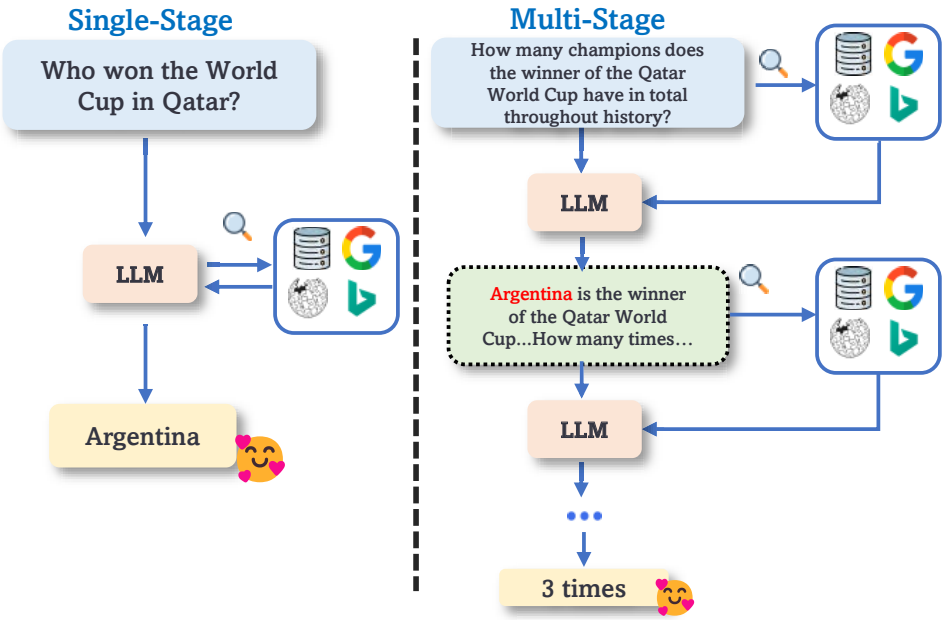


Figure 2.3: **Single-Stage** (left) typically retrieves once, while **Multi-Stage** (right) involves multiple retrievals or revisions to solve complex questions (§2.2.3.2).

Storing Corpus or Documents. *k*NN-LM [122] stores every `<context, token>` as key-value pairs from a corpus in memory. During inference, it calculates the probability of the next token by interpolating a fixed LM with a distribution retrieved from the *k* nearest tokens in the memory. Following this vein, [80, 50, 5] improve the efficiency of *k*NN-LM by skipping unnecessary retrieval. [174] build an additional graph neural network to aggregate information from the retrieved context for better generation. [193] improve the scalability of *k*NN-LM for continual learning, while [228] apply it for zero-shot inference on downstream tasks.

Storing Feedback or Corrections. Inspired by the fact that humans can learn from past mistakes, this line of work stores user feedback in memory to fix the model's problematic predictions and avoid similar errors in the future. By querying the memory, the base LLM gains *editability* to update its outdated knowledge. [116, 236] train an auxiliary corrector to apply feedback to repair the model output. [42] allow users to interact with the system to check its facts and reasoning and correct it when it is wrong. Similarly, [166] equip GPT-3 with a growing memory, where the key is a misunderstanding question, and the value is the corrective feedback. Instead of storing user feedback, [178, 296] explicitly preserve updated knowledge in memory. Given an input, [178] first apply a classifier to determine if a relevant edit exists in the memory and perform knowledge updating through a counter-

factual model. Conversely, [296] decompose complex questions and ask the base model to generate a temporary answer. They revise the model output when the generated answer contradicts the retrieved facts from memory.

2.2.3.2 Retrieval-enhanced

Leveraging an off-the-shelf retriever and the in-context learning ability of LLMs [16], this line of work designs better retrieval strategies to incorporate world knowledge into a fixed LLM through prompting, which can be divided into *single-stage* and *multi-stage* (Fig. 2.3).

Single-Stage. To ground the model with external knowledge during generation, [208, 232] adopt zero-shot and few-shot retrieval respectively and directly prepend the retrieved documents to the input without changing the base LLM. [295] retrieve similar edit demonstrations for each input and perform in-context knowledge editing. Compared with gradient-based knowledge editing (§2.2.2.2), they have competitive editing performance with fewer side effects. Arguing that the general-purpose retrievers could be sub-optimal, [279] adopt a small source LM to provide LM-preferred signals to train an adaptive retriever. [169] employ a heuristic based on entity popularity and only retrieve relevant context when the input questions are less popular, which improves performance and reduces inference costs. Unlike above, to address the limited model's context length, [229] prepend each retrieved document separately to an LLM and then ensemble output probabilities from different passes.

Multi-Stage. When solving complex questions, retrieving information only once based on the input is often inadequate. This branch of work aims to transform single-stage retrieval into multi-stage retrieval in order to solve complex tasks, usually by leveraging reasoning. [244] interleave knowledge retrieval with CoT (Chain-of-Thoughts) [255] generation to solve complex multi-step reasoning questions. Similarly, [197, 124, 268, 104, 223] decompose questions into sub-questions to provide a specific context for retrieval with model generation. [189, 33, 94] further enable the usage of different tools to solve various tasks. Unlike the simple *retrieve-then-read* paradigm, [123] pass intermediate messages between an LLM and a retriever; [60, 79, 293, 277] retrieve after generation and perform post-edit revisions for more faithful outputs. [191] iteratively revise ChatGPT to improve model responses using feedback and external knowledge. [57] teach LLMs themselves to search for knowledge from external knowledge graphs (KGs) via prompting and simplify search-

ing into a multi-hop decision sequence, allowing explainable decision-making of the processes.

2.2.3.3 Internet-enhanced

Prior retrieval-augmented work relies on *static* or *offline* knowledge sources (*e.g.*, Wikipedia dump), which may not be sufficiently up-to-date or complete for tasks that require the latest knowledge [114, 286, 140]. A recent trend uses the whole web as the knowledge source and equips LLMs with the Internet to support real-time information seeking [180, 175, 129, 231, 201, 152]. [132] augment few-shot QA prompting with the context retrieved from Google search. [197, 104] interleave reasoning with web search. Recently, tools such as LangChain [24] and ChatGPT Plugins [185] connect a deployed LLM to the Internet without training, making them more powerful for solving knowledge-intensive tasks. Beyond search engines, [268, 145, 189, 267, 66, 160] treat LLMs as central planners and compose various plug-and-play tools for solving complex questions.

2.2.4 Comparison and Discussion

We present the comparison of different methods in Table 2.1 and in Fig. 2.2, and the characteristics of different methods in Table 2.2. Note that in Table 2.1, 🔥 means the parameters of the original LM are modified, while ❄ means they are unchanged; **Augmentation** means additional components used; **No Training** indicates the method does not require additional training; **Black-box** refers to whether the method suits non-publicly available models (*e.g.*, no model architecture, parameters, activations, or gradients are available).

Category	Large Scale	No Side Effects	Persistent
Knowledge Editing (§2.2.2.2)	✗	✗	✔
Continual Learning (§2.2.2.3)	✔	✗	✔
Retrieval-based (§2.2.3)	✗	✔	✗

Table 2.2: High-level comparison of characteristics of different approaches.

Discussion of Implicit Methods (§2.2.2). Compared to naive re-training or fine-tuning, KE and CL can effectively update obsolete knowledge in LLMs while minimizing interference on irrelevant ones. We identify their major differences: ① **Scale**. Existing KE methods focus on updating small-scale and localized knowledge, typically on synthetic fact pairs

[177, 172]. While one can perform thousands of edits simultaneously [173], updating enormous knowledge in LLMs may be cumbersome. In contrast, CL enhances models' adaptability via tuning larger-scale parameters, thus updating more knowledge at scale [98]. However, KE provides fine-grained controllability when specific knowledge needs to be altered, which is unachievable by CL; ② **Forgetting**. Applying KE methods on LLMs frequently in response to the ever-changing world is sub-optimal due to catastrophic forgetting [93, 76]; CL mitigates this issue when learning new knowledge; ③ **Cost**. CL is generally more computationally expensive than KE due to larger-scale weight updating.

Discussion of Explicit Methods (§2.2.3). Explicit methods use new knowledge retrieved from the world to override old knowledge in an LLM during generation. Despite being effective, memory- and retrieval-enhanced methods must periodically maintain the external memory and the knowledge sources in response to the ever-changing world [114]. Conversely, Internet-enhanced methods enable real-time knowledge seeking, although potentially suffering from noisy and low-quality web content [140, 161]. Compared to single-stage retrieval, multi-stage retrieval can solve more complex problems. Nevertheless, they may interrupt the generation with multiple retrievals or revisions, leading to considerable inference overheads [223].

Updating LLMs Implicitly or Explicitly? We observe an increasing trend of explicitly aligning LLMs with world knowledge while keeping the model untouched (Table 2.1). Compared to explicit approaches: ① **Applicability**. Implicit methods usually require modifying LLM's parameters or gradients, making it challenging to update closed-source models; ② **Side Effects**. Although constraints have been added to avoid editing irrelevant knowledge [177, 173] or forgetting general knowledge [98], modifying the LLM's parameters inevitably has side effects that may hurt the performance, which is hard to estimate due to the complexity of neural networks [15]; ③ **Efficiency**. Implicit methods typically require training, while most explicit methods leverage a fixed LLM and an off-the-shelf retriever, erasing the necessity of training. However, explicit methods do not directly modify the intrinsic knowledge within LLMs; instead, they rely on on-the-fly retrieval during inference, resulting in a notable increase in the computational cost of inference.

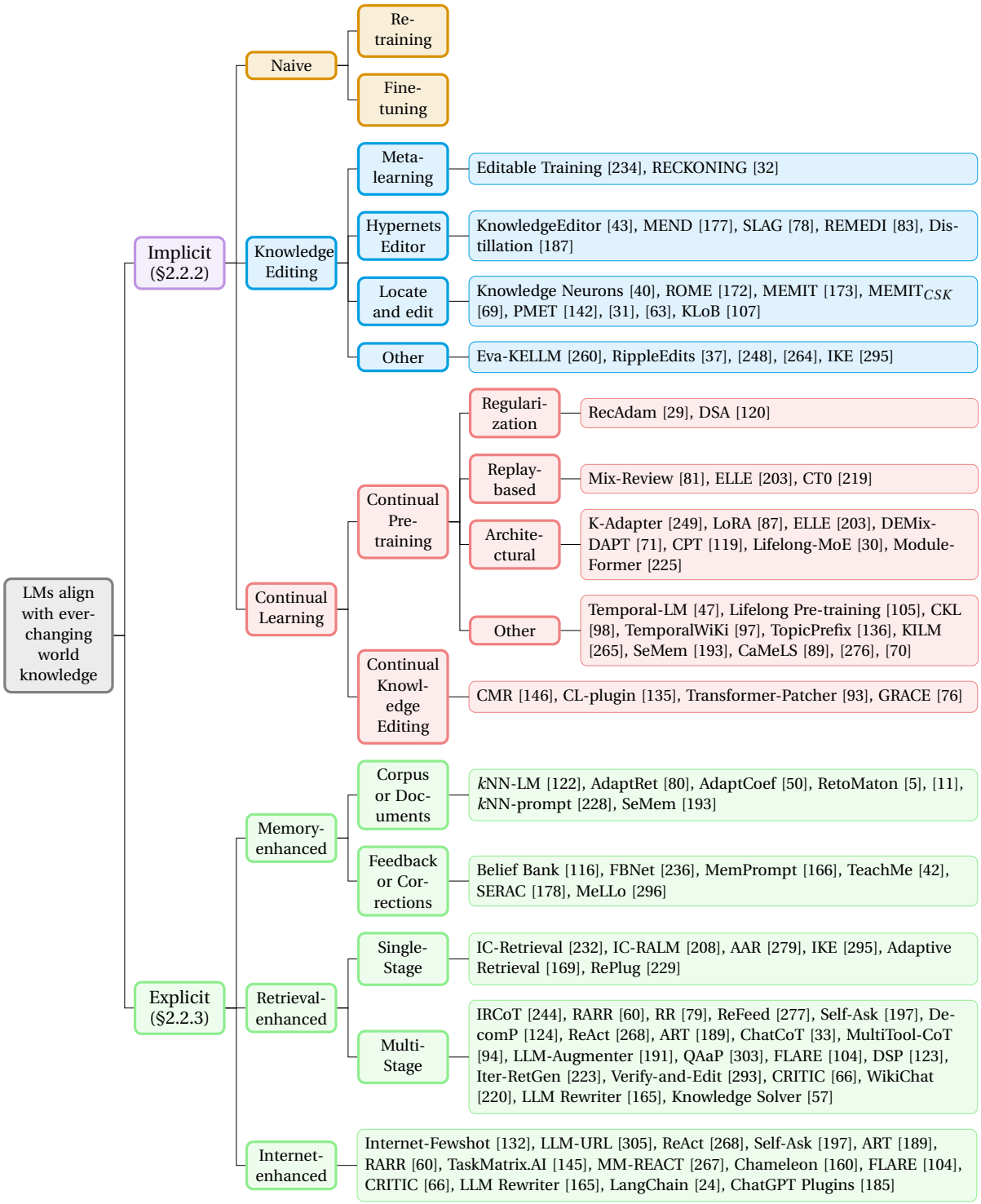


Figure 2.4: Taxonomy of methods to align LMs with the ever-changing world knowledge.

CHAPTER 3

LEARNING KNOWLEDGE VIA DATA EFFICIENT TUNING

In this chapter, we present our data-centric approaches to efficiently and effectively label raw data for training a task-specific language model. Our focus is on the task-oriented DST (Dialogue State Tracking) task, which plays a crucial role in task-oriented dialogue systems widely used in everyday life. As these systems must adapt to emerging data and domains to assist end users effectively, DST serves as an ideal testbed for incorporating new knowledge into a pre-trained language model. This chapter is based on one published paper at EMNLP 2023 – Turn-Level Active Learning for Dialogue State Tracking [291].

3.1 Introduction

Dialogue state tracking (DST) constitutes an essential component of task-oriented dialogue systems. The task of DST is to extract and keep track of the user's intentions and goals as the dialogue progresses [257]. Given the dialogue context, DST needs to predict all *(domain-slot, value)* at each turn. Since the subsequent system action and response rely on the predicted values of specified domain-slots, an accurate prediction of the dialogue state is vital.

Despite the importance of DST, collecting annotated dialogue data for training is expensive and time-consuming, and how to efficiently annotate dialogue is still challenging. It typically requires human workers to manually annotate dialogue states [18] or uses a Machines Talking To Machines (M2M) framework to simulate user and system conversations [222]. Either way, every turn in the conversation needs to be annotated because existing

DST approaches are generally trained in a fully supervised manner, where turn-level annotations are required. However, if it is possible to find the most informative and valuable turn in a dialogue to label, which enables the training of a DST model to achieve comparable performance, we could save the need to annotate the entire dialogue, and could efficiently utilize the large-scale dialogue data collected through API calls.

AL (Active Learning) aims to reduce annotation costs by choosing the most important samples to label [221, 56, 292]. It iteratively uses an acquisition strategy to find samples that benefit model performance the most. Thus, with fewer labelled data, it is possible to achieve the same or better performance. AL has been successfully applied to many fields in natural language processing and computer vision [218, 21, 52, 88]. However, the adoption of AL in DST has been studied very rarely. [263] have studied to use AL to reduce the labelling cost in DST, using a *dialogue-level* strategy. They select a batch of dialogues in each AL iteration and label the entire dialogues (*e.g.*, every turn of each dialogue), which is inefficient to scale to tremendous unlabelled data. To our knowledge, *turn-level* AL remains unstudied for the task of DST.

Turn	User	System	Dialogue State						
1	Can you tell me some info on the Avalon hotel?		**Domain**	hotel				taxi		
			Slot	name	book day	book people	book stay	arriveby	departure	destination
			Value	Avalon	None	None	None	None	None	None
2	The Avalon is a 4 star moderately priced guesthouse in the north with free internet. Would you like to book there?		**Domain**	hotel				taxi		
			Slot	name	book day	book people	book stay	arriveby	departure	destination
	Yes. Can you book it for 5 people on Saturday? We need rooms for 4 nights.		**Value**	Avalon	Saturday	5	4	None	None	None
⋮	⋮		⋮							
8	For what times?		**Domain**	hotel				taxi		
	I need a taxi so that it arrives by the time of my reservation at the restaurant.		**Slot**	name	book day	book people	book stay	arriveby	departure	destination
			Value	Avalon	Saturday	5	2	17:45	Avalon	Frankie and Bennys
9	Your taxi has been booked to take you from Avalon to Frankie and Bennys at 17:45. Your taxi will be a black Tesla and the contact number is 07715682347.		**Domain**	hotel				taxi		
			Slot	name	book day	book people	book stay	arriveby	departure	destination
	That sounds great. Thank you very much.		**Value**	Avalon	Saturday	5	2	17:45	Avalon	Frankie and Bennys
10	Will you need any thing else now?		**Domain**	hotel				taxi		
	No, thank you that will be all for me, goodbye.		**Slot**	name	book day	book people	book stay	arriveby	departure	destination
			Value	Avalon	Saturday	5	2	17:45	Avalon	Frankie and Bennys

All turns are used in training Only selected turns are used in training

Figure 3.1: An example of DST from the MultiWOZ dataset [18].

Furthermore, existing DST approaches [259, 82, 239, 302] treat each dialogue turn as a single, independent training instance with no difference. In fact, in the real-world, utterances in a dialogue have different difficulty levels [41] and do not share equal importance

in a conversation. For example, in Fig. 3.1 [1] , turn-1 is simple and only contains a single domain-slot and value (*i.e.*, *hotel-name=Avalon*), while turn-2 is more complex and generates three new domain-slots, *i.e.*, *hotel-book day, hotel-book people, hotel-book stay*. Given the limited labelling budget, it is an obvious choice to label turn-2 instead of turn-1 since the former is more informative[2]. In addition, we observe that the complete states of the dialogue session are updated at turn-8, while turn-9 and turn-10 simply show humans' politeness and respect without introducing any new domain-slots. Therefore, while the "last turn" has been studied before [148], it is often not the case that only the last turn of a dialogue session generates summary states. Using redundant turns such as turn-9 and turn-10 for training not only requires additional labelling but also possibly distracts the DST model since it introduces irrelevant context information, thus hindering the overall performance [266].

Built on these motivations, we investigate a practical yet rarely studied problem: *given a large amount of unlabelled dialogue data with a limited labelling budget, how can we annotate the raw data more efficiently and achieve comparable DST performance?* To this end, we propose a novel turn-level AL framework for DST that selects the most valuable turn from each dialogue for labelling and training. Experiments on MultiWOZ 2.0 and 2.1 show that our approach outperforms two strong DST baselines in the weakly-supervised scenarios and achieves comparable DST performance with significantly less annotated data, demonstrating both effectiveness and data efficiency. We summarize the main contributions of our work as follows:

- We propose a novel model-agnostic *turn-level* Active Learning framework for dialogue state tracking, which provides a more efficient way to annotate new dialogue data. To our best knowledge, this is the first attempt to apply turn-level AL to DST.
- The superiority of our approach is twofold: firstly, our approach strategically selects the most valuable turn from each dialogue to label, which largely saves annotation costs; secondly, using significantly reduced annotation data, our method achieves the same or better DST performance under the weakly-supervised setting.
- We investigate how turn-level AL can boost the DST performance by analyzing the query sizes, base DST models, and turn selection strategies.

We release our code, data, and model to facilitate future research at `https://github.com/hyintell/AL-DST`.

[1] Utterances at the left and the right sides are from user and system, respectively. Orange color denotes only the selected turn is used in the weakly-supervised training setup. Only two domains (*e.g. hotel, taxi*) are shown due to space limitation. (Best viewed in color).

[2] Here, *informative* refers to the turn that has more valid dialogue states.

3.2 Related Work

3.2.1 Dialogue State Tracking

Dialogue state tracking is an essential yet challenging task in task-oriented dialogue systems [257]. Recent state-of-the-art DST models [259, 126, 82, 270, 239, 134, 302, 90] using different architectures and mechanisms have achieved promising performance on complex multi-domain datasets [18, 54]. However, they are generally trained with extensive annotated data, where each dialogue turn requires comprehensive labelling.

To mitigate the cost of dialogue annotation, some works train DST models on existing domains and perform few-shot learning to transfer prior knowledge to new domains [259, 298], while others further improve transfer learning by pre-training extensive heterogeneous dialogue corpora using constructed tasks [258, 192, 149, 235]. Recently, [144, 148] propose a weakly-supervised training setup, in which only the last turn of each dialogue is used. For instance, **KAGE-GPT2** [148] is a generative model that incorporates a Graph Attention Network to explicitly learn the relationships between domain-slots before predicting slot values. It shows strong performance in both full and weakly-supervised scenarios on MultiWOZ 2.0 [18].

Despite the promising results, we have shown the potential drawbacks of using the last turns in §3.1. In contrast, in this work, we consider the differences between the turns and strategically select the turn that benefits the DST model the most from a dialogue for training.

3.2.2 Active Learning

Active Learning uses an acquisition strategy to select data to minimize the labelling cost while maximizing the model performance [221]. While AL has been successfully used in many fields, such as image segmentation [21], named entity recognition [224], text classification [218], and machine translation [281, 88], rare work has attempted to apply AL to DST. Moreover, recently proposed AL acquisition methods are, unfortunately, not applicable to turn-level DST since they are designed for specific tasks or models. For instance, BADGE [9] calculates gradient embeddings for each data point in the unlabelled pool and uses clustering to sample a batch, whereas we treat each turn within a dialogue as a minimum data unit and only select a single turn from each dialogue; therefore, the diversity-based methods are not applicable to our scenario. ALPS [280] uses the masked language model loss of BERT [46] to measure uncertainty in the downstream text classification task, while CAL [170] se-

lects contrastive samples with the maximum disagreeing predictive likelihood. Both are designed for classification tasks, so these strategies are not directly applicable. Therefore, studying an AL acquisition strategy that is suitable for DST is still an open question.

3.3 Preliminaries

We formalize the notations and terminologies used in the work as follows.

Active Learning (AL). AL aims to strategically select informative unlabelled data to annotate while maximizing a model's training performance [221]. This work focuses on pool-based active learning, where an unlabelled data pool is available. Suppose that we have a model $\mathscr{M}$, a small seed set of labelled data $\mathscr{L}$ and a large pool of unlabelled data $\mathscr{U}$. A typical iteration of AL contains three steps: (1) train the model $\mathscr{M}$ using $\mathscr{L}$; (2) apply an acquisition function $\mathscr{A}$ to select k instances from $\mathscr{U}$ and ask an oracle to annotate them; and (3) add the newly labelled data into $\mathscr{L}$.

Dialogue State Tracking (DST). Given a dialogue $D = \{(X_1, B_1), \ldots, (X_T, B_T)\}$ that contains T turns, X_t denotes the dialogue turn consisting of the user utterance and system response at turn t, while B_t is the corresponding dialogue state. The dialogue state at turn t is defined as $B_t = \{(d_j, s_j, v_j), 1 \leq j \leq J\}$, where d_j and s_j denote domain (*e.g. attraction*) and slot (*e.g. area*) respectively, v_j is the corresponding value (*e.g. south*) of the domain-slot, and J is the total number of predefined domain-slot pairs. Given the dialogue context up to turn t, *i.e.*$H_t = \{X_1, \ldots, X_t\}$, the objective of DST is to predict the value for each domain-slot in dialogue state B_t.

Labelling. Suppose that we have selected a turn t from the dialogue D ($1 \leq t \leq T$) to label. An oracle (*e.g.* human annotator) reads the dialogue history from X_1 to X_t and labels the current dialogue state B_t. In our experiments, we use the gold training set to simulate a human annotator.

Full vs. Weakly-supervised Training. Generally, the training dataset for DST is built in the way that each turn in a dialogue (concatenated with all previous turns) forms an individual training instance. That is, the input of a single training instance for turn t is defined as $M_t = X_1 \oplus X_2 \oplus \cdots \oplus X_t$, where $\oplus$ denotes the concatenation of sequences, and the output is the corresponding dialogue state B_t. By providing the entire dialogue utterances from

the first turn to turn t to the model, the information from the earlier turns is kept in the dialogue history. Let $\mathscr{D}_D$ be the set of training instances created for the dialogue D and t is the selected turn. Given the example in Fig. 3.1, for full supervision, all turns are used for training (*i.e.*, $\mathscr{D}_D = \{(M_1, B_1), \ldots, (M_T, B_T)\}$), whereas in weakly-supervised training, only the selected turn is used (*i.e.*, $\mathscr{D}_D = \{(M_t, B_t)\}$).

3.4 Methodology

In this section, we first define our turn-level AL-based DST framework, followed by the turn selection strategies.

3.4.1 Turn-Level AL for DST

Framework. Our turn-level AL-based DST consists of two parts. First, we use AL to model the differences between turns in a dialogue and find the turn that is the most beneficial to label. The pseudo-code of this step is shown in Algorithm 1. Second, after acquiring all labelled turns, we train a DST model as normal and predict the dialogue states for all turns in the test set for evaluation, as described in §3.3. In this work, we assume the training set is unlabelled and follow the cold-start setting (Algorithm 1 Line 4), where the initial labelled data pool $\mathscr{L} = \emptyset$. We leave the warm-start study for future work.

Active Learning Loop. In each iteration, as shown in Fig. 3.2, we first randomly sample k dialogues from the unlabelled pool $\mathscr{U}$. Then, we apply a turn acquisition function $\mathscr{A}$ and the intermediate DST model trained from the last iteration to each dialogue D to select an unlabelled turn (Algorithm 1 Line 10). It is noteworthy that we consider each turn within a dialogue as a minimum data unit to perform query selection. This is significantly different from [263], where they select a few dialogues from the unlabelled pool and label all turns as the training instances. Orthogonal to [263]'s work, it is possible to combine our turn-level strategy with dialogue-level AL. However, we leave it as future work because the AL strategies to select dialogues and turns could be different to achieve the best performance. In this work, we focus on investigating the effectiveness of AL strategies for turn selection.

To avoid overfitting, we re-initialize the base DST model and re-train it on the current accumulated labelled data $\mathscr{L}$. After R iterations, we acquire the final training set $\mathscr{L}$.

Algorithm 1 Turn-level AL for DST

Require: Initial DST model $\mathcal{M}$, unlabelled dialogue pool $\mathcal{U}$, labelled data pool $\mathcal{L}$, number of queried dialogues per iteration k, acquisition function $\mathcal{A}$, total iterations R

if $\mathcal{L} \neq \emptyset$ **then**
  $\mathcal{M}_0 \leftarrow$ Train $\mathcal{M}$ on $\mathcal{L}$ ▷ Warm-start
else
  $\mathcal{M}_0 \leftarrow \mathcal{M}$ ▷ Cold-start
end if
for iterations $r = 1 : R$ **do**
  $\mathcal{X}_r = \emptyset$
  $\mathcal{U}_r \leftarrow$ Random sample k dialogues from $\mathcal{U}$
  for dialogue $D \in \mathcal{U}_r$ **do**
    $X \leftarrow \mathcal{A}(\mathcal{M}_{r-1}, D)$ ▷ Select a turn
    $\mathcal{X}_r = \mathcal{X}_r \cup \{X\}$
  end for
  $\mathcal{L}_r \leftarrow$ Oracle labels $\mathcal{X}_r$
  $\mathcal{L} = \mathcal{L} \cup \mathcal{L}_r$
  $\mathcal{U} = \mathcal{U} \setminus \mathcal{U}_r$
  $\mathcal{M}_r \leftarrow$ Re-initialize and re-train $\mathcal{M}$ on $\mathcal{L}$
end for
return $\mathcal{L}$ ▷ The final training set

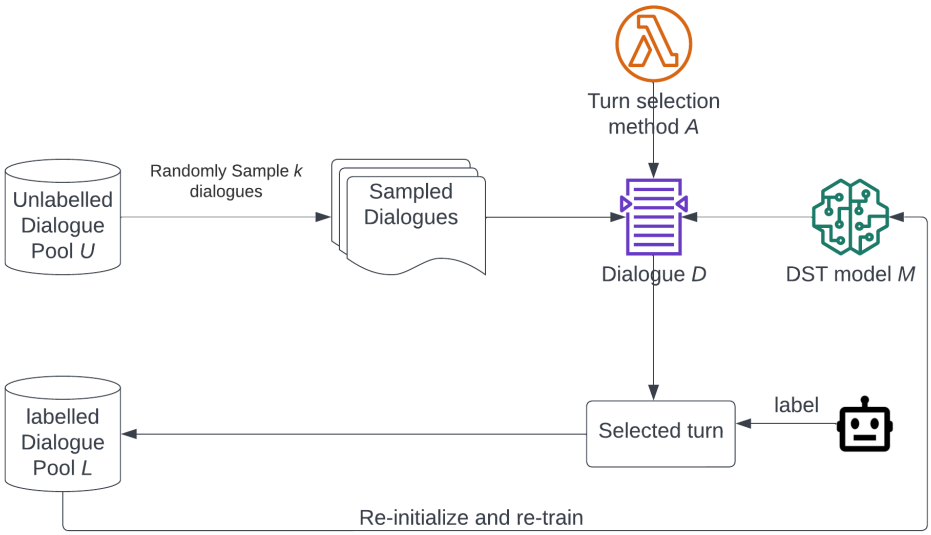


Figure 3.2: A single iteration of AL loop.

3.4.2 Turn Selection Strategies

As mentioned in §3.2.2, recently proposed AL acquisition strategies are not applicable to DST. Therefore, we adapt the common uncertainty-based acquisition strategies to select a turn from a dialogue:

Random Sampling (RS). We randomly select a turn from a given dialogue. Despite its simplicity, RS acts as a strong baseline in literature [221, 263, 52].

$$X = \text{Random}(M_1, \ldots, M_T) \tag{3.1}$$

where T is the total number of turns in the dialogue.

Maximum Entropy (ME). [137] Entropy measures the prediction uncertainty of the dialogue state in a dialogue turn. In particular, we calculate the entropy of each turn in the dialogue and select the highest one. To do that, we use the base DST model to predict the value of the jth domain-slot at turn t, which gives us the value prediction distribution $\mathbf{P}_t^j$. We then calculate the entropy of the predicted value using $\mathbf{P}_t^j$ (Eq. (3.2)):

$$\mathbf{e}_t^j = -\sum_{i=1}^{V} \mathbf{p}_t^j[i] \log \mathbf{p}_t^j[i] \tag{3.2}$$

$$\mathbf{e}_t = \sum_{j=1}^{J} \mathbf{e}_t^j \tag{3.3}$$

$$X = \text{argmax}(\mathbf{e}_1, \ldots, \mathbf{e}_T) \tag{3.4}$$

where V is all possible tokens in the vocabulary. We then sum the entropy of all domain-slots as the turn-level entropy (Eq. (3.3)) and select the maximum dialogue turn (Eq. (3.4)).

Least Confidence (LC). LC typically selects instances where the most likely label has the lowest predicted probability [39]. In DST, we use the sum of the prediction scores for all J domain-slots to measure the model's confidence when evaluating a dialogue turn, and select the turn with the minimum confidence:

$$\mathbf{c}_t = \sum_{j=1}^{J} \mathbf{c}_t^j \tag{3.5}$$

$$X = \text{argmin}(\mathbf{c}_1, \ldots, \mathbf{c}_T) \tag{3.6}$$

where $\mathbf{c}_t^j = \max(\text{logits}_t^j)$ denotes the maximum prediction score of the jth domain-slot at turn t and logits_t^j is the predictive distribution.

3.5 Experiments

3.5.1 Setup

Datasets. We evaluate the weakly-supervised DST performance on the MultiWOZ 2.0 [18] and MultiWOZ 2.1 [54] datasets[3] as they are widely adopted in DST. We use the same pre-processing as [148] and [235], and focus on five domains (*i.e.*, *restaurant, train, hotel, taxi, attraction*). The statistics of the datasets are summarized in Table 3.1.

		MultiWOZ2.0	MultiWOZ2.1
Train	# Dialogues	7888	7888
	# Domains	5	5
	# Slots	30	30
	# Total turns	54945	54961
	# Last turns	7888	7888
	# Avg. turns per dialogue	6.97	6.97
	# Max turns per dialogue	22	22
	# Min turns per dialogue	1	1
Validation	# Dialogues	1000	1000
	# Total turns	7374	7374
Test	# Dialogues	1000	999
	# Total turns	7372	7368

Table 3.1: Statistics of the datasets in the experiments.

Base DST Model. We use **KAGE-GPT2** [148] as the base DST model to implement all experiments. KAGE-GPT2 is a generative model that incorporates a Graph Attention Network to explicitly learn the relationships between domain-slots before predicting slot values. It shows strong performance in both full and weakly-supervised scenarios on MultiWOZ 2.0 [18]. To show that the effectiveness of our AL framework is not tied to specific base models, we also experiment with an end-to-end task-oriented dialogue model **PPTOD** [235]. PPTOD pre-trained on large dialogue corpora gains competitive results on DST in the low-resource settings.

[3]We also tried to use the SGD dataset [210]. However, the PPTOD model is already pre-trained on this dataset, making it unsuitable for downstream evaluation. KAGE-GPT2 requires the predefined ontology to build a graph neural network, but SGD does not provide all possible values for non-categorical slots (See §3.7).

3.5.2 Implementation Details

We use the official release of KAGE-GPT2[4] [148] and PPTOD[5] [235] to implement our turn-level AL framework. All experiments were done with a NVIDIA T4 GPU.

KAGE-GPT2. We use the `L4P4K2-DSGraph` model setup and follow its sparse supervision (last turn) hyperparameter settings. Specifically, the loaded pre-trained GPT-2 model has 12 layers, 768 hidden size, 12 heads and 117M parameters, which is provided by HuggingFace[6]. AdamW optimizer with a linear decay rate 1×10^{-12} is used when training. The GPT-2 component and the graph component are jointly trained, with the initial learning rates are 6.25×10^{-5} and 8×10^{-5} respectively. The training batch size used is 2, while the batch size for validation and evaluation is 16.

PPTOD. We use the released `base` checkpoint, which is initialized with a T5-base model with around 220M parameters. $\text{PPTOD}_{\text{base}}$ is pre-trained on large dialogue corpora, for more details, we refer readers to the original paper. When training, Adafactor optimizer is used and the learning rate is 1×10^{-3}. Both training, validation, and evaluation batch size used is 4.

Turn Selection. During each AL iteration, we use the trained model from the last iteration to evaluate all the turns within a dialogue and then select a turn based on the acquisition strategy.

Training. At the end of each iteration, we re-initialize a new pre-trained GPT-2 model for KAGE-GPT2 or re-initialize a new model from the released pre-trained `base` checkpoint for PPTOD, and then train the model as usual with all current accumulated labelled turns. We train the DST model for 150 epochs using the current accumulated labelled pool $\mathscr{L}$, and early stop when the performance is not improved for 5 epochs on the validation set. Importantly, instead of using the full 7,374 validation set, we only use the last turn of each dialogue to simulate the real-world scenario, where a large amount of annotated validation set is also difficult to obtain [194]. However, we use the full test set when evaluating.

[4] `https://github.com/LinWeizheDragon/Knowledge-Aware-Graph-Enhanced-GPT-2-for-Dialogue-State-Tracking`

[5] `https://github.com/awslabs/pptod`

[6] `https://huggingface.co/models`

3.5.3 Evaluation Metrics

Joint Goal Accuracy (JGA). We use **JGA** to evaluate DST performance, which is the ratio of correct dialogue turns. It is a strict metric since a turn is considered as correct if and only if all the slot values are correctly predicted.

Slot Accuracy (SA). Following the community convention, although it is not a distinguishable metric [127], we also report **SA**, which compares the predicted value with the ground truth for each domain-slot at each dialogue turn.

Reading Cost (RC). Additionally, we define a new evaluation metric, **RC**, which measures the number of turns a human annotator needs to read to label a dialogue turn. As shown in Fig. 3.1, to label the dialogue state B_t at turn t, a human annotator needs to read through the dialogue conversations from X_1 to X_t to understand all the domain-slot values that are mentioned in the dialogue history:

$$\text{RC} = \frac{\sum_{i=1}^{|\mathscr{L}|} \frac{t}{T_{D_i}}}{|\mathscr{L}|} \tag{3.7}$$

where $|\mathscr{L}|$ denotes the total number of annotated dialogues and T_{D_i} is the number of turns of the dialogue D_i. If all last turns are selected, then RC = 1, in which case the annotator reads all turns in all dialogues to label, resulting high cost. Note that we take JGA and RC as primary evaluation metrics.

3.5.4 Baselines

Our main goal is to use AL to actively select the most valuable turn from each dialogue for training, therefore reducing the cost of labelling the entire dialogues. We evaluate the effectiveness of our approach from two angles. First, we compare DST performance of two settings *without* involving AL to show the benefits that AL brings:

- **Full Data (100%)**: all the turns are used for training, which shows the upper limit of the base DST model performance.
- **Last Turn (14.4%[7])**: following [144] and [148], for each dialogue, only the last turn is used for training.

Second, when using AL, we compare our turn-level framework with the dialogue-level approach:

[7] $14.4\% = \frac{\#\text{ turns used}}{\#\text{ total turns}} = \frac{7888}{54945}$

- **CUDS (~14%)** [263]: a dialogue-level method that selects a batch of dialogues in each AL iteration based on the combination of labelling cost, uncertainty, and diversity, and uses all the turns for training. We carefully maintain the number of selected dialogues in each iteration so that the total number of training instances is roughly the same (*i.e.*, $k \simeq 2000$) for a fair comparison.
- **Selected Turn (14.4%)**: we apply Algorithm 1 and set $\mathscr{U} = 7888$, $\mathscr{L} = \emptyset$, $k = 2000$ and use the turn selection methods mentioned in §3.4.2 to conduct experiments. As a trade-off between computation time and DST performance, here we use $k = 2000$; however, we find that a smaller k tends to have a better performance (§3.6.2). Given $k = 2000$, we have selected 7,888 turns after four rounds, and use them to train a final model.

3.6 Results & Analysis

3.6.1 Main Results

We report the final results after the four AL iterations in Table 3.2. We present the intermediate results in Fig. 3.3. **RS**, **LC** and **ME** are active turn selection methods mentioned in §3.4.2. Note that we take JGA and RC as primary evaluation metrics since SA is indistinguishable [127].

Training Data	Model	MultiWOZ 2.0			MultiWOZ 2.1		
		JGA ↑	SA ↑	RC ↓	JGA ↑	SA ↑	RC ↓
		Without Active Learning					
Full Data (100%)	$\text{PPTOD}_{\text{base}}$	53.37±0.46	97.26±0.02	100	57.10±0.51	97.94±0.02	100
	KAGE-GPT2	54.86±0.12	97.47±0.02	100	52.13±0.89	97.18±0.02	100
Last Turn (14.4%)	$\text{PPTOD}_{\text{base}}$-LastTurn	43.83±1.55	96.87±0.06	100	45.94±0.72	97.11±0.04	100
	KAGE-GPT2-LastTurn	50.43±0.23	97.14±0.01	100	49.12±0.13	97.05±0.02	100
		With Active Learning ($k = 2000$)					
CUDS (~14%)*	$\text{PPTOD}_{\text{base}}$+CUDS	43.06±0.04	96.01±0.02	100	43.57±1.16	96.16±0.01	100
	KAGE-GPT2+CUDS	47.06±1.43	96.14±0.07	100	47.56±1.07	96.33	100
Selected Turn (14.4%) (Ours)	$\text{PPTOD}_{\text{base}}$+RS	43.71±0.81	96.64±0.08	58.73±28.7	46.96±0.18	96.56±0.06	**58.55**±28.5
	$\text{PPTOD}_{\text{base}}$+LC	45.79±0.35	97.06±0.04	85.21±19.7	47.37±0.32	96.97±0.05	81.95±24.6
	$\text{PPTOD}_{\text{base}}$+ME	**46.92**±0.79	**97.12**±0.05	**57.37**±32.9	**48.21**±1.00	**97.33**±0.12	67.68±30.1
	KAGE-GPT2+RS	50.37±0.52	97.11±0.06	**58.58**±28.7	46.98±0.64	96.81±0.07	**58.48**±28.5
	KAGE-GPT2+LC	50.56±0.07	97.10±0.01	70.51±30.3	48.13±0.20	96.94±0.01	79.41±24.0
	KAGE-GPT2+ME	**51.34**±0.05	**97.16**±0.05	62.58±28.5	**50.02**±1.10	**97.13**±0.10	71.02±26.7

Table 3.2: The mean and standard deviation of joint goal accuracy (%), slot accuracy (%) and reading cost (%) after the final AL iteration on the test sets. *: we re-implement using [263]'s method.

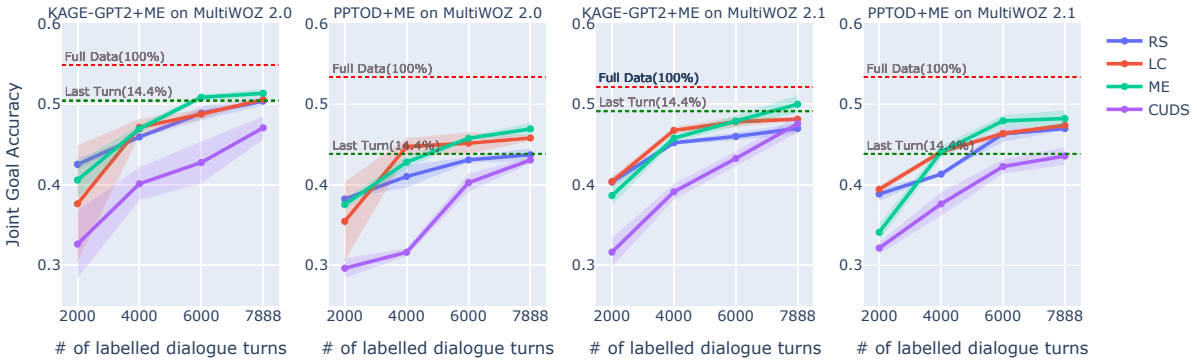


Figure 3.3: Joint goal accuracy on test sets of AL over four iterations with $k = 2000$ dialogues queried per iteration.

Our *turn-level* AL strategy improves DST performance. From Table 3.2, we first observe that, using the same amount of training data (14.4%), our proposed AL approach (*i.e.*, $\texttt{PPTOD}_{\texttt{base}}$+ME and KAGE-GPT2+ME) outperforms the non-AL settings, **Last Turn**, in terms of both joint goal accuracy and slot accuracy. Specifically, compared with $\texttt{PPTOD}_{\texttt{base}}$+LastTurn, our $\texttt{PPTOD}_{\texttt{base}}$+ME significantly boosts the JGA by 3.1% on MultiWOZ 2.0 and 2.3% on MultiWOZ 2.1. KAGE-GPT2+ME also improves its baselines by around 0.9% on both datasets. Compared with the dialogue-level AL strategy **CUDS**, our turn-level methods improve the JGA by a large margin (2.3%~4.3% on both datasets). Considering that DST is a difficult task [18, 259, 134], such JGA improvements demonstrate the effectiveness of our turn-level AL framework, which can effectively find the turns that the base DST model can learn the most from.

Our *turn-level* AL strategy reduces annotation cost. The reading costs (RC) of $\texttt{PPTOD}_{\texttt{base}}$+ME and KAGE-GPT2+ME drop by a large margin (around 29%~43%) compared to the Last Turn and CUDS settings, indicating the benefits and necessity of selecting dialogue turns. This significantly saves the annotation cost because a human annotator does not need to read the entire dialogue to label the last turn but only needs to read until the selected turn.

Our approach uses less annotated data can achieve the same or better DST performance. To further explore the capability of our AL approach, we plot the intermediate DST performance during the four iterations, as shown in Fig. 3.3. Notably, $\text{PPTOD}_{\text{base}}$ with Least Confidence (LC) and Maximum Entropy (ME) turn selection methods surpass the Last Turn baselines at just the second or third iteration on MultiWOZ 2.0 and MultiWOZ 2.1 respectively, showing the large data efficiency of our approach (only 7.3% / 10.9% data are used). This can be explained that $\text{PPTOD}_{\text{base}}$ is fine-tuned on so-far selected turns after each it-

eration and gains a more robust perception of unseen data, thus tending to choose the turns that are more beneficial to the model. In contrast, KAGE-GPT2 underperforms the Last Turn setting in early iterations, achieving slightly higher accuracy in the final round. Despite this, the overall performance of KAGE-GPT2 is still better than $\text{PPTOD}_{\text{base}}$ under the weakly-supervised settings. This is possibly because the additional graph component in KAGE-GPT2 enhances the predictions at intermediate turns and the correlated domain-slots [148]. However, when using CUDS, both DST models underperform a lot on both datasets, especially during early iterations. This indicates that the dialogue-level strategy, which does not distinguish the importance of turns in a dialogue, might not be optimal for selecting training data. In §3.6.2, we show that a smaller query size k can achieve higher data efficiency.

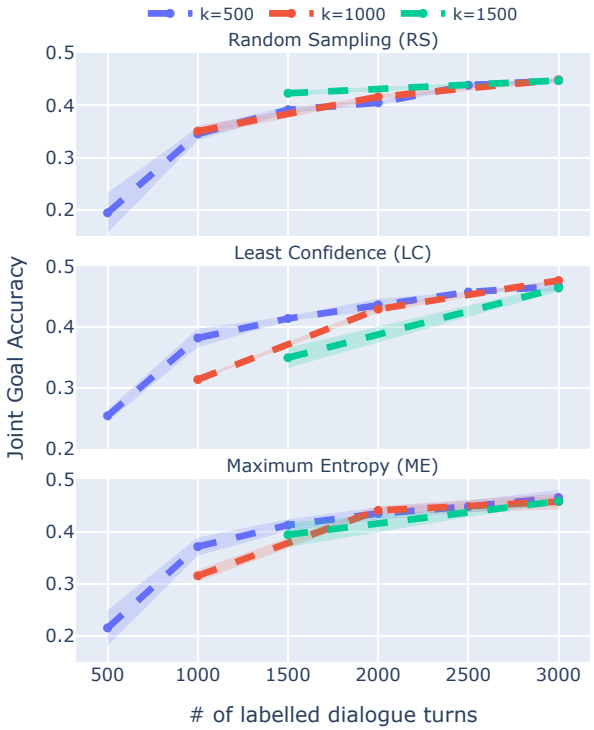


Figure 3.4: Joint goal accuracy on test sets of KAGE-GPT2 on MultiWOZ 2.0 with $k = 500, 1000, 1500$.

3.6.2 Ablation Studies

In this section, we further investigate the factors that impact our turn-level AL framework.

Effect of Dialogue Query Size. Theoretically, the smaller size of queried data per AL iteration, the more intermediate models are trained, resulting the better model performance. Moreover, a smaller query size is more realistic since the annotation budget is generally limited, and there is a lack of enough annotators to label large amounts of dialogues after each iteration. To this end, we initialize the unlabelled pool $\mathscr{U}$ by randomly sampling 3,000 dialogues from the MultiWOZ 2.0 training set, and apply our AL framework to KAGE-GPT2, using different query sizes, *i.e.*, $k = 500, 1000, 1500$, which leads to $6, 3, 2$ rounds respectively.

From Fig. 3.4, we first observe that smaller k improves the intermediate DST performance: when $k = 500$, both LC and ME strategies boost the accuracy by a large margin at the second iteration than $k = 1000$, and at the third iteration than $k = 1500$. This suggests that, with the same number of training data, the multiple-trained DST model gains the ability to have a more accurate perception of the unseen data. By calculating the prediction uncertainty of the new data, the model tends to choose the turns that it can learn the most from. In contrast, RS chooses a random turn regardless of how many AL rounds, therefore not showing the same pattern as LC and ME. Finally, we find a smaller k tends to achieve higher data efficiency when using LC and ME strategies. It is clear from the figure that $k = 500$ uses the least data when reaching the same level of accuracy. However, the drawback of a smaller query size is that it increases overall computation time as more intermediate models have to be trained. We provide a computational cost analysis in §3.6.3.

Effect of Base DST Model. It is no doubt that the base DST model is critical to our turn-level AL framework as it directly determines the upper and lower limit of the overall performance. However, we are interested to see how our approach can further boost the performance of different DST models. We randomly sample $\mathscr{U} = 500$ dialogues from the MultiWOZ 2.0 training set and set the query size $k = 100$ for both models. As shown in Fig. 3.5, we also report the results of the two models using the non-AL strategy of Last Turn, which can be considered as the lower performance baselines.

We first confirm that both $\text{PPTOD}_{\text{base}}$ and KAGE-GPT2 outperform their Last Turn baselines after applying our AL framework, demonstrating both data efficiency and effectiveness of our approach. Secondly, we notice that $\text{PPTOD}_{\text{base}}$ achieves comparable accuracy in the first two rounds, while KAGE-GPT2 nearly stays at 0 regardless of the turn selection methods, showing the superiority of $\text{PPTOD}_{\text{base}}$ under the extreme low-resource scenario. This is possibly because $\text{PPTOD}_{\text{base}}$ is pre-trained on large dialogue corpora thus gains few-shot learning ability [235], whereas only 200 training data are not enough for KAGE-GPT2 to be fine-tuned. However, in the later iterations, the performance of KAGE-GPT2 grows

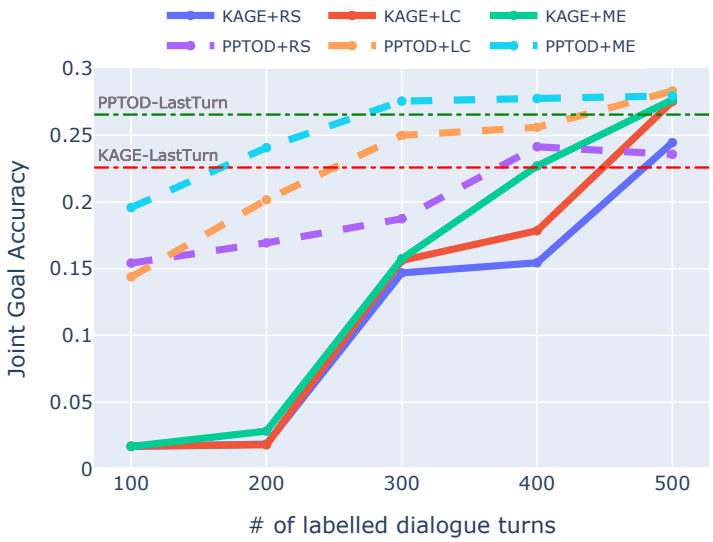


Figure 3.5: Joint goal accuracy on test sets of KAGE-GPT2 and $\text{PPTOD}_{\text{base}}$ on MultiWOZ 2.0 with $k = 100$. Results are averaged over three runs.

Method	KAGE-GPT2	$\text{PPTOD}_{\text{base}}$
LC	$76.51_{\pm 24.7}$	$81.13_{\pm 22.3}$
ME	$68.18_{\pm 29.1}$	$58.68_{\pm 31.5}$

Table 3.3: Reading Cost (RC) (%) of different turn selection methods. The lower the better.

significantly, especially when using the ME strategy, eventually reaching the same level as $\text{PPTOD}_{\text{base}}$. In contrast, the accuracy of $\text{PPTOD}_{\text{base}}$ increases slowly, indicating the model gradually becomes insensitive to the newly labelled data.

Effect of Turn Selection Strategy. From Fig. 3.3, while both ME and LC improve over the RS baseline, ME does not consistently outperform LC during AL iterations in terms of the joint goal accuracy, and vice versa. However, as shown in Table 3.2, LC results in a higher Reading Cost (RC) than ME, which means LC tends to select latter half of turns in dialogues. Conversely, ME significantly reduces RC in the last iteration (Fig. 3.6; Fig. 3.7) and is consistently better than LC and RS for both DST models (Fig. 3.5), which demonstrates the effectiveness of ME under small query size k.

We report their RC in Table 3.3, which also confirms that ME saves reading costs than LC. Examples of the turns selected by ME and LC in a dialogue are shown in §3.6.4 and §3.6.5.

Method	Total Annotation Cost ($) ↓
Full Dialogue	$z * (T * x + T * y)$
Last Turn	$z * (T * x + 1 * y)$
Selected Turn (Ours)	$z * (t * x + 1 * y)$, where $1 \leq t \leq T$

Table 3.4: Annotation cost estimation comparison of different methods.

Method	# of Training data (%) ↓	JGA ↑	RC ↓	Runtime (hour) ↓
Full data	21072 (100%)	46.7	100	2.3
Last Turn	3000 (14.2%)	41.4	100	0.6
ME (Ours)	3000 (14.2%)	44.3	59.3	1.6

Table 3.5: Computational cost comparison using KAGE-GPT2 on MultiWOZ 2.0 with $\mathscr{U}$ = 3000 and k = 1000.

3.6.3 Cost Analysis

Our AL-based method saves annotation costs and achieves comparable DST performance with traditional methods at the expense of increased computation time. In this section, we conduct a cost analysis, including computation and annotation costs.

We initialize the unlabelled pool $\mathscr{U}$ by randomly sampling 3,000 dialogues from the MultiWOZ 2.0 training set, and apply our AL framework to KAGE-GPT2, and set the query size as k = 1000. As shown in Table 3.5, our method improves JGA and RC than the Last Turn baseline, but with an increased runtime since our method requires three rounds of iteration.

Due to a lack of budget, we are unable to employ human annotators to evaluate the actual annotation cost. Instead, we conduct a theoretical cost analysis to show the potential cost reduction of our method. Suppose a dialogue D has T turns in total, and it takes x minutes for a human annotator to read each turn (*i.e.*, reading time), y minutes to annotate a single turn (*i.e.*, annotating time), z dollars per minute to hire a human annotator. Assuming our proposed method selects the tth ($1 \leq t \leq T$) turn to annotate. The total annotation cost, including the reading time and annotating time of three methods, are listed in Table 3.4. Since the Full Dialogue baseline takes each accumulated turn as a training instance (§3.3), it requires the highest annotation cost. Our method only annotates a single turn per dialogue, the same as the Last Turn baseline. Therefore, the annotation cost lies in the selected turn t, which is measured by RC in our experiments. As shown in Table 3.2 and discussed in §3.6.1, our method generally saves RC by a large margin (around 29%~43% across different models) compared to the Last Turn baseline and saves more compared to the Full data setting. Therefore, from a theoretical cost estimation point of view, our pro-

posed method can save annotation costs while maintaining DST performance.

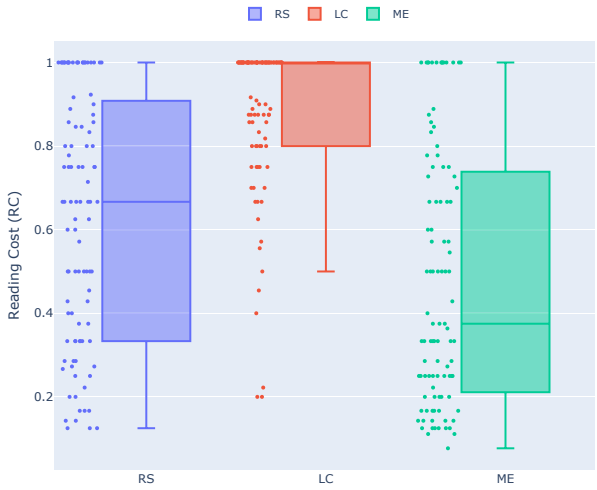


Figure 3.6: Visualization of the turns selected by $\text{PPTOD}_{\text{base}}$ at the final round ($k = 100$). ME reduces RC the most.

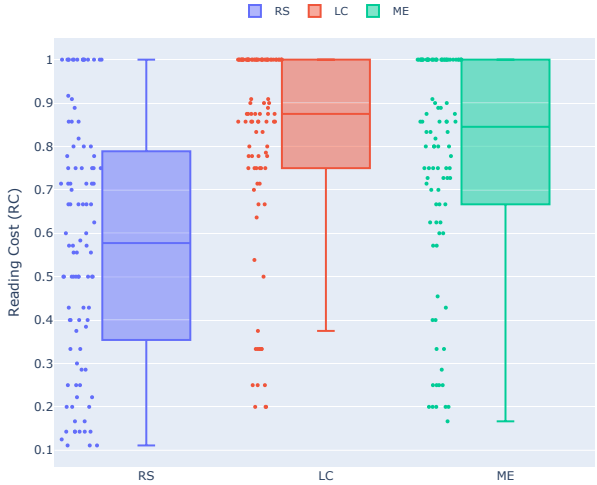


Figure 3.7: Visualization of the turns selected by KAGE-GPT2 at the final round ($k = 100$).

3.6.4 Visualization of Selected Turns

To clearly compare the reading costs of different turn selection methods, we visualize the distributions of the selected turns at the final round for the setting in §3.6.2, as shown in Fig. 3.6 and Fig. 3.7. A dot means a selected turn from a dialogue, while the ends of the box represent the lower and upper quartiles, and the median (second quartile) is marked by a

line inside the box. A higher RC means the turn is selected from the second half of the conversation (RC = 1 means the last turn is selected); thus, a human annotator needs to read most of the conversation to label its state, which is more costly. From the figures, overall, RS distributes randomly, while ME has a much lower reading cost than LC, especially for $\text{PPTOD}_{\text{base}}$.

3.6.5 Example of Selected Turns

Table 3.6, Table 3.7 and Table 3.8 present the examples of selected turns by ME and LC using $\text{PPTOD}_{\text{base}}$ from MultiWOZ 2.0. [S] and [U] denote the system and user utterance respectively, while *State* represents the dialogue states that are mentioned at the current turn. ✓marks the selected turn by the strategy and is the only turn in the dialogue used for training. Although not always the case, we can see that both ME and LC can select the earliest turn that summarizes the entire dialogue, which not only saves the need to read through the whole conversation but also keeps the valuable context information intact as much as possible. However, still, a more suitable AL query strategy for DST is worthy of being studied.

Table 3.6: Example (MUL0295) of the selected turn (marks by ✓) by $\text{PPTOD}_{\text{base}}$ using ME and LC.

	Dialogue MUL0295	**ME**	**LC**
Turn 1	[S]: [U]: i am looking for an expensive place to dine in the centre of town. *State: {restaurant-area=centre, restaurant-pricerange=expensive}*		
Turn 2	[S]: great kymmoy is in the centre of town and expensive. [U]: i want to book a table for 3 people at 14:00 on Saturday. *State: {restaurant-book day=saturday, restaurant-book people=3, restaurant-book time=14:00}*		
Turn 3	[S]: booking was successful. the table will be reserved for 15 minutes. reference number is: vbpwad3j. [U]: thank you so much. i would also like to find a train to take me to kings lynn by 10:15. *State: {train-destination=kings lynn, train-arriveby=10:15}*		✓
Turn 4	[S]: there are 35 departures with those criteria. what time do you want to leave? [U]: the train should arrive by 10:15 please on sunday please. *State: {train-day=sunday}*	✓	
Turn 5	[S]: how many tickets will you need? [U]: just 1 ticket. i will need the train id, cost of ticket and exact departure time as well. *State: {}*		
Turn 6	[S]: there is a train arriving in kings lynn on sunday at 09:58. it departs at 09:11 and costs 7.84 pounds. the train id is tr6088. [U]: great! that s all i needed. thanks a lot for the help. *State: {}*		

3.7 Summary

In this chapter, we present our data-centric approaches to efficiently and effectively label raw data for training a task-specific language model. We tackle the practical dialogue annotation problem by proposing a novel turn-level AL framework for DST, which strategically

Table 3.7: Example (MUL1068) of the selected turn by $\text{PPTOD}_{\text{base}}$ using ME and LC.

	Dialogue MUL1068	**ME**	**LC**
	[S]:		
Turn 1	[U]: i am looking for a gueshouse to stay in during my visit. i would like for it to be in the east section of town.		
	State: {hotel-area=east, hotel-type=guesthouse}		
	[S]: there are 6 entries, do you have any other preferences in hotel choice?		
Turn 2	[U]: i would like a 4 star place that's on the pricier side.		
	State: {hotel-pricerange=expensive, hotel-stars=4}		
	[S]: there are no 4 star expensive guesthouses on the east side of town. would you be interested in a cheap or moderate -ly priced 1 instead?		
Turn 3	[U]: i would like 1 that is cheap then.		
	State: {hotel-pricerangen=cheap}		
	[S]: autumn house is cheap and has free internet and parking . would you like me to book a reservation for you?		
Turn 4	[U]: yes please. can you book the reservation for saturday for 5 guests for a 2 night stay. thanks.		
	State: {hotel-book day=saturday, hotel-book people=5, hotel-book stay=2}		
	[S]: sorry, there are no rooms available for your specification -s. do you want to try a different day or a shorter stay?		
Turn 5	[U]: try for 1 night instead. if that works, can i have the reference number?		
	State: {hotel-book stay=1}		
	[S]: how about the autumn house for 1 night starting saturday and your reference number is svciwlwa.		
Turn 6	[U]: i also need to find something called gonville and caius college, can you help?	✓	
	State: {attraction-name=gonville and caius college}		
	[S]: gonville and caius college is located in the center of town on trinity street . anything else i can do for you?		
Turn 7	[U]: can i get the postcode ?		
	State: {}		
	[S]: no problem , the postal code is cb21ta. did you need the phone number as well?		
Turn 8	[U]: no thanks i am all set . thank you for your help today.		✓
	State: {}		

Table 3.8: Example (PMUL2281) of the selected turn by $\text{PPTOD}_{\text{base}}$ using ME and LC.

	Dialogue PMUL2281	**ME**	**LC**
	[S]:		
Turn 1	[U]: can you help me find a place to dine?		
	State: {}		
	[S]: sure! what type of food would you like to eat in what area?		
Turn 2	[U]: i would like some north indian food that is expensive, in the south.		
	State: {restaurant-food=north indian}		
	[S]: we do not have any north indian restaurant -s, though we do have many indian restaurant -s.		
Turn 3	[U]: indian food would be fine then, can you tell me the name?		
	State: {restaurant-pricerange=expensive, restaurant-food=indian}		
	[S]: there are several indian restaurant -s, may i suggest the golden curry in the centre. it sounds like just what you are looking for.		
Turn 4	[U]: i was actually hoping for a restaurant in the south. are there any available?		✓
	State: {restaurant-area=south}		
	[S]: i found 1 called taj tandoori. want more information?		
Turn 5	[U]: more information please		
	State: {}		
	[S]: it is expensive and in the south.		
Turn 6	[U]: can i get the postcode for that please?	✓	
	State: {}		
	[S]: sure! the post code is cb17aa .		
Turn 7	[U]: thanks! i am also looking for a nightclub. i'll need the postcode, please?		
	State: {attraction-type=nightclub}		
	[S]: the ballare is a nightclub in the centre of town. the entrance fee is 5 pounds.		
Turn 8	[U]: i'll try that. what s the postcode please?		
	State: {}		
	[S]: their postcode is cb23na. can i help you with anything else today?		
Turn 9	[U]: no thanks. that was all i needed today. goodbye.		
	State: {}		

selects the most valuable turn from each dialogue for labelling and training. Experiments show that our approach outperforms strong DST baselines in the weakly-supervised scenarios and achieves the same or better joint goal and slot accuracy with significantly less annotated data. Further analysis are conducted to investigate the impact of AL query sizes, base DST models and turn selection methods.

Limitations. We acknowledge the limitations of this work as follows.

First, our AL approach adds extra computation time compared to directly training a DST model using only the last turns of dialogues. A smaller query size (*e.g.*, k) may further increase the runtime as more intermediate models have to be trained. That is, we achieved similar or even better DST performance with significantly reduced annotation data at the cost of increased computation time. Therefore, the trade-off between computational cost, DST performance, and annotation cost needs to be well-determined.

Second, we are unable to employ human annotators to evaluate the actual cost due to a lack of budget. In practice, the number of annotators required depends on the financial budget, project timeline, and the proficiency of annotators. Estimating the exact number of annotators and the annotation cost is challenging. As a mitigation, we provide a theoretical cost analysis in §3.6.3. However, it is a rough estimation and may not reflect the actual cost.

Third, our experiments are limited to the MultiWOZ 2.0 [18] and MultiWOZ 2.1 [54] datasets. We also tried to use the SGD dataset [210]. However, the PPTOD model is already pre-trained on this dataset, making it unsuitable for downstream evaluation. KAGE-GPT2 requires the predefined ontology (*i.e.*, the all possible domain-slot value pairs in the dataset) to build a graph neural network, but SGD does not provide all possible values for non-categorical slots. For example, MultiWOZ has all possible values predefined for the non-categorical domain-slot *train-arriveBy*, while SGD does not have it since it is innumerable. Our AL framework is built upon the base DST model and thus suffers the same drawbacks; we may try other DST models and datasets in the future.

CHAPTER 4

LEARNING KNOWLEDGE VIA CONTINUAL INSTRUCTION TUNING

In Chapter 3, we focused on acquiring new knowledge for a pre-trained language model within a single task — specifically, the task-oriented dialogue state tracking task. However, language models should adapt to the constant emergence of new data and unseen tasks. In this chapter, we present approaches to continually adapting language models to these new tasks while minimizing the impact on their existing capabilities. This chapter is based on one published paper at Findings of EMNLP 2023 – CITB: A Benchmark for Continual Instruction Tuning [288].

4.1 Introduction

Recent studies have shown that multi-task IT (Instruction Tuning) makes language models better zero-shot learners [253, 216, 250, 36, 158]. IT fine-tunes pre-trained language models (PLMs) on various tasks with natural language instructions (Fig. 4.1) and can achieve remarkably well generalization to unseen tasks.

Despite their impressive performance, these instruction-tuned PLMs still fall short on domain-specific tasks due to the limited exposure to relevant knowledge and vocabulary from the training corpus [163]. Moreover, PLMs are static after deployment, and there is no mechanism to update themselves or adapt to a changing environment [289, 17].

Continual learning (CL) aims to enable information systems to learn from a continuous data stream across time [12]. Therefore, it is promising to leverage CL for instruction-tuned PLMs to continually adapt to new domains and tasks without costly re-training. De-

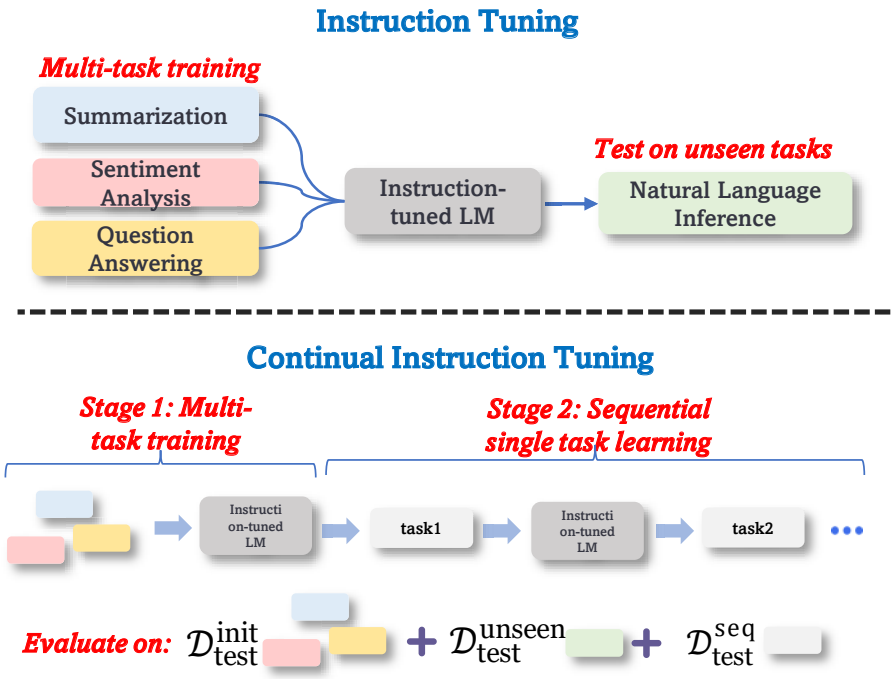


Figure 4.1: Illustration of proposed continual instruction tuning (CIT). Unlike previous works, we evaluate the instruction-tuned model on the initial training, unseen, and newly learned tasks.

spite its importance, it is non-trivial to alleviate *catastrophic forgetting*, a phenomenon in which previously learned knowledge or abilities are degraded due to overwritten parameters [171]. Moreover, enabling knowledge transfer is also essential since many tasks are similar and have common knowledge [121].

Unfortunately, there is little work on applying CL for IT and has only been explored in rather specific settings. [219] continually fine-tune a T0 [216] on eight new tasks with memory replay to avoid forgetting. Despite effectiveness, they need to store a large number of instances per task in memory, which is too costly when scaling to a larger number of tasks. In addition, they do not study knowledge transfer between tasks. [272] propose to use history task instructions to reduce forgetting and enable knowledge transfer. However, they do not compare with commonly adopted CL methods, which makes the effectiveness of other CL methods unknown. Moreover, they only evaluate the model on the newly learned tasks while largely ignoring previously learned tasks during the multi-task training stage (Fig. 4.1). They also overlook the intrinsic ability of the instruction-tuned model on unseen tasks. Lastly, both of them use different evaluation metrics and setups, which creates an obstacle to comparing different techniques and hinders the development of this field.

To this end, we first formulate this practical yet under-explored problem as **Continual**

Instruction Tuning (CIT). Then, we propose a first-ever benchmark suite to study CIT systematically. Our benchmark, CITB, consists of both learning and evaluation protocol and is built on top of the recently proposed SuperNI dataset [250]. We create two CIT task streams: **InstrDialog** stream, which consists of 19 dialogue-related tasks spanning three categories; **InstrDialog++** stream, which includes all the tasks in **InstrDialog** stream and 19 additional tasks selected from broad categories and domains. Using the two long task streams, we implement various CL methods to study forgetting and knowledge transfer under the setup of CIT. We find that directly fine-tuning an instruction-tuned model sequentially yields competitive performance with existing CL methods. With further investigation, we find that rich natural language instructions enable knowledge transfer and reduce forgetting, which is barely fully leveraged by current CL methods. We conduct comprehensive experiments to explore what effects the learning of CIT. We hope our CITB benchmark will serve as a helpful starting point and encourage substantial progress and future work by the community in this practical setting. To summarize, our main contributions are:

- We formulate the problem of CIT and establish a benchmark suite consisting of learning and evaluation protocols.
- We curate two long task streams of various types based on the SuperNI dataset to study different setups of CIT.
- We implement various CL methods of different categories, conduct extensive experiments and ablation studies to analyze the lack of current practices and propose a future direction.

We release our code, data, and model to facilitate future research at `https://github.com/hyintell/CITB`.

4.2 Related Work

4.2.1 Instruction Tuning

Much effort has been made recently to use natural language instructions to solve multiple tasks concurrently or to align with human preferences [243, 297, 185]. Unlike simple and short prompts [153], natural language instructions (Fig. 4.2) can be more comprehensive, including components such as task definition, in-context examples [16], and explanations. Through IT, PLMs learn to complete tasks by following instructions, which enables them to solve *new* tasks by following instructions without learning (*i.e.*, generalization ability). Ideally, we expect the instruction-tuned model to understand any given task instruction so

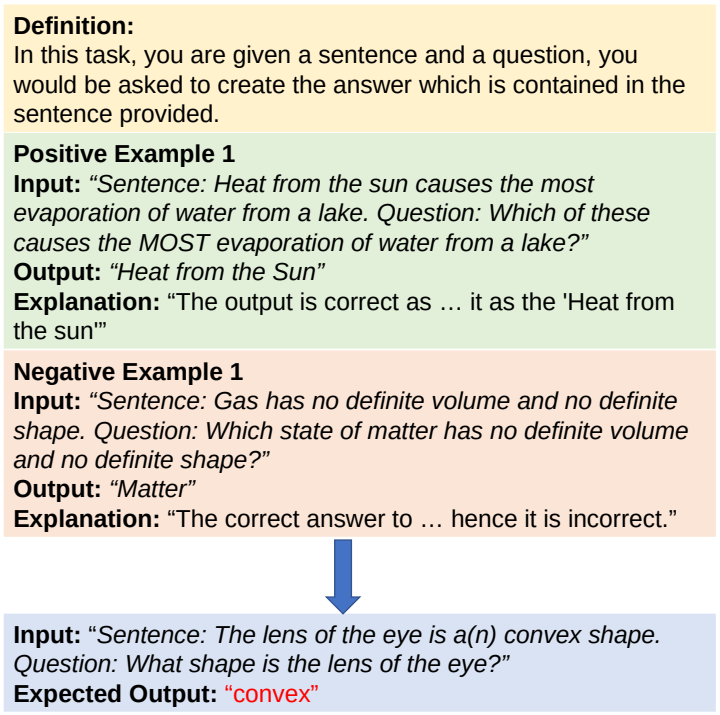


Figure 4.2: An example of natural language instruction that consists of a descriptive task definition, one positive and one negative in-context example with explanation [250].

that an end user can directly leverage the model to solve the task without annotating a large dataset and training it. Unfortunately, despite the instruction-tuned models such as FLAN [253, 158], T0 [216], and Tk-Instruct [250] showing strong generalization performance to their evaluation tasks, there is still a sizeable gap compared with supervised training, which limits the usage of the models. From a practical point of view, a desirable instruction-tuned model should be able to extend its ability by *continually* learning those under-performed tasks or any new task, while not forgetting the old ones.

4.2.2 Continual Learning

In §2.2.2.3, the "continual learning" we discussed is more focused on continual pre-training, where language models are learning new knowledge in an *unsupervised* manner. In this section, we focus on *supervised* continual learning, where language models are provided with labelled data for learning new tasks.

In contrast to multi-task learning, continually fine-tuning a model on tasks might lead to *catastrophic forgetting* [171], where the model forgets previously acquired knowledge after learning new tasks. In CL literature, approaches to overcoming catastrophic forgetting can be grouped into three categories [12, 120]. *Regularization*-based methods use an addi-

tional loss to prevent important parameters of previous tasks from being updated [128, 44]. *Replay*-based methods store and replay a small subset of training data from previous tasks to prevent forgetting [211, 219]; *Architecture*-based methods introduce task-specific components for new tasks and isolate parameters of old tasks [167, 302]. However, the effectiveness of these CL methods for CIT remains unknown. CIT differs from traditional CL in heavily relying on comprehensive instructions. Can previous methods fully leverage the rich instructions to avoid forgetting while facilitating knowledge transfer between tasks? Moreover, many proposed CL methods target tasks of specific types (*e.g.*, text classification, relation extraction) [92, 199, 302], while CIT can learn broad tasks of different categories because of the natural language instructions[1]. [272, 219, 179] study a similar problem using IT in the CL setting, but no benchmarks are built for different categories of CL methods. To tackle CIT, it is essential to establish a unified benchmark to compare existing approaches and promote the development of this field. However, to our best knowledge, CIT is still immature, and no public benchmark is available.

4.3 Preliminaries

Instruction Tuning (IT). Following previous studies [253, 216, 250], each task $t \in \mathcal{T}$ consists of its natural language instruction I^t and a set of N input-output instances $\mathcal{D}^t = \{(x_i^t, y_i^t) \in \mathcal{X}^t \times \mathcal{Y}^t\}_{i=1}^N$, which can be split into the training $\mathcal{D}^t_{\text{train}}$, validation $\mathcal{D}^t_{\text{dev}}$ and test sets $\mathcal{D}^t_{\text{test}}$. Each instance is filled into an instruction template such that different tasks can be transformed into a unified text-to-text format (Fig. 4.2). Given a task instruction and the input of a test instance, a model needs to produce the desired output. IT aims to learn a model $f : I^t \times \mathcal{X}^t \rightarrow \mathcal{Y}^t$ that can predict the output y_i^t given the task instruction I^t and an input x_i^t. In general, the model is first trained on a mixture of tasks ($\mathcal{T}_{\text{seen}}$) and then evaluated for its zero-shot generalization ability on held-out tasks ($\mathcal{T}_{\text{unseen}}$), where $\mathcal{T}_{\text{seen}} \cap \mathcal{T}_{\text{unseen}} = \emptyset$. The model is expected to learn to follow instructions via the training tasks and then solve new tasks with only the help of task instructions.

4.4 Methodology

In this section, we first formalize the CIT problem (Fig. 4.1). Then, we present the learning and evaluation protocol of our framework CITB. Lastly, we describe the data for creating

[1] All tasks can be filled into a natural language instruction template and transformed into a text-to-text format [253].

the benchmark.

4.4.1 Continual Instruction Tuning

In contrast to static IT, which only learns a fixed set of tasks ($\mathcal{T}_{\text{seen}}$), the model should be able to keep learning new tasks without catastrophically forgetting previously learned knowledge and facilitate knowledge transfer if possible. Let us expand the definition such that we have a set of T tasks $\mathcal{T}_{\text{seq}} = \{t_1, \cdots, t_T\}$ that arrives sequentially. Note that the tasks in the stream can be any type and are not restricted to specific categories or domains. Similarly, each task $t_j \in \mathcal{T}_{\text{seq}}$ has a natural language instruction I^{t_j}, training $\mathcal{D}^{t_j}_{\text{train}}$, validation $\mathcal{D}^{t_j}_{\text{dev}}$ and test sets $\mathcal{D}^{t_j}_{\text{test}}$. Likewise traditional CL, the goal of CIT is to learn a *single* model f from $\mathcal{T}_{\text{seq}}$ sequentially.

CIT vs. Traditional CL. While sharing similar desiderata with traditional CL, CIT differs in that: (1) it pays more attention to effectively leveraging the rich natural language instructions to prevent catastrophic forgetting and encourage knowledge transfer; (2) because of the multi-task nature of instructions, all tasks can be formatted in the unified text-to-text format, therefore CIT can learn any task or domain instead of a few specific tasks or domains; (3) after learning a few tasks, the model should have learned how to follow instructions to complete tasks, therefore, we expect fewer training instances required and higher knowledge transfer for future tasks.

4.4.2 Learning Protocol of CIT

A non-instruction-tuned model (*e.g.*, T5; [206]) may struggle to understand instructions if trained only on a task sequentially. It is also against our motivation to extend a model's ability that is already instruction-tuned. Therefore, we separate the learning process into two stages.

Stage 1: Initial Multi-task Fine-tuning. To teach a model a better understanding of task instructions, we first fine-tune the model on instruction data. Suppose we have another group of M tasks $\mathcal{T}_{\text{init}} = \{t_1, \cdots, t_M\}$ that also equips with natural language instructions, where $\mathcal{T}_{\text{init}} \cap \mathcal{T}_{\text{seq}} = \emptyset$. We fine-tune a base pre-trained model on the training set (*i.e.*, $\mathcal{D}^{\text{init}}_{\text{train}} = \bigcup_{i=1}^{M} \mathcal{D}^{t_i}_{\text{train}}$) of the mixed M tasks to get an instruction-tuned model, denoted as f_{init}. After training, most of the training data $\mathcal{D}^{\text{init}}_{\text{train}}$ is unavailable for subsequent sequential learning, but a memory $\mathcal{M}_{\text{init}}$ ($\mathcal{M}_{\text{init}} \ll |\mathcal{D}^{\text{init}}_{\text{train}}|$) that stores a small portion of training

instances is accessible. We use this model as the starting point to conduct the subsequent learning.

Stage 2: Sequential Single Task Fine-tuning. To keep extending knowledge of the instruction-tuned f_{init}, we fine-tune it on the training set $\mathcal{D}_{\text{train}}^{t_j}$ of each task t_j in the stream $\mathcal{T}_{\text{seq}}$. Similarly, when learning the task t_j, the training data of previous tasks in the stream (*i.e.*, $\mathcal{D}_{\text{train}}^{\text{seq}} = \bigcup_{i=1}^{j-1} \mathcal{D}_{\text{train}}^{t_i}$, $i < j < T$) is unavailable, but a small memory $\mathcal{M}_{\text{seq}}$ can be used for training.

4.4.3 Evaluation Protocol of CIT

Evaluation Process. After learning each task t_j ($1 < j < T$) in stream $\mathcal{T}_{\text{seq}}$, we consider three datasets to measure the model's performance. (1) Similar to the standard CL, we evaluate the model on the test sets of all previously learned tasks in the stream, the test set of the current task, and the test set of the next task, denoted as $\mathcal{D}_{\text{test}}^{\text{seq}}$. This helps us measure whether the model forgets previous knowledge and whether it is helpful to learn future tasks; (2) We evaluate the model on the test sets of the M tasks that are used in stage 1 to teach the model how to follow instructions, denoted as $\mathcal{D}_{\text{test}}^{\text{init}} = \bigcup_{i=1}^{M} \mathcal{D}_{\text{test}}^{t_i}$. This is where different from the conventional CL. In CL, previous works only evaluate downstream tasks in the stream but not the tasks during the pre-training phase because such data is generally not accessible to end-users [119]. (3) Since multi-task instruction-tuned models have shown strong zero-shot generalization to unseen tasks [250], our initial model trained in stage 1 might also have zero-shot generalization to some unseen tasks $\mathcal{T}_{\text{unseen}}$, where $\mathcal{T}_{\text{init}} \cap \mathcal{T}_{\text{seq}} \cap \mathcal{T}_{\text{unseen}} = \emptyset$. Let $\mathcal{D}_{\text{test}}^{\text{unseen}}$ be the test sets of all tasks in $\mathcal{T}_{\text{unseen}}$. We evaluate the model on $\mathcal{D}_{\text{test}}^{\text{unseen}}$ if it is available. To sum up, once a new task is learned, the model will be evaluated on:

$$D = \mathcal{D}_{\text{test}}^{\text{seq}} + \mathcal{D}_{\text{test}}^{\text{init}} + \mathcal{D}_{\text{test}}^{\text{unseen}} \tag{4.1}$$

In CIT, it is more critical for the instruction-tuned model to maintain its existing abilities than to learn new ones because it can solve multiple tasks by following instructions. Otherwise, if it forgets many tasks, there is no point in using such a model other than a task-specific one.

Evaluation Metrics. Due to the diversity of the tasks in CIT and the open-ended generation nature of the text-to-text format, we follow [250] to use *ROUGE-L* [147] to measure the

aggregated performance of each task. They have shown that *ROUGE-L* generally works well for both generation and classification tasks.

Following [159] and [12], we also use CL-related metrics to measure the learning procedure. Let $a_{j,i}$ be the *ROUGE-L* score of the model on the test set of task t_i right after training on task t_j, we define the following:

Average *ROUGE-L* (AR), which measures the average performance of the model on all tasks after the final task t_T is learned:

$$\mathbf{AR}_T = \frac{1}{T}\sum_{i=1}^{T} a_{T,i} \tag{4.2}$$

We use **Final *ROUGE-L*** (FR) to measure the performance of the model on $\mathscr{D}_{\text{test}}^{\text{init}}$ and $\mathscr{D}_{\text{test}}^{\text{unseen}}$, respectively, after the final task t_T is learned.

Forward Transfer (FWT), which measures how much the model can help to learn the new task. FWT also tests the model's zero-shot generalization to new tasks:

$$\mathbf{FWT}_T = \frac{1}{T-1}\sum_{i=2}^{T} a_{i-1,i} \tag{4.3}$$

Backward Transfer (BWT), which measures the impact that continually learning on subsequent tasks has on previous tasks:

$$\mathbf{BWT}_T = \frac{1}{T-1}\sum_{i=1}^{T-1} \left(a_{T,i} - a_{i,i}\right) \tag{4.4}$$

Notably, positive BWT indicates that subsequent tasks can improve the performance of previous tasks, while negative value implies knowledge forgetting.

4.4.4 Data Curation

In this work, we adopt the recently proposed SuperNI [250] dataset to establish the benchmark. SuperNI consists of more than 1,600 NLP tasks, spanning a diverse variety of 76 broad task types, such as language generation, classification, question answering, and translation. Moreover, each task is equipped with an instruction and a set of instances, and all the instances can be transformed into the text-to-text format. Therefore, the dataset is suitable for studying CIT. The official training set of SuperNI[2] consists of 756 English tasks spanning

[2] `https://github.com/allenai/natural-instructions`

60 broad NLP categories, while 119 tasks from 12 categories are used for zero-shot evaluation. We keep the official 119 evaluation tasks untouched and create two CIT task streams from the 756 training tasks.

InstrDialog Stream. Dialogue is an important field to study continual learning because new tasks, domains or intents are continuously emerging [167]. To investigate how a model learns new dialogue data under the setup of CIT, we carefully curate *all* dialogue-related tasks from the training set of SuperNI to form the CIT task stream. Specifically, we use 4 tasks from dialogue state tracking, 11 tasks from dialogue generation, and 4 tasks from intent identification, resulting in a total of 19 dialogue tasks, *i.e.*, $|\mathcal{T}_{\text{seq}}| = 19$. We remove tasks that are excluded by the official task splits[3].

InstrDialog++ Stream. Because of the multi-task nature of instructions, an instruction-tuned model can learn any new task with different types (§4.4.1). To study how different types of tasks and how a long-task curriculum affects CIT, we first include all 19 dialogue tasks from the **InstrDialog** stream, then manually select the other 19 tasks from the remaining training task set. We intentionally select tasks from broad categories, including sentence ordering, style transfer, toxic language detection, and others. In total, we have 38 tasks of 18 categories (3 categories from InstrDialog and 15 categories from the new 19 tasks), *i.e.*, $|\mathcal{T}_{\text{seq}}| = 38$.

The remaining training tasks can be used for stage 1 initial multi-task fine-tuning (§4.4.2). In summary, the number of initial fine-tuning tasks available is $M = |\mathcal{T}_{\text{init}}| = 718$, and we use the official 119 test task sets as $\mathcal{T}_{\text{unseen}}$ to evaluate whether the performance deteriorates for unseen tasks after learning new tasks. For all tasks, we fill instances in a natural language instruction template and transform them into a unified text-to-text format (§4.3). Unless otherwise specified, we use the instruction template consisting of the task definition and two positive examples for all tasks because it generally yields the best performance [250]. See an example of natural language instructions in Fig. 4.2, and the selected tasks in Table 4.1. We study the effect of the instruction template in §4.6.3.

4.5 Experiments

Using our CITB benchmark, we conduct experiments on various popular CL methods of different kinds. In this section, we describe our experiment setups and compare methods.

[3] https://github.com/allenai/natural-instructions/tree/master/splits

Task Category	Number	Task Name	Domain
Dialogue Generation	1	task565_circa_answer_generation	Dialogue
	2	task574_air_dialogue_sentence_generation	Dialogue
	3	task576_curiosity_dialogs_answer_generation	Dialogue, Commonsense
	4	task611_mutual_multi_turn_dialogue	Dialogue
	5	task639_multi_woz_user_utterance_generation	Dialogue
	6	task1590_diplomacy_text_generation	Dialogue, Game
	7	task1600_smcalflow_sentence_generation	Dialogue, Commonsense
	8	task1603_smcalflow_sentence_generation	Dialogue
	9	task1714_convai3_sentence_generation	Dialogue
	10	task1729_personachat_generate_next	Dialogue
	11	task1730_personachat_choose_next	Dialogue
Intent Identification	12	task294_storycommonsense _motiv_text_generation	Story
	13	task573_air_dialogue_classification	Dialogue
	14	task848_pubmedqa_classification	Medicine
	15	task1713_convai3_sentence_generation	Dialogue
Dialogue State Tracking	16	task766_craigslist_bargains_classification	Dialogue
	17	task1384_deal_or_no_dialog_classification	Dialogue, Commonsense
	18	task1500_dstc3_classification	Dialogue, Public Places
	19	task1501_dstc3_answer_generation	Dialogue, Public Places
Style Transfer	20	task927_yelp_negative_to_positive_style_transfer	Reviews
Sentence Ordering	21	task1549_wiqa_answer_generation_missing_step	Natural Science
Word Semantics	22	task459_matres_static_classification	News
Text Categorization	23	task379_agnews_topic_classification	News
Pos Tagging	24	task347_hybridqa_incorrect_answer_generation	Wikipedia
Fill in The Blank	25	task1360_numer_sense_multiple_choice_qa_generation	Commonsense
Program Execution	26	task1151_swap_max_min	Mathematics
	27	task636_extract_and_sort_unique_alphabets_in_a_list	Mathematics
Question Generation	28	task301_record_question_generation	News
	29	task082_babi_t1_single_supporting _fact_question_generation	Commonsense
Misc.	30	task306_jeopardy_answer_generation_double	Knowledge Base
	31	task1427_country_region_in_world	Countries
Coherence Classification	32	task298_storycloze_correct_end_classification	Story
Question Answering	33	task864_asdiv_singleop_question_answering	Mathematics
	34	task598_cuad_answer_generation	Law
Summarization	35	task1553_cnn_dailymail_summarization	News
Commonsense Classification	36	task1203_atomic_classification_xreact	Commonsense
Wrong Candidate Generation	37	task967_ruletaker_incorrect_fact _generation_based_on_given_paragraph	Commonsense
Toxic Language Detection	38	task1607_ethos_text_classification	Social

Table 4.1: List of tasks selected from SuperNI [250]

4.5.1 Setup

Model. We use the LM-adapted version of T5-small [206], which is further trained with a language modeling objective. We initialize a T5 model from HuggingFace[4]. Since it is costly to fine-tune on all 718 tasks, we randomly select 100 tasks from $\mathcal{T}_{\text{init}}$ and fine-tune T5 to obtain an instruction-tuned model f_{init} (§4.4.2), which has learned to understand some instructions and can act as a good starting point to conduct subsequent learning. Note that the 100 randomly selected training tasks do not overlap with **InstrDialog** and **InstrDialog++**, but their task categories might overlap with the categories in **InstrDialog++**.

[4] `https://huggingface.co/google/t5-small-lm-adapt`

Train/Dev/Test Splits. Since the number of instances in each task is imbalanced and a large number of training instances do not help generalization in IT [250], we follow [250] use a fixed size of 500/50/100 instances per task as the train/dev/test set for the **InstrDialog** stream. For **InstrDialog++**, since it has a longer task sequence, to save computational cost, we use 100/50/100 instances per task instead. For $\mathcal{T}_{\text{init}}$, we use 100/50/100 instances per task and 100 instances per task for $\mathcal{T}_{\text{unseen}}$. We study the effect of different numbers of training instances in §4.6.3.

4.5.2 Baselines and Compared Methods

We implement widely adopted CL methods from three categories [12] to benchmark CIT.

Regularization-based methods rely on a fixed model capacity with an additional loss term to consolidate previously gained knowledge while learning subsequent tasks. We use **L2** and **EWC** [128], which uses a fisher information matrix to reduce forgetting by regularizing the loss to penalize the changes made to important parameters of previous tasks.

Replay-based methods store a small subset of training instances from previous tasks in memory. The data are replayed later to reduce forgetting. We adopt **Replay**, which saves random instances from each task in memory and then jointly trains the model on new task data and the old data in the memory; **AGEM** [25], which adds a constraint to prevent parameter update from increasing the loss of each previous task. The loss of previous tasks is calculated using the instances stored in the memory.

Architectural-based methods introduce task-specific parameters to the base model to prevent subsequent tasks from interfering with previously learned parameters. We adopt **AdapterCL** [167], which freezes the pre-trained model and trains a residual Adapter [86] for each task independently.

Instruction-based apart from the three categories of CL, we also implement instruction-based baselines because all previous CL methods are not designed for CIT. No prior work had tried to fine-tune an instruction-tuned model sequentially without any mechanism for preventing forgetting or encouraging knowledge transfer. However, it is commonly considered a performance lower bound in CL literature.

To this end, we propose to continually fine-tune the initial instruction-tuned model (§4.5.1) on subsequent tasks, named as **FT-init**. As a direct comparison, we initialize a new

T5 model, which is not tuned on any instruction data, and we continually fine-tune it (**FT-no-init**). In addition, we also report the performance of the initial instruction-tuned model (**Init**), which is the starting point before subsequent learning. Lastly, we jointly fine-tune a T5 model using all the data, including the training data used in stage 1 and the training data of all subsequent tasks in the stream (**Multi**). This is often regarded as the performance upper bound in CL and does not have catastrophic forgetting and knowledge transfer.

4.5.3 Implementation Details

For both **InstrDialog** stream and **InstrDialog++** stream, we conduct continual instruction tuning using the same initial instruction-tuned model for all methods except **FT-no-init**, **AdapterCL**, and **Multi**. For these methods, we initialize a new T5 model. Since AdapterCL needs to train a task-specific adapter for each task, it is too costly to train the 100 adapters for it, therefore we initialize it from a T5 model. For AdapterCL, we use a bottleneck of 100. The initial instruction-tuned model (§4.5.1) is trained on 100 tasks with 100 instances per task (in total 10,000). We use a maximum of 15 epochs with a learning rate of 1e-05. We perform checkpoint selection using the development set.

For **L2** and **EWC**, we tune the regularization term from $[0.001, 0.01, 0.1]$, and use 0.01. For AdapterCL, we use a learning rate of 1e-3. For other methods, we use 1e-5. For the **InstrDialog** stream, we use a batch size of 4 for EWC and 8 for all other methods; for the **InstrDialog++** stream, we use a batch size of 16. For all experiments except AdapterCL, we train a maximum epoch of 15 and perform early stopping with 3 patience to avoid overfitting; for **AdapterCL**, we use a larger epoch of 20 with patience of 5. For **AGEM** and **Replay**, we experiment on a memory size of 10 and 50, *i.e.*, we set the memory size $\mathscr{M}_{\text{init}}$ to 10 and 50, same as $\mathscr{M}_{\text{seq}}$. We jointly train the data in $\mathscr{M}_{\text{init}}$, $\mathscr{M}_{\text{seq}}$, and the new task data for these two methods.

The selected tasks for task stream **InstrDialog** and **InstrDialog++** are listed in Table 4.1. The task orders are listed in Table 4.2. Due to limited computing resources, we randomly permute the task streams and run all experiments using three random seeds. We refer to this as task order 1. We study the effect of task orders in §4.6.3. All experiments are done in an RTX3090 Ti with 24GB VRAM.

Task Order	Task's IDs in order
Order1	848 611 565 1714 574 1590 1730 294 576 1600 1500 639 1729 1501 1713 766 1603 1384 573
Order2	848 1603 1714 565 611 1590 1600 639 294 1500 1384 1713 1501 576 574 1729 766 573 1730
Order3	1713 576 1384 294 573 611 1729 1600 574 1590 848 639 766 1501 565 1603 1730 1500 1714

Table 4.2: Task orders for three runs of the InstrDialog Stream.

Method	InstrDialog AR	InstrDialog FWT	InstrDialog BWT	$\mathcal{T}_{\text{init}}$ FR	$\mathcal{T}_{\text{unseen}}$ FR	Mem.	+P (Tun)	Time
FT-no-init	$29.6_{2.1}$	$8.0_{0.2}$	$-10.8_{2.3}$	$17.3_{0.5}{}^{\dagger}$	$16.5_{1.3}{}^{\dagger}$	0	0 (1)	$0.5_{0.02}$
AdapterCL	$8.1_{0.1}$	$9.4_{0.7}$	$-21.9_{0.9}$	-	-	0	T*0.02 (0.02)	$\mathbf{0.4}_{0.03}$
Init	$22.5^{\dagger}$	-	-	43.5	$\mathbf{36.5}^{\dagger}$	-	-	-
FT-init	$35.7_{0.2}$	$18.5_{0.7}$	$-4.6_{0.2}$	$38.6_{0.3}$	$32.3_{0.6}{}^{\dagger}$	0	0 (1)	$0.6_{0.01}$
L2	$35.6_{0.1}$	$17.5_{0.5}$	$-3.8_{1.2}$	$39.4_{0.4}$	$34.9_{1.2}{}^{\dagger}$	0	1 (1)	$0.6_{0.1}$
EWC	$34.5_{0.6}$	$16.8_{0.4}$	$-6.8_{1.5}$	$37.0_{0.1}$	$32.5_{0.5}{}^{\dagger}$	0	2 (1)	$1.1_{0.2}$
AGEM (10)	$33.2_{0.4}$	$19.1_{0.1}$	$-7.3_{1.0}$	$38.6_{1.1}$	$32.4_{0.0}{}^{\dagger}$	(T+M)*10	0 (1)	$1.1_{0.1}$
AGEM (50)	$34.9_{0.9}$	$18.1_{1.0}$	$-6.0_{0.9}$	$37.7_{0.1}$	$32.6_{1.0}{}^{\dagger}$	(T+M)*50	0 (1)	$1.3_{0.1}$
Replay (10)	$38.4_{0.7}$	$\mathbf{23.7}_{0.0}$	$-1.3_{0.5}$	$42.7_{0.7}$	$32.4_{0.4}{}^{\dagger}$	(T+M)*10	0 (1)	$1.4_{0.04}$
Replay (50)	$40.4_{0.0}$	$22.9_{0.1}$	$\mathbf{1.6}_{1.2}$	$\mathbf{47.1}_{0.5}$	$31.8_{1.0}{}^{\dagger}$	(T+M)*50	0 (1)	$3.2_{0.5}$
Multi	$\mathbf{42.1}_{0.6}$	-	-	$44.7_{1.3}$	$32.8_{0.9}{}^{\dagger}$	0	0 (1)	$1.1_{0.2}$

Table 4.3: Performance of different methods on the **InstrDialog** stream. Means and standard deviations are reported.

4.6 Results & Analysis

In this section, we report the performance of various baselines discussed in §4.5.2 on our benchmark.

4.6.1 Results on InstrDialog Stream

Table 4.3 shows each method's overall performance and resource requirement after continually learning the **InstrDialog** stream. † means zero-shot performance. "Mem." means the number of instances stored in the memory for each task; T is the total number of tasks in the stream and M is the number of tasks used for initial training. "+P" means the percentage of additional parameters added for each task, measured by the total parameters of the base model; "Tun" is the portion of tunable parameters during training. "Time" is the average hours for each method to complete the task stream.

We have the following observations:

First, all methods except AdapterCL have improved AR, compared to the zero-shot performance (22.5) of the starting point model Init. This shows CIT can extend a model's knowledge. In contrast, although AdapterCL is parameter-efficient and does not rely on memory, it performs even worse than Init. We conjecture that AdapterCL fails to learn instructions effectively because it is initialized from a non-instruction-tuned model (T5), and the few tunable parameters restrict it from learning complex instructions.

Second, among all baselines, Replay generally have the best performance. All methods except Replay (50) have negative BWT, meaning that they all suffer from catastrophic forgetting. Furthermore, forgetting on $\mathcal{T}_{\text{init}}$ and $\mathcal{T}_{\text{unseen}}$ is even worse, which demonstrates that the ability of the initial instruction-tuned model Init has deteriorated after learning new dialogue tasks. We also find storing more examples in memory improves Replay but does not significantly help AGEM. It might be because the constraints added to the loss are roughly the same, no matter how many instances are stored. Despite used additional parameters, regularization-based L2 and EWC perform similar to other baselines. Multi overall performs well, with the highest AR and improved FR on $\mathcal{T}_{\text{init}}$, however, it also forgets tasks in $\mathcal{T}_{\text{unseen}}$. Replay (50) has a higher FR on $\mathcal{T}_{\text{init}}$ than Multi because the 5,000 instances stored in $\mathcal{M}_{\text{init}}$ are jointly trained multiple times when learning subsequent tasks (§4.5.3), leading to better data fitting.

Third, FT-init performs surprisingly well on all metrics and is competitive with L2, EWC, and AGEM. This finding contradicts the common sense in CL that simply fine-tuning a model sequentially would lead to catastrophic forgetting because all parameters can be freely updated in learning new tasks [128]. Even for FT-no-init, which is not tuned on any instruction data, shows increased AR after learning the 19 dialogue tasks. This raises a question: do various CL methods truly mitigate forgetting and promote knowledge transfer in CIT? We hypothesize that the rich natural language instructions lead to the remarkable performance of the baselines (§4.6.3).

4.6.2 Results on InstrDialog++ Stream

The performance of all methods after learning the **InstrDialog++** steam is shown in Table 4.4. We observe most of the same findings as in §4.6.1, except that:

First, Init has a higher (30.5 vs. 22.5) zero-shot performance on this long stream than on **InstrDialog**, as in Table 4.3. We analyze that the categories of the selected 100 training tasks (§4.5.1) overlap with the categories in the stream, which enables more knowledge transfer of tasks between the same category because of the similar natural language instructions. For example, both sets have tasks related to sentiment analysis and toxic language detection. In contrast, Init did not learn dialogue tasks, thus showing lower generalization on **InstrDialog**.

Second, we can see improved performance for almost all methods compared to Table 4.3, especially on $\mathcal{T}_{\text{init}}$ and $\mathcal{T}_{\text{unseen}}$. For FT-init and FT-no-init, the improvements of FWT and BWT are particularly significant, reaching the best among all CL methods.

Method	InstrDialog++ AR	InstrDialog++ FWT	InstrDialog++ BWT	$\mathcal{T}_{\text{init}}$ FR	$\mathcal{T}_{\text{unseen}}$ FR
FT-no-init	$31.3_{2.4}$	$26.8_{0.3}$	$-5.2_{1.4}$	$28.9_{0.6}$†	$31.2_{2.1}$†
AdapterCL	$20.7_{0.2}$	$14.3_{0.9}$	$-7.3_{1.1}$	-	-
Init	30.5†	-	-	43.5	**36.5**†
FT-init	$34.8_{0.3}$	$\mathbf{29.8}_{0.5}$	$\mathbf{-2.8}_{0.6}$	$40.8_{0.2}$	$35.9_{0.5}$†
L2	$36.1_{0.2}$	$27.9_{0.3}$	$-4.0_{1.2}$	$41.1_{0.4}$	$34.9_{1.1}$†
EWC	$38.6_{0.4}$	$28.6_{0.5}$	$-4.0_{1.8}$	$41.3_{0.3}$	$34.0_{0.7}$†
AGEM (10)	$39.3_{0.6}$	$28.9_{0.2}$	$-3.8_{1.1}$	$40.3_{0.9}$	$34.1_{0.3}$†
Replay (10)	$43.1_{0.7}$	$29.0_{0.4}$	$-3.6_{0.8}$	$44.0_{0.5}$	$30.1_{0.6}$†
Multi	$\mathbf{44.9}_{0.5}$	-	-	$\mathbf{44.6}_{0.9}$	$33.9_{0.5}$

Table 4.4: Performance of different methods on the **InstrDialog++** stream. † means zero-shot performance.

Combining the results on the two streams from Table 4.3 and Table 4.4, we find that catastrophic forgetting exists in CIT. However, learning a longer task stream and diverse tasks of different types leads to better knowledge transfer and lower forgetting.

4.6.3 Ablation Studies

In this section, we investigate the reason why instruction-based baselines (FT-init and FT-no-init) perform as well as or even better than conventional CL methods (§4.6.1 & §4.6.2). We also explore different aspects that might affect CIT.

Rich instructions enable knowledge transfer and reduce forgetting in CIT. We use the same setup as in Table 4.4, except for using different instruction templates. Results in Table 4.5[5] confirm our hypothesis that the remarkable performance of FT-init and FT-no-init comes from the rich natural language instructions. When only using the task Id (*e.g.*, `task565_circa_answer_generation`) as instruction without descriptive task definition or in-context examples, simply fine-tuning the model sequentially yields low AR, FWT, and BWT, which aligns with the findings in CL literature. Additionally, providing in-context examples (2p or 2p+2n) generally improves performance. However, although task performance (AR) is improved with two additional negative examples, we witness decreased knowledge transfer (FWT and BWT). For the model that is not fine-tuned on any instruction data

[5] "id": only use the short task id; "def": only use descriptive task definitions; "def+2p": use task definition and two positive examples; "def+2p+2n": use additional two negative examples.

(FT-no-init), we find it worse than FT-init, showing the benefits of the initial multi-task training on instruction data.

	FT-init			**FT-no-init**		
Instr.	AR	FWT	BWT	AR	FWT	BWT
id	13.9	10.0	-30.3	11.5	8.0	-27.7
def	38.0	21.7	-6.9	33.8	16.1	-7.3
def+2p	34.8	29.8	-2.8	31.3	26.8	-5.2
def+2p+2n	40.4	28.1	-7.1	33.8	22.3	-7.3

Table 4.5: Effect of instruction templates on **InstrDialog++**.

Similar observations are found on $\mathscr{T}_{\text{init}}$ and $\mathscr{T}_{\text{unseen}}$ in Table 4.6, where the model catastrophically forgets its initial abilities after learning a long stream of tasks. Providing descriptive task definitions significantly boosts task performance as well as facilitates knowledge transfer. Moreover, it also maintains the model's generalization ability on unseen tasks. Combining the results in Table 4.3, Table 4.4, Table 4.5, and Table 4.6, we find that those conventional CL methods do not fully leverage the instructions to reduce forgetting and facilitate knowledge transfer while learning continuously because the naive FT-init and FT-no-init can also achieve the same. This calls for novel CL methods designed for CIT.

	FT-init		**FT-no-init**	
Instr.	$\mathscr{T}_{\text{init}}$	$\mathscr{T}_{\text{unseen}}$	$\mathscr{T}_{\text{init}}$	$\mathscr{T}_{\text{unseen}}$
id	3.6	2.8	1.0	1.7
def	25.8	23.6	20.3	22.4
def+2p	40.8	35.9	28.9	31.2
def+2p+2n	37.7	35.7	27.2	33.1

Table 4.6: Effect of instruction templates on $\mathscr{T}_{\text{init}}$ and $\mathscr{T}_{\text{unseen}}$.

Task types and learning order matter to CIT. To explore how task types and orders in a stream affects CIT, we randomly permute the **InstrDialog** stream to get two new task orders and conduct the same learning as Table 4.3. We present the intermediate learning trends of all 19 tasks in Fig. 4.3 Fig. 4.4, and Fig. 4.5. One can see from the plots that all baselines are highly affected by task orders, fluctuating dramatically when different tasks are learned first. We argue that it is because the task difficulties and similarities vary a lot. Learned tasks transfer knowledge through the instructions to new tasks of the same type, facilitating learning. For example, the last task in order 1 (Fig. 4.3) is a type of dialogue generation,

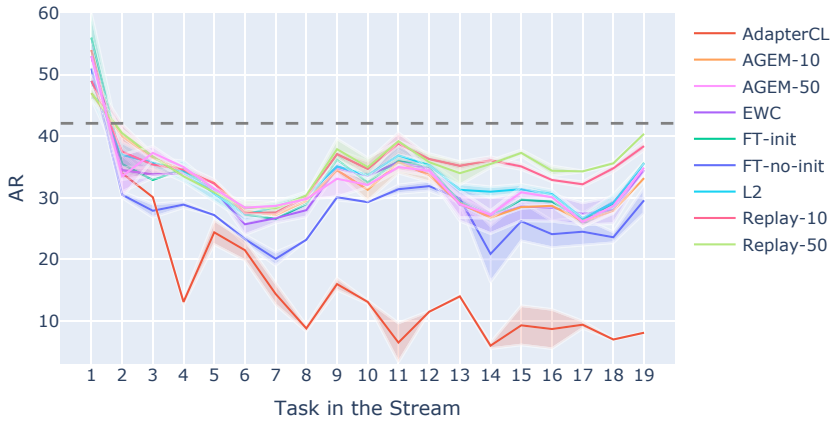


Figure 4.3: AR of each method during learning the **InstrDialog** stream (task order 1).

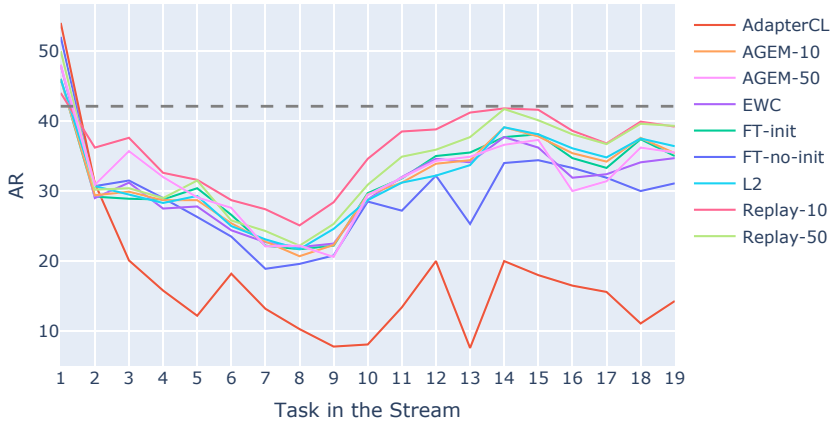


Figure 4.4: AR of each method during learning the **InstrDialog** stream (task order 2).

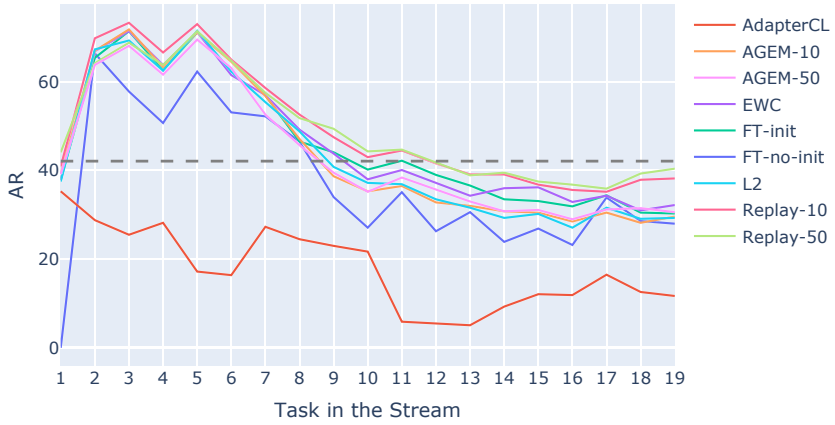


Figure 4.5: AR of each method during learning the **InstrDialog** stream (task order 3).

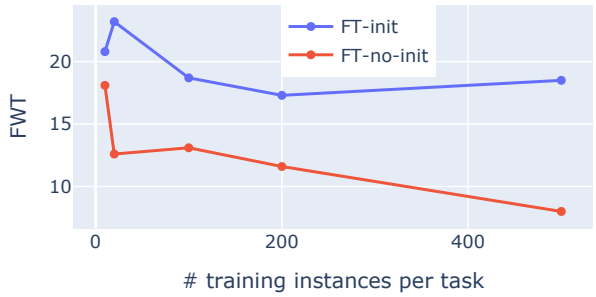


Figure 4.6: Effect of training instances per task on FWT.

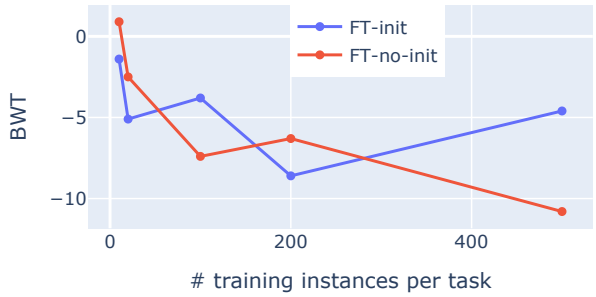


Figure 4.7: Effect of training instances per task on BWT.

which is the dominant task type in the stream (11/19, §4.4.4), therefore all baselines are improved. However, all baselines reach below Multi after learning all 19 tasks, demonstrating knowledge forgetting will eventually appear if learning longer enough tasks.

A large number of training instances do not help knowledge transfer. We vary the number of instances per task used for learning the **InstrDialog** stream from [10, 20, 100, 200, 500]. As shown in Fig. 4.6 and Fig. 4.7, FWT and BWT gradually decrease when the number of training instances is scaled to large values. It aligns with the findings by [250] in standard IT, where a large number of instances do not help generalization to unseen tasks; we also find it is true in CIT. Additionally, we find that instruction-tuned models (FT-init) have better generalization to new tasks (FWT) than the model not fine-tuned on any instruction data (FT-no-init). This shows that, after learning a few tasks, the model have learned how to follow instructions to complete tasks, thus fewer training instances are required for new tasks.

4.7 Summary

In this chapter, we present approaches to continually adapting language models to these new tasks while minimizing the impact on their existing capabilities. We establish a bench-

mark for continual instruction tuning, with two 19 and 38 long task streams to be learned sequentially. We implement and compare various continual learning methods of different types using the benchmark to study their effectiveness under this new domain. We conduct extensive ablation studies to analyze the lack of current practices, and propose a future direction.

Limitations. We identify our limitations as follows. First, due to limited resources, all experiments in this work use T5-small (LM-adapted) as the backbone, which might not entirely reflect continual instruction tuning in general. As [250] points out, there is a sizable gap between the smaller models and the 11B or 3B models in generalizing to new tasks. Second, when creating the two CIT task streams, we only use English tasks from the SuperNI dataset [250]. In future, it can be extended to multilingual task streams to study cross-language continual instruction tuning. Third, we follow the SuperNI dataset to use ROUGE-L as an aggregated metric to evaluate all tasks. Although it acts as a good proxy for the model's overall performance, it might not serve as an effective measurement for some specific tasks. Fourth, while we selected diverse tasks to form the InstrDialog and InstrDialog++ task streams, we did not analyse the characteristics of these tasks [125]. In future, we consider selecting better source tasks and studying how source tasks affect CIT.

CHAPTER 5

LEARNING KNOWLEDGE VIA ADAPTIVE RETRIEVAL-AUGMENTED GENERATION

In Chapter 3 and Chapter 4, we explored how to adapt language models to emerging skills and knowledge through fine-tuning. However, with the advent of LLMs (Large Language Models), which often contain hundreds of billions of parameters, fine-tuning has become prohibitively expensive. Additionally, fine-tuning is not feasible for closed-source models. In this chapter, we present a solution that preserves the general capabilities of LLMs while being efficient and eliminating the need for costly fine-tuning. This chapter is based on one published paper at Findings of ACL 2024 – RetrievalQA: Assessing Adaptive Retrieval-Augmented Generation for Short-form Open-Domain Question Answering [287].

5.1 Introduction

Retrieval-augmented generation (RAG) [73, 138, 209] that augments large language models (LLMs) with retrieval of relevant information has become increasingly popular in knowledge-intensive tasks, including open-domain question-answering (QA) [290, 115, 38, 284]. However, standard RAG methods conduct retrieval *indiscriminately*, irrespective of the input query, which may result in suboptimal task performance and increased inference costs [62]. On the one hand, LLMs encode vast knowledge in parameters through large-scale pre-training, enabling them to effortlessly handle straightforward [1] queries without retrieval

[1] Here, *straightforward* refers to the questions that LLMs can easily answer correctly using their parametric knowledge.

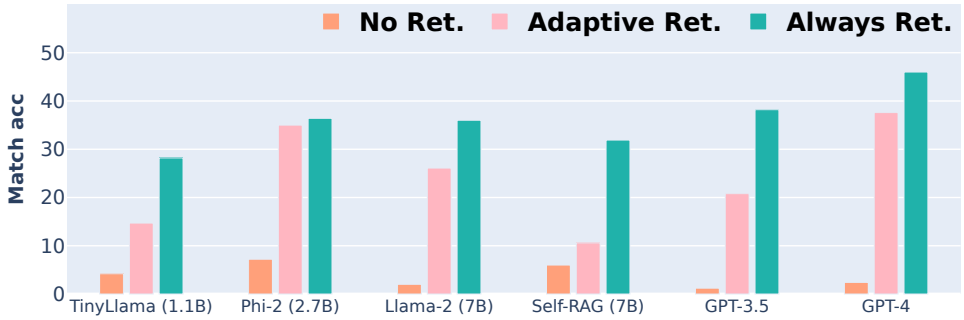
[168]. On the other hand, the retrieved context may contain noise and irrelevant information, and augmenting noisy context can potentially distract LLMs, thereby impeding task performance [227].

To alleviate the limitations of RAG mentioned above, recent studies advocate for **adaptive RAG (ARAG)**, which dynamically determines retrieval necessity and relies only on LLMs' parametric knowledge when deemed unnecessary [58]. However, the effectiveness of these methods is understudied, as there is no suitable benchmark and evaluation. ARAG approaches can be categorized into *calibration-based* and *model-based* judgement. Calibration-based methods [168, 102, 8], while effective, trigger retrieval only when a metric surpasses a pre-defined threshold. For example, [168] heuristically retrieve when the popularity of an entity on Wikipedia is below a certain threshold; [102] trigger retrieval if any token in the temporarily generated sentence has low confidence. Clearly, these ad-hoc calibration-based methods are suboptimal, as we need to tune thresholds for different datasets and models to balance task performance and inference overheads. To obviate the hyperparameter threshold, model-based methods [57, 212] directly prompt LLMs for retrieval decisions, given the observation that LLMs can acknowledge their knowledge boundaries to some extent [108, 273]. These methods undergo separate evaluations, and their effectiveness remains ambiguous due to the limited scope of the assessments.

In this work, we investigate to what extent LLMs can perform calibration-free adaptive retrieval via prompting. To answer this question, we need to evaluate whether LLMs retrieve *only* when necessary. This requests a benchmark that distinguishes between questions that can be answered using LLMs' parametric knowledge and those that require external information through retrieval. Nevertheless, commonly used open-domain QA datasets [207, 106, 130, 168] fail to fulfil this purpose, as various LLMs have distinct sizes and levels of pre-trained knowledge, making them inadequately assess the necessity of external retrieval for LLMs.

To fill this gap, we create `RetrievalQA`, a short-form QA dataset, covering new world and long-tail knowledge and spanning diverse topics. We ensure the knowledge necessary to answer the questions is absent from LLMs. Therefore, LLMs must truthfully decide whether to retrieve to be able to answer the questions correctly. `RetrievalQA` enables us to evaluate the effectiveness of ARAG approaches, an aspect predominantly overlooked in prior studies and recent RAG evaluation systems [27, 215, 55], which focus only on task performance, the relevance of retrieval context or the faithfulness of answers.

Using `RetrievalQA` as a testbed, we benchmark both calibration-based and model-based methods with varying sizes of LLMs. As shown in Fig. 5.1, we find calibration-based



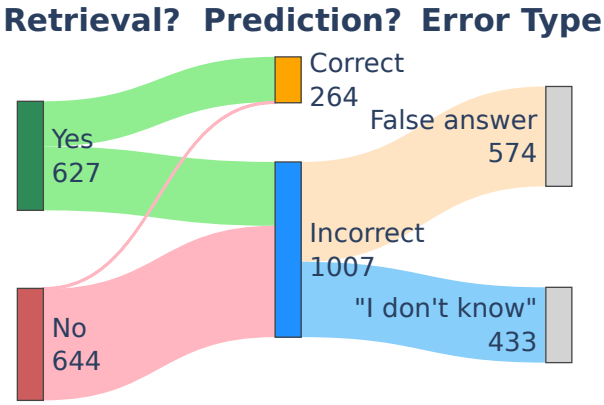


Figure 5.1: **Above**: QA accuracy on our `RetrievalQA` w/, w/o retrieval, and adaptive retrieval. **Below**: an error analysis for GPT-3.5.

Self-RAG requires threshold tuning to balance QA performance and retrieval efficiency, while vanilla prompting is insufficient in guiding LLMs to make reliable retrieval decisions. As an initial effort, we propose **T**ime-**A**ware **A**daptive **RE**trieval (**TA-ARE**), a simple yet effective method to improve ARAG via in-context learning (ICL; [16]), obviating the need for calibration or additional training.

To sum up, this work makes the following contributions:

- We create a new dataset `RetrievalQA` to assess adaptive retrieval-augmented generation for short-form open-domain QA.
- We benchmark existing methods and conduct extensive analysis, finding that vanilla prompting is insufficient in guiding LLMs to make reliable retrieval decisions.
- We propose TA-ARE, a simple yet effective method to help LLMs assess the necessity of retrieval without calibration or additional training.

We release our code, data, and model to facilitate future research at `https://github.com/hyintell/RetrievalQA`.

Please answer the question based on the provided context. Only include the answer in your response and try to be concise. If you do not know the answer, just say "I don't know".

Paragraph:
{retrieved documents}

Question: {question}
Answer:

Figure 5.2: Instruction prompt template for QA with retrieved documents.

5.2 Preliminaries

This section formalises adaptive retrieval-augmented generation (ARAG) for open-domain QA tasks.

Standard RAG. Given a question x, a retriever $\mathscr{R}$, and an external document corpus $\mathscr{D}$ such as Wikipedia, the retriever first retrieves a list of relevant documents $\mathscr{D}_x = \mathscr{R}(x)$, then an LLM needs to generate answer $y = \text{LLM}(I, \mathscr{D}_x, x)$ conditioned on a prompt instruction I, retrieved documents $\mathscr{D}_x$, and the question x. The instruction prompt template for standard RAG is shown in Fig. 5.2.

Adaptive RAG. Standard RAG always retrieves regardless of the input question, while adaptive retrieval only retrieves when necessary. **Calibration-based** methods generally introduce a pre-defined hyperparameter t and only do retrieval when a metric surpasses t:

$$y = \begin{cases} \text{LLM}(I, \mathscr{D}_x, x), & \text{metric} \geq t \\ \text{LLM}(I, x), & \text{otherwise} \end{cases}$$

For **Model-based** methods, we follow [57] and [212] to instruct LLMs to decide whether to retrieve via prompting, obviating the threshold. Specifically, we ask a yes/no question: $r = \text{LLM}(I_{\text{vanilla}}, x)$, where I_{vanilla} = `"Given a question, determine whether you need to retrieve ... answer [Yes] or [No]"`. Retrieval is performed only when LLMs answer yes. We denote this as **Vanilla** prompting. The vanilla prompting template for model-based adaptive RAG is shown in Fig. 5.3.

$$y = \begin{cases} \text{LLM}(I, \mathscr{D}_x, x), & r = \text{Yes} \\ \text{LLM}(I, x), & \text{otherwise} \end{cases}$$

Given a question, determine whether you need to retrieve external resources, such as real-time search engines, Wikipedia, or databases, to answer the question correctly. Only answer "[Yes]" or "[No]".

Question: {question}
Answer:

Figure 5.3: **Vanilla** prompt template for adaptive retrieval.

5.3 RetrievalQA: New Dataset for Open-Domain QA

5.3.1 Design Choice

In this section, we discuss the design choice and rationale of our dataset construction. Our goal is to evaluate adaptive RAG (ARAG) methods and see how good they are at deciding when to retrieve. Therefore, we need the **ground truth labels** for each question's retrieval necessity. Ultimately, there are three kinds of questions here:

- **Case 1**: for all LLMs, questions that can be answered using only their parametric knowledge
- **Case 2**: for all LLMs, questions that can *not* be answered using only the parametric knowledge, therefore requiring external retrieval
- **Case 3**: questions that can be answered with their parametric knowledge for some models but can not be answered for some other models

We do not consider Case 3 because those questions cannot fairly measure whether retrieval is required for different LLMs. For Case 1, it is not trivia to collect questions that can be answered only using the parametric knowledge of LLMs. This is because different LLMs have different levels of pre-trained knowledge, and it is hard to measure [196, 108]. For example, given a question, GPT-3.5 may fail to answer and need the help of external knowledge, while Llama-2 may answer correctly using its own knowledge because it has seen the question in the training data. The pre-training corpora are sometimes unavailable, especially for proprietary models, and we can not guarantee that the collected questions can be 100% answered with their own knowledge, as shown in below Table 5.4.

However, different from Cases 1 and 3, for Case 2, theoretically, it is possible to collect data that guarantees the knowledge to answer the questions is not present in the models. For instance, new world knowledge occurred after model training and long-tail knowledge that did not (or rarely) appear in the training corpora. Therefore, by default, we primarily collect and evaluate questions (Case 2) that are guaranteed cannot be answered without

external information.

5.3.2 Method

Data Collection. Inspired by [304], we aim to collect data such that the knowledge necessary to answer the questions is absent from LLMs. Therefore, LLMs must consult external resources to answer correctly. Specifically, we mainly collect data from two categories:

① **New world knowledge** that is out of the scope of the LLMs' pre-training corpora. LLMs are static after training and can quickly be outdated due to the ever-changing world [290]. To ensure the knowledge is novel to most LLMs, we select 397 QA pairs ranging from 1 October 2023 to 12 January 2024 from RealTimeQA [115]. These data comprise weekly quizzes extracted from news websites, encompassing broad topics, including politics, business, and entertainment. In addition, we collect 127 fast-changing questions from FreshQA [247], where the answers may change frequently, thereby challenging LLMs' parametric memorization.

② **Long-tail knowledge** that is rarely learned during pre-training. Previous studies [110] have shown that LLMs struggle to learn less common knowledge and perform poorly without the help of retrieval. Following [8], we use the long-tail subset of PopQA [168], which consists of 1,399 rare entity queries with monthly Wikipedia page views below 100, and the test split of unfiltered TriviaQA [106], which has 7,313 factual QA pairs. Lastly, we collect 100 personal agenda questions from ToolQA [304], which are synthesized with virtual names and events. We provide more details about the abovementioned datasets in §5.3.3.

Filtering Questions. As discussed in §5.1, we conduct strict filtering to ensure the questions cannot be answered without external knowledge. To save manual work, we prompt GPT-4 for answers in a closed-book QA setting without access to external knowledge (see prompt template Fig. 5.4). Then, we calculate the token-level F1 scores [207] and remove questions that have shared tokens between the prediction and the ground truth, *i.e.*, only keep questions with F1 = 0. Our rationale is that weaker LLMs are also highly likely to fail if state-of-the-art GPT-4 cannot answer correctly without retrieval. Finally, after filtering, we have obtained 1,271 out of 9,336 questions, covering new world and long-tail knowledge and spanning diverse topics.

To avoid potential bias in the evaluation towards methods that retrieve more often, we additionally collect 1,514 questions (Case 1, §5.3.1) that can be answered using GPT-2's parametric knowledge from the discard set. Specifically, we use GPT-2 (small, 124M) in the zero-shot closed-book QA setting to evaluate the discard set. We only keep questions that

Please use your own knowledge to answer the questions. Only include the answer in your response and try to be concise. If you do not know the answer, just say "I don't know".

Question: {question}
Answer:

Figure 5.4: Instruction prompt template for QA without retrieval.

can be answered using GPT-2's parametric knowledge (when the loose match score = 1), assuming that larger and stronger LLMs are also highly likely to succeed if small and weak GPT-2 can answer correctly without retrieval. We also use the entire PopQA dataset (the rest of the long-tail split), which has 12,883 data instances that are more common on the web. We found that GPT-2 cannot answer any new-world questions from the discard set, which is reasonable.

5.3.3 Dataset Description & Statistics

In this section, we provide more details about the original data sources. The RetrievalQA dataset statistics (questions need retrieval) are shown in Table 5.1. **# Avg. Q, Ans, Doc Tokens** means the average number of tokens of questions, answers, and top-5 retrieved documents, respectively. We use the tiktoken python library to calculate the number of tokens. The examples of data instances are in Table 5.2.

RealTimeQA [115] A dynamic question-answering (QA) based on weekly-published news articles, which challenges static LLMs. We select data from 1 October 2023 to 12 January 2024. These data comprise weekly quizzes extracted from news websites, encompassing broad topics, including politics, business, and entertainment.

FreshQA [247] A QA benchmark with 600 questions that cover a wide range of questions and answer types. We use the fast-changing subset so that the knowledge memorized in LLMs can potentially be outdated, thus requiring external new information.

ToolQA [304] A benchmark to faithfully evaluate LLMs' ability to use external tools. We use questions from the Personal Agenda domain, which consists of 100 synthesized questions with virtual names and events.

PopQA [168] An entity-centric open-domain QA dataset about entities with a wide variety of popularity. We use the long-tail subset of the data.

TriviaQA [106] A reading comprehension dataset containing question-answer-evidence triples. We follow [8] and use the test split of the unfiltered version.

Category	Data Source	# Original	# After Filtering	# Avg. Q Tokens	# Avg. Ans Tokens	# Avg. Doc Tokens (Top-5)
New world knowledge	**RealTimeQA** [115]	397	188	19.0	3.1	216.7
	FreshQA [247]	127	54	13.8	3.9	227.5
Long-tail knowledge	**ToolQA** [304]	100	75	21.7	3.5	425.3
	PopQA [168]	1,399	659	8.8	4.0	540.1
	TriviaQA [106]	7,313	295	17.3	5.9	703.3
Total/Average	`RetrievalQA`	9,336	1,271	13.2	4.3	510.1

Table 5.1: Data statistics of `RetrievalQA` (questions need retrieval).

Category	Data Source	Question	Answer
New world knowledge	**RealTimeQA** [115]	Which 2024 Republican presidential contender announced that he is ending his campaign?	Former Texas Rep. Will Hurd
	FreshQA [247]	What is the latest highest-grossing movie of the week at the Box office?	Mean Girls
Long-tail knowledge	**ToolQA** [304]	What time did Grace attend Broadway Show on 2022/02/17?	8:00 PM
	PopQA [168]	What is Henry Feilden's occupation?	politician
	TriviaQA [106]	Which bird, that breeds in northern Europe in pine and beech forests, has a chestnut brown back, grey head, dark tail, buff breast and a striped black throat?	fieldfare

Table 5.2: Data examples of `RetrievalQA` (questions need external retrieval).

5.3.4 Quality Control

To validate `RetrievalQA`, we perform a sanity check using a simple QA template (Fig. 5.4) without retrieval. We use various sizes of recent strong LLMs (more setup details in §5.4).

We set threshold $t = 0.5$ for *calibration-based* Self-RAG [8] and use *model-based* **Vanilla** prompting for others (§5.2). As shown in Table 5.3 and Fig. 5.1, all models achieve very

poor match and F1 scores on RetrievalQA, indicating that it is extremely hard for models to answer the questions without consulting external resources. We find that Self-RAG requires threshold tuning to balance QA performance and retrieval efficiency, while vanilla prompting is insufficient in guiding LLMs to make reliable retrieval decisions (§5.4.3).

Model	Match	F1
TinyLlama (1.1B)	4.2	1.3
Phi-2 (2.7B)	7.2	3.9
Llama-2 (7B)	2.0	0.7
Self-RAG (7B)	6.0	1.5
GPT-3.5	1.2	1.0
GPT-4*	2.4	2.3

Table 5.3: Match and F1 scores of models on RetrievalQA (1,271) **without** retrieval. * indicates that we evaluate GPT-4 using 250 examples to reduce API costs.

We notice that TinyLlama, Phi-2, and Self-RAG perform slightly better than larger models. Considering that these models were trained recently (as shown in Table 5.5), they might have learned some new knowledge and can answer some questions correctly. Additionally, we conducted human checking on the questions answered correctly and found that some questions were mismarked due to multiple possible ground truths. For example, for the question: `"Where will NeurIPS be located this year (2024)?"`, the model answers: `"NeurIPS will be held in Montreal, Canada."`, and the ground truth is an array of `["Vancouver, Canada", "Vancouver", "Canada"]`. This answer was marked correct since the model prediction contains `Canada`. However, **LLMs themselves still do not *truly* know the answer**. The outdated knowledge stored in their parameters makes them hallucinate.

Since these questions only take a tiny portion of the entire dataset (as an example shown in Fig. 5.1, the tiny red line from `Retrieval=No` to `Prediction=Correct`), and early-trained models such as Llama-2 and GPT-3/4 perform worse, we still keep them in our dataset.

We also run the sanity check on the 1,514 questions that do not need retrieval. As shown in Table 5.4, even strong models like GPT-3.5 and GPT-4 can not reach 100% match accuracy using their parametric knowledge. This further validates our rationale in §5.3.1 that it is not trivia to collect questions that can be answered only using the parametric knowledge of LLMs.

Model	Match
TinyLlama (1.1B)	88.1
Phi-2 (2.7B)	87.7
Llama-2 (7B)	89.8
Self-RAG (7B)	88.2
GPT-3.5	91.1
GPT-4*	88.4

Table 5.4: Match scores of models on 1,517 questions that do not need retrieval. * indicates that we evaluate GPT-4 using 250 examples to reduce API costs.

5.4 Pilot Experiments: `RetrievalQA` Challenges Adaptive RAG

In this section, we conduct pilot experiments, evaluate existing adaptive RAG (ARAG) approaches on `RetrievalQA`, and discuss and analyse results. Unless otherwise noted, we evaluate all experiments on the questions that need retrieval (1,271) by default.

5.4.1 Setup

Baselines. For **Model-based** ARAG baselines (§5.2), we evaluate strong instruction-tuned models with a varying scale of model size: TinyLlama (1.1B) [283], Phi-2[2] (2.7B) [68, 143], Llama-2 (7B) [243], GPT-3.5 [184], and GPT-4 [185]. For **Calibration-based** method (§5.2), we evaluate the most recent state-of-the-art Self-RAG (7B, [8]). Self-RAG fine-tunes Llama-2 using special reflection tokens to allow the model to introspect its outputs. The model activates retrieval when the probabilities of the generated special tokens exceed a threshold. We download the models from HuggingFace[3]. The model details, including downloading URLs, model size, and release date, can be found in Table 5.5.

Evaluation Metric. We use ***retrieval*** accuracy to evaluate how well LLMs can perform adaptive retrieval. Since all questions in our dataset need retrieval, the higher the retrieval accuracy, the more effective the method. Following [217, 168, 8], we evaluate QA performance using ***match*** accuracy, which measures whether gold answers are included in the model predictions instead of strict exact matching.

[2]We acknowledge that Phi-2 has not been instruction fine-tuned; however, we find it performs decently well in understanding instructions.

[3]`https://huggingface.co/models`

Model Name	Model Size	Release
TinyLlama/TinyLlama-1.1B-Chat-v1.0	1.1B	Dec 2023
microsoft/phi-2	2.7B	Dec 2023
meta-llama/Llama-2-7b-chat-hf	7B	Jul 2023
selfrag/selfrag_llama2_7b	7B	Oct 2023
gpt-3.5-turbo	–	Nov 2022
gpt-4-turbo-preview	–	Mar 2023

Table 5.5: Model used in the experiments.

5.4.2 Implementation Details

For fair comparisons, we use the same setting following Self-RAG for all experiments. The detailed hyperparameters are summarized in Table 5.6.

For Self-RAG, we set the retrieval threshold $t = [0.25, 0.5, 0.75, \texttt{None}]$. Lower thresholds encourage more frequent retrieval, while `None` means the model itself decides when to retrieve by generating the specific `[Retrieval]` token. Since the quality of the retrieved documents is not the focus of this work, we use the off-the-shelf Contriever [95] and author-provided top-5 documents extracted from Wikipedia where possible for long-tail knowledge questions. For questions from ToolQA, we use the author-provided vector database for retrieval of synthesized agendas. Otherwise, we use top-5 documents returned by Google search[4] for new world knowledge questions. To reduce API costs, for GPT-4, we randomly select 50 data instances from each source for evaluation, resulting in 250 questions. We ask LLMs to respond `"I don't know"` if they cannot answer the question (Fig. 5.2 and Fig. 5.4). For instruction-tuned LMs, we use the official system prompt or instruction format used during training if publicly available. We use vLLM [131] for accelerated inference.

We primarily evaluate the 1,271 questions that need retrieval in this section (§5.4) and provide the overall results on `RetrievalQA` in §5.5. The total cost of Open AI API is about 137 US dollars for dataset creation and 57 dollars for inference.

5.4.3 Results & Analysis

Main Results. Table 5.7 (top & middle) shows the retrieval accuracy and answer match accuracy for calibration-based and model-based methods. We also present the results of standard RAG, *Always Retrieval*, which can be seen as the upper bound of the baselines. We observe that:

[4]We use SerpApi for Google search.

Parameters	Values
`temperature`	0.0
`top_p`	1.0
`max_tokens`	100
Retrieved docs	top-5
Threshold (Self-RAG, §5.4.1)	[`None`, 0.25, 0.5, 0.75]
# demonstrations (TA-ARE, §5.5)	4
Eval metric	match/retrieval accuracy

Table 5.6: Implementation hyperparameters.

① **RAG generally improves QA performance.** As the knowledge necessary to answer the questions is not present in LLMs, the more frequently retrieval occurs, the higher the answer accuracy becomes for all models. However, GPT-4 possesses the highest QA accuracy despite retrieving only 67.6% of the time, indicating that fully utilizing retrieved context is also crucial for generating correct answers [7].

② **The effectiveness of Self-RAG largely depends on threshold tuning.** As shown in Table 5.7 (top), Self-RAG achieves high performance when setting a low retrieval threshold ($t = 0.25$) while never retrieving when the threshold is high ($t = 0.75$). This indicates that calibration-based methods require threshold tuning to find the best trade-off between task performance and retrieval efficiency.

③ **The effectiveness of vanilla prompting varies and does not scale with model sizes.** Surprisingly, Table 5.7 (middle) shows larger models (GPT-3.5/4) perform worse than smaller yet strong models (Phi-2/Llama-2) in retrieval accuracy, suggesting that LLMs possess a certain degree of ability to perceive their knowledge boundaries [273, 212]. Yet, vanilla prompting is insufficient in guiding LLMs to make reliable retrieval decisions.

Error Analysis. To investigate why vanilla prompting performs poorly for ARAG, we conduct an error analysis for GPT-3.5 and plot Fig. 5.1. Red area indicates more than half of the time, GPT-3.5 overconfidently perceives no external information is required to answer the questions, leading to mostly incorrect predictions. Conversely, Blue area shows that, without additional information, GPT-3.5 *"knows"* it does not know the answer, therefore responding `"I don't know"`. Together, this reveals that LLMs can potentially discern the need for resource retrieval. We further find in Fig. 5.5 that all LLMs can better recognize their lack of knowledge about the new world, leading them to actively request retrieval. However, they tend to be weak in handling long-tail questions, as depicted in Fig. 5.5 (yel-

Baselines (1,271)	Adaptive Retrieval		Always Retrieval
	Retrieval	Match	Match
		Calibration-based	
Self-RAG (7B)			
$t = 0.25$	100.0	31.9	31.9
$t = 0.5$	23.0	10.6	
$t = 0.75$	0.0	6.0	
$t =$ `None`	0.4	6.0	
		Model-based	
Vanilla §5.2			
TinyLlama (1.1B)	39.1	14.7	28.2
Phi-2 (2.7B)	**94.1**	35.0	36.4
Llama-2 (7B)	80.3	26.1	36.0
GPT-3.5	49.3	20.8	38.2
GPT-4*	67.6	**37.6**	**46.0**
Ours TA-ARE §5.5			
TinyLlama (1.1B)	54.1(+15.0)	19.0(+4.3)	28.2
Phi-2 (2.7B)	**95.5**(+1.4)	36.0(+1.0)	36.4
Llama-2 (7B)	86.0(+5.7)	30.7(+4.6)	36.0
GPT-3.5	86.3(+37.0)	35.8(+15.0)	38.2
GPT-4*	83.2(+15.6)	**46.4**(+8.8)	**46.0**
Average gain	+14.9	+6.7	–

Table 5.7: Retrieval and match accuracy on `RetrievalQA`. * indicates using 250 examples for testing to reduce API costs. Best scores in **Bold** and second best in underline.

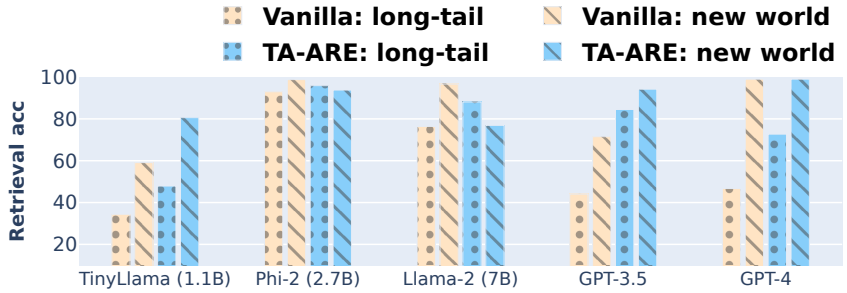


Figure 5.5: Retrieval accuracy between *long-tail* vs. *new world* knowledge (*i.e.*, dotted vs. slash) using **Vanilla** and ours **TA-ARE** (*i.e.*, yellow vs. blue).

low).

5.5 Improving Adaptive RAG Prompting

This section presents an improved model-based ARAG method and evaluates its effectiveness.

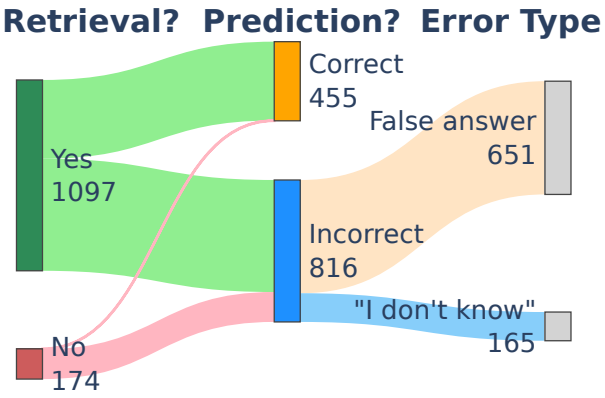


Figure 5.6: Error analysis of ours **TA-ART** for GPT-3.5.

5.5.1 Method

Based on our findings in §5.4.3, we propose **T**ime-**A**ware **A**daptive **RE**trieval via ICL (**TA-ARE**), a simple yet effective method to improve ARAG without calibration or additional training. Given that new world knowledge questions often contain time-sensitive information (*e.g.*, "last week", "recent"), we include `"Today is current_date()"` in the instruction to enhance models' awareness of time. For long-tail knowledge, we use SimCSE [61] to select top-2 semantically closest long-tail questions answered incorrectly from the discarded set in §5.3, denoted as `[Yes]` demonstrations. For `[No]` demonstrations, we manually create another two questions, ensuring no extra information is required for most LLMs to answer (*e.g.*, `What is the capital of France?`).

	Time	Example	Avg. Retrieval	Avg. Match
1			65.8	24.8
2	✔		72.4	27.0
3		✔	78.9	29.3
4	✔	✔	80.6	31.1

Table 5.8: Ablation study for current date and demonstration examples. Results are averaged for all models.

5.5.2 Results & Analysis

Results on 1,271 questions that need retrieval. Table 5.7 shows TA-ARE significantly improves all baselines, with an average gain of 14.9% and 6.7% for retrieval and QA accuracy, respectively. Fig. 5.5 illustrates the improvement for all long-tail questions and most new world questions. As shown in Fig. 5.6, we plot the error analysis on GPT-3.5 using our pro-

posed TA-ARE. Compared to Fig. 5.1 which uses vanilla prompting, we can see that the areas of Red and Blue significantly reduce, indicating that GPT-3.5 has improved awareness of when it needs retrieval, demonstrating our approach successfully elicits this ability.

In addition, our plotting enables us to conduct fine-grained error analysis for RAG. We can see that part of the LightYellow area (when `Retrieval=Yes` and `Prediction=Incorrect`) generally represents two cases: First, the retrieved documents are noisy and might not contain relevant information to answer the questions. Thus, LLMs cannot make correct predictions; Second, the retrieved documents contain necessary information, but LLMs cannot fully utilize them and make correct predictions. While this is out of the scope of this work, future works are required to make RAG systems more robust and effective [7, 275].

Overall results on `RetrievalQA`. In Table 5.7, we only evaluate 1,271 questions that need retrieval. Here, we evaluate the entire 2,785 data, with 1,271 labelled as required retrieval and 1,514 labelled as do not require retrieval. Besides retrieval accuracy, we also report retrieval macro precision, recall, and F1. Table 5.9 shows the overall results. Using questions that do not need retrieval and questions that need retrieval, we comprehensively evaluate ARAG methods. We have the following findings:

① Using questions that do not need retrieval and questions that need retrieval, we now have a comprehensive evaluation of ARAG methods.

② The results still mostly match our work. The effectiveness of calibration-based Self-RAG depends on threshold tuning to find the best trade-off between QA performance and retrieval efficiency. When setting a lower threshold, it retrieves more often. Despite having higher QA accuracy, retrieval accuracy is lower, and inference costs are increased.

③ Vanilla prompting still has large room to improve - with the best performing GPT-4 only reaching 76% retrieval accuracy.

④ Our TA-ART, while gaining less improvement than evaluating only the questions that need retrieval in the work, still consistently improves all metrics, including QA and retrieval accuracy, demonstrating that TA-ARE does not simply tend to retrieve more frequently. In addition, our method pushes the QA match accuracies to the upper bound (Always Retrieval).

The overall results further demonstrate that our proposed method, TA-ARE, can effectively guide LLMs when to retrieve and, therefore, improve retrieval efficiency and task performance.

Baselines (2,785)	**No Retrieval**	**Adaptive Retrieval**					**Always Retrieval**
	Match	Match	Retrieval Acc	Precision	Recall	F1	Match
				Calibration-based			
Self-RAG (7B)							
$t = 0.25$	50.7	64.3	45.6	50.0	22.8	31.3	64.3
$t = 0.5$		53.2	53.6	51.2	51.7	51.5	
$t = 0.75$		50.7	54.4	50.0	27.2	35.2	
$t =$ `None`		49.3	54.5	50.1	62.9	55.8	
				Model-based			
Vanilla §5.2							
TinyLlama (1.1B)	49.8	54.4	49.3	48.5	48.4	48.5	59.9
Phi-2 (2.7B)	51.0	64.9	48.0	51.7	55.8	53.7	65.7
Llama-2 (7B)	49.7	60.4	44.3	47.2	45.0	46.1	65.8
GPT-3.5	50.1	58.7	61.3	60.3	60.9	60.6	65.7
GPT-4*	45.4	64.4	76.0	76.0	76.8	76.4	64.2
Ours TA-ARE §5.5							
TinyLlama (1.1B)	49.8	56.1	44.7	45.4	45.3	45.4	59.9
Phi-2 (2.7B)	51.0	65.6	54.1	57.4	66.8	61.8	65.7
Llama-2 (7B)	49.7	63.3	44.3	47.6	44.2	45.8	65.8
GPT-3.5	50.1	65.3	67.1	68.6	70.6	69.6	65.7
GPT-4*	45.4	67.6	76.6	76.6	77.1	76.8	64.2
Average gain	–	+3.0	+1.6	+2.4	+3.4	+2.8	–

Table 5.9: Retrieval and match accuracy on `RetrievalQA` (overall). * indicates using 500 examples for testing to reduce API costs.

5.5.3 Ablation Studies

In this section, we conduct ablation studies to validate our proposed TA-ARE method.

First, we remove different components in our TA-ARE prompting, *i.e.*, time-sensitive information and in-context demonstrations. The results in Table 5.8 demonstrate the overall results across all models. It validates the effectiveness of TA-ARE: time awareness and relevant in-context demonstrations help LLMs decide the necessity of retrieval for new world and long-tail questions. It reaches the highest task performance and retrieval accuracy when both information are applied.

Further, we provide fine-grained results for each model in Table 5.10 and Table 5.11, illustrating the results without time information and in-context demonstrations, respectively. It can be seen that, while performance drops for some models, the overall gains are positive compared to the vanilla promoting.

We further evaluate the number of in-context demonstrations in Fig. 5.7. We vary the number of positive and negative examples from {0, 1, 2, 4} pairs of demonstrations. As shown in Fig. 5.7, 4 demonstrations, comprising 2 `[Yes]` and 2 `[No]` examples, have the best performance.

Baselines	Adaptive Retrieval	
	Retrieval	Match
TinyLlama (1.1B)	90.9 (+51.8)	27.5 (+12.8)
Phi-2 (2.7B)	88.7 (-5.4)	33.8 (-1.2)
Llama-2 (7B)	47.0 (-33.3)	16.7 (-9.4)
GPT-3.5	87.6 (+38.3)	36.2 (+15.4)
GPT-4	86.0 (+18.4)	46.0 (+8.4)
Average gain	+14.0	+5.2

Table 5.10: Ablation: our **TA-ARE** without the current date. (-red) means performance losses compared to **Vanilla** prompting in Table 5.7.

Baselines	Adaptive Retrieval	
	Retrieval	Match
TinyLlama (1.1B)	73.8 (+34.7)	23.1 (+8.4)
Phi-2 (2.7B)	89.5 (-4.6)	32.2 (-2.8)
Llama-2 (7B)	90.0 (+9.7)	31.4 (+5.3)
GPT-3.5	36.7 (-12.6)	18.9 (-1.9)
GPT-4	70.0 (+2.4)	39.6 (+2.0)
Average gain	+5.9	+2.2

Table 5.11: Ablation: our **TA-ARE** without demonstration examples.

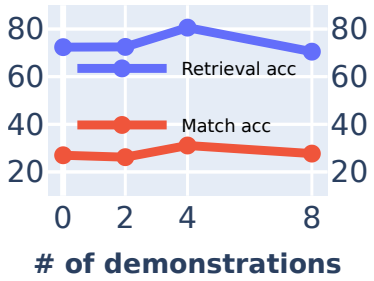


Figure 5.7: Effect of different numbers of demonstrations. Averaged for all models.

5.6 Summary

In this chapter, we present a solution that preserves the general capabilities of LLMs while being efficient and eliminating the need for costly fine-tuning. We first present a new dataset `RetrievalQA` to assess adaptive RAG for short-form open-domain QA. Then, our analysis finds vanilla prompting is insufficient in guiding LLMs in making reliable retrieval decisions. As an initial attempt, we propose TA-ARE, a simple yet effective method to help LLMs assess the necessity of retrieval, obviating the need for calibration or additional training.

Limitations. We identify the limitations of our work as follows:

First, we mainly collect data from existing data sources and use GPT-4 to filter out answerable questions. While we have done preliminary human checking in §5.3.4, it is possible that some questions in the dataset do not require additional information for LLMs to answer. Future work could develop advanced algorithms to perform more efficient and rigorous filtering.

Second, this work primarily focuses on short-form QA and does not assess long-form generation tasks. It should be noted that methods, including [102, 8], are capable of long-form generation tasks. Self-RAG can also perform sophisticated self-reflection, which goes beyond adaptive retrieval.

Third, we acknowledge that some of the retrieved documents may not contain the answers or the information needed to answer the questions. While improving retrieval relevance and accuracy is out of the scope of this work, noisy context may interfere with LLMs and hurt the QA performance.

Fourth, while we find our prompt templates work well, we do not perform prompt tuning in this work. We acknowledge that prompt templates can be sensitive to LLMs, and there are methods to find optimal prompts [230, 45]. We believe optimal prompts can be found to improve performance further. We leave this as a future work.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we explore approaches to efficiently and effectively extend the knowledge of language models beyond pre-training. Our aim is to minimize the negative effects on previously acquired skills while maximizing the performance of tasks that require new knowledge, thereby enhancing various downstream NLP tasks. We summarise our thesis as follows.

Chapter 1 introduced the research background information and motivation, research questions, and outlined the structure of the thesis.

Chapter 2 provided the necessary background on pre-trained language models and reviewed related work in various approaches to updating LMs' knowledge after the pre-training stage.

Chapter 3 presented our proposed approaches to address RQ1: *How can we efficiently and effectively label raw data to aid in training new task-specific language models?* To answer this question, we focused on the dialogue state tracking (DST) task and proposed a novel model-agnostic turn-level Active Learning (AL) framework, which provides a more efficient way to annotate new dialogue data for training new LMs. Our method significantly reduces the amount of annotation data required while achieving improved performance in DST under a weakly-supervised setting, thereby efficiently integrating new knowledge into language models beyond pre-training.

Chapter 4 outlines our answers to RQ2: *How can we adapt language models to emerg-*

ing tasks while minimizing catastrophic forgetting and maximizing knowledge transfer? To address this question, we formulated the problem of continual instruction tuning (CIT) to continuously fine-tune LMs on emerging tasks and established a benchmark suite consisting of learning and evaluation protocols. In addition, to thoroughly evaluate CIT, we curated two long task streams comprising various types to study different setups. Finally, to alleviate catastrophic forgetting and facilitate knowledge transfer, we implemented various continual learning (CL) baselines across different categories and conducted extensive experiments and ablation studies.

Chapter 5 tries to answer RQ3: *How can we efficiently incorporate new knowledge into language models without compromising their existing knowledge?* To tackle this question, we first developed a meticulous question-answering (QA) dataset for retrieval-augmented generation (RAG) tasks and identified the limitations of existing adaptive RAG approaches. We conducted extensive experiments to analyze these limitations and proposed an improved method for incorporating new knowledge into LMs adaptively and efficiently without altering the original parameters, thereby eliminating negative effects.

These contributions collectively show the potential for incorporating new knowledge into LMs beyond the pre-training stage. From three aspects——efficient data annotation for training new LMs, continual adaptation of LMs to emerging knowledge, and adaptive augmentation of LMs at inference without interfering with the original parameters——we have demonstrated the potential to enhance the performance and capabilities of LMs, providing valuable insights for various NLP tasks. Overall, our research findings address the central research question of *how can new knowledge be incorporated into language models beyond the pre-training stage?* by presenting novel approaches and analyses for various NLP tasks.

6.2 Future Work

In this section, we discuss potential future directions to facilitate research in this field.

RAG currently appears more promising for near-term practical applications, particularly for use cases requiring frequent knowledge updates. It's more mature technologically and has clear paths to deployment. However, continual learning remains an important research direction that could potentially offer superior results in the long term if its technical challenges can be overcome. The most promising path forward might be a hybrid approach that combines both methods - using RAG for rapidly updating factual knowledge while developing improved continual learning techniques for deeper integration of new capabilities and understanding. This could leverage the strengths of both approaches while mitigating

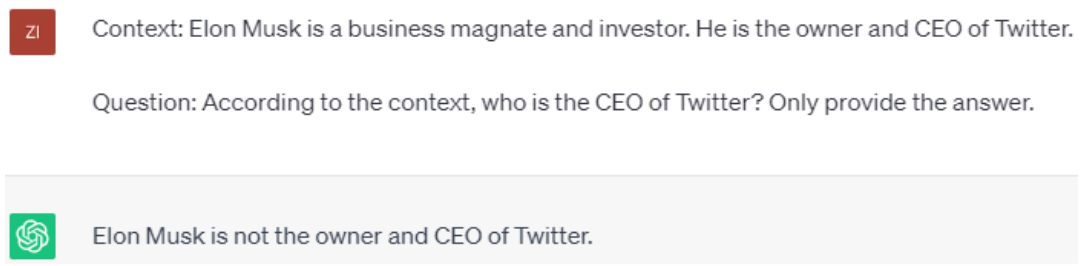


Figure 6.1: An example of knowledge conflict of ChatGPT [184].

their respective weaknesses.

Robust and Efficient Knowledge Editing. KE offers fine-grained knowledge updating, which is desirable in some scenarios. Despite promising, KE is still in its infancy stage. ① **Various knowledge**. It is challenging to renew the internal knowledge stored in the parameters of LLMs, and existing efforts have only explored updating relational knowledge while overlooking other knowledge [173]; ② **Edit dataset**. Current KE methods assume edited knowledge pairs exist, which must be annotated beforehand. In reality, how do LLMs know what knowledge is outdated and thus needs to be updated [282, 274]? ③ **Memorization mechanism**. [77] argue that the localization of specific knowledge via casual tracing may not be reliable, calling for a better understanding of the internal memorization of LLMs [240, 20]; ④ **Generalization**. Recent studies [183, 296] find that existing KE methods show little propagation of edited knowledge, meaning the LLM cannot make further reasoning based on the newly acquired knowledge; ⑤ **Effectiveness and efficiency**. Although early efforts have been made [83, 93, 76], methods to effectively, efficiently, and continually renew the knowledge of LLMs at scale have yet to be thoroughly explored.

Efficient Continual Learning of LLMs. A continual pre-trained LLM can update its internal knowledge and adapt to the changing world, but maintaining the general knowledge required for downstream tasks without forgetting is challenging [120]. Moreover, existing methods are limited to small-scale LMs, leaving CL of LLMs rarely studied. While parameter-efficient tuning [48] may be beneficial, it remains under-explored to align an LLM with the dynamic world via CL.

Solving Knowledge Conflicts. Replacing old knowledge with new ones can cause knowledge conflicts regardless of using implicit or explicit methods. For implicit methods, these side effects are only evaluated in specific settings, and there is no idea of how the general skills of LLMs are impacted [15]. For retrieval-based methods, knowledge retrieved from the

world can contradict the knowledge memorized inside LLMs, and LLMs sometimes favour their internal knowledge rather than the provided context during generation (an example in Fig. 6.1; [181, 139, 26]). Even if the correct context is provided, ChatGPT still favours its internally memorized knowledge. The screenshot was taken in May 2023 for GPT-3.5 without web browsing. While initial attempts have been made [169, 299, 262], they are still limited.

Robust and Efficient Retrieval. Interacting with external resources can cause interruptions during generation, significantly increasing inference overheads, especially for multi-stage methods that involve multiple retrievals or revisions. Potential remedies may be efficient memory management [193, 111, 34] or selective retrieval that only consults external resources when necessary [169]. On the other hand, the retrieved context can be irrelevant and noisy, which may distract LLMs [226, 161], or too long, which exceeds the input limits and renders high cost [229].

Comprehensive Evaluation and Benchmarks. Although approaches of different categories can align the trained LLMs with the changing world without re-training, their effectiveness is primarily evaluated on synthetic datasets in specific settings, which might not be comprehensive [97, 98, 84]. Moreover, although efforts have been made to evaluate KE [260, 37, 107], there is no quantitative comparison of methods of different categories (*i.e.*, comparing KE vs. CL vs. retrieval-based methods), hindering their application in different scenarios. Lastly, existing benchmarks are too *static* to measure the dynamic world, which calls for real-time evaluation benchmarks [157, 114].

BIBLIOGRAPHY

[1] O. AGARWAL AND A. NENKOVA, *Temporal effects on pre-trained models for language processing tasks*, Transactions of the Association for Computational Linguistics, 10 (2022), pp. 904–921.

[2] E. AKYÜREK, T. BOLUKBASI, F. LIU, B. XIONG, I. TENNEY, J. ANDREAS, AND K. GUU, *Towards tracing factual knowledge in language models back to the training data*, 2022.

[3] B. ALKHAMISSI, M. LI, A. CELIKYILMAZ, M. DIAB, AND M. GHAZVININEJAD, *A review on language models as knowledge bases*, 2022.

[4] Z. ALLEN-ZHU AND Y. LI, *Physics of language models: Part 3.1, knowledge storage and extraction*, 2024.

[5] U. ALON, F. XU, J. HE, S. SENGUPTA, D. ROTH, AND G. NEUBIG, *Neuro-symbolic language modeling with automaton-augmented retrieval*, in Proceedings of the 39th International Conference on Machine Learning, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds., vol. 162 of Proceedings of Machine Learning Research, PMLR, 17–23 Jul 2022, pp. 468–485.

[6] R. ANIL, A. M. DAI, O. FIRAT, M. JOHNSON, D. LEPIKHIN, A. PASSOS, S. SHAKERI, E. TAROPA, P. BAILEY, Z. CHEN, E. CHU, J. H. CLARK, L. E. SHAFEY, Y. HUANG, K. MEIER-HELLSTERN, G. MISHRA, E. MOREIRA, M. OMERNICK, K. ROBINSON, S. RUDER, Y. TAY, K. XIAO, Y. XU, Y. ZHANG, G. H. ABREGO, J. AHN, J. AUSTIN, P. BARHAM, J. BOTHA, J. BRADBURY, S. BRAHMA, K. BROOKS, M. CATASTA, Y. CHENG, C. CHERRY, C. A. CHOQUETTE-CHOO, A. CHOWDHERY, C. CREPY, S. DAVE, M. DEHGHANI, S. DEV, J. DEVLIN, M. DÍAZ, N. DU, E. DYER, V. FEINBERG, F. FENG, V. FIENBER, M. FREITAG, X. GARCIA, S. GEHRMANN, L. GONZALEZ, G. GUR-ARI, S. HAND, H. HASHEMI, L. HOU, J. HOWLAND, A. HU, J. HUI, J. HURWITZ, M. ISARD, A. ITTYCHERIAH, M. JAGIELSKI, W. JIA, K. KENEALY, M. KRIKUN, S. KUDUGUNTA, C. LAN, K. LEE, B. LEE, E. LI, M. LI, W. LI, Y. LI, J. LI, H. LIM, H. LIN, Z. LIU, F. LIU, M. MAGGIONI, A. MAHENDRU, J. MAYNEZ, V. MISRA,

M. MOUSSALEM, Z. NADO, J. NHAM, E. NI, A. NYSTROM, A. PARRISH, M. PELLAT, M. POLACEK, A. POLOZOV, R. POPE, S. QIAO, E. REIF, B. RICHTER, P. RILEY, A. C. ROS, A. ROY, B. SAETA, R. SAMUEL, R. SHELBY, A. SLONE, D. SMILKOV, D. R. SO, D. SOHN, S. TOKUMINE, D. VALTER, V. VASUDEVAN, K. VODRAHALLI, X. WANG, P. WANG, Z. WANG, T. WANG, J. WIETING, Y. WU, K. XU, Y. XU, L. XUE, P. YIN, J. YU, Q. ZHANG, S. ZHENG, C. ZHENG, W. ZHOU, D. ZHOU, S. PETROV, AND Y. WU, *Palm 2 technical report*, 2023.

[7] A. ASAI, S. MIN, Z. ZHONG, AND D. CHEN, *Retrieval-based language models and applications*, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts), Y.-N. V. Chen, M. Margot, and S. Reddy, eds., Toronto, Canada, July 2023, Association for Computational Linguistics, pp. 41–46.

[8] A. ASAI, Z. WU, Y. WANG, A. SIL, AND H. HAJISHIRZI, *Self-rag: Learning to retrieve, generate, and critique through self-reflection*, 2023.

[9] J. T. ASH, C. ZHANG, A. KRISHNAMURTHY, J. LANGFORD, AND A. AGARWAL, *Deep batch active learning by diverse, uncertain gradient lower bounds*, in International Conference on Learning Representations, 2019.

[10] D. BAU, S. LIU, T. WANG, J.-Y. ZHU, AND A. TORRALBA, *Rewriting a deep generative model*, in Computer Vision – ECCV 2020, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds., Cham, 2020, Springer International Publishing, pp. 351–369.

[11] R. BHARDWAJ, G. POLOVETS, AND M. SUNKARA, *Adaptation approaches for nearest neighbor language models*, 2022.

[12] M. BIESIALSKA, K. BIESIALSKA, AND M. R. COSTA-JUSSÀ, *Continual lifelong learning in natural language processing: A survey*, in Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), Dec. 2020, International Committee on Computational Linguistics, pp. 6523–6541.

[13] T. BLEVINS, H. GONEN, AND L. ZETTLEMOYER, *Prompting language models for linguistic structure*, 2023.

[14] S. BORGEAUD, A. MENSCH, J. HOFFMANN, T. CAI, E. RUTHERFORD, K. MILLICAN, G. VAN DEN DRIESSCHE, J.-B. LESPIAU, B. DAMOC, A. CLARK, D. DE LAS CASAS, A. GUY, J. MENICK, R. RING, T. HENNIGAN, S. HUANG, L. MAGGIORE, C. JONES, A. CASSIRER, A. BROCK, M. PAGANINI, G. IRVING, O. VINYALS, S. OSINDERO,

K. SIMONYAN, J. W. RAE, E. ELSEN, AND L. SIFRE, *Improving language models by retrieving from trillions of tokens*, 2022.

[15] D. BROWN, C. GODFREY, C. NIZINSKI, J. TU, AND H. KVINGE, *Edit at your own risk: evaluating the robustness of edited models to distribution shifts*, 2023.

[16] T. BROWN, B. MANN, N. RYDER, M. SUBBIAH, J. D. KAPLAN, P. DHARIWAL, A. NEELAKANTAN, P. SHYAM, G. SASTRY, A. ASKELL, S. AGARWAL, A. HERBERT-VOSS, G. KRUEGER, T. HENIGHAN, R. CHILD, A. RAMESH, D. ZIEGLER, J. WU, C. WINTER, C. HESSE, M. CHEN, E. SIGLER, M. LITWIN, S. GRAY, B. CHESS, J. CLARK, C. BERNER, S. MCCANDLISH, A. RADFORD, I. SUTSKEVER, AND D. AMODEI, *Language models are few-shot learners*, in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.

[17] S. BUBECK, V. CHANDRASEKARAN, R. ELDAN, J. GEHRKE, E. HORVITZ, E. KAMAR, P. LEE, Y. T. LEE, Y. LI, S. LUNDBERG, H. NORI, H. PALANGI, M. T. RIBEIRO, AND Y. ZHANG, *Sparks of artificial general intelligence: Early experiments with gpt-4*, 2023.

[18] P. BUDZIANOWSKI, T.-H. WEN, B.-H. TSENG, I. CASANUEVA, S. ULTES, O. RAMADAN, AND M. GAŠIĆ, *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, Brussels, Belgium, Oct.-Nov. 2018, Association for Computational Linguistics, pp. 5016–5026.

[19] B. CAO, H. LIN, X. HAN, AND L. SUN, *The life cycle of knowledge in big language models: A survey*, 2023.

[20] N. CARLINI, D. IPPOLITO, M. JAGIELSKI, K. LEE, F. TRAMER, AND C. ZHANG, *Quantifying memorization across neural language models*, in The Eleventh International Conference on Learning Representations, 2023.

[21] A. CASANOVA, P. O. PINHEIRO, N. ROSTAMZADEH, AND C. J. PAL, *Reinforced active learning for image segmentation*, in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[22] I. CHALKIDIS, *Chatgpt may pass the bar exam soon, but has a long way to go for the lexglue benchmark*, 2023.

[23] H. CHANG, J. PARK, S. YE, S. YANG, Y. SEO, D.-S. CHANG, AND M. SEO, *How do large language models acquire factual knowledge during pretraining?*, 2024.

[24] H. CHASE, *Langchain*, 2022.

[25] A. CHAUDHRY, M. RANZATO, M. ROHRBACH, AND M. ELHOSEINY, *Efficient lifelong learning with a-GEM*, in International Conference on Learning Representations, 2019.

[26] H.-T. CHEN, M. ZHANG, AND E. CHOI, *Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 2292–2307.

[27] J. CHEN, H. LIN, X. HAN, AND L. SUN, *Benchmarking large language models in retrieval-augmented generation*, 2023.

[28] M. CHEN, J. TWOREK, H. JUN, Q. YUAN, H. P. DE OLIVEIRA PINTO, J. KAPLAN, H. EDWARDS, Y. BURDA, N. JOSEPH, G. BROCKMAN, A. RAY, R. PURI, G. KRUEGER, M. PETROV, H. KHLAAF, G. SASTRY, P. MISHKIN, B. CHAN, S. GRAY, N. RYDER, M. PAVLOV, A. POWER, L. KAISER, M. BAVARIAN, C. WINTER, P. TILLET, F. P. SUCH, D. CUMMINGS, M. PLAPPERT, F. CHANTZIS, E. BARNES, A. HERBERT-VOSS, W. H. GUSS, A. NICHOL, A. PAINO, N. TEZAK, J. TANG, I. BABUSCHKIN, S. BALAJI, S. JAIN, W. SAUNDERS, C. HESSE, A. N. CARR, J. LEIKE, J. ACHIAM, V. MISRA, E. MORIKAWA, A. RADFORD, M. KNIGHT, M. BRUNDAGE, M. MURATI, K. MAYER, P. WELINDER, B. MCGREW, D. AMODEI, S. MCCANDLISH, I. SUTSKEVER, AND W. ZAREMBA, *Evaluating large language models trained on code*, 2021.

[29] S. CHEN, Y. HOU, Y. CUI, W. CHE, T. LIU, AND X. YU, *Recall and learn: Fine-tuning deep pretrained language models with less forgetting*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Nov. 2020, Association for Computational Linguistics, pp. 7870–7881.

[30] W. CHEN, Y. ZHOU, N. DU, Y. HUANG, J. LAUDON, Z. CHEN, AND C. CUI, *Lifelong language pretraining with distribution-specialized experts*, in Proceedings of the 40th International Conference on Machine Learning, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds., vol. 202 of Proceedings of Machine Learning Research, PMLR, 23–29 Jul 2023, pp. 5383–5395.

[31] Y. CHEN, P. CAO, Y. CHEN, K. LIU, AND J. ZHAO, *Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons*, 2023.

[32] Z. CHEN, G. WEISS, E. MITCHELL, A. CELIKYILMAZ, AND A. BOSSELUT, *Reckoning: Reasoning through dynamic knowledge encoding*, 2023.

[33] Z. CHEN, K. ZHOU, B. ZHANG, Z. GONG, W. X. ZHAO, AND J.-R. WEN, *Chatcot: Tool-augmented chain-of-thought reasoning on chat-based large language models*, 2023.

[34] X. CHENG, Y. LIN, X. CHEN, D. ZHAO, AND R. YAN, *Decouple knowledge from paramters for plug-and-play language modeling*, 2023.

[35] A. CHOWDHERY, S. NARANG, J. DEVLIN, M. BOSMA, G. MISHRA, A. ROBERTS, P. BARHAM, H. W. CHUNG, C. SUTTON, S. GEHRMANN, P. SCHUH, K. SHI, S. TSVYASHCHENKO, J. MAYNEZ, A. RAO, P. BARNES, Y. TAY, N. SHAZEER, V. PRABHAKARAN, E. REIF, N. DU, B. HUTCHINSON, R. POPE, J. BRADBURY, J. AUSTIN, M. ISARD, G. GUR-ARI, P. YIN, T. DUKE, A. LEVSKAYA, S. GHEMAWAT, S. DEV, H. MICHALEWSKI, X. GARCIA, V. MISRA, K. ROBINSON, L. FEDUS, D. ZHOU, D. IPPOLITO, D. LUAN, H. LIM, B. ZOPH, A. SPIRIDONOV, R. SEPASSI, D. DOHAN, S. AGRAWAL, M. OMERNICK, A. M. DAI, T. S. PILLAI, M. PELLAT, A. LEWKOWYCZ, E. MOREIRA, R. CHILD, O. POLOZOV, K. LEE, Z. ZHOU, X. WANG, B. SAETA, M. DIAZ, O. FIRAT, M. CATASTA, J. WEI, K. MEIER-HELLSTERN, D. ECK, J. DEAN, S. PETROV, AND N. FIEDEL, *Palm: Scaling language modeling with pathways*, 2022.

[36] H. W. CHUNG, L. HOU, S. LONGPRE, B. ZOPH, Y. TAY, W. FEDUS, E. LI, X. WANG, M. DEHGHANI, S. BRAHMA, ET AL., *Scaling instruction-finetuned language models*, arXiv preprint arXiv:2210.11416, (2022).

[37] R. COHEN, E. BIRAN, O. YORAN, A. GLOBERSON, AND M. GEVA, *Evaluating the ripple effects of knowledge editing in language models*, 2023.

[38] J. CUI, Z. LI, Y. YAN, B. CHEN, AND L. YUAN, *Chatlaw: Open-source legal large language model with integrated external knowledge bases*, 2023.

[39] A. CULOTTA AND A. MCCALLUM, *Reducing labeling effort for structured prediction tasks*, in AAAI, vol. 5, 2005, pp. 746–751.

[40] D. DAI, L. DONG, Y. HAO, Z. SUI, B. CHANG, AND F. WEI, *Knowledge neurons in pretrained transformers*, in Proceedings of the 60th Annual Meeting of the Associa-

tion for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 8493–8502.

[41] Y. DAI, H. LI, Y. LI, J. SUN, F. HUANG, L. SI, AND X. ZHU, *Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking*, in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Online, Aug. 2021, Association for Computational Linguistics, pp. 879–885.

[42] B. DALVI MISHRA, O. TAFJORD, AND P. CLARK, *Towards teachable reasoning systems: Using a dynamic memory of user feedback for continual system improvement*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 9465–9480.

[43] N. DE CAO, W. AZIZ, AND I. TITOV, *Editing factual knowledge in language models*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 6491–6506.

[44] M. DE LANGE, R. ALJUNDI, M. MASANA, S. PARISOT, X. JIA, A. LEONARDIS, G. SLABAUGH, AND T. TUYTELAARS, *Continual learning: A comparative study on how to defy forgetting in classification tasks*, arXiv preprint arXiv:1909.08383, 2 (2019), p. 2.

[45] M. DENG, J. WANG, C.-P. HSIEH, Y. WANG, H. GUO, T. SHU, M. SONG, E. XING, AND Z. HU, *RLPrompt: Optimizing discrete text prompts with reinforcement learning*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Y. Goldberg, Z. Kozareva, and Y. Zhang, eds., Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 3369–3391.

[46] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *BERT: Pre-training of deep bidirectional transformers for language understanding*, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, June 2019, Association for Computational Linguistics, pp. 4171–4186.

[47] B. DHINGRA, J. R. COLE, J. M. EISENSCHLOS, D. GILLICK, J. EISENSTEIN, AND W. W. COHEN, *Time-aware language models as temporal knowledge bases*, Transactions of the Association for Computational Linguistics, 10 (2022), pp. 257–273.

[48] N. DING, Y. QIN, G. YANG, F. WEI, Z. YANG, Y. SU, S. HU, Y. CHEN, C.-M. CHAN, W. CHEN, J. YI, W. ZHAO, X. WANG, Z. LIU, H.-T. ZHENG, J. CHEN, Y. LIU, J. TANG, J. LI, AND M. SUN, *Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models*, 2022.

[49] Q. DONG, D. DAI, Y. SONG, J. XU, Z. SUI, AND L. LI, *Calibrating factual knowledge in pretrained language models*, in Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 5937–5947.

[50] A. DROZDOV, S. WANG, R. RAHIMI, A. MCCALLUM, H. ZAMANI, AND M. IYYER, *You can't pick your neighbors, or can you? when and how to rely on retrieval in the kNN-LM*, in Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 2997–3007.

[51] A. DUBEY, A. JAUHRI, A. PANDEY, A. KADIAN, A. AL-DAHLE, A. LETMAN, A. MATHUR, A. SCHELTEN, A. YANG, A. FAN, A. GOYAL, A. HARTSHORN, A. YANG, A. MITRA, A. SRAVANKUMAR, A. KORENEV, A. HINSVARK, A. RAO, A. ZHANG, A. RODRIGUEZ, A. GREGERSON, A. SPATARU, B. ROZIERE, B. BIRON, B. TANG, B. CHERN, C. CAUCHETEUX, C. NAYAK, C. BI, C. MARRA, C. MCCONNELL, C. KELLER, C. TOURET, C. WU, C. WONG, C. C. FERRER, C. NIKOLAIDIS, D. ALLONSIUS, D. SONG, D. PINTZ, D. LIVSHITS, D. ESIOBU, D. CHOUDHARY, D. MAHAJAN, D. GARCIA-OLANO, D. PERINO, D. HUPKES, E. LAKOMKIN, E. ALBADAWY, E. LOBANOVA, E. DINAN, E. M. SMITH, F. RADENOVIC, F. ZHANG, G. SYNNAEVE, G. LEE, G. L. ANDERSON, G. NAIL, G. MIALON, G. PANG, G. CUCURELL, H. NGUYEN, H. KOREVAAR, H. XU, H. TOUVRON, I. ZAROV, I. A. IBARRA, I. KLOUMANN, I. MISRA, I. EVTIMOV, J. COPET, J. LEE, J. GEFFERT, J. VRANES, J. PARK, J. MAHADEOKAR, J. SHAH, J. VAN DER LINDE, J. BILLOCK, J. HONG, J. LEE, J. FU, J. CHI, J. HUANG, J. LIU, J. WANG, J. YU, J. BITTON, J. SPISAK, J. PARK, J. ROCCA, J. JOHNSTUN, J. SAXE, J. JIA, K. V. ALWALA, K. UPASANI, K. PLAWIAK, K. LI, K. HEAFIELD, K. STONE, K. EL-ARINI, K. IYER, K. MALIK, K. CHIU, K. BHALLA, L. RANTALA-YEARY, L. VAN DER MAATEN, L. CHEN, L. TAN, L. JENKINS, L. MARTIN, L. MADAAN, L. MALO, L. BLECHER, L. LANDZAAT,

L. DE OLIVEIRA, M. MUZZI, M. PASUPULETI, M. SINGH, M. PALURI, M. KARDAS, M. OLDHAM, M. RITA, M. PAVLOVA, M. KAMBADUR, M. LEWIS, M. SI, M. K. SINGH, M. HASSAN, N. GOYAL, N. TORABI, N. BASHLYKOV, N. BOGOYCHEV, N. CHATTERJI, O. DUCHENNE, O. ÇELEBI, P. ALRASSY, P. ZHANG, P. LI, P. VASIC, P. WENG, P. BHARGAVA, P. DUBAL, P. KRISHNAN, P. S. KOURA, P. XU, Q. HE, Q. DONG, R. SRINIVASAN, R. GANAPATHY, R. CALDERER, R. S. CABRAL, R. STOJNIC, R. RAILEANU, R. GIRDHAR, R. PATEL, R. SAUVESTRE, R. POLIDORO, R. SUMBALY, R. TAYLOR, R. SILVA, R. HOU, R. WANG, S. HOSSEINI, S. CHENNABASAPPA, S. SINGH, S. BELL, S. S. KIM, S. EDUNOV, S. NIE, S. NARANG, S. RAPARTHY, S. SHEN, S. WAN, S. BHOSALE, S. ZHANG, S. VANDENHENDE, S. BATRA, S. WHITMAN, S. SOOTLA, S. COLLOT, S. GURURANGAN, S. BORODINSKY, T. HERMAN, T. FOWLER, T. SHEASHA, T. GEORGIOU, T. SCIALOM, T. SPECKBACHER, T. MIHAYLOV, T. XIAO, U. KARN, V. GOSWAMI, V. GUPTA, V. RAMANATHAN, V. KERKEZ, V. GONGUET, V. DO, V. VOGETI, V. PETROVIC, W. CHU, W. XIONG, W. FU, W. MEERS, X. MARTINET, X. WANG, X. E. TAN, X. XIE, X. JIA, X. WANG, Y. GOLDSCHLAG, Y. GAUR, Y. BABAEI, Y. WEN, Y. SONG, Y. ZHANG, Y. LI, Y. MAO, Z. D. COUDERT, Z. YAN, Z. CHEN, Z. PAPAKIPOS, A. SINGH, A. GRATTAFIORI, A. JAIN, A. KELSEY, A. SHAJNFELD, A. GANGIDI, A. VICTORIA, A. GOLDSTAND, A. MENON, A. SHARMA, A. BOESENBERG, A. VAUGHAN, A. BAEVSKI, A. FEINSTEIN, A. KALLET, A. SANGANI, A. YUNUS, A. LUPU, A. ALVARADO, A. CAPLES, A. GU, A. HO, A. POULTON, A. RYAN, A. RAMCHANDANI, A. FRANCO, A. SARAF, A. CHOWDHURY, A. GABRIEL, A. BHARAMBE, A. EISENMAN, A. YAZDAN, B. JAMES, B. MAURER, B. LEONHARDI, B. HUANG, B. LOYD, B. D. PAOLA, B. PARANJAPE, B. LIU, B. WU, B. NI, B. HANCOCK, B. WASTI, B. SPENCE, B. STOJKOVIC, B. GAMIDO, B. MONTALVO, C. PARKER, C. BURTON, C. MEJIA, C. WANG, C. KIM, C. ZHOU, C. HU, C.-H. CHU, C. CAI, C. TINDAL, C. FEICHTENHOFER, D. CIVIN, D. BEATY, D. KREYMER, D. LI, D. WYATT, D. ADKINS, D. XU, D. TESTUGGINE, D. DAVID, D. PARIKH, D. LISKOVICH, D. FOSS, D. WANG, D. LE, D. HOLLAND, E. DOWLING, E. JAMIL, E. MONTGOMERY, E. PRESANI, E. HAHN, E. WOOD, E. BRINKMAN, E. ARCAUTE, E. DUNBAR, E. SMOTHERS, F. SUN, F. KREUK, F. TIAN, F. OZGENEL, F. CAGGIONI, F. GUZMÁN, F. KANAYET, F. SEIDE, G. M. FLOREZ, G. SCHWARZ, G. BADEER, G. SWEE, G. HALPERN, G. THATTAI, G. HERMAN, G. SIZOV, GUANGYI, ZHANG, G. LAKSHMINARAYANAN, H. SHOJANAZERI, H. ZOU, H. WANG, H. ZHA, H. HABEEB, H. RUDOLPH, H. SUK, H. ASPEGREN, H. GOLDMAN, I. MOLYBOG, I. TUFANOV, I.-E. VELICHE, I. GAT, J. WEISSMAN, J. GEBOSKI, J. KOHLI, J. ASHER,

J.-B. GAYA, J. MARCUS, J. TANG, J. CHAN, J. ZHEN, J. REIZENSTEIN, J. TEBOUL, J. ZHONG, J. JIN, J. YANG, J. CUMMINGS, J. CARVILL, J. SHEPARD, J. MCPHIE, J. TORRES, J. GINSBURG, J. WANG, K. WU, K. H. U, K. SAXENA, K. PRASAD, K. KHANDELWAL, K. ZAND, K. MATOSICH, K. VEERARAGHAVAN, K. MICHELENA, K. LI, K. HUANG, K. CHAWLA, K. LAKHOTIA, K. HUANG, L. CHEN, L. GARG, L. A, L. SILVA, L. BELL, L. ZHANG, L. GUO, L. YU, L. MOSHKOVICH, L. WEHRSTEDT, M. KHABSA, M. AVALANI, M. BHATT, M. TSIMPOUKELLI, M. MANKUS, M. HASSON, M. LENNIE, M. RESO, M. GROSHEV, M. NAUMOV, M. LATHI, M. KENEALLY, M. L. SELTZER, M. VALKO, M. RESTREPO, M. PATEL, M. VYATSKOV, M. SAMVELYAN, M. CLARK, M. MACEY, M. WANG, M. J. HERMOSO, M. METANAT, M. RASTEGARI, M. BANSAL, N. SANTHANAM, N. PARKS, N. WHITE, N. BAWA, N. SINGHAL, N. EGEBO, N. USUNIER, N. P. LAPTEV, N. DONG, N. ZHANG, N. CHENG, O. CHERNOGUZ, O. HART, O. SALPEKAR, O. KALINLI, P. KENT, P. PAREKH, P. SAAB, P. BALAJI, P. RITTNER, P. BONTRAGER, P. ROUX, P. DOLLAR, P. ZVYAGINA, P. RATANCHANDANI, P. YUVRAJ, Q. LIANG, R. ALAO, R. RODRIGUEZ, R. AYUB, R. MURTHY, R. NAYANI, R. MITRA, R. LI, R. HOGAN, R. BATTEY, R. WANG, R. MAHESWARI, R. HOWES, R. RINOTT, S. J. BONDU, S. DATTA, S. CHUGH, S. HUNT, S. DHILLON, S. SIDOROV, S. PAN, S. VERMA, S. YAMAMOTO, S. RAMASWAMY, S. LINDSAY, S. LINDSAY, S. FENG, S. LIN, S. C. ZHA, S. SHANKAR, S. ZHANG, S. ZHANG, S. WANG, S. AGARWAL, S. SAJUYIGBE, S. CHINTALA, S. MAX, S. CHEN, S. KEHOE, S. SATTERFIELD, S. GOVINDAPRASAD, S. GUPTA, S. CHO, S. VIRK, S. SUBRAMANIAN, S. CHOUDHURY, S. GOLDMAN, T. REMEZ, T. GLASER, T. BEST, T. KOHLER, T. ROBINSON, T. LI, T. ZHANG, T. MATTHEWS, T. CHOU, T. SHAKED, V. VONTIMITTA, V. AJAYI, V. MONTANEZ, V. MOHAN, V. S. KUMAR, V. MANGLA, V. IONESCU, V. POENARU, V. T. MIHAILESCU, V. IVANOV, W. LI, W. WANG, W. JIANG, W. BOUAZIZ, W. CONSTABLE, X. TANG, X. WANG, X. WU, X. WANG, X. XIA, X. WU, X. GAO, Y. CHEN, Y. HU, Y. JIA, Y. QI, Y. LI, Y. ZHANG, Y. ZHANG, Y. ADI, Y. NAM, YU, WANG, Y. HAO, Y. QIAN, Y. HE, Z. RAIT, Z. DEVITO, Z. ROSNBRICK, Z. WEN, Z. YANG, AND Z. ZHAO, *The llama 3 herd of models*, 2024.

[52] L. EIN-DOR, A. HALFON, A. GERA, E. SHNARCH, L. DANKIN, L. CHOSHEN, M. DANILEVSKY, R. AHARONOV, Y. KATZ, AND N. SLONIM, *Active Learning for BERT: An Empirical Study*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Nov. 2020, Association for Computational Linguistics, pp. 7949–7962.

[53] Y. ELAZAR, N. KASSNER, S. RAVFOGEL, A. FEDER, A. RAVICHANDER, M. MOSBACH,

Y. BELINKOV, H. SCHÜTZE, AND Y. GOLDBERG, *Measuring causal effects of data statistics on language model's 'factual' predictions*, 2023.

[54] M. ERIC, R. GOEL, S. PAUL, A. SETHI, S. AGARWAL, S. GAO, A. KUMAR, A. GOYAL, P. KU, AND D. HAKKANI-TUR, *MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines*, in Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, May 2020, European Language Resources Association, pp. 422–428.

[55] S. ES, J. JAMES, L. ESPINOSA-ANKE, AND S. SCHOCKAERT, *Ragas: Automated evaluation of retrieval augmented generation*, 2023.

[56] M. FANG, Y. LI, AND T. COHN, *Learning how to active learn: A deep reinforcement learning approach*, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, Sept. 2017, Association for Computational Linguistics, pp. 595–605.

[57] S. FENG, W. SHI, Y. BAI, V. BALACHANDRAN, T. HE, AND Y. TSVETKOV, *Knowledge card: Filling llms' knowledge gaps with plug-in specialized language models*, 2023.

[58] Z. FENG, W. MA, W. YU, L. HUANG, H. WANG, Q. CHEN, W. PENG, X. FENG, B. QIN, AND T. LIU, *Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications*, 2023.

[59] C. FINN, P. ABBEEL, AND S. LEVINE, *Model-agnostic meta-learning for fast adaptation of deep networks*, in Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, JMLR.org, 2017, p. 1126–1135.

[60] L. GAO, Z. DAI, P. PASUPAT, A. CHEN, A. T. CHAGANTY, Y. FAN, V. Y. ZHAO, N. LAO, H. LEE, D.-C. JUAN, AND K. GUU, *Rarr: Researching and revising what language models say, using language models*, 2023.

[61] T. GAO, X. YAO, AND D. CHEN, *SimCSE: Simple contrastive learning of sentence embeddings*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, eds., Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 6894–6910.

[62] Y. GAO, Y. XIONG, X. GAO, K. JIA, J. PAN, Y. BI, Y. DAI, J. SUN, Q. GUO, M. WANG, AND H. WANG, *Retrieval-augmented generation for large language models: A survey*, 2024.

[63] M. GEVA, J. BASTINGS, K. FILIPPOVA, AND A. GLOBERSON, *Dissecting recall of factual associations in auto-regressive language models*, 2023.

[64] M. GEVA, R. SCHUSTER, J. BERANT, AND O. LEVY, *Transformer feed-forward layers are key-value memories*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 5484–5495.

[65] GOOGLE, *Med-palm 2*, 2023.

[66] Z. GOU, Z. SHAO, Y. GONG, Y. SHEN, Y. YANG, N. DUAN, AND W. CHEN, *Critic: Large language models can self-correct with tool-interactive critiquing*, 2023.

[67] T. GOYAL, J. J. LI, AND G. DURRETT, *News summarization and evaluation in the era of gpt-3*, 2022.

[68] S. GUNASEKAR, Y. ZHANG, J. ANEJA, C. C. T. MENDES, A. D. GIORNO, S. GOPI, M. JAVAHERIPI, P. KAUFFMANN, G. DE ROSA, O. SAARIKIVI, A. SALIM, S. SHAH, H. S. BEHL, X. WANG, S. BUBECK, R. ELDAN, A. T. KALAI, Y. T. LEE, AND Y. LI, *Textbooks are all you need*, 2023.

[69] A. GUPTA, D. MONDAL, A. K. SHESHADRI, W. ZHAO, X. L. LI, S. WIEGREFFE, AND N. TANDON, *Editing commonsense knowledge in gpt*, 2023.

[70] K. GUPTA, B. THÉRIEN, A. IBRAHIM, M. L. RICHTER, Q. ANTHONY, E. BELILOVSKY, I. RISH, AND T. LESORT, *Continual pre-training of large language models: How to (re)warm your model?*, 2023.

[71] S. GURURANGAN, M. LEWIS, A. HOLTZMAN, N. A. SMITH, AND L. ZETTLEMOYER, *DEMix layers: Disentangling domains for modular language modeling*, in Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, United States, July 2022, Association for Computational Linguistics, pp. 5557–5576.

[72] S. GURURANGAN, A. MARASOVIĆ, S. SWAYAMDIPTA, K. LO, I. BELTAGY, D. DOWNEY, AND N. A. SMITH, *Don't stop pretraining: Adapt language models to domains and tasks*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, July 2020, Association for Computational Linguistics, pp. 8342–8360.

[73] K. GUU, K. LEE, Z. TUNG, P. PASUPAT, AND M. CHANG, *Retrieval augmented language model pre-training*, in International conference on machine learning, PMLR, 2020, pp. 3929–3938.

[74] D. HA, A. M. DAI, AND Q. V. LE, *Hypernetworks*, in International Conference on Learning Representations, 2017.

[75] Y. HAO, L. DONG, F. WEI, AND K. XU, *Investigating learning dynamics of BERT fine-tuning*, in Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, K.-F. Wong, K. Knight, and H. Wu, eds., Suzhou, China, Dec. 2020, Association for Computational Linguistics, pp. 87–92.

[76] T. HARTVIGSEN, S. SANKARANARAYANAN, H. PALANGI, Y. KIM, AND M. GHASSEMI, *Aging with grace: Lifelong model editing with discrete key-value adaptors*, 2023.

[77] P. HASE, M. BANSAL, B. KIM, AND A. GHANDEHARIOUN, *Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models*, arXiv preprint arXiv:2301.04213, (2023).

[78] P. HASE, M. DIAB, A. CELIKYILMAZ, X. LI, Z. KOZAREVA, V. STOYANOV, M. BANSAL, AND S. IYER, *Methods for measuring, updating, and visualizing factual beliefs in language models*, in Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, May 2023, Association for Computational Linguistics, pp. 2714–2731.

[79] H. HE, H. ZHANG, AND D. ROTH, *Rethinking with retrieval: Faithful large language model inference*, 2022.

[80] J. HE, G. NEUBIG, AND T. BERG-KIRKPATRICK, *Efficient nearest neighbor language models*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 5703–5714.

[81] T. HE, J. LIU, K. CHO, M. OTT, B. LIU, J. GLASS, AND F. PENG, *Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models*, in Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, Apr. 2021, Association for Computational Linguistics, pp. 1121–1133.

[82] M. HECK, C. VAN NIEKERK, N. LUBIS, C. GEISHAUSER, H.-C. LIN, M. MORESI, AND M. GASIC, *TripPy: A triple copy strategy for value independent neural dialog state*

tracking, in Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 1st virtual meeting, July 2020, Association for Computational Linguistics, pp. 35–44.

[83] E. HERNANDEZ, B. Z. LI, AND J. ANDREAS, *Inspecting and editing knowledge representations in language models*, 2023.

[84] J. HOELSCHER-OBERMAIER, J. PERSSON, E. KRAN, I. KONSTAS, AND F. BAREZ, *Detecting edit failures in large language models: An improved specificity benchmark*, 2023.

[85] N. HOULSBY, A. GIURGIU, S. JASTRZEBSKI, B. MORRONE, Q. DE LAROUSSILHE, A. GESMUNDO, M. ATTARIYAN, AND S. GELLY, *Parameter-efficient transfer learning for nlp*, in International Conference on Machine Learning, PMLR, 2019, pp. 2790–2799.

[86] ——, *Parameter-efficient transfer learning for NLP*, in Proceedings of the 36th International Conference on Machine Learning, K. Chaudhuri and R. Salakhutdinov, eds., vol. 97 of Proceedings of Machine Learning Research, PMLR, 09–15 Jun 2019, pp. 2790–2799.

[87] E. J. HU, YELONG SHEN, P. WALLIS, Z. ALLEN-ZHU, Y. LI, S. WANG, L. WANG, AND W. CHEN, *LoRA: Low-rank adaptation of large language models*, in International Conference on Learning Representations, 2022.

[88] J. HU AND G. NEUBIG, *Phrase-level active learning for neural machine translation*, in Proceedings of the Sixth Conference on Machine Translation, Online, Nov. 2021, Association for Computational Linguistics, pp. 1087–1099.

[89] N. HU, E. MITCHELL, C. D. MANNING, AND C. FINN, *Meta-learning online adaptation of language models*, 2023.

[90] Y. HU, C.-H. LEE, T. XIE, T. YU, N. A. SMITH, AND M. OSTENDORF, *In-context learning for few-shot dialogue state tracking*, in Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 2627–2643.

[91] L. HUANG, W. YU, W. MA, W. ZHONG, Z. FENG, H. WANG, Q. CHEN, W. PENG, X. FENG, B. QIN, AND T. LIU, *A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions*, 2023.

[92] Y. HUANG, Y. ZHANG, J. CHEN, X. WANG, AND D. YANG, *Continual learning for text classification with information disentanglement based regularization*, in Proceed-

ings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, June 2021, Association for Computational Linguistics, pp. 2736–2746.

[93] Z. HUANG, Y. SHEN, X. ZHANG, J. ZHOU, W. RONG, AND Z. XIONG, *Transformer-patcher: One mistake worth one neuron*, in The Eleventh International Conference on Learning Representations, 2023.

[94] T. INABA, H. KIYOMARU, F. CHENG, AND S. KUROHASHI, *Multitool-cot: Gpt-3 can use multiple external tools with chain of thought prompting*, 2023.

[95] G. IZACARD, M. CARON, L. HOSSEINI, S. RIEDEL, P. BOJANOWSKI, A. JOULIN, AND E. GRAVE, *Unsupervised dense information retrieval with contrastive learning*, Transactions on Machine Learning Research, (2022).

[96] G. IZACARD, P. LEWIS, M. LOMELI, L. HOSSEINI, F. PETRONI, T. SCHICK, J. DWIVEDI-YU, A. JOULIN, S. RIEDEL, AND E. GRAVE, *Atlas: Few-shot learning with retrieval augmented language models*, 2022.

[97] J. JANG, S. YE, C. LEE, S. YANG, J. SHIN, J. HAN, G. KIM, AND M. SEO, *TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 6237–6250.

[98] J. JANG, S. YE, S. YANG, J. SHIN, J. HAN, G. KIM, S. J. CHOI, AND M. SEO, *Towards continual knowledge learning of language models*, in International Conference on Learning Representations, 2022.

[99] S. JI, S. PAN, E. CAMBRIA, P. MARTTINEN, AND P. S. YU, *A survey on knowledge graphs: Representation, acquisition, and applications*, IEEE Transactions on Neural Networks and Learning Systems, 33 (2022), pp. 494–514.

[100] Z. JI, N. LEE, R. FRIESKE, T. YU, D. SU, Y. XU, E. ISHII, Y. J. BANG, A. MADOTTO, AND P. FUNG, *Survey of hallucination in natural language generation*, ACM Comput. Surv., 55 (2023).

[101] Z. JIANG, L. GAO, Z. WANG, J. ARAKI, H. DING, J. CALLAN, AND G. NEUBIG, *Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 2336–2349.

[102] Z. JIANG, F. XU, L. GAO, Z. SUN, Q. LIU, J. DWIVEDI-YU, Y. YANG, J. CALLAN, AND G. NEUBIG, *Active retrieval augmented generation*, in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, H. Bouamor, J. Pino, and K. Bali, eds., Singapore, Dec. 2023, Association for Computational Linguistics, pp. 7969–7992.

[103] Z. JIANG, F. F. XU, J. ARAKI, AND G. NEUBIG, *How can we know what language models know?*, Transactions of the Association for Computational Linguistics, 8 (2020), pp. 423–438.

[104] Z. JIANG, F. F. XU, L. GAO, Z. SUN, Q. LIU, J. DWIVEDI-YU, Y. YANG, J. CALLAN, AND G. NEUBIG, *Active retrieval augmented generation*, 2023.

[105] X. JIN, D. ZHANG, H. ZHU, W. XIAO, S.-W. LI, X. WEI, A. ARNOLD, AND X. REN, *Lifelong pretraining: Continually adapting language models to emerging corpora*, in Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, United States, July 2022, Association for Computational Linguistics, pp. 4764–4780.

[106] M. JOSHI, E. CHOI, D. WELD, AND L. ZETTLEMOYER, *TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension*, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), R. Barzilay and M.-Y. Kan, eds., Vancouver, Canada, July 2017, Association for Computational Linguistics, pp. 1601–1611.

[107] Y. JU AND Z. ZHANG, *Klob: a benchmark for assessing knowledge locating methods in language models*, 2023.

[108] S. KADAVATH, T. CONERLY, A. ASKELL, T. HENIGHAN, D. DRAIN, E. PEREZ, N. SCHIEFER, Z. HATFIELD-DODDS, N. DASSARMA, E. TRAN-JOHNSON, S. JOHNSTON, S. EL-SHOWK, A. JONES, N. ELHAGE, T. HUME, A. CHEN, Y. BAI, S. BOWMAN, S. FORT, D. GANGULI, D. HERNANDEZ, J. JACOBSON, J. KERNION, S. KRAVEC, L. LOVITT, K. NDOUSSE, C. OLSSON, S. RINGER, D. AMODEI, T. BROWN, J. CLARK, N. JOSEPH, B. MANN, S. MCCANDLISH, C. OLAH, AND J. KAPLAN, *Language models (mostly) know what they know*, 2022.

[109] E. KAMALLOO, N. DZIRI, C. L. A. CLARKE, AND D. RAFIEI, *Evaluating open-domain question answering in the era of large language models*, 2023.

[110] N. KANDPAL, H. DENG, A. ROBERTS, E. WALLACE, AND C. RAFFEL, *Large language models struggle to learn long-tail knowledge*, in Proceedings of the 40th International Conference on Machine Learning, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds., vol. 202 of Proceedings of Machine Learning Research, PMLR, 23–29 Jul 2023, pp. 15696–15707.

[111] J. KANG, R. LAROCHE, X. YUAN, A. TRISCHLER, X. LIU, AND J. FU, *Think before you act: Decision transformers with internal working memory*, 2023.

[112] V. KARPUKHIN, B. OGUZ, S. MIN, P. LEWIS, L. WU, S. EDUNOV, D. CHEN, AND W.-T. YIH, *Dense passage retrieval for open-domain question answering*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, eds., Online, Nov. 2020, Association for Computational Linguistics, pp. 6769–6781.

[113] J. KASAI, Y. KASAI, K. SAKAGUCHI, Y. YAMADA, AND D. RADEV, *Evaluating gpt-4 and chatgpt on japanese medical licensing examinations*, 2023.

[114] J. KASAI, K. SAKAGUCHI, Y. TAKAHASHI, R. L. BRAS, A. ASAI, X. YU, D. RADEV, N. A. SMITH, Y. CHOI, AND K. INUI, *Realtime qa: What's the answer right now?*, 2022.

[115] J. KASAI, K. SAKAGUCHI, YOICHI TAKAHASHI, R. L. BRAS, A. ASAI, X. V. YU, D. RADEV, N. A. SMITH, Y. CHOI, AND K. INUI, *Realtime QA: What's the answer right now?*, in Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023.

[116] N. KASSNER, O. TAFJORD, H. SCHÜTZE, AND P. CLARK, *BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 8849–8861.

[117] J. KAUR, S. BHATIA, M. AGGARWAL, R. BANSAL, AND B. KRISHNAMURTHY, *LM-CORE: Language models with contextually relevant external knowledge*, in Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, United States, July 2022, Association for Computational Linguistics, pp. 750–769.

[118] A. KAZEMNEJAD, M. REZAGHOLIZADEH, P. PARTHASARATHI, AND S. CHANDAR, *Measuring the knowledge acquisition-utilization gap in pretrained language models*, 2023.

[119] Z. KE, H. LIN, Y. SHAO, H. XU, L. SHU, AND B. LIU, *Continual training of language models for few-shot learning*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 10205–10216.

[120] Z. KE AND B. LIU, *Continual learning of natural language processing tasks: A survey*, 2023.

[121] Z. KE, B. LIU, N. MA, H. XU, AND L. SHU, *Achieving forgetting prevention and knowledge transfer in continual learning*, in Advances in Neural Information Processing Systems, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds., 2021.

[122] U. KHANDELWAL, O. LEVY, D. JURAFSKY, L. ZETTLEMOYER, AND M. LEWIS, *Generalization through memorization: Nearest neighbor language models*, in International Conference on Learning Representations, 2020.

[123] O. KHATTAB, K. SANTHANAM, X. L. LI, D. HALL, P. LIANG, C. POTTS, AND M. ZAHARIA, *Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp*, 2023.

[124] T. KHOT, H. TRIVEDI, M. FINLAYSON, Y. FU, K. RICHARDSON, P. CLARK, AND A. SABHARWAL, *Decomposed prompting: A modular approach for solving complex tasks*, in The Eleventh International Conference on Learning Representations, 2023.

[125] J. KIM, A. ASAI, G. ILHARCO, AND H. HAJISHIRZI, *Taskweb: Selecting better source tasks for multi-task nlp*, 2023.

[126] S. KIM, S. YANG, G. KIM, AND S.-W. LEE, *Efficient dialogue state tracking by selectively overwriting memory*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, July 2020, Association for Computational Linguistics, pp. 567–582.

[127] T. KIM, H. YOON, Y. LEE, P. KANG, AND M. KIM, *Mismatch between multi-turn dialogue and its evaluation metric in dialogue state tracking*, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 297–309.

[128] J. KIRKPATRICK, R. PASCANU, N. RABINOWITZ, J. VENESS, G. DESJARDINS, A. A. RUSU, K. MILAN, J. QUAN, T. RAMALHO, A. GRABSKA-BARWINSKA, D. HASSABIS, C. CLOPATH, D. KUMARAN, AND R. HADSELL, *Overcoming catastrophic forgetting*

in neural networks, Proceedings of the National Academy of Sciences, 114 (2017), pp. 3521–3526.

[129] M. KOMEILI, K. SHUSTER, AND J. WESTON, *Internet-augmented dialogue generation*, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 8460–8478.

[130] T. KWIATKOWSKI, J. PALOMAKI, O. REDFIELD, M. COLLINS, A. PARIKH, C. ALBERTI, D. EPSTEIN, I. POLOSUKHIN, J. DEVLIN, K. LEE, K. TOUTANOVA, L. JONES, M. KELCEY, M.-W. CHANG, A. M. DAI, J. USZKOREIT, Q. LE, AND S. PETROV, *Natural questions: A benchmark for question answering research*, Transactions of the Association for Computational Linguistics, 7 (2019), pp. 452–466.

[131] W. KWON, Z. LI, S. ZHUANG, Y. SHENG, L. ZHENG, C. H. YU, J. GONZALEZ, H. ZHANG, AND I. STOICA, *Efficient memory management for large language model serving with pagedattention*, in Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23, New York, NY, USA, 2023, Association for Computing Machinery, p. 611–626.

[132] A. LAZARIDOU, E. GRIBOVSKAYA, W. STOKOWIEC, AND N. GRIGOREV, *Internet-augmented language models through few-shot prompting for open-domain question answering*, 2022.

[133] A. LAZARIDOU, A. KUNCORO, E. GRIBOVSKAYA, D. AGRAWAL, A. LISKA, T. TERZI, M. GIMENEZ, C. DE MASSON D'AUTUME, T. KOČISKÝ, S. RUDER, D. YOGATAMA, K. CAO, S. YOUNG, AND P. BLUNSOM, *Mind the gap: Assessing temporal generalization in neural language models*, in Advances in Neural Information Processing Systems, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds., 2021.

[134] C.-H. LEE, H. CHENG, AND M. OSTENDORF, *Dialogue state tracking with a language model using schema-driven prompting*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 4937–4949.

[135] K. LEE, W. HAN, S.-W. HWANG, H. LEE, J. PARK, AND S.-W. LEE, *Plug-and-play adaptation for continuously-updated QA*, in Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 438–447.

[136] N. LEE, W. PING, P. XU, M. PATWARY, P. N. FUNG, M. SHOEYBI, AND B. CATANZARO, *Factuality enhanced language models for open-ended text generation*, in Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., vol. 35, Curran Associates, Inc., 2022, pp. 34586–34599.

[137] D. D. LEWIS AND W. A. GALE, *A sequential algorithm for training text classifiers*, in SIGIR'94, Springer, 1994, pp. 3–12.

[138] P. LEWIS, E. PEREZ, A. PIKTUS, F. PETRONI, V. KARPUKHIN, N. GOYAL, H. KÜTTLER, M. LEWIS, W.-T. YIH, T. ROCKTÄSCHEL, ET AL., *Retrieval-augmented generation for knowledge-intensive nlp tasks*, Advances in Neural Information Processing Systems, 33 (2020), pp. 9459–9474.

[139] D. LI, A. S. RAWAT, M. ZAHEER, X. WANG, M. LUKASIK, A. VEIT, F. YU, AND S. KUMAR, *Large language models with controllable working memory*, 2022.

[140] J. LI, T. TANG, W. X. ZHAO, J. WANG, J.-Y. NIE, AND J.-R. WEN, *The web can be your oyster for improving large language models*, 2023.

[141] S. LI, X. LI, L. SHANG, Z. DONG, C. SUN, B. LIU, Z. JI, X. JIANG, AND Q. LIU, *How pre-trained language models capture factual knowledge? a causal-inspired analysis*, 2022.

[142] X. LI, S. LI, S. SONG, J. YANG, J. MA, AND J. YU, *Pmet: Precise model editing in a transformer*, 2023.

[143] Y. LI, S. BUBECK, R. ELDAN, A. D. GIORNO, S. GUNASEKAR, AND Y. T. LEE, *Textbooks are all you need ii: phi-1.5 technical report*, 2023.

[144] S. LIANG, L. PODDAR, AND G. SZARVAS, *Attention guided dialogue state tracking with sparse supervision*, arXiv preprint arXiv:2101.11958, (2021).

[145] Y. LIANG, C. WU, T. SONG, W. WU, Y. XIA, Y. LIU, Y. OU, S. LU, L. JI, S. MAO, Y. WANG, L. SHOU, M. GONG, AND N. DUAN, *Taskmatrix.ai: Completing tasks by connecting foundation models with millions of apis*, 2023.

[146] B. Y. LIN, S. WANG, X. LIN, R. JIA, L. XIAO, X. REN, AND S. YIH, *On continual model refinement in out-of-distribution data streams*, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 3128–3139.

[147] C.-Y. LIN, *ROUGE: A package for automatic evaluation of summaries*, in Text Summarization Branches Out, Barcelona, Spain, July 2004, Association for Computational Linguistics, pp. 74–81.

[148] W. LIN, B.-H. TSENG, AND B. BYRNE, *Knowledge-aware graph-enhanced GPT-2 for dialogue state tracking*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 7871–7881.

[149] Z. LIN, B. LIU, A. MADOTTO, S. MOON, Z. ZHOU, P. CROOK, Z. WANG, Z. YU, E. CHO, R. SUBBA, AND P. FUNG, *Zero-shot dialogue state tracking via cross-task transfer*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 7890–7900.

[150] A. LISKA, T. KOCISKY, E. GRIBOVSKAYA, T. TERZI, E. SEZENER, D. AGRAWAL, C. DE MASSON D'AUTUME, T. SCHOLTES, M. ZAHEER, S. YOUNG, E. GILSENAN-MCMAHON, S. AUSTIN, P. BLUNSOM, AND A. LAZARIDOU, *StreamingQA: A benchmark for adaptation to new knowledge over time in question answering models*, in Proceedings of the 39th International Conference on Machine Learning, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds., vol. 162 of Proceedings of Machine Learning Research, PMLR, 17–23 Jul 2022, pp. 13604–13622.

[151] J. LIU, A. LIU, X. LU, S. WELLECK, P. WEST, R. L. BRAS, Y. CHOI, AND H. HAJISHIRZI, *Generated knowledge prompting for commonsense reasoning*, 2022.

[152] N. F. LIU, T. ZHANG, AND P. LIANG, *Evaluating verifiability in generative search engines*, 2023.

[153] P. LIU, W. YUAN, J. FU, Z. JIANG, H. HAYASHI, AND G. NEUBIG, *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*, CoRR, abs/2107.13586 (2021).

[154] P. LIU, W. YUAN, J. FU, Z. JIANG, H. HAYASHI, AND G. NEUBIG, *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*, ACM Comput. Surv., 55 (2023).

[155] T. LIU AND B. K. H. LOW, *Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks*, 2023.

[156] Y. LIU, M. OTT, N. GOYAL, J. DU, M. JOSHI, D. CHEN, O. LEVY, M. LEWIS, L. ZETTLEMOYER, AND V. STOYANOV, *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692, (2019).

[157] A. LIŠKA, T. KOČISKÝ, E. GRIBOVSKAYA, T. TERZI, E. SEZENER, D. AGRAWAL, C. DE MASSON D'AUTUME, T. SCHOLTES, M. ZAHEER, S. YOUNG, E. GILSENAN-MCMAHON, S. AUSTIN, P. BLUNSOM, AND A. LAZARIDOU, *Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models*, 2022.

[158] S. LONGPRE, L. HOU, T. VU, A. WEBSON, H. W. CHUNG, Y. TAY, D. ZHOU, Q. V. LE, B. ZOPH, J. WEI, AND A. ROBERTS, *The flan collection: Designing data and methods for effective instruction tuning*, 2023.

[159] D. LOPEZ-PAZ AND M. A. RANZATO, *Gradient episodic memory for continual learning*, in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., vol. 30, Curran Associates, Inc., 2017.

[160] P. LU, B. PENG, H. CHENG, M. GALLEY, K.-W. CHANG, Y. N. WU, S.-C. ZHU, AND J. GAO, *Chameleon: Plug-and-play compositional reasoning with large language models*, 2023.

[161] H. LUO, Y.-S. CHUANG, Y. GONG, T. ZHANG, Y. KIM, X. WU, D. FOX, H. MENG, AND J. GLASS, *Sail: Search-augmented instruction learning*, 2023.

[162] Y. LUO, Z. YANG, F. MENG, Y. LI, J. ZHOU, AND Y. ZHANG, *An empirical study of catastrophic forgetting in large language models during continual fine-tuning*, 2024.

[163] Z. LUO, C. XU, P. ZHAO, X. GENG, C. TAO, J. MA, Q. LIN, AND D. JIANG, *Augmented large language models with parametric knowledge guiding*, 2023.

[164] K. LUU, D. KHASHABI, S. GURURANGAN, K. MANDYAM, AND N. A. SMITH, *Time waits for no one! analysis and challenges of temporal misalignment*, in Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, United States, July 2022, Association for Computational Linguistics, pp. 5944–5958.

[165] X. MA, Y. GONG, P. HE, H. ZHAO, AND N. DUAN, *Query rewriting for retrieval-augmented large language models*, 2023.

[166] A. MADAAN, N. TANDON, P. CLARK, AND Y. YANG, *Memory-assisted prompt editing to improve GPT-3 after deployment*, in Proceedings of the 2022 Conference on Em-

pirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 2833–2861.

[167] 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*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 7452–7467.

[168] A. MALLEN, A. ASAI, V. ZHONG, R. DAS, D. KHASHABI, AND H. HAJISHIRZI, *When not to trust language models: Investigating effectiveness of parametric and non-parametric memories*, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), A. Rogers, J. Boyd-Graber, and N. Okazaki, eds., Toronto, Canada, July 2023, Association for Computational Linguistics, pp. 9802–9822.

[169] A. MALLEN, A. ASAI, V. ZHONG, R. DAS, D. KHASHABI, AND H. HAJISHIRZI, *When not to trust language models: Investigating effectiveness of parametric and non-parametric memories*, 2023.

[170] K. MARGATINA, G. VERNIKOS, L. BARRAULT, AND N. ALETRAS, *Active learning by acquiring contrastive examples*, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 650–663.

[171] M. MCCLOSKEY AND N. J. COHEN, *Catastrophic interference in connectionist networks: The sequential learning problem*, in Psychology of learning and motivation, vol. 24, Elsevier, 1989, pp. 109–165.

[172] K. MENG, D. BAU, A. ANDONIAN, AND Y. BELINKOV, *Locating and editing factual associations in gpt*, in Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., vol. 35, Curran Associates, Inc., 2022, pp. 17359–17372.

[173] K. MENG, A. S. SHARMA, A. J. ANDONIAN, Y. BELINKOV, AND D. BAU, *Mass-editing memory in a transformer*, in The Eleventh International Conference on Learning Representations, 2023.

[174] Y. MENG, S. ZONG, X. LI, X. SUN, T. ZHANG, F. WU, AND J. LI, *GNN-LM: Language modeling based on global contexts via GNN*, in International Conference on Learning Representations, 2022.

[175] J. MENICK, M. TREBACZ, V. MIKULIK, J. ASLANIDES, F. SONG, M. CHADWICK, M. GLAESE, S. YOUNG, L. CAMPBELL-GILLINGHAM, G. IRVING, AND N. MCALEESE, *Teaching language models to support answers with verified quotes*, 2022.

[176] G. MIALON, R. DESSÌ, M. LOMELI, C. NALMPANTIS, R. PASUNURU, R. RAILEANU, B. ROZIÈRE, T. SCHICK, J. DWIVEDI-YU, A. CELIKYILMAZ, E. GRAVE, Y. LECUN, AND T. SCIALOM, *Augmented language models: a survey*, 2023.

[177] E. MITCHELL, C. LIN, A. BOSSELUT, C. FINN, AND C. D. MANNING, *Fast model editing at scale*, in International Conference on Learning Representations, 2022.

[178] E. MITCHELL, C. LIN, A. BOSSELUT, C. D. MANNING, AND C. FINN, *Memory-based model editing at scale*, in Proceedings of the 39th International Conference on Machine Learning, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds., vol. 162 of Proceedings of Machine Learning Research, PMLR, 17–23 Jul 2022, pp. 15817–15831.

[179] J. MOK, J. DO, S. LEE, T. TAGHAVI, S. YU, AND S. YOON, *Large-scale lifelong learning of in-context instructions and how to tackle it*, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, July 2023, Association for Computational Linguistics, pp. 12573–12589.

[180] R. NAKANO, J. HILTON, S. BALAJI, J. WU, L. OUYANG, C. KIM, C. HESSE, S. JAIN, V. KOSARAJU, W. SAUNDERS, X. JIANG, K. COBBE, T. ELOUNDOU, G. KRUEGER, K. BUTTON, M. KNIGHT, B. CHESS, AND J. SCHULMAN, *Webgpt: Browser-assisted question-answering with human feedback*, 2022.

[181] E. NEEMAN, R. AHARONI, O. HONOVICH, L. CHOSHEN, I. SZPEKTOR, AND O. ABEND, *Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering*, 2022.

[182] H. NORI, N. KING, S. M. MCKINNEY, D. CARIGNAN, AND E. HORVITZ, *Capabilities of gpt-4 on medical challenge problems*, 2023.

[183] Y. ONOE, M. J. Q. ZHANG, S. PADMANABHAN, G. DURRETT, AND E. CHOI, *Can lms learn new entities from descriptions? challenges in propagating injected knowledge*, 2023.

[184] OPENAI, *Introducing chatgpt*, 2022.

[185] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens,

J. SITKIN, K. SLAMA, I. SOHL, B. SOKOLOWSKY, Y. SONG, N. STAUDACHER, F. P. SUCH, N. SUMMERS, I. SUTSKEVER, J. TANG, N. TEZAK, M. B. THOMPSON, P. TILLET, A. TOOTOONCHIAN, E. TSENG, P. TUGGLE, N. TURLEY, J. TWOREK, J. F. C. URIBE, A. VALLONE, A. VIJAYVERGIYA, C. VOSS, C. WAINWRIGHT, J. J. WANG, A. WANG, B. WANG, J. WARD, J. WEI, C. WEINMANN, A. WELIHINDA, P. WELINDER, J. WENG, L. WENG, M. WIETHOFF, D. WILLNER, C. WINTER, S. WOLRICH, H. WONG, L. WORKMAN, S. WU, J. WU, M. WU, K. XIAO, T. XU, S. YOO, K. YU, Q. YUAN, W. ZAREMBA, R. ZELLERS, C. ZHANG, M. ZHANG, S. ZHAO, T. ZHENG, J. ZHUANG, W. ZHUK, AND B. ZOPH, *Gpt-4 technical report*, 2024.

[186] L. OUYANG, J. WU, X. JIANG, D. ALMEIDA, C. WAINWRIGHT, P. MISHKIN, C. ZHANG, S. AGARWAL, K. SLAMA, A. RAY, J. SCHULMAN, J. HILTON, F. KELTON, L. MILLER, M. SIMENS, A. ASKELL, P. WELINDER, P. F. CHRISTIANO, J. LEIKE, AND R. LOWE, *Training language models to follow instructions with human feedback*, in Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., vol. 35, Curran Associates, Inc., 2022, pp. 27730–27744.

[187] S. PADMANABHAN, Y. ONOE, M. J. Q. ZHANG, G. DURRETT, AND E. CHOI, *Propagating knowledge updates to lms through distillation*, 2023.

[188] S. PAN, L. LUO, Y. WANG, C. CHEN, J. WANG, AND X. WU, *Unifying large language models and knowledge graphs: A roadmap*, 2023.

[189] B. PARANJAPE, S. LUNDBERG, S. SINGH, H. HAJISHIRZI, L. ZETTLEMOYER, AND M. T. RIBEIRO, *Art: Automatic multi-step reasoning and tool-use for large language models*, 2023.

[190] D. PATTERSON, J. GONZALEZ, Q. LE, C. LIANG, L.-M. MUNGUIA, D. ROTHCHILD, D. SO, M. TEXIER, AND J. DEAN, *Carbon emissions and large neural network training*, 2021.

[191] B. PENG, M. GALLEY, P. HE, H. CHENG, Y. XIE, Y. HU, Q. HUANG, L. LIDEN, Z. YU, W. CHEN, AND J. GAO, *Check your facts and try again: Improving large language models with external knowledge and automated feedback*, 2023.

[192] B. PENG, C. LI, J. LI, S. SHAYANDEH, L. LIDEN, AND J. GAO, *Soloist: Building task bots at scale with transfer learning and machine teaching*, Transactions of the Association for Computational Linguistics, 9 (2021), pp. 807–824.

[193] G. PENG, T. GE, S.-Q. CHEN, F. WEI, AND H. WANG, *Semiparametric language models are scalable continual learners*, 2023.

[194] E. PEREZ, D. KIELA, AND K. CHO, *True few-shot learning with language models*, Advances in Neural Information Processing Systems, 34 (2021).

[195] F. PETRONI, A. PIKTUS, A. FAN, P. LEWIS, M. YAZDANI, N. DE CAO, J. THORNE, Y. JERNITE, V. KARPUKHIN, J. MAILLARD, V. PLACHOURAS, T. ROCKTÄSCHEL, AND S. RIEDEL, *KILT: a benchmark for knowledge intensive language tasks*, in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, June 2021, Association for Computational Linguistics, pp. 2523–2544.

[196] F. PETRONI, T. ROCKTÄSCHEL, S. RIEDEL, P. LEWIS, A. BAKHTIN, Y. WU, AND A. MILLER, *Language models as knowledge bases?*, in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), K. Inui, J. Jiang, V. Ng, and X. Wan, eds., Hong Kong, China, Nov. 2019, Association for Computational Linguistics, pp. 2463–2473.

[197] O. PRESS, M. ZHANG, S. MIN, L. SCHMIDT, N. A. SMITH, AND M. LEWIS, *Measuring and narrowing the compositionality gap in language models*, 2023.

[198] S. QIAO, Y. OU, N. ZHANG, X. CHEN, Y. YAO, S. DENG, C. TAN, F. HUANG, AND H. CHEN, *Reasoning with language model prompting: A survey*, 2023.

[199] C. QIN AND S. JOTY, *Continual few-shot relation learning via embedding space regularization and data augmentation*, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 2776–2789.

[200] G. QIN AND J. EISNER, *Learning how to ask: Querying lms with mixtures of soft prompts*, 2021.

[201] Y. QIN, Z. CAI, D. JIN, L. YAN, S. LIANG, K. ZHU, Y. LIN, X. HAN, N. DING, H. WANG, R. XIE, F. QI, Z. LIU, M. SUN, AND J. ZHOU, *Webcpm: Interactive web search for chinese long-form question answering*, 2023.

[202] Y. QIN, S. HU, Y. LIN, W. CHEN, N. DING, G. CUI, Z. ZENG, Y. HUANG, C. XIAO, C. HAN, Y. R. FUNG, Y. SU, H. WANG, C. QIAN, R. TIAN, K. ZHU, S. LIANG, X. SHEN, B. XU, Z. ZHANG, Y. YE, B. LI, Z. TANG, J. YI, Y. ZHU, Z. DAI, L. YAN, X. CONG, Y. LU, W. ZHAO, Y. HUANG, J. YAN, X. HAN, X. SUN, D. LI, J. PHANG,

C. YANG, T. WU, H. JI, Z. LIU, AND M. SUN, *Tool learning with foundation models*, 2023.

[203] Y. QIN, J. ZHANG, Y. LIN, Z. LIU, P. LI, M. SUN, AND J. ZHOU, *ELLE: Efficient lifelong pre-training for emerging data*, in Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 2789–2810.

[204] A. RADFORD, K. NARASIMHAN, T. SALIMANS, I. SUTSKEVER, ET AL., *Improving language understanding by generative pre-training*, (2018).

[205] A. RADFORD, J. WU, R. CHILD, D. LUAN, D. AMODEI, AND I. SUTSKEVER, *Language models are unsupervised multitask learners*, (2019).

[206] C. RAFFEL, N. SHAZEER, A. ROBERTS, K. LEE, S. NARANG, M. MATENA, Y. ZHOU, W. LI, P. J. LIU, ET AL., *Exploring the limits of transfer learning with a unified text-to-text transformer.*, J. Mach. Learn. Res., 21 (2020), pp. 1–67.

[207] P. RAJPURKAR, J. ZHANG, K. LOPYREV, AND P. LIANG, *SQuAD: 100,000+ questions for machine comprehension of text*, in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, J. Su, K. Duh, and X. Carreras, eds., Austin, Texas, Nov. 2016, Association for Computational Linguistics, pp. 2383–2392.

[208] O. RAM, Y. LEVINE, I. DALMEDIGOS, D. MUHLGAY, A. SHASHUA, K. LEYTON-BROWN, AND Y. SHOHAM, *In-context retrieval-augmented language models*, 2023.

[209] O. RAM, Y. LEVINE, I. DALMEDIGOS, D. MUHLGAY, A. SHASHUA, K. LEYTON-BROWN, AND Y. SHOHAM, *In-context retrieval-augmented language models*, Transactions of the Association for Computational Linguistics, 11 (2023), pp. 1316–1331.

[210] A. RASTOGI, X. ZANG, S. SUNKARA, R. GUPTA, AND P. KHAITAN, *Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset*, Proceedings of the AAAI Conference on Artificial Intelligence, 34 (2020), pp. 8689–8696.

[211] S.-A. REBUFFI, A. KOLESNIKOV, G. SPERL, AND C. H. LAMPERT, *icarl: Incremental classifier and representation learning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[212] R. REN, Y. WANG, Y. QU, W. X. ZHAO, J. LIU, H. TIAN, H. WU, J.-R. WEN, AND H. WANG, *Investigating the factual knowledge boundary of large language models with retrieval augmentation*, 2023.

[213] A. ROBERTS, C. RAFFEL, AND N. SHAZEER, *How much knowledge can you pack into the parameters of a language model?*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Nov. 2020, Association for Computational Linguistics, pp. 5418–5426.

[214] P. RÖTTGER AND J. PIERREHUMBERT, *Temporal adaptation of BERT and performance on downstream document classification: Insights from social media*, in Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, Nov. 2021, Association for Computational Linguistics, pp. 2400–2412.

[215] J. SAAD-FALCON, O. KHATTAB, C. POTTS, AND M. ZAHARIA, *Ares: An automated evaluation framework for retrieval-augmented generation systems*, 2023.

[216] V. SANH, A. WEBSON, C. RAFFEL, S. BACH, L. SUTAWIKA, Z. ALYAFEAI, A. CHAFFIN, A. STIEGLER, A. RAJA, M. DEY, M. S. BARI, C. XU, U. THAKKER, S. S. SHARMA, E. SZCZECHLA, T. KIM, G. CHHABLANI, N. NAYAK, D. DATTA, J. CHANG, M. T.-J. JIANG, H. WANG, M. MANICA, S. SHEN, Z. X. YONG, H. PANDEY, R. BAWDEN, T. WANG, T. NEERAJ, J. ROZEN, A. SHARMA, A. SANTILLI, T. FEVRY, J. A. FRIES, R. TEEHAN, T. L. SCAO, S. BIDERMAN, L. GAO, T. WOLF, AND A. M. RUSH, *Multitask prompted training enables zero-shot task generalization*, in International Conference on Learning Representations, 2022.

[217] T. SCHICK, J. DWIVEDI-YU, R. DESSÌ, R. RAILEANU, M. LOMELI, L. ZETTLEMOYER, N. CANCEDDA, AND T. SCIALOM, *Toolformer: Language models can teach themselves to use tools*, 2023.

[218] R. SCHUMANN AND I. REHBEIN, *Active learning via membership query synthesis for semi-supervised sentence classification*, in Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Hong Kong, China, Nov. 2019, Association for Computational Linguistics, pp. 472–481.

[219] T. SCIALOM, T. CHAKRABARTY, AND S. MURESAN, *Fine-tuned language models are continual learners*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 6107–6122.

[220] S. J. SEMNANI, V. Z. YAO, H. C. ZHANG, AND M. S. LAM, *Wikichat: A few-shot llm-based chatbot grounded with wikipedia*, 2023.

[221] B. SETTLES, *Active learning literature survey*, Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[222] P. SHAH, D. HAKKANI-TÜR, B. LIU, AND G. TÜR, *Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning*, in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), New Orleans - Louisiana, June 2018, Association for Computational Linguistics, pp. 41–51.

[223] Z. SHAO, Y. GONG, Y. SHEN, M. HUANG, N. DUAN, AND W. CHEN, *Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy*, 2023.

[224] Y. SHEN, H. YUN, Z. LIPTON, Y. KRONROD, AND A. ANANDKUMAR, *Deep active learning for named entity recognition*, in Proceedings of the 2nd Workshop on Representation Learning for NLP, Vancouver, Canada, Aug. 2017, Association for Computational Linguistics, pp. 252–256.

[225] Y. SHEN, Z. ZHANG, T. CAO, S. TAN, Z. CHEN, AND C. GAN, *Moduleformer: Modularity emerges from mixture-of-experts*, 2023.

[226] F. SHI, X. CHEN, K. MISRA, N. SCALES, D. DOHAN, E. CHI, N. SCHÄRLI, AND D. ZHOU, *Large language models can be easily distracted by irrelevant context*, 2023.

[227] F. SHI, X. CHEN, K. MISRA, N. SCALES, D. DOHAN, E. H. CHI, N. SCHÄRLI, AND D. ZHOU, *Large language models can be easily distracted by irrelevant context*, in Proceedings of the 40th International Conference on Machine Learning, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds., vol. 202 of Proceedings of Machine Learning Research, PMLR, 23–29 Jul 2023, pp. 31210–31227.

[228] W. SHI, J. MICHAEL, S. GURURANGAN, AND L. ZETTLEMOYER, *Nearest neighbor zero-shot inference*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 3254–3265.

[229] W. SHI, S. MIN, M. YASUNAGA, M. SEO, R. JAMES, M. LEWIS, L. ZETTLEMOYER, AND W. TAU YIH, *Replug: Retrieval-augmented black-box language models*, 2023.

[230] T. SHIN, Y. RAZEGHI, R. L. LOGAN IV, E. WALLACE, AND S. SINGH, *AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts*, in

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, eds., Online, Nov. 2020, Association for Computational Linguistics, pp. 4222–4235.

[231] K. SHUSTER, J. XU, M. KOMEILI, D. JU, E. M. SMITH, S. ROLLER, M. UNG, M. CHEN, K. ARORA, J. LANE, M. BEHROOZ, W. NGAN, S. POFF, N. GOYAL, A. SZLAM, Y.-L. BOUREAU, M. KAMBADUR, AND J. WESTON, *Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage*, 2022.

[232] C. SI, Z. GAN, Z. YANG, S. WANG, J. WANG, J. L. BOYD-GRABER, AND L. WANG, *Prompting GPT-3 to be reliable*, in The Eleventh International Conference on Learning Representations, 2023.

[233] K. SINGHAL, S. AZIZI, T. TU, S. S. MAHDAVI, J. WEI, H. W. CHUNG, N. SCALES, A. TANWANI, H. COLE-LEWIS, S. PFOHL, P. PAYNE, M. SENEVIRATNE, P. GAMBLE, C. KELLY, N. SCHARLI, A. CHOWDHERY, P. MANSFIELD, B. A. Y ARCAS, D. WEBSTER, G. S. CORRADO, Y. MATIAS, K. CHOU, J. GOTTWEIS, N. TOMASEV, Y. LIU, A. RAJKOMAR, J. BARRAL, C. SEMTURS, A. KARTHIKESALINGAM, AND V. NATARAJAN, *Large language models encode clinical knowledge*, 2022.

[234] A. SINITSIN, V. PLOKHOTNYUK, D. PYRKIN, S. POPOV, AND A. BABENKO, *Editable neural networks*, in International Conference on Learning Representations, 2020.

[235] Y. SU, L. SHU, E. MANSIMOV, A. GUPTA, D. CAI, Y.-A. LAI, AND Y. ZHANG, *Multi-task pre-training for plug-and-play task-oriented dialogue system*, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 4661–4676.

[236] N. TANDON, A. MADAAN, P. CLARK, AND Y. YANG, *Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback*, in Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, United States, July 2022, Association for Computational Linguistics, pp. 339–352.

[237] R. TEEHAN, M. CLINCIU, O. SERIKOV, E. SZCZECHLA, N. SEELAM, S. MIRKIN, AND A. GOKASLAN, *Emergent structures and training dynamics in large language models*, in Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models, A. Fan, S. Ilic, T. Wolf, and M. Gallé, eds., virtual+Dublin, May 2022, Association for Computational Linguistics, pp. 146–159.

[238] I. TENNEY, P. XIA, B. CHEN, A. WANG, A. POLIAK, R. T. MCCOY, N. KIM, B. V. DURME, S. R. BOWMAN, D. DAS, AND E. PAVLICK, *What do you learn from context? probing for sentence structure in contextualized word representations*, 2019.

[239] X. TIAN, L. HUANG, Y. LIN, S. BAO, H. HE, Y. YANG, H. WU, F. WANG, AND S. SUN, *Amendable generation for dialogue state tracking*, in Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, Online, Nov. 2021, Association for Computational Linguistics, pp. 80–92.

[240] K. TIRUMALA, A. MARKOSYAN, L. ZETTLEMOYER, AND A. AGHAJANYAN, *Memorization without overfitting: Analyzing the training dynamics of large language models*, in Advances in Neural Information Processing Systems, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., vol. 35, Curran Associates, Inc., 2022, pp. 38274–38290.

[241] K. TIRUMALA, A. H. MARKOSYAN, L. ZETTLEMOYER, AND A. AGHAJANYAN, *Memorization without overfitting: Analyzing the training dynamics of large language models*, 2022.

[242] H. TOUVRON, T. LAVRIL, G. IZACARD, X. MARTINET, M.-A. LACHAUX, T. LACROIX, B. ROZIÈRE, N. GOYAL, E. HAMBRO, F. AZHAR, A. RODRIGUEZ, A. JOULIN, E. GRAVE, AND G. LAMPLE, *Llama: Open and efficient foundation language models*, 2023.

[243] H. TOUVRON, L. MARTIN, K. STONE, P. ALBERT, A. ALMAHAIRI, Y. BABAEI, N. BASHLYKOV, S. BATRA, P. BHARGAVA, S. BHOSALE, D. BIKEL, L. BLECHER, C. C. FERRER, M. CHEN, G. CUCURULL, D. ESIOBU, J. FERNANDES, J. FU, W. FU, B. FULLER, C. GAO, V. GOSWAMI, N. GOYAL, A. HARTSHORN, S. HOSSEINI, R. HOU, H. INAN, M. KARDAS, V. KERKEZ, M. KHABSA, I. KLOUMANN, A. KORENEV, P. S. KOURA, M.-A. LACHAUX, T. LAVRIL, J. LEE, D. LISKOVICH, Y. LU, Y. MAO, X. MARTINET, T. MIHAYLOV, P. MISHRA, I. MOLYBOG, Y. NIE, A. POULTON, J. REIZENSTEIN, R. RUNGTA, K. SALADI, A. SCHELTEN, R. SILVA, E. M. SMITH, R. SUBRAMANIAN, X. E. TAN, B. TANG, R. TAYLOR, A. WILLIAMS, J. X. KUAN, P. XU, Z. YAN, I. ZAROV, Y. ZHANG, A. FAN, M. KAMBADUR, S. NARANG, A. RODRIGUEZ, R. STOJNIC, S. EDUNOV, AND T. SCIALOM, *Llama 2: Open foundation and fine-tuned chat models*, 2023.

[244] H. TRIVEDI, N. BALASUBRAMANIAN, T. KHOT, AND A. SABHARWAL, *Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions*, 2022.

[245] M. TÄNZER, S. RUDER, AND M. REI, *Memorisation versus generalisation in pre-trained language models*, 2022.

[246] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. U. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., vol. 30, Curran Associates, Inc., 2017.

[247] T. VU, M. IYYER, X. WANG, N. CONSTANT, J. WEI, J. WEI, C. TAR, Y.-H. SUNG, D. ZHOU, Q. LE, AND T. LUONG, *Freshllms: Refreshing large language models with search engine augmentation*, 2023.

[248] J. WANG, Y. LIANG, Z. SUN, Y. CAO, AND J. XU, *Cross-lingual knowledge editing in large language models*, 2023.

[249] R. WANG, D. TANG, N. DUAN, Z. WEI, X. HUANG, J. JI, G. CAO, D. JIANG, AND M. ZHOU, *K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters*, in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, Aug. 2021, Association for Computational Linguistics, pp. 1405–1418.

[250] Y. WANG, S. MISHRA, P. ALIPOORMOLABASHI, Y. KORDI, A. MIRZAEI, A. NAIK, A. ASHOK, A. S. DHANASEKARAN, A. ARUNKUMAR, D. STAP, E. PATHAK, G. KARAMANOLAKIS, H. LAI, I. PUROHIT, I. MONDAL, J. ANDERSON, K. KUZNIA, K. DOSHI, K. K. PAL, M. PATEL, M. MORADSHAHI, M. PARMAR, M. PUROHIT, N. VARSHNEY, P. R. KAZA, P. VERMA, R. S. PURI, R. KARIA, S. DOSHI, S. K. SAMPAT, S. MISHRA, S. REDDY A, S. PATRO, T. DIXIT, AND X. SHEN, *Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 5085–5109.

[251] Z. WANG, E. YANG, L. SHEN, AND H. HUANG, *A comprehensive survey of forgetting in deep learning beyond continual learning*, 2023.

[252] Z. WANG, G. ZHANG, K. YANG, N. SHI, W. ZHOU, S. HAO, G. XIONG, Y. LI, M. Y. SIM, X. CHEN, Q. ZHU, Z. YANG, A. NIK, Q. LIU, C. LIN, S. WANG, R. LIU, W. CHEN, K. XU, D. LIU, Y. GUO, AND J. FU, *Interactive natural language processing*, 2023.

[253] J. WEI, M. BOSMA, V. ZHAO, K. GUU, A. W. YU, B. LESTER, N. DU, A. M. DAI, AND Q. V. LE, *Finetuned language models are zero-shot learners*, in International Conference on Learning Representations, 2022.

[254] J. WEI, Y. TAY, R. BOMMASANI, C. RAFFEL, B. ZOPH, S. BORGEAUD, D. YOGATAMA, M. BOSMA, D. ZHOU, D. METZLER, E. H. CHI, T. HASHIMOTO, O. VINYALS, P. LIANG, J. DEAN, AND W. FEDUS, *Emergent abilities of large language models*, 2022.

[255] J. WEI, X. WANG, D. SCHUURMANS, M. BOSMA, BRIAN ICHTER, F. XIA, E. H. CHI, Q. V. LE, AND D. ZHOU, *Chain of thought prompting elicits reasoning in large language models*, in Advances in Neural Information Processing Systems, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds., 2022.

[256] X. WEI, S. WANG, D. ZHANG, P. BHATIA, AND A. ARNOLD, *Knowledge enhanced pre-trained language models: A compreshensive survey*, 2021.

[257] J. WILLIAMS, A. RAUX, D. RAMACHANDRAN, AND A. BLACK, *The dialog state tracking challenge*, in Proceedings of the SIGDIAL 2013 Conference, Metz, France, Aug. 2013, Association for Computational Linguistics, pp. 404–413.

[258] C.-S. WU, S. C. HOI, R. SOCHER, AND C. XIONG, *TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Nov. 2020, Association for Computational Linguistics, pp. 917–929.

[259] C.-S. WU, A. MADOTTO, E. HOSSEINI-ASL, C. XIONG, R. SOCHER, AND P. FUNG, *Transferable multi-domain state generator for task-oriented dialogue systems*, in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 2019, Association for Computational Linguistics, pp. 808–819.

[260] S. WU, M. PENG, Y. CHEN, J. SU, AND M. SUN, *Eva-kellm: A new benchmark for evaluating knowledge editing of llms*, 2023.

[261] Y. WU, M. N. RABE, D. HUTCHINS, AND C. SZEGEDY, *Memorizing transformers*, in International Conference on Learning Representations, 2022.

[262] J. XIE, K. ZHANG, J. CHEN, R. LOU, AND Y. SU, *Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge clashes*, 2023.

[263] K. XIE, C. CHANG, L. REN, L. CHEN, AND K. YU, *Cost-sensitive active learning for dialogue state tracking*, in Proceedings of the 19th Annual SIGdial Meeting on

Discourse and Dialogue, Melbourne, Australia, July 2018, Association for Computational Linguistics, pp. 209–213.

[264] Y. XU, Y. HOU, W. CHE, AND M. ZHANG, *Language anisotropic cross-lingual model editing*, 2023.

[265] Y. XU, M. NAMAZIFAR, D. HAZARIKA, A. PADMAKUMAR, Y. LIU, AND D. HAKKANI-TÜR, *Kilm: Knowledge injection into encoder-decoder language models*, 2023.

[266] P. YANG, H. HUANG, AND X.-L. MAO, *Comprehensive study: How the context information of different granularity affects dialogue state tracking?*, in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, Aug. 2021, Association for Computational Linguistics, pp. 2481–2491.

[267] Z. YANG, L. LI, J. WANG, K. LIN, E. AZARNASAB, F. AHMED, Z. LIU, C. LIU, M. ZENG, AND L. WANG, *Mm-react: Prompting chatgpt for multimodal reasoning and action*, 2023.

[268] S. YAO, J. ZHAO, D. YU, N. DU, I. SHAFRAN, K. R. NARASIMHAN, AND Y. CAO, *React: Synergizing reasoning and acting in language models*, in The Eleventh International Conference on Learning Representations, 2023.

[269] Y. YAO, P. WANG, B. TIAN, S. CHENG, Z. LI, S. DENG, H. CHEN, AND N. ZHANG, *Editing large language models: Problems, methods, and opportunities*, 2023.

[270] F. YE, J. MANOTUMRUKSA, Q. ZHANG, S. LI, AND E. YILMAZ, *Slot self-attentive dialogue state tracking*, in Proceedings of the Web Conference 2021, WWW '21, New York, NY, USA, 2021, Association for Computing Machinery, p. 1598–1608.

[271] D. YIN, L. DONG, H. CHENG, X. LIU, K.-W. CHANG, F. WEI, AND J. GAO, *A survey of knowledge-intensive nlp with pre-trained language models*, 2022.

[272] W. YIN, J. LI, AND C. XIONG, *ConTinTin: Continual learning from task instructions*, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 3062–3072.

[273] Z. YIN, Q. SUN, Q. GUO, J. WU, X. QIU, AND X. HUANG, *Do large language models know what they don't know?*, in Findings of the Association for Computational Linguistics: ACL 2023, A. Rogers, J. Boyd-Graber, and N. Okazaki, eds., Toronto, Canada, July 2023, Association for Computational Linguistics, pp. 8653–8665.

[274] Z. YIN, Q. SUN, Q. GUO, J. WU, X. QIU, AND X. HUANG, *Do large language models know what they don't know?*, 2023.

[275] O. YORAN, T. WOLFSON, O. RAM, AND J. BERANT, *Making retrieval-augmented language models robust to irrelevant context*, 2023.

[276] P. YU AND H. JI, *Self information update for large language models through mitigating exposure bias*, 2023.

[277] W. YU, Z. ZHANG, Z. LIANG, M. JIANG, AND A. SABHARWAL, *Improving language models via plug-and-play retrieval feedback*, 2023.

[278] W. YU, C. ZHU, Z. LI, Z. HU, Q. WANG, H. JI, AND M. JIANG, *A survey of knowledge-enhanced text generation*, ACM Comput. Surv., 54 (2022).

[279] Z. YU, C. XIONG, S. YU, AND Z. LIU, *Augmentation-adapted retriever improves generalization of language models as generic plug-in*, 2023.

[280] M. YUAN, H.-T. LIN, AND J. BOYD-GRABER, *Cold-start active learning through self-supervised language modeling*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Nov. 2020, Association for Computational Linguistics, pp. 7935–7948.

[281] X. ZENG, S. GARG, R. CHATTERJEE, U. NALLASAMY, AND M. PAULIK, *Empirical evaluation of active learning techniques for neural MT*, in Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), Hong Kong, China, Nov. 2019, Association for Computational Linguistics, pp. 84–93.

[282] M. J. Q. ZHANG AND E. CHOI, *Mitigating temporal misalignment by discarding outdated facts*, 2023.

[283] P. ZHANG, G. ZENG, T. WANG, AND W. LU, *Tinyllama: An open-source small language model*, 2024.

[284] Q. ZHANG, S. CHEN, D. XU, Q. CAO, X. CHEN, T. COHN, AND M. FANG, *A survey for efficient open domain question answering*, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), A. Rogers, J. Boyd-Graber, and N. Okazaki, eds., Toronto, Canada, July 2023, Association for Computational Linguistics, pp. 14447–14465.

[285] S. ZHANG, S. ROLLER, N. GOYAL, M. ARTETXE, M. CHEN, S. CHEN, C. DEWAN, M. DIAB, X. LI, X. V. LIN, T. MIHAYLOV, M. OTT, S. SHLEIFER, K. SHUSTER, D. SIMIG, P. S. KOURA, A. SRIDHAR, T. WANG, AND L. ZETTLEMOYER, *Opt: Open pre-trained transformer language models*, 2022.

[286] T. ZHANG, H. LUO, Y.-S. CHUANG, W. FANG, L. GAITSKELL, T. HARTVIGSEN, X. WU, D. FOX, H. MENG, AND J. GLASS, *Interpretable unified language checking*, 2023.

[287] Z. ZHANG, M. FANG, AND L. CHEN, *Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering*, 2024.

[288] Z. ZHANG, M. FANG, L. CHEN, AND M.-R. NAMAZI-RAD, *CITB: A benchmark for continual instruction tuning*, in Findings of the Association for Computational Linguistics: EMNLP 2023, H. Bouamor, J. Pino, and K. Bali, eds., Singapore, Dec. 2023, Association for Computational Linguistics, pp. 9443–9455.

[289] Z. ZHANG, M. FANG, L. CHEN, M.-R. NAMAZI-RAD, AND J. WANG, *How do large language models capture the ever-changing world knowledge? a review of recent advances*, 2023.

[290] Z. ZHANG, M. FANG, L. CHEN, M.-R. NAMAZI-RAD, AND J. WANG, *How do large language models capture the ever-changing world knowledge? a review of recent advances*, in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, H. Bouamor, J. Pino, and K. Bali, eds., Singapore, Dec. 2023, Association for Computational Linguistics, pp. 8289–8311.

[291] Z. ZHANG, M. FANG, F. YE, L. CHEN, AND M.-R. NAMAZI-RAD, *Turn-level active learning for dialogue state tracking*, in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, H. Bouamor, J. Pino, and K. Bali, eds., Singapore, Dec. 2023, Association for Computational Linguistics, pp. 7705–7719.

[292] Z. ZHANG, E. STRUBELL, AND E. HOVY, *A survey of active learning for natural language processing*, in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, Dec. 2022, Association for Computational Linguistics, pp. 6166–6190.

[293] R. ZHAO, X. LI, S. JOTY, C. QIN, AND L. BING, *Verify-and-edit: A knowledge-enhanced chain-of-thought framework*, 2023.

[294] C. ZHEN, Y. SHANG, X. LIU, Y. LI, Y. CHEN, AND D. ZHANG, *A survey on knowledge-enhanced pre-trained language models*, 2022.

[295] C. ZHENG, L. LI, Q. DONG, Y. FAN, Z. WU, J. XU, AND B. CHANG, *Can we edit factual knowledge by in-context learning?*, 2023.

[296] Z. ZHONG, Z. WU, C. D. MANNING, C. POTTS, AND D. CHEN, *Mquake: Assessing knowledge editing in language models via multi-hop questions*, 2023.

[297] C. ZHOU, P. LIU, P. XU, S. IYER, J. SUN, Y. MAO, X. MA, A. EFRAT, P. YU, L. YU, S. ZHANG, G. GHOSH, M. LEWIS, L. ZETTLEMOYER, AND O. LEVY, *Lima: Less is more for alignment*, 2023.

[298] L. ZHOU AND K. SMALL, *Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering*, arXiv preprint arXiv:1911.06192, (2019).

[299] W. ZHOU, S. ZHANG, H. POON, AND M. CHEN, *Context-faithful prompting for large language models*, 2023.

[300] C. ZHU, A. S. RAWAT, M. ZAHEER, S. BHOJANAPALLI, D. LI, F. YU, AND S. KUMAR, *Modifying memories in transformer models*, 2020.

[301] F. ZHU, W. LEI, C. WANG, J. ZHENG, S. PORIA, AND T.-S. CHUA, *Retrieving and reading: A comprehensive survey on open-domain question answering*, 2021.

[302] Q. ZHU, B. LI, F. MI, X. ZHU, AND M. HUANG, *Continual prompt tuning for dialog state tracking*, in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, May 2022, Association for Computational Linguistics, pp. 1124–1137.

[303] X. ZHU, C. YANG, B. CHEN, S. LI, J.-G. LOU, AND Y. YANG, *Question answering as programming for solving time-sensitive questions*, 2023.

[304] Y. ZHUANG, Y. YU, K. WANG, H. SUN, AND C. ZHANG, *ToolQA: A dataset for LLM question answering with external tools*, in Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023.

[305] N. ZIEMS, W. YU, Z. ZHANG, AND M. JIANG, *Large language models are built-in autoregressive search engines*, 2023.