A Unified Adapted-based Framework towards Unbiased and Robust Factual Knowledge Extraction

Yijun YANG

Master of Science
School of Informatics
University of Edinburgh
2023

Abstract

Factual knowledge extraction for generating knowledge triples from large language models (LLMs) has raised surging interest in the natural language processing (NLP) community in recent years. A variety of works have been proposed in order to improve the accuracy of extracted knowledge based on optimizing the prompts since prompting is a computationally cheap way to interact with LLMs. However, an increasing number of researchers point out that there exists severe undesirable bias among prompt-based models such as prompt preference bias and prompt verbalization bias. Besides, as far as we learn, there is no work investigating other tuning methods except prompt tuning such as adapter tuning, which is a recent popular parameter-efficient tuning method. In this thesis, to make a comprehensive measure of the prompt verbalization bias, we first create **ParaTrex** dataset utilizing the large language models and through strict human supervision. ParaTrex is shown to have better diversity and larger scales than the existing paraphrased dataset. Secondly, to mitigate the biases in prompt-based models as well as fill in the gap of research of adapter-tuning on knowledge probing, we propose a unified adapter-based framework **Uni-Arkex** for mitigating both prompt preference bias and prompt verbalization bias. Experimental results show the competitive performance of adapter tuning. Moreover, they present that adapter-tuning helps our proposed framework achieve new state-of-the-art results in extraction accuracy while simultaneously successfully reducing both of these biases. Sufficient analyses are conducted to show the adapter's excellent compatibility with multi-task frameworks and the synergizing effect of synchronously optimizing those two biases through our proposed framework.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yijun YANG*)

Acknowledgements

This is the toughest year during my study journey. My dissertation could not have been successfully completed without so much support and company I received from my supervisor, my friends, my parents, and my girlfriends.

First and foremost, I would like to express my appreciation to my supervisor Jeff Z. PAN. His invaluable advice and feedback not only broadened my horizons over the whole domain but also gave me much encouragement to finish all the experiments and writing for this dissertation. I learned a lot during the three-month meeting time.

Then, I would like to extend my sincere thanks to all my friends. Junjie XU, Zhanghao HU, and Chenmien TAN were my true comrades in this journey of growth and knowledge, and I am profoundly grateful for all the precious time we spent learning and growing together. It was our mutual encouragement and assistance during challenging moments that gave me the courage and ability to complete this project. I want to express my heartfelt appreciation to Ph.D. student Jie HE for the meticulous suggestions and support that enhanced my whole project. Similarly, my gratitude extends to Ph.D. student Ricky ZHU for the insightful advice provided during the initial phases of the project. Patrick CHEN is both my advisor and friend, and I sincerely thank him for all the advice he provided regarding the dataset portion of my paper. I'm grateful to all my encountered friends who supported me during difficult times, enabling me to push my boundaries and accomplish this extensive project in such a challenging three-month period.

I would also like to thank my parents for their patience and love throughout the journey of studying abroad. Although physical distance kept us apart, their boundless love became my driving force, propelling me to conquer the hurdles of challenging studies and ensuring that I never felt isolated or alone.

Finally, I want to express my sincerest gratitude to my girlfriend Yanran NI. Our journey began in the first year of college, and since then, she has stood beside me, an unwavering presence through my entire university experience and my adventure studying abroad. Her constant companionship and love are always my spring of confidence and courage, infusing my world with assurance during every tough period filled with uncertainty and anxiety. It's my greatest fortune to meet her.

Table of Contents

1 Introduction **1**

1.1 Contributions 3

1.2 Outline 4

2 Background and Related works **5**

2.1 Access knowledge in Large Language models 5

2.2 Adapter-based Tuning 7

2.3 Bias Study 9

2.3.1 Prompt Preference Bias 9

2.3.2 Prompt Verbalization Bias 10

3 Extending dataset through large language models **11**

3.1 Motivation 11

3.2 Dataset Creation 12

3.3 Dataset evaluation 13

3.3.1 Automatic evaluation 14

3.3.2 Human evaluation 15

4 Task definition **16**

4.1 Problem definition 16

4.2 Datasets 18

4.3 Evaluation Metrics 20

4.3.1 Accuracy measurement 20

4.3.2 Prompt preference bias measurement 20

4.3.3 Prompt verbalization bias measurement 21

5 Methodology **22**

5.1 Comparision models 22

5.2 Overall Framework of proposed models 23
5.2.1 Improve the accuracy of factual extraction 23
5.2.2 Alleviating prompt preference bias 24
5.2.3 Alleviating prompt verbalization bias 25

6 Experiment and results **27**
6.1 Experiment setup 27
6.1.1 Datasets 27
6.1.2 Model implementation details 28
6.2 Main results 28
6.2.1 Results for LAMA 29
6.2.2 results for LM-KBC 32
6.2.3 Scaling results for models with different sizes 32
6.3 Ablation Study 33
6.4 Case Study 35
6.5 Discussion 37

7 Conclusions **39**

Bibliography **41**

A Data Extension details for ParaTrex **47**
A.1 Details of generated templates for an example relation 47
A.2 Human evaluation 47

B Full results for LAMA **50**
B.1 Specific results for all relations of our proposed method 50

Chapter 1

Introduction

Pretrained large language models (LLMs) are now widely employed in the field of natural language processing(NLP) and have achieved impressive capabilities across various downstream applications [31] [30]. They are called large language models since these models contain millions, or billions of parameters and are pre-trained on huge amounts of public corpus. A crucial reason behind the success of LLMs is shown to be the inherent knowledge stored in their parameters learned through pre-training, which includes world knowledge [39], relational knowledge [46], commonsense knowledge[8] and etc. However, as neural networks are widely considered as a black box system, the inherent knowledge within language models is typically encoded in a diffused manner, leading to challenges in both interpretation and updating the knowledge inside. Contrarily, Knowledge Bases(KBs) are easier to modify and more trustable to access the required knowledge in practice. Hence, there is a rising interest from researchers to investigate how to treat LLMs as KBs by measuring and extracting factual knowledge directly from LLMs.

LAMA [39] is the first and most popular benchmark for measuring the extracted factual knowledge from LLMs. In LAMA, factual knowledge is represented as triples <subject, relation, object> and is extracted through the query <subject, relation, ?> and a manually designed prompt template. For example, regarding a specific query <Barack Obama, place of birth, ?> , we query LLMs using the prompt:"*Barack Obama was born in [MASK]*" to extract factual knowledge. Since searching for optimal prompts has long been a problem within prompt-based models [31], massive existing research focuses on automatically optimizing the prompt templates. For discrete prompts such as natural language prompts, [47] proposed AutoPrompt and aimed at generating discrete prompts through gradient optimization. In contrast, [32] argue that soft prompts, which

are formed by continuous vectors, are more effective and propose P-tuning to optimize the prompt template through an inner Bi-LSTM module from scratch. In addition, [29] proposed prefix-tuning to tune merely task-related prefix for the input prompts. Aside from prompt-tuning methods, fine-tuning-based methods such as [28] are widely used in practice as a powerful baseline or solution for extracting challenging relations like [49]. Despite fine-tuning and prompt-tuning solutions, few research investigate how adapter tuning, which is another popular parameter-efficient fine-tuning method, performs in probing knowledge from LLMs. That motivates us to first compare adapter-tuning methods with existing prompt-tuning and fine-tuning baselines. Extensive experiments in this paper show that for accuracy performance, adapter-tuning can perform even better than fine-tuning with less tuned parameters in most cases and is consistently better than P-tuning methods. That becomes the first motivation for designing an adapter-based framework for factual knowledge extraction.

With the rising interest in knowledge-probing measurements, some researchers begin to focus on the underlying rationales behind the answers generated by LLMs. [5] firstly points out that prompt-based models have severe prompt preference bias. That means prompt-based models generate answers mostly based on their preference for specific prompt templates instead of their true inherent knowledge. They show that the prediction distribution from prompt-only inputs such as "*[MASK] was born in [MASK]*" has an extremely high correlation with the original inputs "*Barack Obama was born in [MASK]*", which shows that the prediction is dominated by prompt templates. [53] tries to mitigate this problem by automatically picking potential objects from prompt-only inputs through a classifier and maximizing their entropy. However, our experiments show that the neural classifier may not be trustworthy and we propose a simplified and interpretable version based on [53] to mitigate prompt preference bias.

In addition to prompt preference bias, recent research points out that inconsistency between semantically similar prompts is another severe and undesirable problem. In the remaining parts of this paper, we refer to this inconsistency as prompt verbalization bias, which means that models may favor specific verbalization of prompt templates and give a biased distribution. [12] first put emphasis on this problem and tries to mitigate it through additional paraphrased datasets. [36] further proposed P-adapters. P-adapters insert one adapter layer to map different paraphrases into the same space to improve the robustness of the model outputs. Regarding this bias, we propose a novel self-augmentation method to improve the inner consistency of our models. Specifically, we augment the original inputs with the prefix "It is true that" and "It is false that" to

help the model recheck their output answers. Experiments show that through simple self-augmentation, the prompt verbalization bias can be significantly reduced.

Motivated by [11], which shows that adapters have good compatibility with multi-task settings. We then try to design a Unified Adapter-based framework for unbiased and Robust factual Knowledge Extraction(**Uni-ARKEx**), with the primary aim of using a unified framework to mitigate both prompt preference bias as well as prompt verbalization bias and have better accuracy over knowledge probing tasks. We demonstrate in our experiments that our proposed framework achieves a new state-of-the-art performance on probing accuracy. This achievement is coupled with the advantages of parameter-efficient tuning, low prompt-preference bias, and a notable level of consistency. We provide a detailed analysis of our proposed methods in chapter 6.

Besides our proposed models, additional paraphrased datasets are important components necessary for measuring the prompt verbalization bias. [12] first proposed a paraphrased version of the LAMA benchmark called ParaRel. However, due to the constraints of NLP tools at that time, the scale and diversity still have a large potential to improve. In this thesis, we proposed a more diverse and high-quality paraphrased dataset through LLMs GPT-3.5[37]. We report both automatic and human evaluations on our proposed dataset to ensure its practicality.

1.1 Contributions

We summarise the main contributions of this dissertation as follows.

1. We propose ParaTrex, a large-scale challenging paraphrased dataset based on the LAMA benchmark. Our evaluations show that our proposed dataset is more lexically and syntactically diverse than the currently available dataset ParaRel[12]. Our human evaluation shows that ParaTrex has a high agreement with humans.

2. We fill in the gap of research on adapter tuning for factual knowledge extraction and show that adapter tuning is able to perform nearly or better than fine-tuning in most cases due to its parameter efficiency and preservation of inner knowledge in LLMs.

3. We propose two modules on mitigating prompt preference bias and prompt verbalization bias and design a unified framework for alleviating both of these biases. Our proposed framework beats the current SOTA MeCoD [53] on BERT-large and RoBERTa-large settings.

4. We made ablation studies and case studies to validate the effectiveness of each module in our proposed framework. It is shown that separate modules within our models may have synergized effects on mitigating certain biases instead of working separately.

1.2 Outline

The main structure of this thesis is as follows:

- **Chapter 2** We introduce the background knowledge behind factual knowledge extraction, current methods for accessing the knowledge, and the bias study of factual knowledge extraction tasks.
- **Chapter 3** We explain the motivation of our proposed dataset ParaTrex, the construction of ParaTrex, and both automatic and human evaluations for ParaTrex.
- **Chapter 4** We give a formal definition of our task and evaluation matrices.
- **Chapter 5** Our proposed framework Uni-Arkex will be explained, including four modules: Adapters, maxing entropy. self-augmentation and paraphrased augmentation.
- **Chapter 6** The experiment results, including the ablation study and case study, will be discussed.
- **Chapter 7** We conclude this project and discuss the potential limitations and future work.

Chapter 2

Background and Related works

This chapter mainly introduces the essential background information and recent works related to this project. Section 2.1 introduces the origin of the factual knowledge extraction task and recent relevant research including the main ways for accessing the knowledge in language models. Section 2.2 introduces the background knowledge of parameter-efficient fine-tuning instead of fine-tuning. Section 2.3 discusses recent bias studies related to this task.

2.1 Access knowledge in Large Language models

In the context of this paper, a large language model(LLM) refers to a deep neural language model pre-trained on a large amount of unlabeled text in a self-supervised setting such as masked language modeling (BERT[10]) and next-word prediction (GPT[3]). Although LLMs have already achieved great breakthroughs in huge amounts of NLP tasks, we still do not have full control over the behavior of LLMs to let them give trustable answers[30]. However, knowledge bases are an existing solution to accessing specific gold-standard relation information. Knowledge bases(KBs) usually represent a manually engineered schema that prescribes the potential set of entities and relationships, along with their interconnections. This schema guarantees precise, consistent, and explainable outcomes. Therefore, how can we control the repository of knowledge stored and, especially, extract the knowledge in the weights of an LLM similar to KBs, emerges as a compelling avenue for research.

Fine-tuning: A dominant approach to obtaining particular pieces of information from LLMs is by means of fine-tuning the model on a pertinent downstream task, such as commonsense question answering. Fine-tuning LLMs for specific downstream tasks

has proven to be a successful approach for refining and eliciting specific knowledge for evaluation on these tasks [42]. This is because the majority of knowledge that is encoded in an LLM is garnered during pretraining, with fine-tuning merely acquiring an interface to access such accumulated knowledge [8]. However, recent works point out that fine-tuning for factual knowledge extraction may have some potential problems. An example is frequency shock [24], where, in testing time, the model over-predicts rare entities in the training set and under-predicts common entities that do not appear in the training set in enough times. Besides, fine-tuning is also generally considered to suffer from catastrophic forgetting, which means the LLM forgets the previously learned knowledge while fine-tuning. This may not be optimal when considering the utilization of LLMs as generalized KBs or for the purpose of general intelligence.

Prompting: At the same time, although the paradigm of pre-training and fine-tuning is popular among pre-trained language models such as BERT-large [10] with 340 million parameters, it becomes hard to fine-tune larger LLMs such as GPT-3 [3] and LLAMA [51] due to huge computation costs, who have 175 billion and 7 to 65 billion parameters respectively. Fortunately, recent findings by [31] suggest that prompts offer a promising avenue for directly accessing this knowledge without the need for extra fine-tuning. The prompting paradigm provides the model with a familiar query format, such as a cloze-style format for BERT, thereby resulting in improved responses. Prompting is generally separated into Discrete Prompts and Soft Prompts.

Discrete prompts usually refer to prompts that may not be optimized like continuous vectors after being tokenized by language models. Many papers tackle prompting from the view of cloze-style like in [39]. For example, "The capital of United Kingdom is <mask>" is prompted for BERT if we want to extract the capital knowledge from LLMs. In Radford's studies [42], prompting was first introduced. They showed that it could achieve satisfactory zero-shot performance with the use of well-crafted prompts. Other researchers have also capitalized on this enhanced performance and have evaluated various discrete prompting methods, including entailment [52] and label token optimization [55]. Nevertheless, the process of manually creating the most effective prompt for specific tasks presents a formidable challenge. To address this issue, AutoPrompt [47] tackles prompt creation automatically using gradient-based search, while a more hands-on approach to prompt crafting was proposed by [34]. Despite these innovations, the quality of discrete prompts still remains uncertain.

Therefore, soft prompts have been introduced. These soft prompts are formulated using continuous and learnable word vectors as input. Throughout the training process,

gradient descent updates the parameters associated with the soft prompts, while the core model parameters remain fixed. Prefix-tuning, as suggested by [29], firstly focuses on tuning several task-specific vectors as soft prompts and demonstrates comparable generation outcomes while modifying only a limited subset of the model's parameters. In contrast to prefix-tuning, which incorporates adjustable prefixes throughout every Transformer layer, prompt tuning [27] presents a simpler approach involving the inclusion of soft prompts solely at the input layer. They showed that as the model's size grows, the performance disparity between prompt tuning and complete fine-tuning diminishes. [32] further proposes a method, making all tokens within prompt templates as learnable soft prompts and showing similar scaling results on larger language models. As for the factual knowledge extraction tasks, [41] and [56] have determined that soft prompts offer distinct advantages over discrete prompts since soft prompts exhibit enhanced expressiveness, enabling them to encapsulate multiple contexts simultaneously. However, recent research has begun to challenge the notion that soft prompts consistently outperform straightforward manual prompts and suggests using discrete prompts as a baseline before using soft prompts [58]. It's worth noting that all prompt-based models suffer from certain issues. For instance, they can be sensitive to the choice of initialization, be unstable to optimize, and can exhibit inconsistency when dealing with semantically similar prompts [4]. These challenges remain to be effectively addressed.

2.2 Adapter-based Tuning

Shared the same motivation with prompt tuning to overcome the problems of expensive fine-tuning and pre-training, adapters are proposed for parameter-efficient transfer learning. Adapter-based techniques inject compact neural components (known as adapters) into the layers of the Transformer model and only fine-tune these adapters for the purpose of model adaptation. Among them, Houlsby [19] gives the first instantiation, which is shown in Fig 2.1. Specifically, the adapter module is inserted between each feed-forward layer and the layer norm layer within each transformer layer. One adapter module contains a down-projection and an up-projection neural layer. For an input feature $\mathbf{h} \in \mathbb{R}^d$, a down-projection parameter matrix $\mathbf{W}_d \in \mathbb{R}^{d \times r}$ is first applied to map the input into a r-dimensional bottleneck space, where r is usually far less than initial dimension d. A nonlinear function is applied after that and then the up-projection matrix $\mathbf{W}_u \in \mathbb{R}^{r \times d}$ is used to project the vectors back into the d-dimension space. Finally, a residual connection is added. Within each transformer block, the adapter module is

inserted following the multi-head self-attention and feed-forward network sublayers. This arrangement leads to a reduction in the number of tuned parameters per layer to $2 \times (2dr$ for projection matrices $+ d$ for residual connections $+ r$ for bias terms$)$. In practical terms, this strategy involves merely fine-tuning about 0.5% to 8% of the entire model's parameters. This leads to about 60% faster than vanilla fine-tuning.

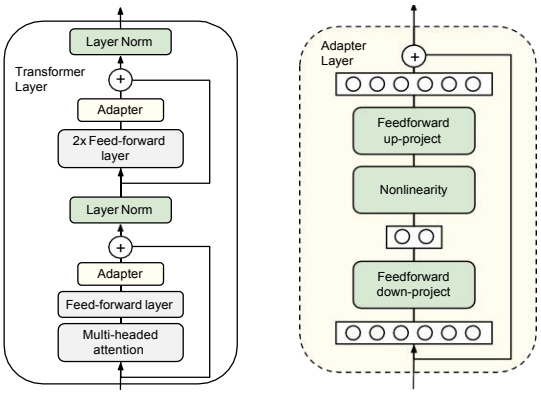

Figure 2.1: Illustration of adapter architecture [19] in transformer blocks.

Except for parameter efficiency, adapters are also known for their high modularity. Trained adapters can be inserted in pre-trained LLMs for specific tasks without the need to revisit LLMs and fine-tune a new model. Adapter-based fine-tuning offers the benefit of simultaneously incorporating multiple instances of adapters onto a pre-trained model, which is valuable in various application scenarios such as multi-task learning [50] [35]. By integrating adapter modules alongside the self-attention module in a parallel fashion, pre-trained language models (PLMs) can showcase remarkable representational capacity within the multi-task learning framework. AdapterFusion [40] is also shown to be effective in debiasing on multitask-debiasing framework [26], which also motivates our idea on using adapter-based tuning as a unified framework on unbiased factual knowledge extraction.

Although the training of adapters is faster than fine-tuning, we note that its inference time will be 4% or 6% slower [19]. However, this computational cost could be reduced by dropping adapters dynamically at the low transformer layers [45]. Recent studies indicate that adapter-based fine-tuning exhibits better robustness compared to traditional fine-tuning approaches [11]. Notably, in scenarios involving few-shot learning and cross-lingual tasks, adapter-based fine-tuning has been shown to outperform standard fine-tuning methods [17]. Furthermore, it has been found to be more robust when subjected to adversarial attacks [15]. This robustness also motivates us to apply adapter-based tuning for an unbiased and robust framework for knowledge probing.

2.3 Bias Study

Although prompt-based factual knowledge extraction is able to achieve decent performance, it is crucial to comprehend the reason behind the specific predictions generated by LLMs in order to attain more accurate outcomes. Recent research reveals that LLMs occasionally formulate predictions without being based on particular knowledge, which makes the probing results biased and unreliable. Such biases are mainly divided into prompt preference bias and prompt verbalization bias in this thesis.

2.3.1 Prompt Preference Bias

Prompt preference bias was first introduced in [5], which means that the prediction of prompt-based models is severely prompt-biased. Specifically, for prompt-only inputs such as "*<mask> was born in <mask>*." and raw inputs such as "*Steve Jobs was born in <mask>*.", we expect the distributions of predictions stemming from two distinct inputs to show a significant dissimilarity. This is because prompt-only inputs lack the key subject information in the input, so models can only depend on prompts to make predictions instead of their internal knowledge. However, after analyzing the correlations between these two distributions, [5] shows that correlation coefficients between these two inputs exceed 0.6 in more than half of the relations. This indicates that the distribution derived from prompt-only inputs holds greater influence over the final prediction distribution, implying that prompt-based knowledge retrieval largely relies on informed assumptions drawn from these prompt-influenced distributions. In simpler terms, the predictions are produced by sampling from prompt-biased distributions, guided by the moderate influence of subjects. [4] further applied causal analysis on this bias, showing that this bias stems from the underlying linguistic correlation between PLMs and prompts, and giving a causal framework solution by manually intervening to block the observed back door path in the causal model.

Recent work [53] also proposes a new metric on positioning the prompt preference bias as well as another solution based on neural models. Under the assumption that unbiased models should output a nearly uniform distribution over potential candidates under prompt-only inputs, this paper evaluates prompt-preference bias through the entropy of the top 10 predictions from prompt-only inputs, which is called counterfactual entropy (See section 4.3 for details). Meanwhile, they provide a muti-task and contrastive learning framework to mitigate the prompt preference bias. However, their method solely alleviates prompt preference bias. This inspires us to design a unified

framework targeting both prompt preference bias and prompt verbalization bias.

2.3.2 Prompt Verbalization Bias

Prompt Verbalization Bias is also known as the inconsistency of LLMs on prompts. Note that this is different from prompt preference bias in this paper. Prompt preference bias represents the model excessively depending on the initial distribution established by prompts to formulate predictions, rather than relying on genuine internal knowledge, while prompt verbalization bias puts emphasis on the inconsistency of LLMs on semantically similar queries with different verbalizations. Research has shown that LLMs suffer from a lack of consistency in their answers [12]. They may output different distributions of answers when queried for the same fact but under a different verbalization such as paraphrases. Therefore, a strategy to assess the consistency of a model involves probing LLMs by a paraphrase of the identical relation for a specific subject and checking whether the model consistently generates the same predictions [14]. Several benchmarks have been proposed to measure the consistency of LLMs [12] [44], where [12] tries to improve the consistency of the model by minimizing the Kullback-Leibler(**KL**) divergence of output distributions between paraphrases. [4] further applies causal analysis to show that this inconsistency sources from the same linguistic regularity with the pre-training corpus. Besides, [36] makes the first step of inserting adapters between the embedding layer and the first transformer layer in order to map different paraphrases into the same embedding space, which gives additional insight into the advantages of adapters in mitigating prompt verbalization bias.

In addition to insensitivity under paraphrases, previous research delves into the fragility of Language Models and examines the impact of incorporating negation such as "*not*" into prompts [23] [14]. They show that an LM can maintain contradictory beliefs within its parameters, such as simultaneously holding "*Birds can fly*" and "*Birds cannot fly*", indicating insensitivity to the contextual nuances of negation. Furthermore, [23] demonstrate a comparable effect when misguiding the probe with a misleading distractor (e.g., "Talk? Birds can [MASK]"). Thus, robust LLMs are expected to exhibit consistency not only across varied paraphrases but also negations and entailments. [16] quantify consistency within entailment, encompassing contrapositives, after updating the LM's beliefs. Similar to [12]'s effort on overcoming inconsistency under paraphrases, [16] includes another loss function to their objective function to minimize the error across entailed data, which coincides with our proposed multi-task framework.

Chapter 3

Extending dataset through large language models

This chapter discusses our motivation and methodology for expanding knowledge probing datasets into their paraphrased versions leveraging large language models. These datasets will be then used for measuring consistency and training in the following chapters. Section 3.1 explains our motivation for proposing a new dataset. Section 3.2 elaborates on the methods for creating datasets. The evaluation matrices are introduced in section 3.3, including automatic matrices and human evaluation.

3.1 Motivation

As mentioned in section 2.3.2, we measure prompt verbalization bias by checking whether LLMs can provide consistent predictions based on different prompt paraphrases. There already exists several benchmarks for paraphrasing to measure the consistency of LLMs [44] and methods for generating paraphrases [2]. For instance [9] employs back translation to generate paraphrases for measuring consistency after modifying the factual knowledge in LLMs. However, they do not perform on the LAMA dataset. Therefore, before conducting our experiment, a paraphrased version of our factual knowledge benchmark LAMA [39] is necessary.

[12] and [36] make the first trial to generate paraphrased versions based on the LAMA dataset. [12] generates a high-quality paraphrase dataset called ParaRel, with 328 distinct paraphrases over 38 relations. They use back-translation to augment each base pattern of prompt templates and further do a systematic exploration of Wikipedia sentences containing the identical subject-object tuple as in LAMA datasets. Then

they manually extract their templates. However, we argue that there are two main limitations of this dataset. Firstly, given the limited accuracy of automatic methods like back-translation and the labor constraints of human annotation, it is contended that the scale of their generated paraphrases remains somewhat limited. To illustrate, a mere two paraphrases are generated for one of the relations as an instance of their limitation. Secondly, since they leverage a syntax-based search engine SPIKE [48] to search for patterns, the lexical diversity of the paraphrase is not guaranteed. That may not be ideal when simulating the true situation when humans query for certain factual knowledge. For example, people may a variety of sentences such as "*[X] was situated in [Y].*", "*[X] could be observed in [Y].*" to ask for the location of a certain item, not only "*[X] is located in [Y].*". [36] claim to extend paraphrases into about 81 paraphrases per relation. However, their datasets are not available as far as we know.

Fortunately, recent breakthroughs in LLMs in NLP make it practical to generate large-scale and high-quality paraphrases with affordable costs. LLMs such as GPT-3.5 and GPT-4 have been shown to have a surprising agreement with human beings after instruction tuning and reinforcement learning with human feedback [37]. This motivates us to construct a large-scale, highly diverse, and comprehensive paraphrased dataset based on LAMA benchmark. Our primary objective is to contribute a more comprehensive dataset that not only poses a challenge but also offers an effective evaluation benchmark for assessing the consistency of LLMs when extracting the inherent knowledge.

3.2 Dataset Creation

Formally, for a specific relation such as 'Captital of', where we extract the factual information given prompt '[X] is the capital of [Y]', we want to generate several paraphrases such as '[Y]'s capital city is [X]', or more complicatedly, '[X] is the administrative center of [Y]'. We construct our paraphrased version of LAMA datasets called **ParaTrex** with the following steps: (1) We began with the patterns provided by LAMA [39]. Here each relation has one prompt template called base-pattern. For example, the base pattern of relation "*Capital Of*" is "[X] is the capital of [Y]." (2) For each relation, we extract its base pattern and the corresponding description of this relation such as "country, state, department, canton or other administrative division of which the municipality is the governmental seat" for relation "*Capital Of*" so as to make the generation more specific. (3) we formulate a meticulously crafted manual prompt,

directing ChatGPT (GPT-3.5) to produce a total of 40 paraphrases. This includes 5 succinct paraphrases, each comprising no more than 7 words, as well as 5 extended paraphrases, each encompassing fewer than 15 words. An illustrative instance of this paraphrase generation process is illustrated in Figure 3.1. (4) Through human inspection, we remove inappropriate paraphrases characterized by excessive ambiguity or excessive similarity to preceding generations. (5) We execute steps 3 and 4 iteratively until satisfying answers are achieved. We ensure that for each relation, we have at least 25 paraphrases, 5 short paraphrases less than 7 words, and 5 long paraphrases less than 15 words. Furthermore, we introduce a random division of our paraphrases into two distinct sets: an in-domain set comprising 50% of the entire dataset, and an out-of-domain set constituting the remaining 50% of the original data. Notably, the out-of-domain set encompasses all long and short-version paraphrases. This is because we want to simulate the situation where individuals seek to extract specific knowledge by inputting a concise query or an exceptionally long query for seeking specific knowledge. We provide an example of a specific relation 'Capital of' in ParaTrex in Appendix A.1.

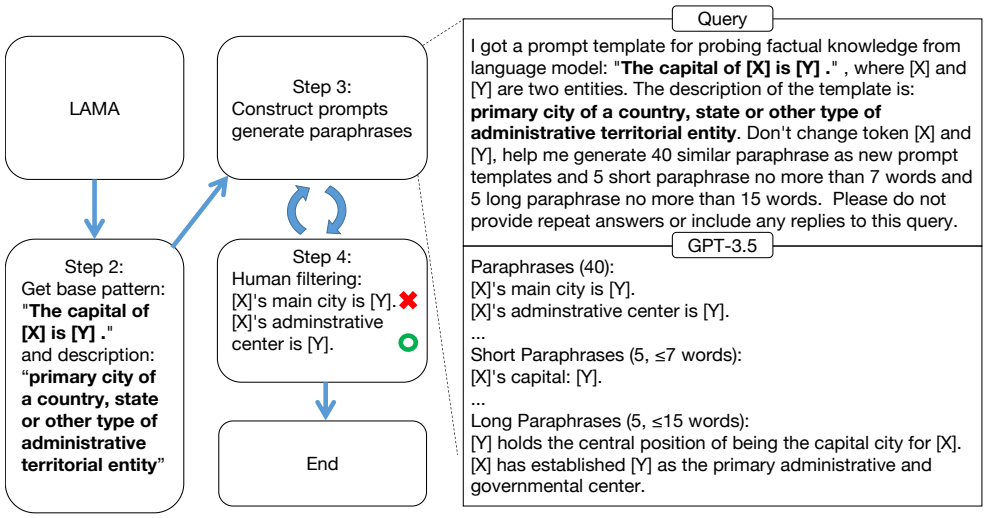

Figure 3.1: Illustration of our workflow to generate ParaTrex, a paraphrased version of prompt templates. Here we present a specific relation 'capital of' in LAMA [39].

3.3 Dataset evaluation

In this section, we perform an evaluation of ParaTrex, our proposed paraphrased version of the LAMA dataset and compare them with the existing datasets ParaRel [12]. We follow [12] to carry out the evaluation based on both automatic metrics and human judgment. Overall, the statistics of ParaTrex and ParaRel are illustrated in Table 3.1.

	ParaRel [12]	ParaTrex(Ours)
# Relations	39	40
# Patterns	329	1544
Min # patterns per rel.	1	27
Max # patterns per rel.	20	47
Avg # patterns per rel.	8.3	38.6
Avg lexical per rel	5.73	8.42

Table 3.1: Statistics of ParaRel and ParaTrex.

3.3.1 Automatic evaluation

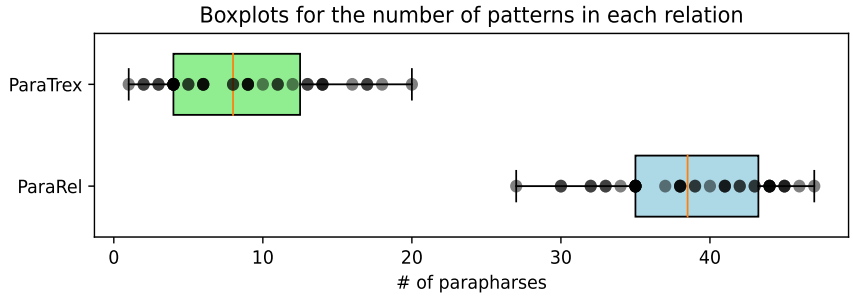

Figure 3.2: Boxplot of the size comparison between ParaRel. ParaTrex contains 40 relations in ParaTrex and ParaRel consists of 39 relations in total. It shows that the size of data in ParaTrex is far larger than in ParaRel.

Size. We first report the size of our generated dataset. We count the number of generated templates in each relation and show the boxplot for the comparison between ParaRel [12] and ParaTrex in Fig 3.2. Generally, we can observe that the average of templates in our dataset is approximately 4.5 times bigger than ParaRel. Furthermore, our dataset consists of more relations and exhibits a more extended average lexical content within the templates, as detailed in Table 3.1.

Diversity. We then illustrate the diversity of our proposed ParaTrex. Specifically, we first listed all pair-wise permutations of n templates for each relation, getting n(n-1) sentence pairs. Then pair-wise n-gram BLEU score [38] was calculated on these pairs to evaluate their diversity. BLEU is an automatic score widely used for evaluating the similarity between the target and reference sentences among machine translation. It measures the precision of the n-gram span in target sentences and that in reference sentences. Given that the sentence pairs have similar semantics (evaluated by humans in the next section), the average score of the lower-order n-gram score tends to represent lexical diversity more and the average score of the higher-order n-gram score tends to capture the diversity of complex syntactic structures. Fig 3.3 shows the trend over

n-gram average pairwise BLEU scores of all relations. Here we omit the n-gram order greater than 4 since the value becomes too tiny to observe. We find that the BLEU scores of ParaTrex perform consistently lower than ParaRel, which depicts that our proposed dataset has a better lexical and syntactical diversity of generated sentences.

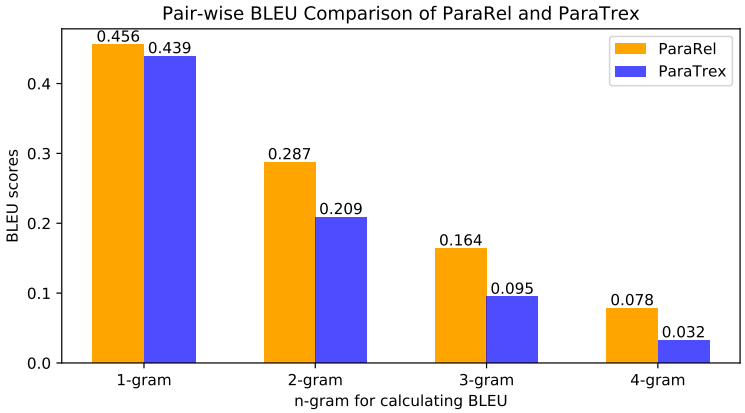

Figure 3.3: Bar chart of the pair-wise BLEU comparison between ParaRel. The scores are the average BLEU scores between all relations. ParaTrex gets a consistently lower score than ParaRel, representing that the templates in ParaTrex are more lexically and syntactically diverse.

3.3.2 Human evaluation

In addition, we conduct a human evaluation study to rate the quality of our generated paraphrases. Specifically, we examine whether the meaning of the raw inputs remains the same as the generated outputs. Due to the restriction of funding and time, we recruited five judges with diverse backgrounds to evaluate our datasets[1]. Following [12], we randomly picked 82 paraphrases in the ParaTrex dataset and 42 wrong paraphrases sampling from the paraphrases of wrong relations. We ask the evaluators to **select candidates that are not the paraphrase of the given inputs**. The participants need to pick out the wrong paraphrases. We consider the remaining answers as what they think to be the correct paraphrases of the given inputs. In Appendix A.2, we show the questionnaire designed for evaluating our datasets. Results show that on average, human evaluators get 96.88% accuracy in successfully identifying inaccurate paraphrases and a 92% accuracy in selecting the true paraphrases provided by ParaTrex, which shows that our proposed datasets have a satisfying agreement with human beings, thus proving the favorable quality of our datasets.

[1]Those bilingual speakers excluded the author and included four MSc students in the University of Edinburgh and one MPhil student majoring in Applied Linguistics in the University of Cambridge.

Chapter 4

Task definition

In this chapter, we will provide a formal definition of our tasks. Section 4.1 will introduce the specific definition of the problems we want to solve/mitigate. In section 4.2, we elaborate on the details of the used datasets. Finally, in section 4.3, we will carefully discuss the evaluation metrics we employed for measuring our models.

4.1 Problem definition

We first give a formal definition of the three main parts we want to focus on in this thesis, which are factual knowledge extraction, prompt preference bias, and prompt verbalization bias.

Factual Knowledge Extraction. Let $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$ be a set of entities and $\mathcal{R} = \{r_1, r_2, \cdots, r_n\}$ be a set of relation. A knowledge graph(KG) is made up of triples (subject, relation, object) denoted as (e_i, r_j, e_k), where $e_i, e_k \in \mathcal{E}$ are subject and object entities and $r_j \in \mathcal{R}$ is the relation. Factual knowledge Extraction aims to extract such triples within LLMs $\mathcal{M}$. Specifically, we let $\mathcal{M}$ make predictions based on incomplete triples $(e_i, r_j, ?)$ after being converted into natural language queries through a designed prompt template $\mathcal{P}_j$. We denote the converted query by $\mathcal{P}_j(e_i)$. For instance, suppose we want to query the profession of Obama (i.e. (*Barack Obama*, *profession*, ?)). We achieve this by converting the template of relation "*profession*", which is "*The profession of [X] is [Y]*" into "*The profession of Barack Obama is <mask>*" by replacing [X], [Y] tokens with the subject and <mask> respectively. The masked token is then predicted by LLMs based on their output probabilities and the top predictions are adopted as the

answer $\hat{e}_k$. Mathematically:

$$\hat{e}_k = \underset{o}{\operatorname{argmax}} P_{\mathcal{M}}(o|\mathcal{P}_j(e_i)) \tag{4.1}$$

$$o = \mathcal{M}(\mathcal{P}_j(e_i)) \tag{4.2}$$

where o is the generated word, a random variable whose distribution is estimated by an LLM $\mathcal{M}$. Our task is to let the generated answer $\hat{e}_k$ be close to the reference answer e_k. We apply the top 1 hit rate and mean reciprocal rank (MRR) to evaluate the quality of our knowledge extraction (see Section 4.3.1 for details).

Prompt Preference bias. Based on [53]'s definition of object bias, for target triples (subject, relation, object), prompt Preference bias refers to the phenomenon that: (1) LLMs with prompts retrieve object candidates unequally when only subject-masked prompt $\mathcal{P}_j(<mask>)$ is given. (2) The divergence between the distribution generated by the subject-masked prompt and the normal prompt becomes too small, which means that the model relies too much on the prior distribution provided by the prompt templates. For example, given the template "*The native language of [X] is [Y].*", the model prefers "*French*" to "*English*" when subject [X] is not assigned, which further makes the model tend to predict "*French*" when given subject such as "*J.K. Rowling*". Our task to mitigate such bias is to both maximize the divergence between subject-masked prompts and smooth the prior object distribution of LLMs' output through subject-masked prompts. We use three measurements for this task, which are counterfactual hit rate, counterfactual entropy, and the Kullback–Leibler(KL) divergence. These will further be explained in section 4.3.2.

Prompt verbalization bias We follow [12]'s definition of consistency as prompt verbalization bias. Formally, prompt verbalization bias refers to the inconsistency of the model in responding to semantically similar prompts, stemming from the model's potential favor of specific verbalization for various prompts. We define a model to be inconsistent when, given a pair of quasi-paraphrased cloze-phrases like "*Seinfeld originally aired on [MASK]*" and "*Seinfeld premiered on [MASK]*", it produces logically conflicting predictions for N-1 relationships across an extensive array of entities such as NBC and ABC. We define the model predicting both NBC and ABC for the aforementioned patterns as lacking consistency due to the contradiction of these two words. Notably, consistency does not require strict factual accuracy, although factual correctness remains an essential attribute for knowledge bases (KBs). We, therefore, measure them separately, which will be discussed in section 4.3.3. Our task to alleviate prompt verbalization bias is

Dataset	Relation	Query	Answer
T-REx[39]	P1412(Languages spoken)	Carl III used to communicate in [MASK].	Swedish
	P19(Place of birth)	Francesco Bartolomeo Conti was born in [MASK]	Florence
	P176(Manufacturer)	iPod Touch is produced by [MASK].	Apple
LM-KBC[49] [1]	CountryOfficialLanguage	The official language of Philippines is [MASK].	Filipino, English
	PersonInstrument	Chris Daughtry plays [MASK], which is an instrument.	guitar
		Bang Yong-guk plays [MASK], which is an instrument.	-
ParaRel[12]	P1412(Languages spoken)	Carl III used [MASK] to communicate.	Swedish
		Carl III communicated in [MASK].	Swedish
		Carl III typically used [MASK] to communicate.	Swedish
ParaTrex	P1412(Languages spoken)	Carl III employed [MASK] for communication.	Swedish
		Carl III spoke [MASK] for their communication needs.	Swedish
		Carl III engaged in communication through [MASK] as their primary language.	Swedish

Table 4.1: Examples of each dataset. We give 3 examples for each dataset we used within our experiments. The first and second rows illustrate two knowledge-probing datasets used for measuring extraction accuracy. The third and fourth rows show the instances from the paraphrased version of T-REx dataset, which is employed on measuring the consistency of models.

therefore to make LLMs give identical predictions given different quasi-paraphrased cloze-phrases prompts.

In summary, we introduce task formulations for factual knowledge extraction, prompt preference bias and prompt verbalization bias. Since probing factual knowledge from LLMs requires both precision and robustness [1], the final goal of this project is to develop a unified framework for unbiased, robust, and precise factual knowledge extraction. That means to both give precise predictions and remain robust on counterfactual subject-masked inputs as well as semantically similar paraphrases.

4.2 Datasets

For assessing the knowledge in LLMs, lots of benchmarks have been proposed for probing knowledge contained in LLMs. For example, linguistic knowledge [14] [54], syntactic knowledge [7], factual knowledge [39] [20] [22], and commonsense knowledge [57]. Here we introduce two basic benchmarks we use for assessing our models and two paraphrased extensions we use for measuring the consistency of our models. Specific examples for each dataset are shown in Table 4.1.

LAMA [39]. LAMA (LAnguage Model Analysis) probing [39] is the first dataset invented for testing the factual and commonsense knowledge in language models. It

provides a set of knowledge sources that are composed of a corpus of facts. Each fact is converted into a clozed-phrase statement which is used to query the language model, as is shown in Table 4.1. Here we use the T-REx knowledge source, which is a subset of Wikidata triples derived from the T-REx dataset [13]. It contains a total of 34039 facts for 41 relations. To make the consistency results comparable, we follow [12] and remove all N-M relations when calculating consistency between models (31 relations remained).

LM-KBC [49]. Although much follow-up work reporting further improvements of models for factual knowledge extraction using LAMA dataset [32] [53] as well as criticism recently [4] [5] [23] [24] [53], it's worth noting that these studies do not extend their results on the common other datasets except LAMA T-Rex. Based on the need for broader evaluations across diverse datasets, we, therefore, use additional latest datasets called LM-KBC(Knowledge Base Construction from Pre-Trained Language Models) to extend our results. LM-KBC is a challenge at the 21st International Semantic Web Conference (ISWC 2022). This challenge exhibits a parallel pattern and task formulation when compared to LAMA T-Rex while being more complicated and challenging. The key difference is that LM-KBC made no assumptions on relations cardinalities, which means that a subject entity could stand in relation with zero, one, or many object entities as shown in Table 4.1. This dataset consists of 12 relations, each comprising 100 subjects for training and 50 samples as validation sets.

ParaRel [12]. ParaRel was a paraphrased version of LAMA designed for measuring the probing consistency of LLMs, consisting of 38 relations over 328 distinct paraphrases on the relation in LAMA T-REx. It is constructed using paraphrased from LPAQA [21] and the syntax-based search engine SPIKE[48] to augment the original prompt template. Although the scale of the dataset is not large, it takes the first step in constructing paraphrase datasets on factual knowledge extraction tasks and shares a high agreement with human evaluations. Three specific examples in ParaRel for relation "*Language spoken*" are shown in Table 4.1, where we can observe that most paraphrases are syntax-based. We argue that this dataset can further be enhanced with both more lexical diversity and quantities, which motivates our construction for ParaTrex.

ParaTrex. We construct ParaTrex, another paraphrased version of LAMA, with the primary objective of providing a dataset with greater diversity and complexity on measuring the consistency in the knowledge probing task. The construction details can be seen in chapter 3. ParaTrex is formed by a total of 1544 facts from 40 relations, with greater lexical can syntactical diversity than ParaTrex and good human agreements.

4.3 Evaluation Metrics

4.3.1 Accuracy measurement

We use the top-1 hit rate(**Hit@1**) and mean reciprocal rank (**MRR**) to evaluate the accuracy of our extracted results. Hit@1 measures the accuracy of the answer provided by LLMs with the highest probability and MRR measures the average reciprocal rank of the golden answer in the output distribution of LLMs. Specifically for each relation consisting n_i number of samples:

$$\text{Hit@1}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbb{I}(\hat{e}_{ij} = e_{ij}) \tag{4.3}$$

$$\text{MRR}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\frac{1}{rank_j} \tag{4.4}$$

, where n_{rel} is the number of relations in the dataset, n_i refers to the samples in a specific relation i, $\mathbb{I}()$ is an indicator function which outputs 0 if the condition in the bracket is satisfied otherwise 0. Here $\hat{e}_{ij}$ and e_{ij} refer to the output prediction and golden-truth entity respectively the same as equation 4.1. $rank_j$ is the rank of the golden-truth entity within the output distribution of LLMs. Notably, the hit rate metric focuses on the performance of models on retrieving the most confident results while MRR gives a more general overall evaluation on the retrieving performance of our model.

4.3.2 Prompt preference bias measurement

For measuring Prompt preference bias, we use three matrices: **counterfactual hitting rate**, **counterfactual entropy**, and the Kullback–Leibler divergence (**KL divergence**). Counterfactual hitting rate and counterfactual entropy are evaluated based on the counterfactual subject-masked input. This metric intuitively shows the abnormal accuracy of the models' prediction based on the subject-masked dataset. The counterfactual entropy is calculated by the entropy of the probability among the first K predictions. This measurement captures, more generally, the extent to which the model exhibits bias towards candidate objects without giving the key information of the subject. We choose the K to be 10 here following [53] since based on manual observations, the first 10 outputs are unlikely to include irrelevant candidates such as stopwords. Specifically, for the i-th relation with n_i number of samples:

$$\text{Counter Factual Entropy}_i = -\frac{1}{n_i}\sum_{j=1}^{n_i}\{\sum_{k=1}^{K} p(\hat{e}_{ik})\log_2 p(\hat{e}_{ik})\} \tag{4.5}$$

Besides, note that our definition of prompt preference bias also comprises the divergence between subjected-masked bias and subject-unmasked bias, which indicates the degree to which the output distribution is dominated by the prompts. We also employ the KL divergence between raw and counterfactual inputs as a measurement:

$$\mathrm{KLD}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\{\sum_{k=1}^{n_{vocab}} p(\hat{e}_{ik})\log\frac{p(\hat{e}_{ik})}{p_{counter}(\hat{e}_{ik})}\} \tag{4.6}$$

, where n_{vocab} is the vocabulary size of the model and $p_{counter}(\hat{e}_{ik})$ is the output probability of entity e_{ik} from the subject-masked inputs.

4.3.3 Prompt verbalization bias measurement

Following [12], we measure prompt verbalization bias through the **consistency** among different paraphrases. The Consistency measures the proportion of pairs of prompts where the model makes the same prediction. Formally, given a set of unordered paraphrase pairs P_i of relation r_i, consisting of n distinct prompts. We then have totally $\frac{1}{2}n(n-1)$ number of permutations. For the j-th sample in the i-th relation, we define the consistency between all paraphrases (**All-Consist**) as:

$$\text{All-Consist}(\text{Top-1},\mathrm{P}_i)_j = \frac{\sum_{p_m,p_n\in P_i}\mathbb{I}[\hat{e}^m_{ij}=\hat{e}^n_{ij}]}{\frac{1}{2}n(n-1)} \tag{4.7}$$

, where $\mathbb{I}$ is the indicator function, $\hat{e}^m_{ij}$ and $\hat{e}^n_{ij}$ refer to the predicted entity given by LLMs from prompt p_m and p_n respectively. For the reason of simplicity and intuition, we also consider the combination of the unique raw prompt template from LAMA, and templates from paraphrased LAMA $p_m \in P_i$, getting n combinations in total. The consistency between raw prompts and paraphrased prompts(**Raw-vs-para-Consist**) will be degraded to:

$$\text{Raw-vs-para-Consist}(\text{Top-1},\mathrm{P}_i)_j = \frac{\sum_{p_m\in P_i,p}\ \mathbb{I}[\hat{e}_{ij}=\hat{e}^m_{ij}]}{n} \tag{4.8}$$

Besides, as mentioned in section 4.1, previous consistency does not require strict factual accuracy. However, factual correctness remains a crucial attribute for KBs. We, therefore, additionally measure the consistency over factual correct prediction and refer to it as **Acc-Consist**. Formally:

$$\text{Acc-Consist}(\text{Top-1},\mathrm{P}_i)_j = \frac{\sum_{p_m,p_n\in P_i}\mathbb{I}[\hat{e}^m_{ij}=\hat{e}^n_{ij}=e_{ij}]}{\frac{1}{2}n(n-1)} \tag{4.9}$$

To sum up, we employ **Raw-vs-para-Consist**, **All-Consist**, and **Acc-Consist** as consistency measurements to have a comprehensive evaluation of the consistency of LLMs.

Chapter 5

Methodology

In this chapter, we show the details of the comparison models and the architecture of our proposed models. Our baselines include P-tuning [32] and MeCoD [53], which are expounded upon in section 5.1. We will show the holistic framework of our proposed method in section 5.2, encompassing two distinct components aimed at mitigating prompt preference bias and prompt verbalization bias respectively.[1]

5.1 Comparision models

We choose the following two baselines because of their great contribution to improving the accuracy of factual knowledge extractions and mitigating biases within the factual knowledge extraction tasks respectively.

P-tuning [32] . P-tuning can be one of the most representative works of tuning soft-prompt for extracting knowledge from LLMs. Unlike Prefix-tuning [29] and prompt-tuning [47] which freezes part of the inputs embedding, P-tuning extends the prompt searching space by making all embeddings of inputs tunable except the subject and object mask. For instance, the traditional prompt template for probing capital knowledge serves as "The capital of Britain is [MASK]" while the input of P-tuning is "h_0 h_1 h_2 capital Britain h_3 $\cdots$ h_i [MASK].", where all h_i is learnable prompts tuned by an inherent bi-LSTMs. P-tuning shows significant improvements against other prompting methods such as manual prompt or discrete prompt searching. However, from our implementation, the performance of P-tuning still has a gap compared with fine-tuning the whole LLMs. This inspires us to find a new tuning method

[1] In this chapter, some explanations are constructed using descriptions from the students' Progress Report and Informatics Project Proposal.

MeCoD. [53] MeCoD was proposed for the purpose of mitigating object bias when probing knowledge. They focus mainly on relieving prompt preference bias through maximizing the entropy of the output distribution. Their key novalty is based on the idea that only relevant object should be taken into consideration, such as the city objects when asking for the capital. They inserted a tiny multiple-layer perceptron(MLP) as a binary classifier for each output candidate and let it automatically decide whether the object is relevant to the answer. After filtering irrelevant objects through MLP, they use the following two methods to debias the prompt preference bias: (1) maximize the entropy of the distribution of the remaining objects to force the distribution close to a uniform distribution, (2) use contrastive learning loss to push the output embedding away from the biased object and pull the embedding close to the ground-truth embedding. Their method reaches the latest **SOTA** as well as significantly mitigating the prompt preference bias. However, they do not relieve the prompt verbalization bias. In addition, with our implementation, we find that the inserted MLP does not actually work for classifying the relevant candidates. It picks only a tiny part of all relevant candidates. Therefore, there still exists potential improvements based on MeCoDs.

5.2 Overall Framework of proposed models

Here we proposed a unified adapter-based framework that can alleviate both the prompt preference bias and prompt verbalization bias within the factual knowledge extraction tasks, named Uni-Arkex. Our goal is to (1) improve the accuracy of the extraction results from LLMs and (2) make the model less suffer from the domination of prompt and its inconsistency. The basic idea is simply to leverage augmented data on accomplishing separate tasks so that our model no can perform well on both sides. We will explain our methods in the following three parts. The overall architecture of our method is shown in Fig 5.1.

5.2.1 Improve the accuracy of factual extraction

To make further improvments on the accuracy of factual extraction, we choose adapter-based tuning methods, which is a more powerful parameter-efficient fine-tuning method compared with prompt tuning. Adapter tuning is also shown to have better performance over fully fine-tuning when we do not have access to a large scale of data [6]. Moreover, adapter-based tuning is also shown to have a good performance on multi-task

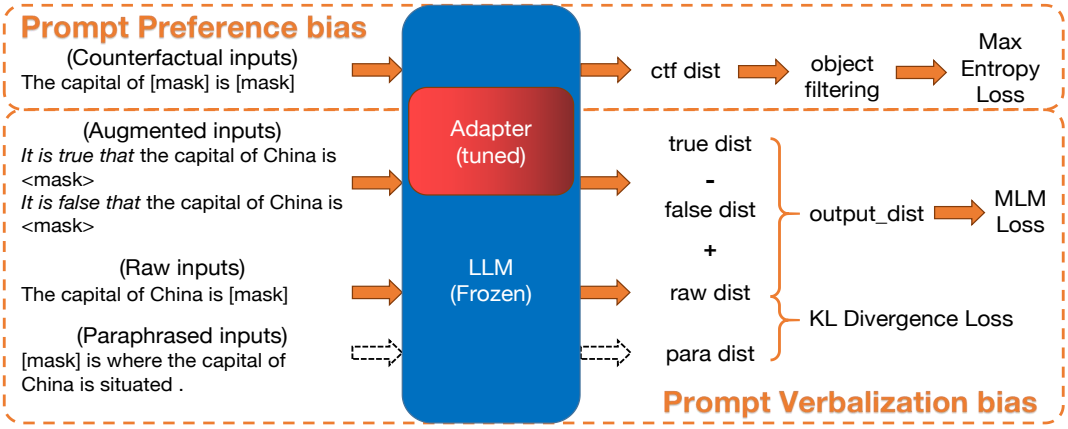

Figure 5.1: Overall architecture of the proposed Unified Adapter-based framework of unbiased and robust factual Knowledge Extraction (Uni-Arkex). In the figure, "dist" refers to the output distribution of candidate objects from LLM. The dashed arrow means an optional module when paraphrased inputs are not provided.

settings[11]. It also has other advantages such as preserving the internal knowledge within LLMs by freezing all of the parameters. Here in our models, we simply insert an adapter before each feedforward network (FFN) in each transformer layer. Specifically, for each input $\mathbf{h} \in \mathbb{R}^d$, our adapters make the following transformation:

$$\mathbf{h} \leftarrow \mathrm{GELU}(\mathbf{h}\mathbf{W}_d)\mathbf{W}_u + \mathbf{h} \tag{5.1}$$

where GELU[18] is a non-linear activate function, $\mathbf{W}_d \in \mathbb{R}^{d\times k}$ and $\mathbf{W}_u \in \mathbb{R}^{k\times d}$ are two learnable parameter matrix in adapters. They are used for first down-projecting the hidden states into dimension $k<<d$, and then projecting them back to d-dimension spaces. Here k is a hyperparamter.

5.2.2 Alleviating prompt preference bias

According to the problem definition in section 4.1, for subject-masked prompts, the unbiased output distribution should ideally satisfy the following two results (1) the output probability of relevant candidates should be equal (2) the KL divergence of distributions between subject-masked prompts and original prompt should not be too small. For (2), we find that it can be hard to formally define a certain threshold of small KL divergence. We also don't want the KL divergence to be as large as possible because it is not necessary to ignore all valuable prior knowledge brought by prompt templates. We, therefore, choose to design an object only on optimizing problem (1). We also

report the KL divergence when making an evaluation to see if the models work in the way we expected. Specifically, similar to [53], we construct an additional loss L_{me} to maximize the entropy over all relevant candidates in order to encourage the model to assign equal probability to each relevant candidate. Here an object filtering process is necessary since not all objects are related and need to be smoothed. For example, suppose we have prompts:"The capital of [X] is [MASK]", we prefer the model to have an equal probability over output city entities but not stopwords like "*and*". In practice, we simply choose the top 300 candidates for BERT and 30 candidates for Roberta and remove the stopwords among them. This is because, from our empirical observation, the top k words include most of the relevant candidates except some common stopwords. Formally, given the output probability of object i: $p(i), i = 1, 2, \cdots, k$ and the stopwords set S, the max entropy loss is calculated by:

$$L_{me} = -\sum_{i=1,\ i\notin S}^{k} p(i)\log_2(p(i)) \tag{5.2}$$

It is worth noting that in [53], a simple multiple-layer perceptron is used as a classifier to automatically choose the relevant candidates. However, based on our re-implementation, we find that this MLP only picks a tiny part of all relevant candidates. Their neural classifier lacks interpretability on the criteria for filtering objects. In addition, this MLP will bring additional costs on both training and inference. We will show later that simply removing stopwords instead neural classifier can have the same or even better performance than [53]

5.2.3 Alleviating prompt verbalization bias

We mitigate prompt verbalization bias through paraphrased inputs and augmented inputs. As for paraphrased inputs, we choose the same strategy as [12], which minimizes the kl divergence of the distribution between the raw input and paraphrased input, specifically:

$$L_{kld} = \sum_{i=1}^{n_{vocab}} p(i)\log\frac{p(i)}{p_{para}(i)} \tag{5.3}$$

where $p_{para}(i)$ is the probability of candidate object i from the paraphrased inputs. Here in order to constraint the training and inference time, we randomly pick two paraphrases among the in-domain paraphrased dataset and average their KL divergence loss when training.

Although optimize the kl divergence explicitly through paraphrased input is helpful, it's worth noting that real-world applications may often suffer from a lack of diverse

and high-quality paraphrases. That motivates us to propose a self-augmentation method through the raw inputs. We augment our raw data with prefix "*It is true that*" and "*It is false that*" and encourage the model's self-consistency by combining there output distribution to make final predictions. Specifically, the output probability $p_{o_m}(i)$ for object candidate i and the masked language model(MLM) loss L_{mlm} are calculated by:

$$p_o(i) = \text{Softmax}(logit(i) + \lambda_{aug}(logit_{true}(i) - logit_{false}(i))) \tag{5.4}$$

$$L_{mlm} = -\sum_{i=1}^{n_{vocab}} y(i) \log p_o(i) \tag{5.5}$$

where logit refers to the logit of output before softmax layer, $logit_{true}$ and $logit_{false}$ are logits from prompt with prefix 'It's true' and 'It's false'. λ_{aug} is the hyperparameter used for controlling the impact of augmented inputs.
During training, the model is optimized by jointly minimizing the following loss:

$$L = L_{mlm} - \lambda_{me} L_{me} + \lambda_{kld} L_{kld} \tag{5.6}$$

where λ_{me} and λ_{kld} are hyperparameters used for balancing the impact of three losses. During the process of inference, we note that the input of paraphrased inputs is not necessary(shown in the dashed arrow in Fig 5.1). This is designed to make our models more general in cases lacking diverse and high-quality paraphrased queries.

Chapter 6

Experiment and results

This chapter introduces the experiment setups for comparison between models and related results. In section 6.1, we first explain the details of our implementation of both models and evaluation pipelines. For section 6.2, we show and discuss the main results of the experiment. Section 6.3 displays our ablation study on each proposed module in our models. In section 6.4, we perform a specific case study on our models, and in section 6.5, we make discussions on other remaining problems and give a summary of all experiments.

6.1 Experiment setup

We will introduce our preprocessing for background datasets introduced in section 4.2 and the detailed implementations and hyperparameters of our models in this section.

6.1.1 Datasets

We adopted the LAMA-TREx [39] as our main testing set using its official train-test splits, comprising 41 relations and 29,500 testing triples in total. In addition, we expand our experiments in LM-KBC challenge datasets [49], which totally includes 12 relations. It's important to note that we exclusively use 6 out of the 12 relations among LM-KBC official datasets within our experiment. This is driven by the fact that most of the objects in other relations either exhibit in multiple terms, which is not supported by the mask language models or cannot be properly tokenized with BERT or RoBERTa tokenizers, which may potentially lead the model to overfit by predicting "[UNK]" token regardless of inputs. Besides, as the testing sets remain private in the

LM-KBC challenge, we split 50% of the development set to create test sets following the work from [28], which shows the agreement between this splitting method and the official testing set. For paraphrased version datasets, we use our proposed ParaTrex and ParaRel[12] to train and measure the consistency of our models. We note that N-M relations are omitted because measuring consistency when there are several correct answers can be hard. Among the remaining 25 relations, we split 50% of paraphrased templates as out-of-domain templates, which is not seen by models during the training phases and constructing 3 settings: (1) In-Domain(**ID**): where all prompts and their paraphrases are seen by models during training phases. (2) Out-of-Domain(**OOD**): where all prompts are unseen by the models. This setting is designed for simulating situations when LLMs receive unseen queries for factual knowledge by humans in real life. (3) Pararel(**PR**): where we use Pararel datasets[12] as an out-of-domain datasets. We apply three consistency evaluation matrices (explained in section 4.3) for each setting.

6.1.2 Model implementation details

For LLMs in our experiments, we employ BERT-large[10], and RoBERTa-large[33] as our based models. We discuss the scalability of our methods through the comparison between BERT-base/RoBERTa-base and BERT-large/RoBERTa-large. Due to the time and computing constraints in this project. We leave larger models such as GPT-2[43] and Llama [51] for future works. For P-tuning[32], we follow their default setting in Liu's paper. For our proposed method, we use Adam optimizer [25] with its default configuration and set the learning rate to 1e-5 to optimize the adapters. We set the hidden dimension of adapters to 256 and optimize it freezing all pretrained parameters in LLMs. We set λ_{aug}, λ_{me}, λ_{kl} to be 0.5, 0.2 and 0.2 respectively.

6.2 Main results

In this section, we provide the results of experiments and show the effectiveness of our proposed method by comparing with SOTA models P-tuning [32] and MeCoD[53], which has good performance on accuracy and mitigating prompt preference bias respectively. We compare them in both soft prompt settings and manually designed prompt settings. We evaluate the accuracy, prompt preference bias and prompt verbalization bias(consistency) in LAMA benchmark in section 6.2.1. Notably, we do not evaluate

Accuracy&PP bias		BERT-Large					RoBERTa-Large				
		Accuracy		Prompt preference bias			Accuracy		Prompt preference bias		
Prompt	Method	hit@1	MRR	ct_entropy	ct_hit@1	KLD	hit@1	MRR	ct_entropy	ct_hit@1	KLD
Soft Prompt	P-tuning [32]	0.529	0.624	1.7812	0.1618	2.9546	0.4286	0.5143	1.6798	0.1625	2.0157
	+Adapters	**0.5334**	**0.6309**	1.719	0.14	3.3961	**0.5085**	**0.6085**	1.786	0.1582	2.7552
	+MeCod [53]	0.5303	0.6288	2.2386	0.0092	8.884	0.47	0.5713	2.0721	0.0567	5.2588
	+Uni-Arkex (w/o aug/para)	0.53	0.6249	**2.2932**	**0**	**13.618**	0.5024	0.6021	**2.2863**	**0.017**	**11.0267**
Manual Prompt	LAMA [39]	0.3445	0.4106	1.8408	0.0419	3.5449	0.2255	0.2709	1.9661	0.0523	1.7514
	Fine-Tune	0.5351	0.6293	1.3957	0.1338	5.063	0.5109	0.6096	1.6507	0.1736	3.1701
	Adapters	**0.546**	**0.6411**	1.6027	0.142	4.129	**0.5257**	**0.6233**	1.8021	0.1833	2.5435
	+MeCod(OI)	0.5432	0.6389	**2.2969**	0.0087	6.1119	0.5195	0.6173	**2.2801**	**0.0128**	4.5231
	Uni-Arkex	0.5439	0.6381	2.2698	**0.0023**	**13.3672**	0.5246	0.6223	2.1119	0.0403	**10.5302**

Table 6.1: Main results for accuracy and prompt preference bias on LAMA benchmarks. We use P-tuning [32]'s soft prompts as initialization in the soft prompt settings. In this setting, we do not add augmentation and paraphrase modules in our proposed model since all prompts are automatically optimized. In manual prompt settings, MeCoD(OI) is our adapter-based re-implementation of [53] based on the manual prompt. In each group, the best score is marked bold and the second-best result is underlined.

the consistency for LM-KBC benchmark because of lacking the paraphrased datasets. So we only provide results for accuracy and prompt preference bias of the LM-KBC benchmark in section 6.2.2. For both benchmarks, we report soft prompt settings and manual prompt settings. We note that for soft prompt settings, we use our proposed Uni-Arkex without the augmenting and paraphrasing module since all prompts here are learnable continuous vectors, which is to some extent different from natural language. Besides, in the manual prompt settings, we use our implemented adapter-based version of MeCoD [53] because (1) we want to make a fair comparison among adapter-based models and (2) the implementation in [53] is based on the pre-trained soft-prompts, which does not support manual prompt inputs well.

6.2.1 Results for LAMA

Table 6.1 shows the experiment results for average accuracy and prompt preference bias among all relations. Full results including all relations are provided in Appendix B.

We first focus on the comparison between four different downstream tuning methods: P-tuning, adapter-tuning, manual prompt(LAMA), and fine-tuning. We conclude that (1) adapters outperform all other tuning methods in accuracy and (2) all traditional tuning methods suffer from prompt preference bias. Firstly, observing the columns

Consistency		ID_raw	ID_all	ID_acc	OOD_raw	ood_all	ood_acc	PR_raw	PR_all	PR_acc
BERT -Large	LAMA	0.3358	0.2837	0.1576	0.2764	0.253	0.1449	0.5494	0.4661	0.2504
	Adapters	0.6092	0.5341	0.3909	0.5296	0.4903	0.3576	0.7212	0.6523	0.4581
	+MeCod (OI)	0.6339	0.5648	0.4124	0.5648	0.5276	0.3847	0.735	0.6733	0.4715
	Uni-Arkex	**0.6841**	**0.6222**	**0.4443**	**0.6185**	**0.5796**	**0.4188**	**0.7642**	**0.7098**	**0.4957**
RoBERTa -Large	LAMA	0.2393	0.2061	0.008	0.1965	0.1709	0.0059	0.33	0.2828	0.0047
	Adapters	0.6185	0.5524	**0.0132**	0.564	0.5029	**0.0116**	0.6691	0.6044	0.0064
	+MeCod (OI)	0.6166	0.5482	0.0131	0.5641	0.5025	0.0115	0.679	0.612	**0.0065**
	Uni-Arkex	**0.6614**	**0.6106**	0.0099	**0.6137**	**0.5654**	0.0088	**0.7168**	**0.6596**	0.0045

Table 6.2: Main results for Consistency on ParaTrex(Ours) and ParaRel[12] benchmarks. LAMA here refers to the manual prompt template given by LAMA benchmark [39]. Here MeCoD is our implementation of the manual prompt. In each group, the best score is marked bold and the second best result is underlined.

of accuracy, we find that adapter tuning performs significantly better than any other tuning methods evaluated by top 1 precision and MRR in both soft prompt and manual prompt cases. Fine-tuning and P-tuning get similar scores and outperform manual prompts significantly. This shows that tuning is necessary for extracting specific knowledge from LLMs. Adapters stand out possibly because of their advantages in both preserving the internal knowledge of LLMs and accessing the reasoning process between LLM's internal layers. In contrast, fine-tuning may potentially suffer from catastrophic forgetting due to modifying all parameters within LLMs. P-tuning merely learns optimal prompts before embedding layers and it is not able to improve the model's reasoning process by optimizing the reasoning process through the layers between LLMs. We then report the prompt preference bias over these traditional tuning methods. We find that all of these traditional tuning methods suffer from prompt preference bias. Their low counterfactual entropy and high counterfactual hitting rate indicate that LLMs have the ability to guess certain objects even though they are not given the corresponding subject. The low KL divergence value also denotes that the prediction distribution is dominated by the prompt template instead of the true knowledge within LLMs. Here we find that all tuning methods except manual prompt have obvious lower value at counterfactual entropy and larger value at counterfactual hitting rate, this shows that model tends to link specific prompt templates to some favored objects. In addition, we argue that this bias may come from the tuning process instead of prompt types or pretraining steps. This is based on the observation that the value of prompt preference bias matrices is close to each other across both prompt settings and both BERT and RoBERTa models. This observation motivates our trial on debiasing through additional

loss function during the training process.

We then work on mitigating the bias through our proposed method Uni-Arkex. For soft prompt settings, our method has significant improvements on all prompt preference biases, outperforming current SOTA MeCOD [53]. Notably, it is surprising that the kl divergence over subject-masked counterfactual inputs and raw inputs is not explicitly optimized when training our models, nonetheless, it demonstrates a pronounced increase together with the improvements of counterfactual entropy and counterfactual hitting rate. These synchronous improvements indicate that while maximizing the entropy, the model also tends to learn to make predictions less constrained to the prior distribution given by the prompt templates. Although we still need concrete evidence to show that LLMs employ their inherent knowledge to make predictions, we take a first step to show that LLMs have the ability to make correct factual answers unconstrained by the prior distribution given by the prompt template. Besides, the performance drop in the hit rate is less than 0.005 in all settings, which can be neglectable. In manual prompt cases, our proposed Uni-Arkex can even perform better than the SOTA model MeCoD [53], with both less prompt preference bias and substantially better consistency on prompt verbalization (Table 6.2), which shows the effectiveness of our proposed methods.

The final finding from Table 6.1 is that the trend of accuracy and prompt preference bias are consistent across BERT-Large and RoBERTa large, both showing that Uni-Arkex reduces the prompt preference bias while maintaining good performance from adapter-tuning, which shows that our methods can generalize well between different models.

Table 6.2 shows the prompt verbalization bias measurements. Less prompt verbalization bias means better consistency. Overall for different settings, we can observe the trend: "Out of domain consistency of ParaTrex (OOD) $<$ in domain consistency of ParaTrex (ID) $<$ ParaRel consistency (PR)". These observations precisely agree with our expectations as we show in section 3.3 that ParaTrex is more diverse than ParaTrex. That also explains why on unseen ParaRel paraphrases, the model has better consistency over in-domain settings, which is seen by models during the training process. Within each setting, we can observe the trend: "consistency over accurate predictions(acc) $<$ consistency between all input permutations (all) $<$ consistency between raw inputs and paraphrased inputs (raw)". This is because the difficulty of these three tasks is increasing. As for the results, it is shown that our method performs consistently better in an average 4% percent over other methods, which shows that together with debiasing on prompt preference bias our method does well in relieving prompt verbalization bias

and producing robust results. Note that the consistency among accurate predictions of RoBERTa is abnormally slow, we argue that this is because we do not search for appropriate hyperparameters such as specific learning rate for Roberta. However, we can still observe the same trend in the (all) and (raw) settings.

6.2.2 results for LM-KBC

Table 6.3 shows our extended experiments over LM-KBC [49] datasets. We do not report the measurement of prompt verbalization bias and the Uni-Arkex framework with the paraphrased module due to the lack of paraphrased datasets. Overall, we can observe a similar tendency as in LAMA benchmarks. Specifically, all traditional tuning methods such as P-tuning, fine-tuning, and adapter-tuning suffer from prompt preference bias since we can observe high counterfactual hit rates. Among them, adapters perform relatively better at retrieving accurate answers than P-tuning and LAMA. Our proposed method is capable of both maintaining the accuracy performance of adapters and alleviating the prompt preference bias as all of three prompt preference bias measurements are significantly improved with our Uni-Arkex method. This tendency shares with both two LLMs. However, there exist some differences such as here fine-tuning achieves good scores in Roberta-Large and the accuracy performance of Roberta is considerably greater than BERT-large. We conjecture that this is because of the lack of enough training data since we omit a lot of relations and bad samples during the training process. Despite these differences, it is still obvious that our proposed method is able to effectively mitigate prompt preference bias or even improve the accuracy performance in some cases such as our Uni-Arkex by BERT-Large.

6.2.3 Scaling results for models with different sizes

After analyzing the effectiveness of our model, we then try to investigate whether our model can scale well among different sizes of models. We perform experiments on our proposed Uni-Arkex using BERT-base, RoBERTa-base, and BERT-Large. We compare our results with the adapter-tuning-only results to see whether Uni-Arkex can still have good performance under different scales of models. We report the bar chart of the comparison outcomes in Fig 6.1. Firstly, comparing the results among the same base models such as BERT-base-cased and RoBERTa-base-cased, we can conclude that the trend of performance of accuracy and consistency remains the same. The larger models always get better accuracy and consistency under both adapter-tuning and

LM-KBC		RoBERTa-Large					BERT-Large				
		Accuracy		Prompt Preference Bias			Accuracy		Prompt Preference Bias		
Prompt	Method	hit@1	MRR	ct_entropy	ct_hit@1	kld	hit@1	MRR	ct_entropy	ct_hit@1	kld
Soft Prompt	P-tuning	0.4148	0.4813	1.4923	0.3666	1.2922	0.3663	0.5377	1.7696	0.149	2.4417
	+Adapters	**0.6485**	0.741	1.3019	0.4612	1.2013	**0.3768**	**0.543**	1.8043	0.1882	2.8659
	Uni-Arkex (w/o aug/para)	0.6347	**0.7411**	**2.2815**	**0.0088**	**8.5797**	0.3697	0.5367	**2.2989**	**0**	**13.0276**
Manual Prompt	LAMA	0.1535	0.2341	1.9261	0.0152	1.3608	0.3054	0.4395	1.9733	0.0741	3.781
	Fine-Tune	**0.6391**	**0.7548**	0.8604	0.3035	2.3351	0.3644	0.5436	1.6981	0.1829	2.7415
	Adapters	0.6098	0.7293	1.2258	0.4735	1.3473	0.3708	0.539	2.2554	0.0246	4.3199
	+MeCod	0.5877	0.7105	**2.2612**	0.1181	4.3556	0.3735	0.545	1.9598	0.1784	2.5762
	Uni-Arkex (w/o para)	0.607	0.7298	1.9422	**0.0667**	**7.9745**	**0.3744**	**0.5464**	**2.2726**	**0**	**13.5184**

Table 6.3: Main results for accuracy and prompt preference bias on LM-KBC benchmarks. We use P-tuning [32]'s soft prompts as initialization in the soft prompt settings. In manual prompt settings, MeCoD(OI) is our adapter-based re-implementation of [53] based on the manual prompt. In each group, the best score is marked in bold and the second-best result is underlined.

Uni-Arkex. However, for prompt preference measurements including KL divergence and counterfactual hit rate of the Uni-Arkex model, we observe the inverse trend. This represents that our method tends to work relatively better for smaller models. A potential reason behind this is that smaller models are less robust in extracting factual knowledge, and thus it can be easier for us to modify their extraction results through additional loss functions. Secondly, comparing all results between adapter-tuning and Uni-Arkex, it is possible to conclude that our proposed Uni-Arkex can generalize well among all different sizes of models. Specifically, the bar chart indicates that we have the same or even better accuracy performance compared with adapter-tuning while having incredibly less prompt preference bias and prompt verbalization bias.

6.3 Ablation Study

In this section, we perform an ablation study to test which module of our proposed framework contributes most to improving accuracy, mitigating prompt preference bias or prompt verbalization bias.

Table 6.4 presents the result of our ablation on accuracy and prompt preference bias. We do not ablate the adapter-tuning module here since we have already compared it with other tuning methods and shown its effectiveness in Table 6.1 and Table 6.2. From the accuracy measurements, it is shown that each of our modules does not have

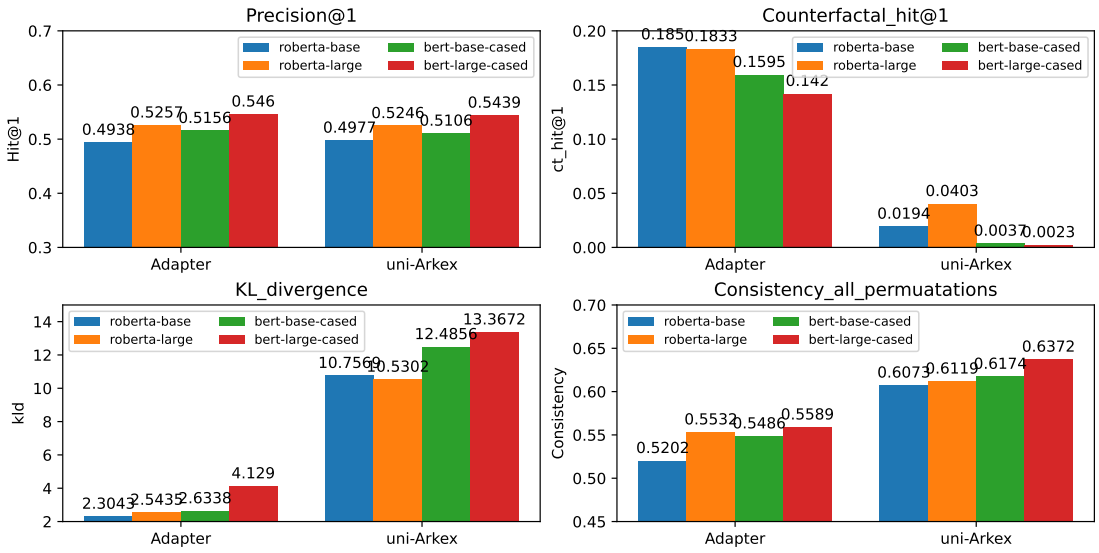

Figure 6.1: Bar chart of our results for both accuracy and bias measurements on models with different scales. Here we want to test whether our previous results can scale as the parameter of models grows. We use BERT-base-cased, BERT-large-cased, Roberta-base, and Roberta-large to simulate the scaling of the model sizes. The take-away message is that our methods remain effective when the size of the model grows. We maintain the accuracy performance while significantly reducing prompt preference bias and prompt verbalization bias.

a significant impact on the probing accuracy. Similarly, for prompt preference bias, our paraphrased and augmentation modules do not make a negative effect on prompt preference bias. This meets our expectations since they are not designed for mitigating prompt preference bias. In contrast, when the max entropy loss is removed, a significant decrease in counterfactual entropy and KL divergence is witnessed. At the same time, the counterfactual hit rate boosts a lot. This represents that the max entropy module is a crucial part of debiasing prompt preference bias

The ablation study for prompt verbalization bias is shown in Table 6.5. We make the following three conclusions based on the results in Table 6.5. (1) The paraphrasing module helps relieve the prompt verbalization bias, but not in the dominant rule. Comparing the first row vs. the second row and the third row vs. the fourth row, we can observe actual but not significant drops in consistency scores. The drops are consistent in out-of-domain settings, representing that the paraphrased inputs module can help models generalize more on unseen data. However, the improvement is not so significant and that's why we make this module an optional choice both for training and inference. (2) The augmentation inputs module takes a crucial role in improving the consistency

	Accuracy		Prompt preference bias		
Method	test_hit@1	MRR	ct_entropy	ct_hit@1	KLD
Uni-Arkex	0.5439	0.6381	**2.2698**	0.0023	**13.3672**
w/o aug & para	0.5448	0.6392	2.2499	**0.0014**	13.2938
w/o me & aug & para	**0.546**	**0.6411**	1.6027	0.142	4.129

Table 6.4: Ablation study of the effects of each our module on accuracy and prompt preference bias.

Consistency	ID_raw	ID_all	ID_acc	OOD_raw	ood_all	ood_acc	PR_raw	PR_all	PR_acc
Uni-Arkex	0.6841	0.6222	**0.4443**	**0.6185**	**0.5796**	**0.4188**	**0.7642**	0.7098	**0.4957**
w/o para	**0.686**	**0.623**	0.4433	0.6128	0.5767	0.4143	0.7631	**0.7113**	**0.4957**
w/o aug	0.6376	0.5673	0.4132	0.5712	0.5232	0.3823	0.7325	0.6668	0.4677
w/o aug & para	0.6305	0.5602	0.4092	0.5595	0.5191	0.3792	0.7284	0.6633	0.4656
w/o me & aug & para	0.6092	0.5341	0.3909	0.5296	0.4903	0.3576	0.7212	0.6523	0.4581

Table 6.5: Ablation study of the effects of each our module on prompt verbalization bias

of our models. This conclusion can be found by comparing the first row vs. the third row and the second row vs. the fourth row, which presents a significant decrease when removing the augmentation module. (3) The max entropy module, which is designed for mitigating the prompt preference bias, is able to help relieve the prompt verbalization bias as well. By observing the last two rows, we notice a considerable drop in performance of consistency when we ablate the max entropy module. This shows that our modules do not take effect separately. They can have synergistic contributions to relieving a certain type of bias.

6.4 Case Study

We perform a case study on the BERT-Large model to make a qualitative analysis of our proposed methods. Firstly, we focus on specific cases on how the models make the correct prediction due to the prompt preference bias. We show two specific cases from relation 'P37', which asks for the official language of a specific item. The prompt template used is: "*The official language of [sub] is [obj].*". The detailed results are shown in table 6.6.

Method	Inputs	Top-5 candidates/logits from (subject: Rwanda, object: French)				
Uni-Arkex	raw inputs	**French**	Rwanda	English	Portuguese	Italian
		21.2631	21.1482	19.3928	16.6334	16.1467
	subject	Georgian	Azerbaijan	Portuguese	Turkish	Myanmar
	masked	11.2759	11.2169	11.172	11.1536	11.1424
MeCoD	raw inputs	Rwanda	Congo	English	Georgian	Cameroon
		15.2095	10.3738	10.119	10.0667	10.0087
	subject	Armenian	Georgian	Azerbaijan	Myanmar	Turkish
	masked	11.9405	11.0925	10.9124	10.7832	10.5903
LAMA	raw inputs	English	**French**	Rwanda	Latin	Mon
		13.9086	13.1605	12.3816	10.0265	10.016
	subject	**French**	Spanish	English	Portuguese	German
	masked	13.5287	13.1813	13.0275	12.4855	12.4292

Table 6.6: Case study on top-5 objects and their logits extracted by LLMs through original prompt template: "*The official language of Rwanda is [MASK].*" and subject-masked prompt: "*The official language of [MASK] is [MASK].*". The bold candidates are the ground truth objects.

The last row in Table 6.6 indicates that for the vanilla LLMs without tuning, the LLM suffers from prompt preference bias on objects such as *French* and *English* make predictions based on this prior distribution. The specific logits of object *French* and *English* of LAMA methods are close to each other, which means that the model is not confident with their predictions, which potentially shows that probably the vanilla model is, to some extent, guessing from the prior distribution. MeCoD [53] is the SOTA model developed for relieving this problem. However, since they apply a neural gate to automatically classify which object to be debiased, the gate may force the model to underfit some objects, which may be harmful. For instance, *French* has a relatively high logit from the vanilla model with subject-masked prompts. MeCoD successfully smooths this high counterfactual logit but causes the model to underfit this object so that it cannot recall the correct object *French*. In contrast, our proposed Uni-Arkex is capable of making accurate predictions while having an unbiased prediction distribution under subject-masked inputs. Moreover, we can observe that our model has far larger logits for prediction than other baselines, possibly meaning that our model makes the prediction more confidently after excluding biased objects through the debiasing process. We, therefore, conclude that our unified unbiased framework is able to debias

Inputs (Subject: Vesanto, Object: Finnish)		Predictions	
Type	Prompt template	Adapter-Tuning	Uni-Arkex
raw	The official language of [X] is [MASK].	**Finnish**	**Finnish**
paraphrased	[X] designates [MASK] as the official language .	Italian	**Finnish**
	[X] has [MASK] as its official language .	It	**Finnish**
	[MASK] has been declared as the recognized language in [X] .	Finland	**Finnish**
	In [X], [MASK] is acknowledged as the prescribed language by the government.	It	Finland
	The officially recognized language in [X] is [MASK] .	Italian	Italian
	[X] recognizes [MASK] as its official language .	Italian	**Finnish**

Table 6.7: Case study on top-5 objects and their logits extracted by LLMs through original prompt template: "*The official language of Rwanda is [MASK].*" and subject-masked prompt: "*The official language of [MASK] is [MASK].*". The bold candidates are the ground truth objects.

on specific objects while not letting the model underfit them, which helps the model output knowledge based on truly understanding relationships between objects and subjects.

Table 6.7 gives a specific example of the consistency study. We provide an instance where adapter-tuning and Uni-Arkex are both correct on original prompts. We sample several cases where our models make correct predictions while adapter-tuning cannot investigate why our models perform better. Based on the shown results, we conclude that our proposed model is more robust over both syntactically and lexically diverse prompt templates. For example, from the second and fourth rows of paraphrased prompt templates, we can observe the different syntax over the raw templates. Our model maintains well on outputting language objects instead of stopwords like 'it'. In addition, for the first and the last rows of paraphrased templates, new terms such as 'designates' and 'recognized' are added to the templates, which are more diverse from the raw inputs. Our model still outputs consistent outputs in this case. Although our model may still make mistakes such as retrieving wrong answers or wrong lexical forms, overall we can observe that it provides consistent and correct results under most circumstances. Therefore, based on these cases, we can speculate that our framework makes the model provide more robust results over syntactical and lexical diverse paraphrases of queries.

6.5 Discussion

In this thesis, we propose a unified adapter-based framework for unbiased and robust factual knowledge extraction. Our ablation study shows that each of our proposed

modules has positive effects on mitigating potential bias when retrieving factual knowledge from LLMs. The maxing extropy module takes a crucial role in relieving prompt preference bias while the self-augmentation module has a vital impact on mitigating the prompt verbalization bias. Surprisingly, we also observe a synergizing effect of maxing entropy modules and self-augmentation modules on prompt verbalization bias, showing the effectiveness of our unified framework. We then carry out a specific case study, qualitatively showing how prompt preference bias and prompt verbalization bias harm the quality of the extracted knowledge. Specifically, we find that our methods improve the model to be (1) more robust among both syntactically and lexically diverse prompt paraphrases and (2) more unbiased in favoring specific objects from the prior distribution from prompt templates. Meanwhile, we do not see any drops in the accuracy performance, which shows that those potential biases can be mitigated without having much impact on the model's performance. This can be explained by the good compatibility of adapters for muti-task learning settings [11]. Despite these improvements, we note that our work is still not performed on larger LLMs such as Llama[51] and GPT [37], which has proved to have emerging capabilities over BERT-large and RoBERTa-large. We will leave them as future works.

Compared with other existing baselines, as far as we know, our framework is the first proposed framework trying to reduce both prompt preference bias and prompt verbalization bias. In addition, there are also no related works on investigating adapter-tuning for factual knowledge extraction. Regarding each module within our framework, our maxing entropy module is similar to MeCoD [53] but we choose a different strategy for filtering the object candidates. In MeCoD, they employ a simple MLP layer to automatically learn to filter the objects. However, we suspect that the black-box MLP layer may not work as we expect since we found that the classification results only choose a small portion among all valid candidates. We also give an example in the case study when MeCoD makes the model underfit the ground truth objects so that the model cannot extract the correct knowledge. Therefore in our methods, we choose to merely remove common stopwords instead of using a neural classifier, which is easier to interpret and simpler to implement. As for the paraphrasing module, we apply the same idea as [12] to minimize the KL divergence between paraphrases. However, this is an optional choice within our proposed framework and we also show in ablation that this method does not contribute to the main improvements of our experiment results.

Chapter 7

Conclusions

In this project, we focus on the factual knowledge extraction tasks that regard large language models(LLMs) as knowledge bases and extract specific knowledge triples <subject, relation, object> via prompt-based methods. We aim to improve the accuracy of existing knowledge probing methods and mitigate two potential biases existing in prompt-based models, which are prompt preference bias and prompt verbalization bias respectively. We summarise the following conclusions we reached in this project:

- We propose an extended large-scale paraphrasing dataset ParaTrex based on LAMA benchmarks in order to make a more comprehensive evaluation of prompt verbalization bias on the existing models. Automatic evaluations show that ParaTrex is more diverse than the existing benchmark ParaRel both lexically and syntactically. We hope that ParaTrex can make valuable contributions as a resource for future research.

- We prove that simple adapter-tuning has a competitive performance on probing factual knowledge and is able to outperform all existing tuning methods such as fine-tuning and P-tuning [32]. This is possibly due to the fact that (1) additional adapters can have access to the feed-forward process, thus may help LLMs reason when making inferences; (2) freezing parameters within original LLMs preserves the learned knowledge from pre-training when tuning with adapters.

- We propose a simple but effective unified adapter-based framework for unbiased and robust factual knowledge extraction (Uni-Arkex). For retrieval accuracy, Uni-Arkex outperforms the current state-of-the-art(SOTA) model MeCoD [53] on BERT-large and RoBERTa large-settings. Meanwhile, intensive experiments demonstrate that our proposed method significantly mitigates both prompt pref-

erence bias and prompt verbalization bias. In addition, we show that these improvements remain consistent when scaling the size of LLMs.

- We provide analysis and conclude that the success of our framework is mainly because (1) adapter-based tuning is compatible with multi-task settings; (2) mitigating prompt preference bias and prompt verbalization bias are not two separate tasks. They may have a positive synergizing effect on each other when we try to optimize these two biases simultaneously.

Limitations and future works: Despite the achievements above, there are still limitations in this project. Firstly, the human evaluation of our proposed dataset ParaTrex is based on merely 5 bilingual speakers. More judgments with diverse backgrounds are necessary to further prove the high quality of our proposed datasets. It is worth inviting more people to evaluate our proposed datasets. Secondly, due to the constraints of time and computing resources, we do not perform experiments on super large LLMs such as Llama [51] and GPT-4 [37], which is more popular among real applications. It is worth extending our experiments on these LLMs to further investigate whether the improvements of our proposed modules such as maxing entropy and self-augmentation are scalable among larger LLMs. Thirdly, we just take a straightforward step to propose a unified framework in order to ensure that it generalizes well across various scenarios. This framework has the potential to be further optimized. For example, in our self-augmentation modules, we simply make mathematical addition and minus between output distributions. Other methods such as a contrastive learning framework may yield additional enhancements. These improvements and experiments also present a practical direction for future works.

Bibliography

[1] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*, 2022.

[2] Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. Paracrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, 2020.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Boxi Cao, Hongyu Lin, Xianpei Han, Fangchao Liu, and Le Sun. Can prompt probe pretrained language models? understanding the invisible risks from a causal view. *arXiv preprint arXiv:2203.12258*, 2022.

[5] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or educated guess? revisiting language models as knowledge bases. *arXiv preprint arXiv:2106.09231*, 2021.

[6] Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. Revisiting parameter-efficient tuning: Are we really there yet? *arXiv preprint arXiv:2202.07962*, 2022.

[7] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

[8] Jeff Da, Ronan Le Bras, Ximing Lu, Yejin Choi, and Antoine Bosselut. Analyzing commonsense emergence in few-shot knowledge models. *arXiv preprint arXiv:2101.00297*, 2021.

[9] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

[12] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

[13] Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[14] Allyson Ettinger. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020.

[15] Wenjuan Han, Bo Pang, and Yingnian Wu. Robust transfer learning with pretrained language models through adapters. *arXiv preprint arXiv:2108.02340*, 2021.

[16] Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654*, 2021.

[17] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*, 2021.

[18] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[20] Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. X-factr: Multilingual factual knowledge retrieval from pretrained language models. *arXiv preprint arXiv:2010.06189*, 2020.

[21] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[22] Nora Kassner, Philipp Dufter, and Hinrich Schütze. Multilingual lama: Investigating knowledge in multilingual pretrained language models. *arXiv preprint arXiv:2102.00894*, 2021.

[23] Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. *arXiv preprint arXiv:1911.03343*, 2019.

[24] Mehran Kazemi, Sid Mittal, and Deepak Ramachandran. Understanding fine-tuning for factual knowledge extraction from language models. *arXiv preprint arXiv:2301.11293*, 2023.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Deepak Kumar, Oleg Lesota, George Zerveas, Daniel Cohen, Carsten Eickhoff, Markus Schedl, and Navid Rekabsaz. Parameter-efficient modularised bias mitigation via adapterfusion. *arXiv preprint arXiv:2302.06321*, 2023.

[27] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[28] Tianyi Li, Wenyu Huang, Nikos Papasarantopoulos, Pavlos Vougiouklis, and Jeff Z Pan. Task-specific pre-training and prompt decomposition for knowledge graph population with language models. *arXiv preprint arXiv:2208.12539*, 2022.

[29] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[30] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[31] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[32] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021.

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[34] Robert L Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*, 2021.

[35] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*, 2021.

[36] Benjamin Newman, Prafulla Kumar Choubey, and Nazneen Rajani. P-adapters: Robustly extracting factual information from language models with diverse prompts. *arXiv preprint arXiv:2110.07280*, 2021.

[37] OpenAI. Gpt-4 technical report, 2023.

[38] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[39] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[40] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

[41] Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021.

[42] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[43] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[44] Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. On the systematicity of probing contextualized word representations: The case of hypernymy in bert. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, 2020.

[45] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*, 2020.

[46] Tara Safavi and Danai Koutra. Relational world knowledge representation in contextual language models: A review. *arXiv preprint arXiv:2104.05837*, 2021.

[47] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[48] Micah Shlain, Hillel Taub-Tabib, Shoval Sadde, and Yoav Goldberg. Syntactic search by example. In *Proceedings of the 58th Annual Meeting of the Association*

for Computational Linguistics: System Demonstrations, pages 17–23, Online, July 2020. Association for Computational Linguistics.

[49] Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski. Lm-kbc: Knowledge base construction from pre-trained language models, 2022.

[50] Asa Cooper Stickland and Iain Murray. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning*, pages 5986–5995. PMLR, 2019.

[51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[52] Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*, 2021.

[53] Yuhang Wang, Dongyuan Lu, Chao Kong, and Jitao Sang. Towards alleviating the object bias in prompt tuning-based factual knowledge extraction. *arXiv preprint arXiv:2306.03378*, 2023.

[54] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020.

[55] Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. *arXiv preprint arXiv:2108.13161*, 2021.

[56] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. *arXiv preprint arXiv:2104.05240*, 2021.

[57] Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740, 2020.

[58] Yulin Zhou, Yiren Zhao, Ilia Shumailov, Robert Mullins, and Yarin Gal. Revisiting automated prompting: Are we actually doing better? *arXiv preprint arXiv:2304.03609*, 2023.

Appendix A

Data Extension details for ParaTrex

A.1 Details of generated templates for an example relation

We provided a full example in ParaTrex for relation 'P1376': 'CapitalOf' in Table A.1.

A.2 Human evaluation

We give a screenshot of our questionnaire for human evaluation on ParaTrex. The questionnaire includes 40 questions in total and we show 2 examples of the questions.

Templates	inhouse split	paraphrase type
The capital of [Y] is [X] .	test	short paraphrase
[X] is [Y]'s capital .	test	short paraphrase
[X] serves as [Y]'s capital .	test	short paraphrase
[Y]'s capital city is [X] .	test	short paraphrase
[X] acts as [Y]'s capital .	test	short paraphrase
[X] is the administrative division where the municipality of [Y] serves as the capital .	test	long paraphrase
The governmental seat of [Y] is located in [X], which is the capital city .	test	long paraphrase
[X] holds the status of being the capital city and administrative center of [Y] .	test	long paraphrase
The capital of [Y] is none other than [X], where the government operates .	test	long paraphrase
The administrative hub of [Y] is [X], which holds the position of being the capital cit .	test	long paraphrase
[X] is the official capital of [Y] .	test	normal paraphrase
The capital city of [Y] goes by the name of [X] .	test	normal paraphrase
[X] is the designated capital city of [Y] .	test	normal paraphrase
[X] serves as the principal capital city of [Y] .	test	normal paraphrase
[X] is the administrative capital and governmental seat of [Y] .	test	normal paraphrase
[X] is the principal administrative center of [Y] .	test	normal paraphrase
[X] serves as the capital city and governmental hub of [Y] .	test	normal paraphrase
[X] holds the official status of being [Y]'s capital city .	test	normal paraphrase
[X] acts as the administrative capital of [Y] .	test	normal paraphrase
[X] serves as the capital city of [Y] .	test	normal paraphrase
[X] is the primary governing capital and administrative center of [Y] .	test	normal paraphrase
[X] is the primary political center of [Y] .	test	normal paraphrase
[X] holds the title of being [Y]'s capital .	test	normal paraphrase
[X] serves as the seat of government for [Y] .	test	normal paraphrase
[X] is the city that serves as [Y]'s capital .	test	normal paraphrase
The government of [Y] is headquartered in [X], its capital .	test	normal paraphrase
[X] acts as the political center of [Y] .	test	normal paraphrase
[X] holds the official position of being [Y]'s capital .	train	normal paraphrase
[X] serves as the governing center of [Y] .	train	normal paraphrase
The capital city of [Y] is [X] .	train	normal paraphrase
[X] is the administrative center of [Y] .	train	normal paraphrase
The seat of administration in [Y] is [X] .	train	normal paraphrase
The designated capital city of [Y] is [X] .	train	normal paraphrase
The governmental headquarters of [Y] is located in [X] .	train	normal paraphrase
[X] holds the status of being [Y]'s capital .	train	normal paraphrase
The government of [Y] is headquartered in [X] .	train	normal paraphrase
[X] is where the governing body of [Y] is located .	train	normal paraphrase
[X] holds the position of being [Y]'s capital city .	train	normal paraphrase
[X] holds the official governmental seat and capital status of [Y] .	train	normal paraphrase
[X] serves as the governing capital of [Y] .	train	normal paraphrase
The capital city of [Y] is none other than [X] .	train	normal paraphrase
The political center of [Y] is [X] .	train	normal paraphrase
The administrative capital of [Y] is [X] .	train	normal paraphrase
The government headquarters of [Y] can be found in [X] .	train	normal paraphrase
[X] is where the government of [Y] is based .	train	normal paraphrase

Table A.1: A specific example of relation 'Capital of' in our proposed ParaTrex. The original prompt template in LAMA is "*[X] is the capital of [Y] .*"

```
In the following questions, we provide 1 original input and 3
probable paraphrases. Please choose the sentances you think that
are NOT paraphrases of the original inputs. For example, please
answer 1-1 if you think the first sentance of the first question
is NOT the paraphrase of the orignal sentance. Please answer 1-0
if you think all candidates of the first question are the
paraphrase of the question.
Note that there may be several or no answer for a certain
question.
You can use translation machine to translate a certain word if
you do not understand it. But please write answers based on your
own understanding. DO NOT translate the whole sentance and make
predictions using automatic machines!

1: Original sentence: "[X] died in [Y] ."
Example:  "Otto Brahm died in Berlin . || Nicholas V died in
Rome ."
Example [X]: "Otto Brahm || Berlin"
Example [Y]: "Nicholas V || Rome"
Description: "most specific known (e.g. city instead of country,
or hospital instead of city) death location of a person, animal
or fictional character"
Paraphrase candidates:
1. The final moments of [X] took place in [Y] .
2. [Y] was the means of expression for [X] .
3. [X]'s passing occurred in [Y] .

2: Original sentence: "[X] is a subclass of [Y] ."
Example:  "quarter note is a subclass of note . || Doublecortin
is a subclass of protein ."
Example [X]: "quarter note || note"
Example [Y]: "Doublecortin || protein"
Description: "all instances of these items are instances of those
items; this item is a class (subset) of that item. Not to be
confused with P31 (instance of)"
Paraphrase candidates:
1. [X] is an offshoot of [Y] .
2. [X] used [Y] as their language of interaction .
3. [X] is grouped within [Y] .
```

Figure A.1: Examples of questions for human evaluation on ParaTrex

Appendix B

Full results for LAMA

B.1 Specific results for all relations of our proposed method

We show the full results of all relations for our proposed Uni-Arkex on the LAMA dataset based on the BERT-large model.

Relation	Test_acc	MRR	CT_Entropy	CT_Hit1	KLD
P159	0.4257	0.5099	2.2999	0	13.0318
P138	0.7714	0.7962	2.2982	0	14.482
P20	0.4015	0.5081	2.3015	0	13.0068
P361	0.5503	0.649	2.3	0	13.347
P495	0.4486	0.5733	2.3022	0	13.6849
P413	0.4485	0.618	2.3017	0	14.7436
P190	0.0477	0.1343	2.3011	0	11.7378
P108	0.119	0.2201	2.3007	0	13.2785
P103	0.8966	0.9359	2.3021	0	14.6807
P178	0.6871	0.767	2.3018	0	15.0095
P937	0.5217	0.6245	2.2998	0	14.2565
P27	0.5313	0.6594	2.3019	0	14.7093
P176	0.9195	0.9424	2.3008	0	15.6398
P740	0.1886	0.2925	2.3011	0	12.0164
P39	0.7052	0.8095	2.2956	0	14.6951
P136	0.7253	0.8196	2.3006	0	14.9541
P131	0.4735	0.5791	2.302	0	13.9597
P276	0.5392	0.5987	2.3	0	13.5306
P30	0.9468	0.9694	2.301	0	15.5741
P140	0.8433	0.8991	2.2997	0	15.2921
P364	0.5847	0.7114	2.299	0	13.4268
P449	0.4196	0.6232	2.301	0	12.9184
P37	0.6989	0.797	2.3011	0	14.0201
P127	0.5735	0.6436	2.2989	0	13.409
P530	0.0379	0.1142	2.3001	0	11.9911
P1303	0.4815	0.6911	2.2936	0	14.1848
P19	0.2577	0.3508	2.2528	0	10.0559
P463	0.7389	0.8149	2.2976	0	15.1337
P36	0.6773	0.7073	1.6855	0	7.4246
P264	0.0189	0.1685	2.3004	0	11.9502
P106	0.4275	0.5829	2.3017	0	13.8305
P101	0.2059	0.3465	1.7977	0.089	2.1489
P407	0.7515	0.8204	2.2978	0	13.1364
P279	0.7087	0.7713	2.3014	0	14.7263
P1376	0.8	0.8426	2.2833	0	13.8507
P47	0.3113	0.4927	2.3002	0	13.5657
P17	0.6294	0.7196	2.3016	0	14.1329
P1001	0.8795	0.9048	2.296	0	15.0948
P1412	0.8193	0.8757	2.3016	0	14.6903
Average	0.5439	0.6381	2.2698	0.0023	13.3672

Table B.1: Full results of accuracy and prompt-preference bias for our Uni-Arkex methods.

relation	ID_consist	ID_all_consist	ID_acc_consist	OOD_consist	ood_all_consist	ood_acc_consist	PR_consist	PR_consist	PR_consist
P159	0.5786	0.4855	0.2428	0.4126	0.3512	0.1738	0.5913	0.4734	0.247
P138	0.8225	0.7977	0.6658	0.8296	0.805	0.6755	0.8935	0.8734	0.7194
P20	0.5199	0.4387	0.2146	0.492	0.3689	0.1796	0.6388	0.5416	0.2567
P361	0.5967	0.4988	0.3319	0.547	0.5403	0.3617	0.9405	0.9206	0.5322
P495	0.521	0.4067	0.2239	0.5654	0.5154	0.2856	0.7643	0.7398	0.3878
P413	0.4173	0.2933	0.1692	0.4753	0.3009	0.1793	0.7092	0.6105	0.3721
P103	0.8607	0.8133	0.7519	0.8202	0.7126	0.6614	0.9557	0.9456	0.8762
P176	0.9248	0.9117	0.8766	0.9294	0.9368	0.8969	0.9415	0.924	0.884
P740	0.5603	0.4538	0.1099	0.4321	0.3424	0.0945	0.6877	0.5952	0.134
P136	0.755	0.8341	0.609	0.6827	0.7559	0.5599	0.6589	0.5521	0.4354
P131	0.6005	0.5501	0.3378	0.5467	0.4875	0.3117	0.8327	0.7882	0.4295
P276	0.5515	0.4644	0.3121	0.5055	0.4224	0.3034	0.8832	0.8442	0.5191
P30	0.9387	0.9333	0.8966	0.9267	0.9192	0.8841	0.9666	0.9615	0.9278
P140	0.8367	0.7792	0.6848	0.8234	0.8087	0.7131	0.9159	0.8903	0.7783
P364	0.7392	0.7039	0.4779	0.7687	0.7388	0.4968	0.8505	0.8352	0.5583
P449	0.6358	0.5415	0.2419	0.5736	0.4858	0.2161	0.427	0.2737	0.1261
P37	0.8846	0.8566	0.6251	0.8671	0.8571	0.6334	0.916	0.9047	0.656
P127	0.6567	0.629	0.4619	0.6094	0.6055	0.4419	0.538	0.3956	0.3013
P19	0.3338	0.1999	0.0678	0.3467	0.2563	0.0722	0.2793	0.198	0.0406
P36	0.7862	0.7539	0.5828	0.722	0.6697	0.5245	0.833	0.7844	0.6049
P264	0.5649	0.4837	0.026	0.1161	0.3174	0.0182	0.3596	0.2962	0.009
P407	0.8536	0.8206	0.6491	0.6689	0.7165	0.5672	0.8042	0.7236	0.57
P279	0.705	0.586	0.4572	0.3917	0.3324	0.1903	0.9316	0.9157	0.6825
P1376	0.8293	0.8291	0.7223	0.8406	0.8136	0.7071	0.8659	0.8489	0.7361
P17	0.6294	0.4905	0.3675	0.5695	0.4308	0.3225	0.9207	0.9075	0.6076
Average	0.6841	0.6222	0.4443	0.6185	0.5796	0.4188	0.7642	0.7098	0.4957

Table B.2: Full results of prompt verbalization bias for our Uni-Arkex methods