Towards Alleviating the Object Bias in Prompt Tuning-based Factual Knowledge Extraction

Anonymous ACL submission

Abstract

Many works employed prompt tuning methods to automatically optimize prompt queries and extract the factual knowledge stored in Pretrained Language Models. In this paper, we observe that the optimized prompts, including discrete prompts and continuous prompts, exhibit undesirable object bias. To handle this problem, we propose a novel prompt tuning method called MeCoD consisting of three modules: Prompt Encoder, Object Equalization and Biased Object Obstruction. Experimental results show that MeCoD can significantly reduce the object bias and at the same time improve accuracy of factual knowledge extraction.

1 Introduction

Pretrained language models (PLMs) have become a standard practice in NLP and achieved strong performance on many downstream tasks (Qiu et al., 2020) (Liu et al., 2021a). A recognized reason why PLMs are so powerful is the knowledge learned from a large amount of public corpus (Liu et al., 2019a). Recently, researchers have taken interest in measuring and extracting the factual knowledge in PLMs. Petroni et al. (2019) first formally proposed the LAMA benchmark, which employs handcrafted prompts to retrieve factual knowledge in the form of < subject, relation, object > triples. For example, regarding a factual knowledge triple < *Douglas Adams*, *native language*, *English* >, LAMA can query PLMs with "*The native language of Douglas Adams is [MASK]*" to extract the native language of *Douglas Adams*, where "*The native language of [X] is [MASK]*" is a manual prompt for the relation "*native language*" and "*[MASK]*" is a placeholder for the object to predict.

In order to extract factual knowledge more effectively, many works take a step toward automatically tuning prompts with additional training set. Shin et al. (2020) proposed AutoPrompt to generate discrete prompts automatically based on gradient optimization by maximizing the expected likelihood of the ground truth object. Instead of searching discrete prompts, a more flexible research line is tuning continuous prompts directly in the input embedding space. For example, Liu et al. (2021b) proposed P-tuning to optimize a continuous prompt for each factual relation, and achieved SOTA performance. Li and Liang (2021) proposed a semi-automatic method called Prefix-tuning to learn a prefix to add to manual prompts. Newman et al. (2022) applied Prefix-tuning to improve the robustness of factual knowledge extraction.

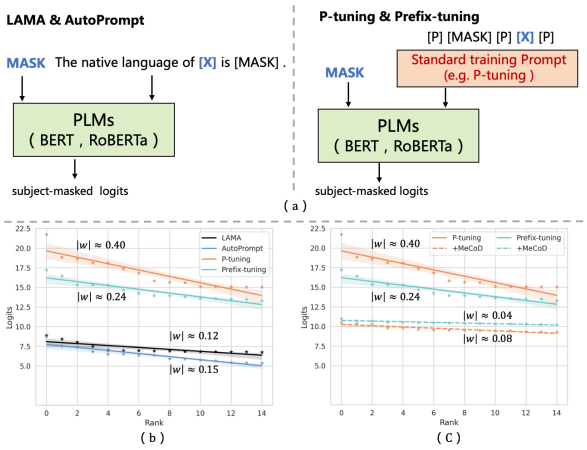

Figure 1: *Object bias* in different prompt-based knowledge extraction methods: LAMA, AutoPrompt, Prefix-tuning and P-tuning. (a) demonstrates how to construct subject-masked prompts. (b), (c) show the derived logits of top-retrieved objects for original and our proposed prompt-tuning methods, respectively

Although the above prompt tuning methods achieve good performance, we discuss in this paper that they suffer from severe *object bias* problem. As illustrated in Figure 1(a), we construct subject-masked prompts for different prompt-based knowledge extraction methods. Example prompts for relation P103 are illustrated in Table 1. We conduct experiments on LAMA benchmark (Petroni et al., 2019) which consists of 41 fact relations. Figure 1(b) shows the derived logits of top-k retrieved

Extraction Method		Prompt Template
LAMA	*Original*	The native language of Pierre Messmer is **[MASK]** .
	Subject-masked	The native language of [MASK] is **[MASK]** .
AutoPrompt	*Original*	Pierre Messmer [T] **[MASK]** .
	Subject-masked	[MASK] [T] **[MASK]**
Prefix-tuning	*Original*	[P] The native language of Pierre Messmer is **[MASK]** .
	Subject-masked	[P] The native language of MASK is **[MASK]** .
P-tuning	*Original*	[P] **[MASK]** [P] Pierre Messmer [P]
	Subject-masked	[P] **[MASK]** [P] [MASK] [P]

Table 1: Example of original and subject-masked prompt templates for relation P103. "[T]","[P]" indicate discrete and continuous optimizable prompt token, respectively. The number of [P] and [T] can be customized. "[MASK]" in bold is the placeholder for the object to predict.

objects in descending order. Since the subject is masked in the issued prompt template, no context is provided and an even logit distribution for different object candidates is expected. Taking the fact < *Douglas Adams*, *native language*, *English* > for example, objects like "French", "English" and "Russian" should be treated equally when *Douglas Adams* is masked. However, we observe non-trivial slopes ($|w|$ in Figure 1 (b)) of the regression lines in the 4 examined knowledge extraction methods, i.e., they all exhibit bias towards specific objects. Notably, the 3 prompt-tuning methods of AutoPrompt, P-tuning and Prefix-tuning, have more inclined slopes and thus exhibit more severe object bias. More object bias measurement results are available at Section 2.1. Given the observed object bias in prompt tuning methods, we further design analysis experiments and find the negative influence of object bias on knowledge extraction accuracy (detailed in Section 2.2). This motivates us to develop solutions to both alleviate the object bias problem and contribute to more accurate factual knowledge extraction.

In this paper, to address the object bias problem in prompt-tuning stage, we propose **MeCoD** (**M**aximum **e**ntropy and **C**ontrastive learning for **o**bject **D**ebiasing) towards unbiased factual knowledge extraction [1]. The basic idea is deriving equalized object predictions with subject-masked prompt, and at the same time disencouraging the biased objects with original prompt. These goals are realized by a maximum entropy-based Object Equalization module and contrastive learning-based Biased Object Obstruction module, respectively. Figure 1 (c) illustrates the intuitive effect of object bias alleviation.

Contributions. We summarize the main contributions of this paper as follows:

- We position the object bias problem in prompt tuning-based factual knowledge extraction. The influence of object bias on knowledge extraction accuracy is also discussed.
- We propose an object debiasing method at the prompt tuning stage to alleviate the object bias and improve accuracy of factual knowledge extraction.
- The effectiveness of the proposed method is validated with sufficient qualitative and quantitative experiments.

2 Data Analysis

Object Bias Definition. Factual knowledge can be represented in form of <subject, relation, object> triples. *Object bias* in factual knowledge extraction refers to the phenomenon that the pretrained language model with prompts retrieves object candidates unequally when subject is not assigned, e.g., prefering "French" to "English" in the prediction of person's native language when the person is not specified.

2.1 Object Bias Measurement.

Object bias inherently considers the uncertainty of retrieved objects with subject-masked prompt queries. We thus employ entropy (Shannon, 1948) in this work to measure object bias. Specifically,

[1] Since continuous prompts are more effective and widely adopted, MeCoD is designed to improve continuous prompt tuning methods, e.g., P-tuning, Prefix-tuning.

Method	Entropy	Comparison with 2.305
LAMA	2.077	-9%
AutoPrompt	1.901	-17%
P-tuning	1.754	-23%
Prefix-tuning	2.002	-13%

Table 2: The averaged object bias entropy over 41 relations of different knowledge extraction methods on LAMA benchmark.

Method	All	Incorrectly predicted
LAMA	0.260	0.258
AutoPrompt	0.380	0.416
P-tuning	0.432	0.479
Prefix-tuning	0.374	0.394

Table 3: The Pearson correlation coefficient between the rank corresponding to original prompt and subject-masked prompt.

Dateset	P@1	Entropy
Original	49.413	1.754
Undersampled	48.776	1.924

Table 4: Results for P-tuning fit to original dataset and undersampled dataset.

we define *object bias entropy* in terms of the relation R as:

$$entropy(R) = -\sum_{i=1}^{k} p_R(i) log_2(p_R(i)), \quad (1)$$

where p_R is obtained by selecting top-k subject-masked logit values and normalizing with softmax function, k denotes the number of logit values used to calculate entropy. In our subsequent analysis, we set k to 10, and $entropy(R)$ will achieve a maximum value of about 2.305 when the object logits are equal. The smaller the value, the more significant the object bias.

We measure the 4 typical factual knowledge extraction methods on the LAMA benchmark according to Eqn.1. The averaged result over 41 relations is shown in Table 2. It is easy to find that the observation in the form of object bias entropy is consistent with that of slope in Figure 1: (1) The 4 methods all exhibit object bias, the entropy values noticeable decrease from 2.305 by 9%, 17%, 23% and 13%, respectively. (2) The object bias entropy of prompt tuning methods, including AutoPrompt, Prefix-tuning and P-tuning, is more smaller than that of manual prompts, LAMA. This observation further demonstrates that prompt tuning methods suffer more serious object bias than manual prompts. Note that the object bias entropy of Prefix tuning falls in between that of manual prompts, LAMA and full-automatic prompts, AutoPrompt and P-tuning. A possible reason is that the manual template in the Prefix-tuning limits the learning of object bias.

2.2 Influence on Knowledge Extraction

In order to investigate the influence of object bias on knowledge extraction, we compare the retrieved object candidates from original prompts and subject-masked prompts. Specifically, we examine the rank of ground-truth object in the retrieved object lists and calculate Pearson correlation coefficient between the rank corresponding to the original prompt and the subject-masked prompt.

According to the result on all testing samples, we find that the correlations of prompt tuning methods are higher than that of LAMA. This illustrates the prediction results of prompt tuning methods are influenced more significantly by object bias than that of manual prompts. Furthermore, comparing between the results of all testing samples and incorrectly predicted samples, it is easy to observe that correlation coefficient of LAMA remains almost unchanged, while those of the 3 prompt tuning methods increase obviously in incorrectly predicted samples. This suggests that some of the incorrect predictions can be attributed to the bias towards specific objects, and motivates us to address the object bias problem to further improve knowledge extraction accuracy.

2.3 Undersampling-based Preliminary Attempts on Object Debiasing.

The above observations demonstrate the necessity for object debiasing. Object bias is attributed to both the pretraining stage of PLMs and prompt tuning stage. The observed object bias of LAMA mainly origins from the pre-training stage. While some pilot studies (Guo et al., 2022) are devoted to reducing bias in the pre-trained language models, we found in the above analysis that the object bias at prompt-tuning stage is more severe than that at the pre-training stage, and thus focus on object debiasing at the prompt tuning stage in this paper.

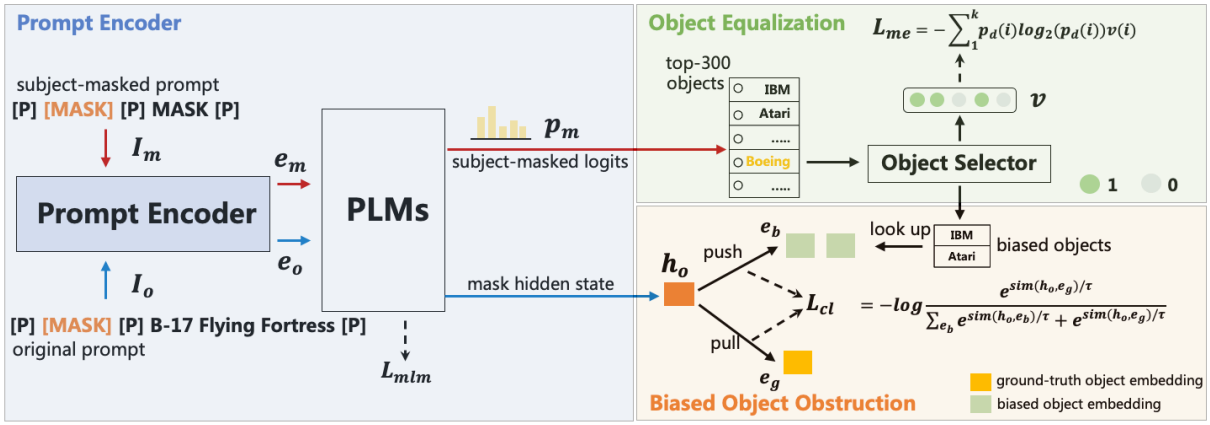

Figure 2: Overall architechture of the proposed MeCoD.

The straightfoward cause of object bias at the prompt tuning stage is imbalanced training data for optimizing prompts. Take P-tuning as an example, we make a preliminary attempt by retraining it with undersampled balanced training set. In order not to excessively reduce the number of samples, we first group the training samples according to objects, and then randomly undersample the two groups with the largest number of samples. In this case, their numbers are consistent with the number of the third largest group. Table 4 summarizes the performance of P-tuning trained with different training sets.

We observe that object bias is alleviated on undersampled training set, and this validates the attribution of imbalanced tuning set in deriving object bias. However, we find that the P@1 drops clearly due to insufficient use of data, that is, accuracy is sacrificed for debiasing. This inspires us to design an effective prompt tuning method, instead of simply balancing training data, to alleviate object bias as well as improve accuracy performence.

3 Methodology

We present the overall framework, MeCoD, as illustrated in Figure 2. The basic idea is to improve prompt encoder so that issuing the resultant embeddings to popular PLMs no longer exhibits object bias. The goals are two-fold: (1) Objects should have equal opportunities to be extracted from PLMs by subject-masked prompt; (2) The biased objects should be prevented from being extracted by the original prompt with specified subject. Correspondingly, MeCoD includes three modules. The first module is Prompt Encoder to be optimized, which takes original and subject-masked prompts as inputs and issues the resultant embeddings to PLMs to obtain mask hidden state h_o and subject-masked logits p_m respectively (see Section 3.1). The second module is Object Equalization, which takes subject-masked logits p_m as input and forces model to treat objects equally, when the subject is masked (see Section 3.2). The third module is Biased Object Obstruction which further prevents the biased objects from being extracted by forcing the mask hidden state h_o away from biased object embeddings and close to ground-truth object embedding (see Section 3.3). We will take P-tuning as an example to elaborate the details of each module below.

3.1 Prompt Encoder

In this module, we first construct subject-masked input by replacing "*[X]*" with "*[MASK]*" , as shown in Table 1. The number of "*[MASK]*" is set as the number of tokenized subject. As shown in Figure 2 (left), given original prompt I_o and subject-masked prompt I_m, we use Prompt Encoder to get input embeddings e_o and e_m for PLMs. Then, mask hidden state h_o, subject-masked logits p_m and the MLM (Masked Language Modeling) loss $\mathcal{L}_{mlm}$, can be obtained from PLMs as follows:

$$\begin{aligned} h_o &= \text{PLMs}(e_o), h_m = \text{PLMs}(e_m), \\ p_o &= \text{MLM-head}(h_o), \\ p_m &= \text{MLM-head}(h_m), \\ \mathcal{L}_{mlm} &= -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_{o_i}), \end{aligned} \tag{2}$$

where y_i denotes the ground truth. p_m will be used to equalize the objects with respect to subject-masked prompt in Section 3.2. Both h_o and p_m will be employed to obstruct the influence of biased objects in Section 3.3.

3.2 Object Equalization

According to data analysis in Section 2, we consider that the probabilities of object candidates should be equalized when issuing the subject-masked prompt to PLMs. In this subsection, we will introduce a method based on maximum entropy to force objects to be treated equally when subject is not given. Note that only the fact-related objects need to be considered, e.g., regarding relation P103 (native language), objects with risk to bias the prediction results are like "English", "French" instead of "apple". To filter out the unrelated objects, we first sort subject-masked logits p_m with descending order to get p_d, and empirically reserve the top-300 objects c_o. Then, we further employ a linear layer as a binary classifier named Object Selector to identify the objects to be equalized. Specifically, the object selector takes the object embeddings as input and returns a binary vector $v \in \{0,1\}^{300}$ with gumbel softmax (Jang et al., 2017) as follows:

$$v = \text{gumbel-softmax}(\text{Linear}(E(c_o))), \quad (3)$$

where E denotes embedding layer of PLMs. The object sets corresponding to $v(i) = 1, i = 1, 2, ..., 300$ are selected to be equalized. Finally, we construct the loss $\mathcal{L}_{me}$ based on Maximum Entropy:

$$\mathcal{L}_{me} = -\sum_{i=1}^{k} p_d(i) log_2(p_d(i)) v(i). \quad (4)$$

3.3 Biased Object Obstruction

This module further reduce the probability of retrieving biased objects when issuing prompt with specified subjects, and we introduce a module based on contrastive learning. The key idea is to simultaneously minimize the representation gap between "*[MASK]*" and ground-truth object and maximize that between "*[MASK]*" and irrelevant biased objects. Specifically, we regard the objects corresponding to $v(i) = 1, i = 1, 2, ..., 300$ as biased objects except for the ground-truth object. We formalize it as a contrastive learning problem and propose to minimize the following loss (van den Oord et al., 2018):

$$\mathcal{L}_{cl} = -log \frac{e^{sim(h_o, e_g)/\tau}}{\sum_{e_b} e^{sim(h_o, e_b)/\tau} + e^{sim(h_o, e_g)/\tau}}, \quad (5)$$

where sim(·) calculates the consine similarity of different representations, e_g and e_b denote the word embeddings of ground-truth object and biased objects, respectively. τ is the temperature, controlling the difficulty of distinguishing between positive and negative samples. Intuitively, the contrastive loss forces the model to push the mask hidden state away from the embeddings of biased objects, and pull it to the ground-truth object embedding.

During training, the model is optimized by jointly minimizing loss $\mathcal{L}$ as follows:

$$\mathcal{L} = \mathcal{L}_{mlm} - \lambda_1 \mathcal{L}_{me} + \lambda_2 \mathcal{L}_{cl}, \quad (6)$$

where λ_1 and λ_2 are the coefficients to balance the three training losses.

4 Experiments

4.1 Experimental Settings

Datasets. We adopt the LAMA-TREx (Petroni et al., 2019) as the main testing set which consists of 41 Wikidata relations and altogether 29,500 testing triples. Besides, we also evaluate our method on WIKI-UNI (Cao et al., 2021) which is constructed to ensure each object to appear the same times for each relation. As for training, we use the data collected by Shin et al. (2020) which contains 800 training samples and 200 developing samples for each fact relation.

Evaluation metrics and baselines. In addition to object bias entropy, we also evaluate the performance on knowledge extraction with metrics of precision-at-1 (P@1) and mean reciprocal rank (MRR). We report average performance over 41 relations. In order to evaluate the effectiveness of the proposed MeCoD, we implement the LAMA, Prefix-tuning, P-tuning and Undersampling-based solutions (see Section 2) as baselines. Specifically, LAMA provides hand-crafted prompts that are less object-biased than prompt tuning methods. P-tuning and Prefix-tuning are the representatives of continuous prompt, which are more effective and widely used.

Implementation details. For PLMs in our experiments, we investigate BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) [2]. The prompt encoder is initialized by standard training prompt encoder, e.g.,P-tuning based on LSTM (Shi et al., 2015). We use the Adam optimizer (Kingma and Ba, 2014) with its default configuration. For gradient training, we fix parameters of PLMs, and set

[2]We use the the implementations of huggingface. https://huggingface.co

Method	BERT-base			RoBERTa-base		
	P@1	MRR	Entropy	P@1	MRR	Entropy
LAMA	29.641	39.312	2.077	15.206	22.791	1.972
Prefix-tuning	47.472	57.522	2.002	45.828	56.473	1.904
+ Undersampling	42.254	52.914	2.159	40.742	52.086	2.100
+ MeCoD	48.329(2% ↑)	58.553	**2.145**(7% ↑)	46.281(1% ↑)	57.072	**2.298**(16% ↑)
P-tuning	49.413	59.419	1.754	44.828	54.955	1.655
+ Undersampling	48.776	58.696	1.926	42.498	53.203	2.020
+ MeCoD	**50.438**(2% ↑)	**60.335**	2.141(22% ↑)	**46.813**(4% ↑)	**57.474**	2.109(27% ↑)

Table 5: Results on the LAMA benchmark using the BERT-base-cased and RoBERTa-base model.

Method	BERT-base		RoBERTa-base	
	P@1	Entropy	P@1	Entropy
P-tuning + MeCoD	**50.438**	2.141	**46.813**	**2.109**
w/o Object Equalization	50.301	1.808	46.471	1.661
w/o Biased Object Obstruction	50.317	**2.212**	46.625	1.984

Table 6: Ablation study.

the learning rate to 1e-5 to jointly optimize Prompt Encoder and Object Selector. We set λ_1 and λ_2 to 0.2, 0.1 respectively. As for training time, take MeCoD on P-tuning as an example, it spends about 40 minutes for each relation on 1 GPU device of RTX A4000.

4.2 Quantitative Results

Table 5 shows the performance of each method on the two selected PLMs. Briefly, MeCoD outperforms the baselines on both object bias and knowledge extraction performance. Take P-tuning as an example, enhanced with MeCoD, object bias entropy increases by 22% and 27%, and P@1 increases by 2% and 4% for BERT and RoBERTa, respectively. This demonstrates that alleviating bias contributes to improving the accuracy performance. Similar results are observed in Prefix-tuning. This demonstrates the generality and effectiveness of our method. The results of Undersampling-based methods illustrate that accuracy is sacrificed for debiasing. The improvement is also reflected in the evaluation results on WIKI-UNI dataset, as shown in Table 9 and Table 10 in AppendixA.1.

4.3 Ablation Study

In order to clarify the source of performance improvement in MeCoD, we take P-tuning as an example and conduct ablations by removing particular modules from MeCoD. The ablation study results are shown in Table 6. We can conclude that (1) Object Equalization plays a crucial role in alleviating bias, as removing the module causes object bias entropy to decrease. The decreased accuracy shows that Object Equalization module is helpful to improve the accuracy by alleviating the object bias. (2) Biased Object Obstruction is useful for ensuring accuracy, because the contrastive loss forces model not to be affected by biased objects. But it does not alleviate object bias clearly, which may cause other unexpected problems, so it's not recommended to be used alone.

4.4 Case Study

In order to better understand how MeCoD contributes to alleviating object bias, as shown in Figure 3, we visualize the regression lines of subject-masked logits of P-tuning and Prefix-tuning on relation P178 (developer) and relation P103 (native language), respectively. The lines corresponding to MeCoD are relatively flatter, that is, the object bias is alleviated clearly by our method, which is consistent with quantitative results in Table 5. Furthermore, we investigate the influence of alleviating object bias on knowledge extraction by observing the top-k candidates extracted by original prompt and subject-masked prompt respectively on two fact samples, as shown in Table 7. Take P-tuning

Method	Top-k Candidates					
	original prompt			subject-masked prompt		
P-tuning	Atari(1)	**Boeing(2)**	IBM(4)	IBM(1)	Atari(2)	Boeing(22)
	13.789	**12.980**	12.662	11.851	11.846	7.071
P-tuning + MeCoD	**Boeing(1)**	Atari(3)	IBM(4)	Atari(21)	IBM(27)	Boeing(53)
	13.501	11.720	11.124	8.510	8.396	7.717
Prefix-tuning	French(1)	**English(2)**	Russian(3)	French(1)	Russian(2)	English(5)
	19.316	18.383	18.036	16.584	16.067	14.894
Prefix-tuning + MeCoD	**English(1)**	French(2)	Russia(3)	Russian(213)	French(257)	English(279)
	20.834	20.532	19.242	7.013	6.526	6.269

Table 7: Case Study: Top-k object candidates extracted by original prompt and subject-masked prompt about two fact samples. The first (middle row) is the result about "the developer of B-17 Flying Fortress". The second (bottom row) is the result about "the native language of Douglas Adams". The bold fonts indicate ground truth, the numbers in parentheses are ranks of object candidates and the numbers below objects are the corresponding logit values.

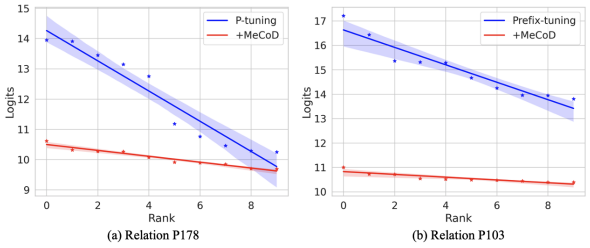

Figure 3: Case Study: The regression lines of subject-masked logits. (a) shows the result predicted by P-tuning and MeCoD on relation P178. (b) shows the result predicted by Prefix-tuning and MeCoD on relation P103.

for example, by observing the results extracted by subject-masked prompt, we find that the candidates' logits of MeCoD are more even, which means less object bias. However, P-tuning shows obvious bias on some objects like "Atari", "IBM". Correspondingly, as for the results extracted by original prompt, MeCoD correctly predicts "Boeing", but P-tuning predicts incorrectly on biased object "Atari", and ranks the correct object at the second place. Similar results are observed in Prefix-tuning. Therefore, we conclude that object bias is responsible to such incorrect predictions.

4.5 Discussions

Our experiments show that object bias of prompt tuning methods is undesirable. This inspires us to investigate how prompt tuning methods extract factual knowledge, and explore the potential cause of object bias. Specifically, we take P-tuning for example and illustrate in the following discussions on relation P19 (*place-of-birth*).

Finding Nearest Neighbors. In order to figure out the implication of the prompt token embeddings, we follow (Lester et al., 2021) and find their nearest neighbors from the frozen model's vocabulary. As shown in Table 11 [3], we observe that the prompt tokens in close position exhibit similar patterns. This indicates the tokens in different positions probably play different roles, but we can not understand the concrete meaning by the observation of their nearest neighbors. Therefore, we further analyze the the candidate words about prompt tokens, which is returned by masked language modeling (MLM) of PLMs.

Checking MLM Candidate Words. Table 8 illustrates the MLM candidate words of prompt tokens. Specifically, the subject is set to "Claude Arrieu" who was a prolific French composer born in "Paris". Two interesting observations include: (1) The MLM candidate words of front and back prompt tokens are mostly punctuations, while the front also include articles. This indicates that prompt tuning methods mimic human linguistic expression to some extent. For example, we often start a sentence with the article "the" and end it with the punctuation ".". (2) The results of middle prompt tokens exhibit literal correlations between "subject" and "[MASK]". Specifically, it is easy to find that the candidate words of Middle-1 are related to the object, for example "London" and even the ground-truth object "Paris". Note that the Middle-1 is the closest to "[MASK]". As for Middle-3, some candidate words can form the names of famous people with the first token of

[3]We show Table 11 in Appendix A.2.

Prompt token	MLM candidate words								
Front-1	.	\	,	the	)	;	of	and	?
Front-2	.	,	of	\	-	)	;	the	?
Front-3	.	of	,	the	in	;	The	\	a
Middle-1	.	London	:	##ville	Amsterdam	,	##s	**Paris**	-
Middle-2	Albert	Victor	Max	**Paris**	.	Robert	Amsterdam	##s	?
Middle-3	Albert	Max	Victor	Robert	.	Eric	Ann	Raymond	and
Back-1	,	:	##il	-	.	;	##l	##el	##lyn
Back-2	:	##il	-	,	.	July	August	operator	;
Back-3	.	;	?	!	|	...	ŏ964	c	-

Table 8: MLM candidate words of prompt tokens. Prompt template used in this case is "[P][P][P] [MASK] [P][P][P] Claude Arrieu [P][P][P]". [P] indicates prompt token, and we use "Front", "Middle", "Back" to represent their positions. The words in bold indicate ground-truth object.

the subject, e.g., "Albert Claude", "Victor Claude", "Robert Claude". Note that the Middle-3 is the closest to "subject". The candidate words of Middle-2 can be seen as a mixture of the above two type words. By combining the above two observations, we conclude that prompt tuning methods extract factual knowledge by depending on shallow literal correlations rather than factual relations. Furthermore, we consider that the shallow correlations is one of the potential cause of object bias. This needs to be demonstrated by rigorous experiments and analysis in the further works, and we just conjecture it intuitively here.

5 Related work

Language Models as Knowledge Base. Since the birth of Pretrained Language Models (PLMs), researchers have observed that there are much knowledge in PLMs. A typical research line is to explore whether the pre-training model can serve as a knowledge base. Petroni et al. (2019) demonstrated the existence of factual knowledge in PLMs and probed the factual knowledge with cloze-style prompts. Recently, Dai et al. (2021) proposed a knowledge attribution method to identify the factual knowledge neurons that store facts in PLMs. However, Cao et al. (2021) and Li et al. (2022) questioned the previous conclusion by investigating the behaviors of MLMs, that current MLMs can potentially serve as reliable factual knowledge bases. In this paper, instead of knowledge storage mechanism of the PLMs, we mainly investigate the object bias problem in factual knowledge extraction during prompt tuning stage.

Factual Knowledge Extraction. In addition to manual prompts, many researchers explored more effective methods for factual knowledge extraction. Jiang et al. (2020) mined prompts through text mining and paraphrasing. Recently, researchers engage in prompt tuning methods (Haviv et al., 2021) (Qin and Eisner, 2021). For example, Shin et al. (2020) trained a model to generate prompts automatically based on gradient optimization. Liu et al. (2021b) proposed P-tuning which completely abandoned natural language forms and optimized a continuous prompt for each factual relation. Li and Liang (2021) proposed a semi-automatic method called Prefix-tuning to learn a prefix to add to manual prompts. Newman et al. (2022) applied Prefix-tuning to improve the robustness of factual knowledge extraction. In this paper, we observe that prompt tuning methods suffer serious object bias, and propose a framework MeCoD to alleviate it.

6 Conclusion

In this work, we position the object bias problem in prompt tuning-based factual knowledge extraction, and propose MeCoD, a framework for alleviating the object bias and improving accuracy of factual knowledge extraction. Experimental results demonstrate the usefulness and generality of MeCoD. Besides, we argue that the shallow association learned by prompt tuning is one potential cause of object bias. In the future, we are working towards exploring the mechanism behind deriving object bias, designing more reliable prompt tuning methods for factual knowledge extraction and investigating the problems on generative pre-trained models, e.g., GPT2 (Radford et al., 2019), GPT3 (Brown et al., 2020).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

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

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Yue (Sophie) Guo, Yi Yang, and A. Abbasi. 2022. Autodebias: Debiasing masked language models with automated biased prompts. In *Annual Meeting of the Association for Computational Linguistics*.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. Bertese: Learning to speak to bert. *arXiv preprint arXiv:2103.05327*.

Eric Jang, Shixiang Shane Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *ArXiv*, abs/1611.01144.

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

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

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *ArXiv*, abs/2104.08691.

Shaobo Li, Xiaoguang Li, Lifeng Shang, Zhenhua Dong, Chengjie Sun, Bingquan Liu, Zhenzhou Ji, Xin Jiang, and Qun Liu. 2022. How pre-trained language models capture factual knowledge? a causal-inspired analysis. *arXiv preprint arXiv:2203.16747*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, abs/2101.00190.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. *ArXiv*, abs/1903.08855.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys (CSUR)*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *ArXiv*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Benjamin Newman, Prafulla Kumar Choubey, and Nazneen Rajani. 2022. P-adapters: Robustly extracting factual information from language models with diverse prompts. *ArXiv*, abs/2110.07280.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *ArXiv*, abs/1909.01066.

Guanghui Qin and Jas' Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *ArXiv*, abs/2104.06599.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *ArXiv*, abs/2003.08271.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang chun Woo. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting knowledge from language models using automatically generated prompts. *ArXiv*, abs/2010.15980.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748.

A Appendix

A.1 Quantitative Results on WIKI-UNI Datasets

Method	P@1	MRR	Entropy
LAMA	14.785	21.178	2.083
Prefix-tuning	21.806	28.342	2.000
+ Undersampling	21.427	27.846	2.150
+ MeCoD	22.745	29.479	**2.283**
P-tuning	22.310	29.251	1.776
+ Undersampling	22.467	28.982	1.964
+ MeCoD	**22.935**	**29.971**	2.092

Table 9: Results on the WIKI-UNI dataset using the BERT-base-cased model, averaged over relations.

Method	P@1	MRR	Entropy
LAMA	8.411	12.920	1.960
Prefix-tuning	20.536	27.485	1.908
+ Undersampling	21.379	28.163	2.118
+ MeCoD	**22.199**	**29.0.53**	**2.298**
P-tuning	19.240	25.934	1.675
+ Undersampling	19.351	25.782	2.020
+ MeCoD	20.521	27.813	2.044

Table 10: Results on the WIKI-UNI dataset using the RoBERTa-base model, averaged over relations.

A.2 Nearest Neighbors Case

Prompt token	Nearest words			
Front-1	Discovery	##cam	##final	Kathy
Front-2	Kathy	##cam	=	##cam
Front-3	=	:	##cam	com
Middle-1	=	:	Kathy	based
Middle-2	=	Kathy	:	actress
Middle-3	=	actress	suitcase	divorced
Back-1	divorced	Buildings	tells	suitcase
Back-2	divorced	suitcase	Shannon	psychiatrist
Back-3	Shannon	Cheryl	divorced	interception

Table 11: Top-4 nearest words of prompt tokens. Prompt template used in this case is "[P][P][P] [MASK] [P][P][P] Claude Arrieu [P][P][P]". [P] indicates prompt token, and we use "Front", "Middle", "Back" to represent their positions.