Calibrating Factual Knowledge in Pretrained Language Models

Qingxiu Dong[1] *, **Damai Dai**[1] *, **Yifan Song**[1], **Jingjing Xu**[2], **Zhifang Sui**[1] **and Lei Li**[3]
[1] MOE Key Lab of Computational Linguistics, School of Computer Science, Peking University
[2] Shanghai AI Lab [3] University of California, Santa Barbara
dqx@stu.pku.edu.cn, {daidamai,yfsong,jingjingxu, szf}@pku.edu.cn,
lilei@cs.ucsb.edu

Abstract

Previous literature has proved that Pretrained Language Models (PLMs) can store factual knowledge. However, we find that facts stored in the PLMs are not always correct. It motivates us to explore a fundamental question: How do we calibrate factual knowledge in PLMs without re-training from scratch? In this work, we propose a simple and lightweight method CALINET to achieve this goal. To be specific, we first detect whether PLMs can learn the right facts via a contrastive score between right and fake facts. If not, we then use a lightweight method to add and adapt new parameters to specific factual texts. Experiments on the knowledge probing task show the calibration effectiveness and efficiency. In addition, through closed-book question answering, we find that the calibrated PLM possesses knowledge generalization ability after fine-tuning. Beyond the calibration performance, we further investigate and visualize the knowledge calibration mechanism. The code and data are available at https://github.com/dqxiu/CaliNet.

1 Introduction

Recently, Pretrained Language Models (PLMs) have improved performance on various Natural Language Processing (NLP) tasks (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020). Probing tasks like LAMA (Petroni et al., 2019; Elazar et al., 2021; Jiang et al., 2020) have shown that PLMs can store factual knowledge and act as knowledge bases. Leveraging knowledge in PLMs can benefit knowledge-intensive downstream tasks such as fact checking and question answering (Lee et al., 2020; Bouraoui et al., 2020; Roberts et al., 2020a). However, knowledge stored in PLMs may have factual errors, which hinder the performance in downstream tasks (Elazar et al., 2021; Cao et al., 2021a). It is essential and fundamental to detect and calibrate false facts stored in a PLM.

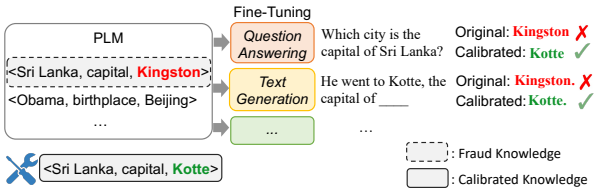

Figure 1: Illustration of knowledge calibration. Knowledge stored in PLMs have factual errors, which impairs model performance on question answering or generation. Knowledge calibration aims to rectifie these wrong knowledge.

In order to deal with the false facts, previous work focuses on complementing or modifying knowledge for a specific downstream task. Yao et al. (2022) proposed retrieving external knowledge during fine-tuning. Cao et al. (2021b) modified specific knowledge after finetuning. However, these methods do not generalize to multiple tasks.

In this paper, we explore a task-agnostic method to directly calibrate general factual knowledge in PLMs without re-training from scratch. We aim to correct the false facts in PLMs. Since every single fact has multiple surfaces, we also expect that the calibrated knowledge should be generalizable to various text surfaces. Figure 1 illustrates the process of calibration. First, we detect the false knowledge in PLMs with a Contrastive Knowledge Assessing (CKA) method (demonstrated in Figure 2). Since PLMs make black-box decisions, we evaluate PLMs via their predictions for simplification. The key motivation behind CKA is a plain argument that a PLM correctly learns a fact if and only if the model assigns the right fact higher scores than possible negative facts. For that false knowledge, we then propose CALINET to calibrate them by telling PLMs what the right fact is. Without compromising parameters in the original PLM, our approach calibrates the false knowledge by fine-tuning new parameters while the original parameters are fixed during calibration. Inspired by Dai

*Equal contribution.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5937–5947
December 7-11, 2022

et al. (2022) who state that the Feed-Forward Networks (FFNs) in PLMs store factual knowledge, we extend a specific FFN in the PLM with a calibrating FFN, which consists of several calibration memory slots. As shown in Figure 3, without modifying parameters in the original PLM, our approach calibrates the false knowledge through paraphrased natural sentences that express the corresponding correct facts.

Extensive experiments on probing tasks and question answering tasks demonstrate that CALINET calibrates false facts in PLMs efficiently and exhibits a remarkable generalization ability. We also analyze the calibration memory slots and the calibration mechanism to better understand how the proposed method works. Further, we explain how and where CALINET calibrates the factual knowledge in a PLM by tracing the evolution of the model prediction.

In summary, our contributions are three-fold:

- We propose a Contrastive Knowledge Assessment to evaluate factual knowledge stored in PLMs. The assessment shows that nearly 50% of facts randomly sampled from T-REx (El-Sahar et al., 2018) are stored incorrectly in PLMs.
- We propose CALINET to calibrate incorrect factual knowledge in PLMs. Without compromising parameters in original PLMs, our method can rectify incorrect knowledge and broadly generalizes well.
- We also investigate how CALINET works via calibration memory slots.

2 Contrastive Knowledge Assessment

The first step for calibration is to detect which wrong facts are learned by PLMs. We propose Contrastive Knowledge Assessment (CKA) and implement it to identify false knowledge in PLMs.

Traditional evaluation usually adopts rank-based metrics. It evaluates a PLM based on how highly it ranks the ground truth entity against other entities. However, it comes with two main problems. One is the problem of **inexhaustible answers**. The rank-based method fails to assess PLMs on multiple valid predictions. The top-1 only has one prediction, but the right predictions can be multiple. The other one is the problem of **frequency bias**. The ranking is particularly susceptible to the token frequency in the pretraining corpus. When the tail entity o frequently coexists with a head entity s, even if they express nothing about a specific fact, the model will still assign o a high rank when assessing this fact.

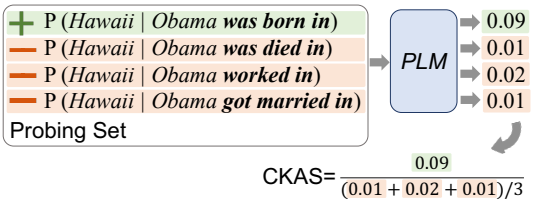

Figure 2: CKA assesses the knowledge stored in PLMs in a contrastive manner. The probing set includes the one positive probing prompt and several negative probing prompts. For simplification, we set $\alpha = 0$.

To address these limitations, we propose CKA to detect the false factual knowledge stored in PLMs. The core idea is assessing model prediction under a positive right fact and negative wrong facts in a contrastive manner. For each fact, we sample a prompt to transform it into natural text.

Let the triplet $\langle s, r, o\rangle$ denote a correct fact, where s, and o denote the subject entity and the object entity, respectively. We define r as the correct relation in a positive probing prompt, r' as the incorrect relation in a negative probing prompt.[1] For a PLM M, we consider the probability it assigns to o given $\langle s, r\rangle$ and $\langle s, r'\rangle$. As $\langle s, r, o\rangle$ is correct and $\langle s, r', o\rangle$ is erroneous, $P_M(o|s, r)$ should be larger than $P_M(o|s, r')$ if M knows the fact. Thus, CKA calculates the factual correctness of a fact $\langle s, r, o\rangle$ for the model M by

$$\text{CKA}_\text{M}(s, r, o) = \frac{P_M(o|s, r) + \alpha}{\mathbb{E}_{r'}\left[P_M(o|s, r')\right] + \alpha}, \quad (1)$$

where α is a smoothing factor. For a more stable comparison, we sample multiple erroneous relations r' for negative probing prompts and calculate the expectation of various $P_M(o|s, r')$.

In our implementation, the templates of the positive prompts come from LAMA (Petroni et al., 2019) and the the templates of the negative prompts are manually designed for quality guarantee. The negative prompts have contradictory semantics with the positive prompts but still prompt the same type of entities. For example, the positive prompt

[1] Our contrastive assessing framework is not limited to which part to be replaced for contrast. But relation replacement is more practical than entity replacement as relations are limited compared with entities.

template of <x, subclass of, y> is "[X] is the subclass of [Y]", and the negative prompt template can be "[X] is the parent class of [Y]".

An example of calculating the CKA score is shown in Figure 2. Further, we can set a threshold (usually < 1.0) for the CKA score to detect false knowledge in PLMs.

We compare the CKA score with the rank-based assessment used by previous work (Petroni et al., 2019) to show our advantages. As shown in Table 1, the rank-based knowledge assessment suffers from inexhaustible answers and frequency bias. In contrast, CKA evaluates each tail entity o independently, so we no longer need to know all the other valid objects. In addition, s appears in both the numerator and the denominator of the CKA score, which neutralizes the influence of the frequency bias.

3 Knowledge Calibration

The CKA method outputs which wrong facts a PLM learns. This section describes how we calibrate them.

Suppose that we have detected k false facts in a PLM. We aim to calibrate them to the correct ones so that the downstream tasks will not access false factual knowledge from the PLM. Previous work (Geva et al., 2021; Dai et al., 2022) point out that FFNs in Transformers can be regarded as key-value memories that store factual knowledge. Inspired by this, we design an FFN-like CALINET and take advantage of the properties of FFN to calibrate factual knowledge in PLMs directly. It is also important to note that the proposed method can be used to any part of the parameters. In this work, we apply the method on FFN because FFN is proven to take more responsibility when storing facts. In this section, we introduce the architecture of CALINET, the construction of the calibration data, and how to perform calibration on a pretrained model.

3.1 CALINET

In order to calibrate factual knowledge in PLMs, we propose a lightweight CALINET to adjust the output of FFNs in a pretrained Transformer. Let $H \in \mathbb{R}^{n \times d}$ denote the output of the attention layer in a Transformer block, the original FFN layer can be formulated as follows:

$$\mathrm{FFN}(H) = \mathrm{GELU}\left(HK^T\right)V,$$

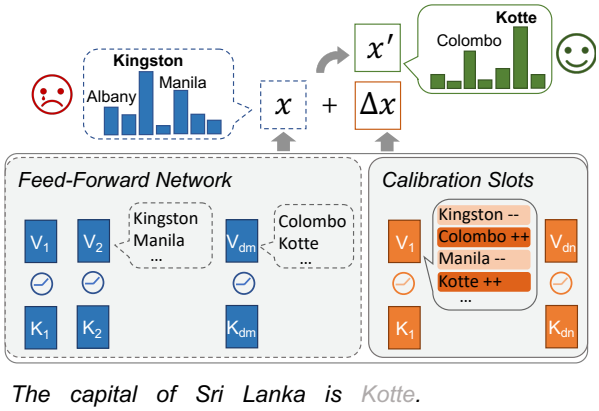

Figure 3: Illustration of CALINET. Calibration memory slots calibrate the erroneous knowledge stored in FFN by adjusting its predicted token distributions.

where $K, V \in \mathbb{R}^{d_m \times d}$ are parameter matrices of the first and second linear layers in FFN, respectively.

Our CALINET shares the same architecture with FFN but with a smaller intermediate dimension d_c. As shown in Figure 3, we deem each key-value pair as a calibration memory slot that stores factual knowledge. When computing the final FFN output, we add the output of CALINET to the original FFN output as an adjustment term for knowledge calibration, namely:

$$\Delta\mathrm{FFN}(H) = \mathrm{GELU}\left(H\tilde{K}^T\right)\tilde{V},$$
$$\mathrm{FFN}'(H) = \mathrm{FFN}(H) + \Delta\mathrm{FFN}(H),$$

where $\tilde{K}, \tilde{V} \in \mathbb{R}^{d_c \times d}$ are parameter matrices of CALINET, and $\mathrm{FFN}'(H)$ is the calibrated FFN output. Note that $d_c \ll d_m$, so our method just introduces quite a small number of parameters.

3.2 Calibration Data Construction

A fact can be expressed in multiple surface forms. For example, *"Obama was born in Hawaii."* and *"The birthplace of Obama is Hawaii"* describe the same factual knowledge. In order to calibrate a fact instead of merely fitting a specific surface form, we consider multiple paraphrased expressions for each fact. To be specific, we construct the calibration data based on the PARAREL dataset (Elazar et al., 2021), which contains various surface form templates for 38 relations. First, for each of the k detected false triplets, we fill the head entity or the tail entity into more than five paraphrased templates of its relation. Then, we replace the other entity with a mask token to be predicted. In this

Fact	Rank-based Assessment		CKA	
	Assess	Top-3 Prediction	Assess	Score
Inexhaustible Answers				
Germany shares border with *Czech Republic*.	✗	France, Russia, Austria	✓	4.45
India is a member of *UN*.	✗	NATO, India, AS	✓	2.27
Frederick was born in *Berlin*.	✗	Frederick, 18, Baltimore	✓	3.52
Frequency Bias				
Adi Shankara is affiliated with the *Hindu* religion.	✓	Hindu, Ko, Si	✗	0.98
Adi Shankara is against the *Hindu* religion.	-	Hindu, religion, Buddhist	-	-

Table 1: Instances of knowledge assessment to show the advantages of CKA from two aspects. Non-entity predictions are excluded. For CKA, we set a threshold that the model has a false fact if it gets a CKA score lower than 1. The rank-based method fails in assessing knowledge with multiple right answers (inexhaustible answers). For example, rank-based methods only filter knowledge with top-1 prediction for "Germany shares borders with [MASK].", the right answer "Czech Republic" will be ignored even if "Czech Republi" is in top-k predictions. The ranking is particularly susceptible to the entity co-occurrence during pretraining (frequency bias). For example, since "Hindu" coexists frequently with the "Adi Shanka", even if the prompt expresses nothing about a fact, the model ranks "Hindu" top-1. The instance in the last line is a control example about this situation but not a fact-probing instance, so there is no outcome.

Split	Source	Target
Train	[MASK] was born in Hawaii.	Obama
	Obama is originally from [MASK].	Hawaii
	[MASK] was originally from Hawaii.	Obama
	Obama is native to [MASK].	Hawaii
Valid	[MASK] originates from Hawaii.	Obama
	Obama originated from [MASK].	Hawaii
Test	Obama is a/an [MASK]-born person.	Hawaii
	[MASK] was native to Hawaii.	Obama
	Obama, a [MASK]-born person.	Hawaii

Table 2: Example of knowledge-intensive data for training CALINET. We generate multiple texts via templates for each triple where the templates in training, validation, and test are not sharing.

way, we obtain various paraphrased expressions for each fact. We divide these data into training, validation, and test sets where the templates in any two sets do not overlap. We show the example data for a fact <*Obama, born in, Hawaii*> in Table 2.

3.3 Model Calibration

With calibration data, we train CALINET via a masked language modeling objective. We freeze the original parameters of PLMs and only optimize the calibration memory slots to calibrate hidden states to factually correct ones. Only the new parameters are updated. In this way, the update will not affect the storage of other knowledge. During training, we also consider multiple paraphrased expressions for each fact such that the knowledge calibrated by CALINET can be generalized to various expressions.

4 Experiments

4.1 False Knowledge Detection

Datasets and Models We sample various scales of factual triplets from the T-REx dataset (ElSahar et al., 2018). For each triplet, we fill its head entity and tail entity into the template in LAMA (Petroni et al., 2019) according to the relation. As a result, we constructed datasets containing 100 facts and 1000 facts for false knowledge detection, where facts contain multiple sentences in their paraphrased form. We consider detecting the factual knowledge in $T5_{base}$ and $T5_{large}$ (Raffel et al., 2020) in our experiments.

False Rate We implement CKA for knowledge assessment and detection in PLMs. We use the False Rate to denote the proportion of false knowledge in PLMs. False Rate is the proportion of instances that have a CKA score lower than 1.0, which represents that the fact is not correctly learned by the model.

Experimental Settings We first calculate the CKA to detect false knowledge in T5. For each relation in LAMA, we manually write 3 erroneous relation templates. Then, for each fact, we fill the head entity into these templates to generate various negative probing prompts used in CKA. After that, we calculate the CKA score for each fact following Equation (1), where $\mathbb{E}\left[P_M(o|s,r')\right]$ is computed

Model	# Facts	Method	# Calibration Params	False Rate(↓)	Ori (↓)	Adv (↑)	LM(↓)	EM(↑)	F1(↑)
T5-base	10^2	Vanilla	0	48.10%	87.21	219.18	89.21	0.63	7.48
		CALINET	0.1M	17.09%	1.22	>1000	54.45	81.65	84.58
		C. P.	220M	13.29%	1.15	>1000	116.52	87.34	89.85
	10^3	Vanilla	0	51.34%	90.61	208.90	60.64	0.94	6.51
		CALINET	0.5M	18.30%	1.26	>1000	46.71	71.18	73.48
		C. P.	220M	18.23%	1.28	>1000	139.96	78.15	80.35
T5-large	10^2	Vanilla	0	46.20%	34.36	116.38	92.52	2.53	7.23
		CALINET	0.5M	15.19%	1.30	>1000	44.21	81.65	85.11
		C. P.	770M	14.56%	1.21	>1000	477.24	87.97	90.49
	10^3	Vanilla	0	45.04%	31.44	93.77	58.78	2.48	6.86
		CALINET	1.0M	20.84%	1.32	>1000	43.04	70.84	72.92
		C. P.	770M	17.16%	1.28	>1000	154.52	78.22	80.57

Table 3: False knowledge detection and calibration for 100 facts and 1000 facts. "Ori." and "Adv." refer to the original test set (contains true facts) and the adversarial test set (contains false facts), respectively. ↑ denotes that higher is better and ↓ denotes that lower is better. # Facts represents the scale of facts and # Calibration Params represents the number of parameters that participate in knowledge calibration. C. P. denotes the continue pretraining method for knowledge calibration. With adding only a few parameters, our CALINET achieves comparable performance on knowledge calibration compared with C. P. and has less negative impacts on the generalization ability.

by the average probability of the negative probing prompts. Finally, we identify the false factual knowledge in the PLM whose CKA score is lower than one and calculate the overall False Rate for the PLM.

Results As shown in Table 3, we find that the false facts account for nearly half of all the facts for T5-base based on the CKA metric. As for T5-large, which has a larger model capacity, its False Rate is slightly lower than T5-base but still relatively high. The disappointingly high False Rate in PLMs embodies the necessity to calibrate factual knowledge.

4.2 Calibrating False Factual Knowledge

4.2.1 Experimental Settings

For the detected false knowledge in PLMs, we construct the calibration data following Section 3.2. Our CALINET consists of 64 and 256 calibration memory slots for 100 and 1000 target facts, respectively. We concatenate CALINET to the last layer of the T5 decoder in our experiments. Following Gururangan et al. (2020), we continue pretraining on the calibration data (i.e., optimizing all the parameters) as an upper bound to reach. Appendix A shows detailed hyper-parameter settings.

4.2.2 Metrics

We evaluate the calibrated model from two aspects, the **knowledge modeling** ability and the **language modeling** ability.

For knowledge modeling ability, a model with good knowledge modeling ability should know which sentences are factually correct and which ones are factually wrong. For the former, we calculate the model perplexity on the original test set where the target is the correct entity. For the latter, we calculate the model perplexity on an adversarial test set whose target entity is replaced by a false one in the same entity type. In addition, we use Exact Match (EM) and F1 to further evaluate the generation correctness.

In order to evaluate the language modeling ability, we randomly mask the test data in the same manner as that in the pretraining stage and denote it as the LM test set.

4.2.3 Results

We show the results for knowledge calibration in Table 3. The calibration makes the model perplexity decrease on the original test set and increases on the adversarial test set. That is, compared to the original model, our method adjusts the model to "know" about the given facts. In addition, our method has little effect on the model perplexity on the general test set because the model parameters

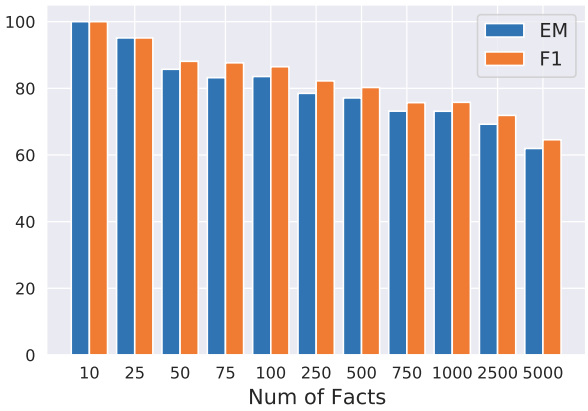

Figure 4: Calibration results for different scales of facts. Given 5000 facts, our method can calibrate more than 60% of facts in PLMs at once.

# Slots	100 Facts		1000 Facts	
	EM	F1	EM	F1
16	72.16	76.33	17.63	21.00
64	81.65	84.58	50.87	53.65
256	82.91	85.74	71.18	73.48
1024	82.91	85.43	72.92	75.21
3072	83.54	86.48	73.12	75.80

Table 4: Calibration ability with different numbers of calibration memory slots.

are not destroyed like fine-tuning; thus its semantic understanding ability is well-retained.

We also assess the knowledge correctness of the calibrated model. The improvement of Top1 prediction EM and F1 indicates that knowledge calibration enables the model to generate factually correct predictions. The overall False Rate calculated via CKA score decreases from 48.10% to 17.09%, which further validates the effectiveness of the CALINET.

4.2.4 Scalability of Knowledge Calibration

In order to delve deeper into the scale limitation of knowledge calibration, we apply our method to different scales of facts to be calibrated. As Figure 4 shows, when the number of facts to be calibrated is 10, the calibration EM score is 100%, i.e., the factual knowledge is perfectly calibrated. As the number of facts increases, the EM score will gradually decrease. Surprisingly, when the number reaches 5000, our method can still calibrate more than 60% of the facts in PLMs at once.

Compared with previous work on similar topics like knowledge editing (Cao et al., 2021b; Zhu et al., 2020; Cao et al., 2021a), we make huge progress in the amount of knowledge that can be calibrated at once. Mitchell et al. (2022) prove that batched editing for factual knowledge in PLMs is difficult. More concretely, when they modify more than 125 facts at once, the success rate of model editing has already been less than 70%. By contrast, in our method, the calibration EM score for 1000 facts is still greater than 70%.

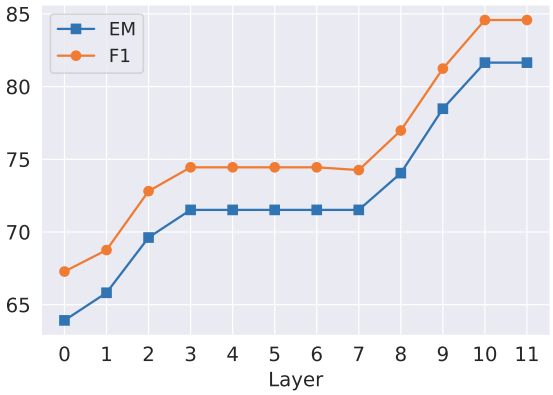

Figure 5: Calibration ability of concatenating CaliNet in different layers.

Model	Cali. Set		Uncali. Set		Overall	
	EM	F1	EM	F1	EM	F1
$T5_{WQ}$	0.00	7.95	32.41	38.24	29.94	35.93
$T5_{C.P.+WQ}$	8.46	14.27	32.72	38.58	30.88	36.73
$T5_{Cali+WQ}$	10.77	18.34	31.65	37.57	30.06	36.11
$T5_{TQ}$	0.00	14.01	23.53	29.75	21.63	28.47
$T5_{C.P.+TQ}$	6.91	20.18	22.35	28.65	21.09	27.96
$T5_{Cali+TQ}$	6.78	18.72	23.02	29.64	21.70	28.74

Table 5: Generalization ability of the calibrated knowledge in PLMs, evaluated by open-domain question answering. Cali. Set denotes the calibration subset, Uncali. Set denotes the subset without calibration. For WebQuestions (WQ), Cali. Set includes 81 questions, for TriviaQA (TQ) Cali. Set includes 811 questions.

4.2.5 Architectures of CALINET

Number of Calibration Memory Slots We conduct experiments with different calibration memory slots and show the results in Table 4. For calibrating 100 facts, we find that only 64 calibration memory slots is sufficient to achieve a performance close to that of 3072 slots. In terms of 1000 facts, 256 calibration memory slots are almost enough. In practice, we take the smallest number of calibration memory slots that can achieve relatively high performance for better calibration efficiency.

Position to Concatenate CALINET We concatenate CALINET to each FFN layer in the T5 decoder to study the difference on the calibration

ability. Figure 5 shows that deeper layers maintain stronger calibration ability and the last two layers achieve comparable calibration performance. We think this is because the knowledge calibration in the deeper layers will be affected less by other information in the model. This finding is also consistent with Dai et al. (2022), who find that the deeper layers store more factual knowledge.

4.3 Calibration Generalizability

Data Construction We validate the generalization ability of the calibrated knowledge in PLMs on two open-domain question answering datasets WebQuestions (Berant et al., 2013) and TriviaQA (Joshi et al., 2017). In order to obtain the facts to be calibrated, we fine-tune the T5 model on WebQuestions and TriviaQA without retrieving external knowledge bases. In this stage, the model learned to answer questions with its internal knowledge. According to their prediction correctness on the test set, we aggregate the questions that the PLM answers incorrectly. Then, we retrieve all the triplets, which include any entity in these questions from T-REx. Like in Section 3.2, we transform the triplets into paraphrased natural sentences for training CALINET.

Settings According to the facts to be calibrated, 64 calibration memory slots are trained for WebQuestions, and 256 calibration memory slots are trained for TriviaQA. After knowledge calibration, the calibrated PLM is further fine-tuned on the question answering tasks. We also use the continue pretraining method (C. P.) as an upper bound. Our hyper-parameter settings follow Roberts et al. (2020b).

Results The results are demonstrated in Table 5. We have the following findings. Firstly, with CALINET, the model performance improves on the calibration subset, which consists of the questions that T5 cannot correctly answer. It indicates that the calibrated knowledge in PLMs can be generalized to the question answering tasks. Secondly, the performance on the remaining questions (Uncali. Set) is hardly impacted. Thirdly, with only a few calibration memory slots, our method achieves a comparable knowledge calibration effect as continuing pretraining all the parameters. In addition, continuing pretraining will affect the language modeling ability of PLMs (refer to Table 3) while our method will not.

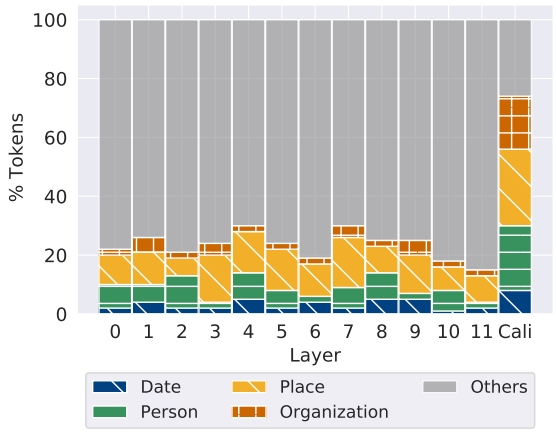

Figure 6: Meaning of values in original FFNs and CALINET. Nearly 80% of values in CALINET correspond to meaningful concepts used for knowledge calibration.

5 Interpretability of CALINET

In this section, we analyze CALINET on the memory slot level to interpret its meaning and working mechanism.

5.1 Meanings of FFN Values

Inspired by Geva et al. (2021, 2022), we cast each value vector in FFNs or CALINET as an input-independent distribution over the output vocabulary for analyzing it meaning:

$$\mathbf{p}_i^\ell = \text{softmax}(E\mathbf{v}_i^\ell),$$
$$\mathbf{p}_k^c = \text{softmax}(E\mathbf{v}_k^c),$$

where $\mathbf{v}_i^\ell$ denotes the i-th value in the ℓ-th FFN layer, $\mathbf{v}_k^c$ denotes the k-th calibration memory slot in CALINET, E denotes the output embedding matrix.

In order to reveal what kinds of knowledge are stored in FFNs and calibrated by CALINET, we manually annotate the meaning of each value according to its top-ranked tokens. Specifically, we randomly sample 100 values from each FFN layer in the original T5 decoder and 100 values from CALINET For each value, we examine the top-30 tokens with the highest probabilities according to $\mathbf{p}_i^\ell$ or $\mathbf{p}_k^c$. Following Geva et al. (2022), we manually identify patterns that occur in at least 4 tokens and categorize them into "person", "place", "organization", "date" and "others".

We illustrate the annotation results in Figure 6 and find that the values in CALINET are more knowledge-intensive compared with values in the original FFNs. Specifically, nearly 80% of values

Input	Alice Hollister is a <extra_id_0> by profession. (Target output: film actress)
Layer 8	writer, professional, musician, journalist, freelance, **lawyer**, doctor, woman, retired, scientist
Layer 9	**lawyer**, writer, journalist, professional, freelance, scientist, doctor, teacher, pharmacist, musician
Layer 10	writer, **lawyer**, professional, freelance, journalist, doctor, teacher, veterinarian, psychologist, nurse
Layer 11	**lawyer**, writer, nurse, doctor, journalist, teacher, professional, psychologist, social, solicitor
Layer 11 w/ CaliNet	**film**, Film, films, filmmaker, movie, journalist, actor, cinema, theatre, **actress**
Input	Le Matin, an <extra_id_0>-language work. (Target output: francophone)
Layer 8	**English**, independent, artist, American, experimental, international, original, award, example, ethno
Layer 9	**English**, international, American, Italian, ethno, Australian, experimental, original, independent, art
Layer 10	**English**, Italian, ethno, international, American, African, art, Irish, experimental, original
Layer 11	**English**, French, Italian, original, interpret, Arabic, American, expressive, in, early
Layer 11 w/ CaliNet	**francophone**, French, L, Franco, theatre, **English**, Le, Italian, Toulouse, french

Table 6: Evolution of the output token distributions. Bold red tokens refer to wrong tokens predicted without calibration. Bold and light blue tokens refer to correct tokens predicted after calibration and their semantic-related tokens, respectively.

in CaliNet correspond to meaningful concepts used for adjusting the hidden states to calibrate the factual knowledge.

5.2 Working Mechanism of CaliNet

We further reveal the working mechanism of knowledge calibration by tracing the evolution of the output distribution in different layers. Let $\mathbf{x}$ be the input hidden state of an FFN layer, $\hat{\mathbf{x}}$ be the output of the FFN. Following Geva et al. (2022) and taking the residual connection into consideration, we define the output token distribution of this FFN layer by

$$\mathbf{y} = \text{softmax}\left(E(\hat{\mathbf{x}} + \mathbf{x})\right).$$

Let $\tilde{\mathbf{x}}^c$ denotes the output of CaliNet. If we concatenate CaliNet to this FFN layer, the output token distribution will become

$$\tilde{\mathbf{y}} = \text{softmax}\left(E(\hat{\mathbf{x}} + \mathbf{x} + \tilde{\mathbf{x}}^c)\right).$$

For the last four FFN layers, we show the top-10 tokens with the highest probabilities according to the output token distribution in Table 6. Also, we provide the top-10 tokens after knowledge calibration. We find that the factually incorrect predictions are usually high-frequency tokens like "English" or "lawyer". However, the original FFNs in the PLM have little effect on the output token distribution, especially on the top-ranked tokens. By contrast, CaliNet can adjust the output token distribution greatly and produce the correct result. More notably, CaliNet not only increases the probability of the factually correct token but also increases the probability of tokens that are synonyms of the correct token. This indicates that our method can calibrate the factual knowledge in a generalized way instead of just learning the surface forms of a fact.

6 Related Work

Knowledge Correctness in PLMs Large-scale pretrained language models are commonly seen as non-symbolic KBs containing factual knowledge. To assess the knowledge stored in PLMs, Petroni et al. (2019) introduce the rank-based LAMA probing and define that a PLM knows a fact if it successfully predicts masked objects in cloze-style sentences. Jiang et al. (2020) give a tighter lower bound than LAMA(Petroni et al., 2019) on what PLMs know by designing better prompts. However, Elazar et al. (2021) observe that rank-based probing methods are not robust against paraphrased context, leading to inconsistent results. Some other work (Pörner et al., 2019; Cao et al., 2021a) points out that the ability of PLMs to store knowledge is overestimated due to biased prompts and golden answer leakage.

Knowledge Injection into PLMs Many studies have explored integrating external knowledge into PLMs to enhance their performance on knowledge-intensive tasks. ERNIE (Zhang et al., 2019) and KnowBERT (Peters et al., 2019) incorporate knowledge graphs to provide structured knowledge during pretraining. K-adapter (Wang et al., 2021) injects factual and linguistic knowledge into PLM with adapters, which are pretrained on two structured prediction tasks. Kformer (Yao et al., 2022) also extends FFN in PLMs. In their work, the knowledge is converted into dense embedding and directly injected into the extended FFN. In contrast to all previous work, CaliNet is pretrained

with paraphrased natural sentences to fully exploit the semantic modeling capability of PLMs, and the calibrated knowledge can be utilized in any downstream tasks.

Knowledge Editing Given a revised fact set, the objective of knowledge editing is to seek alternative parameters so that the model can make new predictions on revised instances while keeping all the other predictions unchanged. Zhu et al. (2020) formulate the knowledge editing task as a constrained optimization problem and create a benchmark to evaluate the effectiveness of knowledge editing methods. Cao et al. (2021a); Mitchell et al. (2022) introduce a hypernetwork to modify a fact without affecting the rest of the knowledge. Meng et al. (2022) develop a causal intervention for locating and editing knowledge in GPT-style models. Current knowledge editing approaches mainly aim to modify the model after fine-tuning, which will hinder the generalization of knowledge stored in PLMs. In contrast, through calibrating factual knowledge before fine-tuning, our proposed method can rectify the knowledge in models and broadly generalizes the calibrated knowledge for downstream tasks.

7 Conclusion

In this paper, we reassess the knowledge stored in PLMs in a contrastive manner and detect the incorrect knowledge stored in PLMs. We propose CALINET, which adds new parameters to calibrate the knowledge stored in PLMs at scale without updating the original model parameters. The knowledge-calibrated PLMs generalize calibrated knowledge well and perform better than original PLMs on various downstream tasks like open-domain QA. We further provide neuron-level investigations on the calibration mechanism and study how calibration works.

Limitations and Future Work

Despite the effectiveness of knowledge calibration, our current studies still have several limitations.

First, our knowledge assessing and knowledge calibration approach relies on existing knowledge bases and synthetic data. It is a long-term goal to achieve a full-scale knowledge assessment or knowledge calibration because knowledge is complicated. Compared to inaccurate remote supervision and expensive human annotation, our template-filling solution is a relatively efficient solution for calibration data generation. However, our template-filling solution still builds synthetic test data rather than real test data for CKA. To explore the applicability of CALINET in practice, we recruit three human annotators to write 50 test facts. Specifically, following the contrastive framework in CKA, annotators write one positive sentence and three negative sentences for each fact. The positive sentence state a true fact. The negative sentence must contain the same relation as the positive sentence but a false object entity. Experiments show that CALINET effectively reduces the False Rate by 35.61% on real test data, consistent with our results on test data construct via template-filling. However, this work still has a lot of room for improving the calibration applicability in reality.

Second, We evaluate PLMs via their predictions. It is somehow a biased approach. Appendix B provides some negative cases of the CKA score. It is an open research question to assess the factual knowledge correctness in PLMs accurately.

Third, the current method cannot completely calibrate all the factual errors in PLMs. We expect that future work can present more advanced knowledge calibration methods.

Acknowledge

This paper is supported by the National Key Research and Development Program of China 2020AAA0106700 and NSFC project U19A2065.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of EMNLP*.

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of AAAI*.

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, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey 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. In *Proceedings of NeurIPS*.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021a.

Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of ACL*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021b. Editing factual knowledge in language models. In *Proceedings of EMNLP*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *TACL*, 9.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of LREC*.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *CoRR*, abs/2203.14680.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of EMNLP*.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *TACL*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*.

Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language models as fact checkers? In *Proceedings of FEVER*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual knowledge in GPT. *CoRR*, abs/2202.05262.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *Proceedings of ICLR*.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of EMNLP-IJCNLP*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP*.

Nina Pörner, Ulli Waltinger, and Hinrich Schütze. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *CoRR*, abs/1911.03681.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020a. How much knowledge can you pack into the parameters of a language model? In *Proceedings of EMNLP*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020b. How much knowledge can you pack into the parameters of a language model? In *Proceedings of EMNLP*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of ACL-IJCNLP*, pages 1405–1418.

Yunzhi Yao, Shaohan Huang, Ningyu Zhang, Li Dong, Furu Wei, and Huajun Chen. 2022. Kformer: Knowledge injection in transformer feed-forward layers. *CoRR*, abs/2201.05742.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of ACL*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *CoRR*, abs/2012.00363.

Appendix

A Implementation Details

We conduct experiments based on HuggingFace[2] and follow their default hyperparameter settings unless noted otherwise. We use grid search for learning rate from {1e-2, 1e-3, . . . , 1e-4}. We conduct all the experiments on a single A40 GPU.

For knowledge calibration, we use a constant learning rate scheduler and the Adafactor optimizer. The training and evaluating batch size is 512, with gradient accumulation steps set to 4. The max sequence length of the source sentence is 64, and that of the target length is 8. Our warm-up steps are 100. Our CALINET Training and continue pretraining steps are 5000 steps for 100 facts and 50000 steps for 1000 facts.

For fine-tuning on WebQuestions and TriviaQA, our hyperparameter follows the setting of Roberts et al. (2020b). The max training steps are 4000 steps.

B Negative Case of CKA

Although the CKA score solves the problems of rank-based metrics towards inexhaustible answers and frequency bias, it may fail to make an accurate assessment in some situations. Especially when the number of negative probing prompts is small, the CKA score can be easily biased. For example, for the relation 'P103' on native language, our positive template is "The native language of [X] is [Y] .", our negative templates are "[X] cannot speak [Y] .", "[X] have learned [Y] ."," [X] is teaching [Y] ." The average CKA score of 1,000 probing facts is 13.92. This surprisingly high score overestimates the knowledge of T5 in the native language because the second and the third negative templates have a larger scope than the positive template, resulting in a low negative score.

[2] https://github.com/huggingface/transformers