Assessing Knowledge Editing in Language Models via Relation Perspective

Anonymous ACL submission

Abstract

Knowledge Editing (KE) for modifying factual knowledge in Large Language Models (LLMs) has been receiving increasing attention. However, existing knowledge editing methods are entity-centric, and it is unclear whether this approach is suitable for a relation-centric perspective. To address this gap, this paper constructs a new benchmark named **RaKE**, which focuses on **R**elation b**a**sed **K**nowledge **E**diting. In this paper, we establish a suite of innovative metrics for evaluation and conduct comprehensive experiments involving various knowledge editing baselines. We notice that existing knowledge editing methods exhibit the potential difficulty in their ability to edit relations. Therefore, we further explore the role of relations in factual triplets within the transformer. Our research results confirm that knowledge related to relations is not only stored in the FFN network but also in the attention layers. This provides experimental support for future relation-based knowledge editing methods.

1 Introduction

Large Language Models (LLMs), trained on large-scale knowledge corpora such as Wikipedia, exhibit remarkable performance across various natural language processing tasks (Ma et al., 2023; Lei et al., 2023). However, current LLMs face challenges posed by errors, biases, and inappropriate information (Neeman et al., 2022; Guo et al., 2022). Meanwhile, LLMs need to adapt to emerging knowledge over time and eliminate outdated knowledge (Kasai et al., 2022; Wei et al., 2023). To maintain the accuracy and reliability of LLMs, the task of Knowledge Editing (KE)[1], which involves modifying and updating the internal knowledge of language models, has recently gained significant attention.

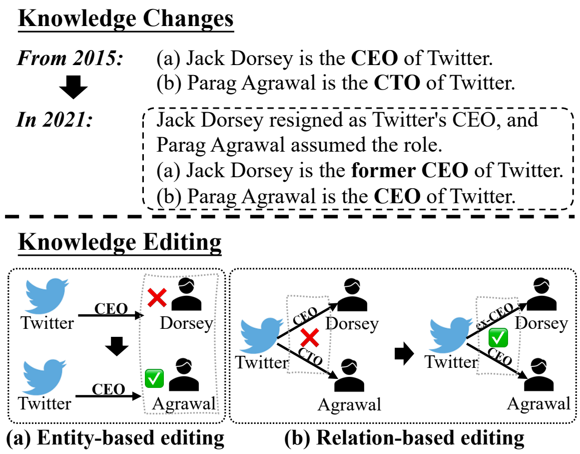

Figure 1: As time progresses, relationships between entities undergo continuous changes. In real-world scenarios, such as Wikipedia, updating factual knowledge sometimes necessitates the modification of relationships to accurately reflect evolving information.

The factual knowledge encapsulated in language models can be represented as the relation between subject and object in the form of (s, r, o)[2]. As time progresses, the relations between entities also undergo changes, as illustrated in Figure 1 (b). For instance, consider the evolution of Parag Agrawal's role at Twitter[3]: *"From 2015, Parag Agrawal is the **CTO** of Twitter,"* transforms into *"In 2021, Parag Agrawal is the **CEO** of Twitter."* The intuitive need arises to directly modify the relation ("CTO" to "CEO") to accurately reflect this evolving knowledge. However, existing attempts focuses on editing from the entity perspective (De Cao et al., 2021; Mitchell et al., 2021, 2022; Dong et al., 2022; Huang et al., 2023; Dai et al., 2021; Meng et al., 2022a,b; Zheng et al., 2023; Zhong et al., 2023), ignoring the modification of factual knowledge from the relation perspective.

To fill this gap, we construct a **R**elation-b**a**sed

[1] In this paper, the term "knowledge editing" is equivalent to "model editing" and "memory editing".

[2] Knowledge triples: (subject entity, relation, object entity).

[3] https://en.wikipedia.org/wiki/Twitter,_Inc.

Knowledge **E**diting benchmark called **RaKE**, and extend previous evaluation principles (Mitchell et al., 2021; Elazar et al., 2021) to the perspective of relation. Then, we empirically investigate the outcomes of existing methods on relation-based editing. Surprisingly, the experimental results reveal that relation-based editing lags far behind entity-based editing, contradicting the expectation of their consistency as they pertain to the same factual knowledge. To delve into the reasons causing such inconsistency, we conduct a causal tracing analysis on the relation r within the factual knowledge and investigate how and where the relation memories are stored in LLMs. The results show evidence that the relation memories are not only related to the feed-forward network (FFN) but also to the attention layer. Due to the fact that entity-based methods primarily modify parameters within the feed-forward network (FFN), our experiments indicate that the underperformance of current relation-based editing stems from a lack of modification to knowledge neurons associated with the attention layer. We hope that our work can provide the NLP community with insights.

Our main contributions are summarized as follows:

- For the first time, we identify the importance of knowledge editing from a relational perspective and construct a new benchmark, RaKE, tailored for relation-based editing.

- We conduct extensive experiments using various baseline methods, and the results reveal significant limitations in the current approaches to relation-based editing.

- Our results confirm the crucial role of not only the feed-forward network but also the attention modules in storing relational knowledge. This insight provides valuable guidance for future KE research.

2 Preliminaries

In this section, we will illustrate the proposed relation-based editing task in Figure 2. We will discuss the task definition (§2.1), and explain the evaluation metrics (§2.2).

2.1 Task Definition

Following the work of (Petroni et al., 2019), we adopt the definition that a large language model possesses knowledge of a fact P in the form of (s, r, o). In this context, s represents a subject entity (e.g., Lyon), r represents a relation (e.g., twin city), and o represents an object (e.g., Beirut). We also use a few variations of the data for the fact (s, r, o). The additional variables include:

1. s^* represents a neighboring entity to the subject s (e.g. *"Cairo"* is a neighboring entity to *"Lyon"*), for which (s^*, r, o) is a true fact like (s, r, o).

2. r^* is a paraphrase of the relation r between the subject s and object o, such as *"[s] works in the field of [o]"* for *"[s] works in the area of [o]."*

3. o^c is the original object that correctly completes the fact $(s, r, \cdot)$, and o^* is a new object after editing updates.

As show in Figure 2, we can establish the logical equivalence of the factual knowledge P between entity perspective and relation perspective. In this paper, we propose that the fact P signifies the natural language prompt "The relation between Lyon and Beirut is ___" where the relation r needs to be completed. The main objective of the model editing task is to modify a base model f_θ, parameterized by θ, to gain control over the model's prediction outputs. Specifically, the base model f_θ is represented by a function $f : \mathbb{X} \mapsto \mathbb{Y}$ that associates an input P with its corresponding prediction r, as show in Equation 1.

$$f_\theta(P) = \begin{cases} \underset{\theta}{\mathrm{argmax}} p_\theta(r \mid s, o) & \text{if } o \in o^* \\ \underset{\theta}{\mathrm{argmin}} p_\theta(r \mid s, o) & \text{if } o \in o^c \end{cases} \quad (1)$$

To achieve control over the model's output, we aim for the model's conditional probability $p_\theta(r|s, o^*)$ to be maximized and $p_\theta(r|s, o^c)$ to be minimized. Here, o^c represents the original object entity, and o^* represents the modified object entity.

2.2 Evaluation Metrics

Model editing methods are commonly evaluated according to three aspects: Efficacy: their effectiveness in altering the model prediction for the input prompt P. Generalization: generalize to paraphrases of the prompt P. Specificity: avoid side effects on irrelevant fact knowledge.

Entity-based Editing	Input Prompt		Objective
Delete Object o^c	What is the twin city of Lyon? It is ______	$\longrightarrow$	$\underset{\theta}{\text{argmin}}\, p_\theta(o^c \mid \text{Input})$
Add Object o^*	What is the twin city of Lyon? It is ______	$\longrightarrow$	$\underset{\theta}{\text{argmax}}\, p_\theta(o^* \mid \text{Input})$
Relation-based Editing	**Input Prompt**		**Objective**
Delete Relation r	The relation between Lyon and $\overbrace{\text{Beirut}}^{o^c}$ is ___	$\longrightarrow$	$\underset{\theta}{\text{argmin}}\, p_\theta(r \mid \text{Del Input})$
Add Relation r	The relation between Lyon and $\overbrace{\text{Manila}}^{o^*}$ is ___	$\longrightarrow$	$\underset{\theta}{\text{argmax}}\, p_\theta(r \mid \text{Add Input})$

Figure 2: Depiction of editing problem variants, where r represents the relation P190 "twin city," o^c and o^* respectively represent the original object and the new object after editing. We can establish the logical equivalence of the editing results from both perspectives. Instead of modifying a new object fact within the model (Entity-based Editing), we consider directly modifying the relation output (Relation-based Editing).

In particular, we gather a set of more difficult false facts (s, r, o^*), these counterfactuals start with low scores compared to the correct facts (s, r, o^c). Our editing objective is to establish a relationship r between s and o^* while severing the connection r between s and o^c. To assess the efficacy of changes about relation, we divide the evaluation metrics into two: Success and Magnitude. The Success is the proportion of cases for which we have $p(r^*) > p(r^c)$ (or $p(o^*) > p(o^c)$) post-edit, and Magnitude is the average difference $p(r^*) - p(r^c)$ (or $p(o^*) - p(o^c)$). In details, we report Efficacy Success (ES) and Efficacy Magnitude (EM) to assess the efficacy of changes about relation, we collect a set of rephrased prompts equivalent to P and report Paraphrase Scores (PS) and (PM), we collect a set of nearby subjects s_n for which (s_n, r, o^c) holds true to measure Neighborhood Score NS and NM , computed similarly to ES and EM. To test three metrics tradeoff, we report the harmonic mean of ES, PS, NS as Score (S).

3 RaKE: Relation-based Knowledge Editing

A factual knowledge can be represented by a triplet (s, r, o). In the entity perspective, the current approach predicts the object based on the given prompt (s, r). In the relation perspective, it is equivalent to completing the relationship between the subject and object given (s, o). For example, "What is the twin city of Lyon? It is ___", for which the expected completion is o = "Beirut". Equivalent to: "The relation between Lyon and Beirut is ___", for which the expected completion is r = "twin city". To evaluate the editing capability of the current editing method for relation knowledge, we follow the dataset COUNTERFACT (Meng et al., 2022a) and construct an equivalent relation perspective dataset named RaKE. We first present the data construction process for the dataset. Then, we present the data statistics and evaluation settings of the RaKE, followed by evaluation metrics in the end.

3.1 Dataset Construction

Generalization Dataset Construction. To compare and assess semantic generalization of the language model in the relation perspective, we collect relations based on Wikidata (Vrandečić and Krötzsch, 2014), a knowledge base consisting of fact triples associated with thousands of relations. We first manually select 34 common relations from wikidata and then leverage the PARAREL dataset (Elazar et al., 2021) to get paraphrase for relations. Finally, we construct relation paraphrase prompts using manually designed templates, such as: "When it comes to subject and object, the relation can be defined as ___". We also adopt GPT3.5-turbo model to ensure that the sampled fact triples are coherent and lead to natural questions about relations, such as: "What is the correlation between Danielle Darrieux and English?".

Efficay Dataset Construction. In this paper, we define the knowledge editing task from a relational perspective using two atomic operations. 1) Delete operation: Removing the relation r between s and o. 2) Add operation: Adding the rela-

Criterion	zsRE	PARAREL	COUNTERFACT	Calibration	MQuAKE	RIPPLEEDITS	RaKE
Entity Efficacy	✓	✓	✓	✓	✓	✓	✓
Entity Paraphrase	✗	✓	✓	✓	✓	✓	✓
Specificity	✗	✗	✓	✗	✓	✓	✓
Multi-hop	✗	✗	✗	✗	✓	✗	✗
Relation Efficacy	✗	✗	✗	✗	✗	✗	✓
Relation Paraphrase	✗	✗	✗	✗	✗	✗	✓

Table 1: Comparison to Existing Benchmarks. While previous benchmarks have defined factual knowledge in the form of triples (s, r, o), existing paradigms assess whether an "entity-based" edit $(s, r \to o^*)$ is successful, but lack evaluation for the equivalent knowledge $(s, o^* \to r)$.

tion r between s and o^*, as illustrated in Figure 2. By utilizing these two atomic operations, we have achieved the logical equivalence to the entity-based editing method. We manually designed templates for these two atomic operations and constructed efficacy prompts for all facts by filling the slots.

3.2 Dataset Comparison

Table 1 shows a comprehensive comparison of related datasets. RaKE is the first dataset to study relation-based knowledge editing over language models. Due to the fact that factual knowledge is composed of tuple (s, r, o), any change in one of these components will result in a transformation of the associated knowledge. Therefore, for the expression of the same factual knowledge, there are two perspectives: the relation perspective and the entity perspective. There exists a mutual dependence and feedback relationship between these two perspectives. Compared with previous benchmarks, RaKE takes into account editing problem variants and incorporates evaluation prompts related to relation editing. It assesses the effectiveness of edits from a relational perspective, rather than solely measuring the accuracy of predicting the tail entity.

3.3 Dataset Statistics

The RaKE dataset consists of 21,919 editing samples, each of which can be categorized as either entity-based or relation-based. Each sample includes editing prompts for modifying knowledge, as well as Paraphrase Prompts and Neighborhood Prompts. Specifically, the entity-based category contains 21,919 Edit Prompts, 82,650 Neighborhood Prompts, and 43,838 Paraphrase Prompts. The relation-based category includes 43,838 Edit Prompts, 284,102 Paraphrase Prompts. Both categories share the entity-based Neighborhood Prompts to assess the impact on unrelated knowledge. The dataset statitics are summarized in Table 2.

Type	N_{Entity}	$N_{Relation}$
Edit Prompts	21919	43838
Neighborhood Prompts	82650	-
Paraphrase Prompts	43838	284102

Table 2: Statistics of RaKE. N_{Entity} and $N_{Relation}$ represent the number of samples in the entity perspective and relation perspective, respectively.

4 Experiments

In this section, we compare the performance differences between entity-based editing and relation-based editing and identify weaknesses in LLMs with respect to editing relations. The results of these comparisons are displayed in Table 3. Furthermore, we analyze the storage and recall of relation memory in LLMs through Casual Tracing, as show in Figure 4.

4.1 Experimental Setup

Language models. We use GPT-2 XL (1.5B) and GPT-J (6B) as the baseline models to assess model editing methods. In our experiment, we utilize four NVIDIA RTX A6000 GPUs and ten NVIDIA GeForce RTX 3090 GPUs to run model editing approaches.

Model editing methods. In this paper, our focus is on transformer-based language models, specifically exploring the connection between model parameters and memory. Therefore, we employ memory-based and Locate-Then-Edit paradigms as our model editing methods.

- **Finetune.** Fine-tuning is a commonly used approach for adapting pre-trained language models to specific tasks or domains.In this paper, we compare with a naive fine-tuning approach that uses weight decay to prevent forgetfulness (FT).

Editor		Score	E-Efficacy		R-Efficacy		Specificity		E-Generalization		R-Generalization	
		S ↑	ES ↑	EM ↑	ES ↑	EM ↑	NS ↑	NM ↑	PS ↑	PM ↑	PS ↑	PM ↑
Entity Perspective												
GPT-2 XL	FT	72.98	99.28	92.1	97.19	0.12	70.06	3.6	48.21	0.38	76.14	0.09
	KN	46.42	30.45	-2.08	83.42	0.08	69.19	1.98	28.8	-1.92	72.93	0.05
	MEND	67.81	93.8	45.27	97.91	0.12	44.44	-6.61	58.0	7.88	76.22	0.08
	ROME	87.01	99.93	97.94	96.12	0.17	75.36	4.4	96.6	62.91	74.46	0.09
	MEMIT	83.78	93.88	64.06	97.28	0.13	76.75	4.97	79.6	26.24	76.0	0.09
GPT-J	MEND	69.0	97.43	72.12	91.91	0.11	53.15	-5.44	53.53	11.12	72.34	0.08
	ROME	87.51	99.99	99.49	91.37	0.13	78.61	5.3	99.49	77.21	74.52	0.09
	MEMIT	87.43	99.81	97.05	92.36	0.14	80.97	6.81	95.07	50.73	74.2	0.10
Relation Perspective												
GPT-2 XL	FT	42.76	23.92	-4.76	98.79	29.19	76.69	5.05	25.44	-4.13	79.03	2.19
	KN	41.23	22.53	-4.92	97.52	0.12	77.72	5.17	24.61	-4.09	76.16	0.08
	MEND	41.57	22.33	-4.94	100.0	14.78	77.63	5.2	24.63	-4.13	83.24	1.7
	ROME	47.27	27.92	-3.7	99.99	86.7	77.88	5.09	28.12	-3.76	84.47	15.16
	MEMIT	42.03	24.15	-4.11	91.36	3.84	77.66	5.13	24.63	-4.04	76.24	0.73
GPT-J	MEND	32.38	15.51	-7.26	100.0	45.96	82.77	7.58	17.99	-6.65	81.52	5.11
	ROME	51.98	30.95	-3.83	100.0	98.51	82.75	7.54	31.87	-3.76	95.97	28.18
	MEMIT	36.27	18.92	-7.62	100.0	91.76	82.81	7.54	19.37	-7.82	88.5	13.21

Table 3: The performance of knowledge editing approaches. In the table, the prefix R represents relation, and E represents Entity.

- **KN.** The Knowledge Neuron (KN) method (Dai et al., 2021) introduces a knowledge attribution technique to identify the "knowledge neuron" (a key-value pair in the Feed-Forward Network matrix) encapsulate important memory. These neurons are then updated to incorporate relevant knowledge.

- **MEND.** Model Editor Networks with Gradient Decomposition (Mitchell et al., 2021) enables efficient local edits to language models by transforming the gradients of fine-tuned models. It achieves this by utilizing a low-rank decomposition of the gradients.

- **ROME.** Meng et al. (2022a) applies causal mediation analysis to locate the specific areas requiring modifications. Instead of modifying individual knowledge neurons in the FFN, ROME iteratively updates one fact at a time by altering the entire matrix.

- **MEMIT.** Meng et al. (2022b) is a method that allows for simultaneous modification of a sequence of layers in a language model. It facilitates thousands of alterations to be executed efficiently.

4.2 Results and Analysis

Efficacy. Entity-based editing centers on the task of completing the object based on a prompt comprising a subject and relation. Conversely, relation-based editing pertains to the task of finalizing the relationship between a subject and object, given a prompt containing the subject and object. Grounded in the presumption of an inherent equivalence relationship between these two editing paradigms, we posit that **altering the object is fundamentally tantamount to introducing a relation between the head entity and the tail entity**. However, according to Table 3, we observe that current model editing methods are not applicable to relation perspective. Specifically, R-Efficacy shows a significant decrease in performance compared to E-Efficacy in terms of the EM metric. This suggests that the existing editing methods, which work well for entity perspective tasks, do not effectively handle relation perspective tasks. There is a clear performance gap when it comes to editing relations, indicating the limitations of LLMs in accurately capturing and generating complex relationships between entities. This finding highlights the need for further research and development of editing methods specifically tailored for relation perspective tasks, aiming to improve the performance and efficacy of LLMs in relation completion and understanding.

Geralization. We evaluate all methods on GPT-2 XL with knowledge edit in RaKE. The evaluation results are shown in Table 3. From the results of the Entity based Generalization and Relation based

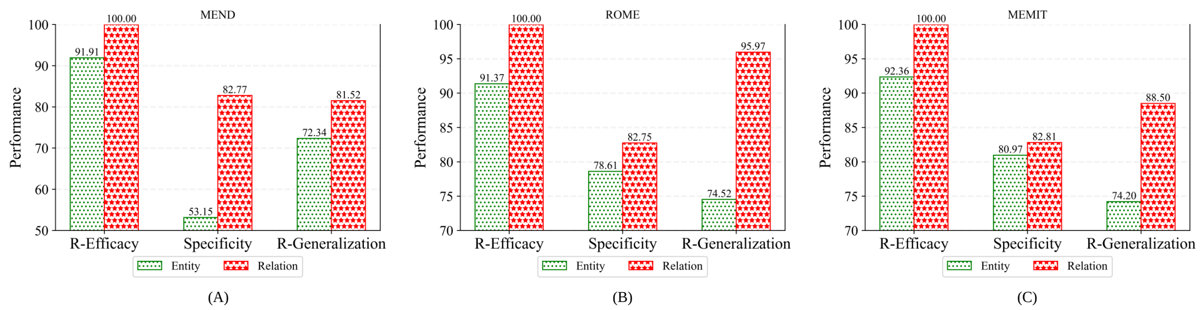

Figure 3: Figures (A), (B), and (C) demonstrate the advantages of relation editing over entity editing, based on the GPT-J model.

Generalization metrics, we can conclude that both entity-based and relation-based methods improve the generalization within their respective perspectives. However, their impact on generalization from the other perspective is relatively limited. Despite the logical equivalence of the knowledge edited from the entity and relation perspectives in terms of triple representation, they exhibit surprising differences in effectiveness. This leads us to speculate that **entity-based knowledge and relation-based knowledge are not equivalent in language models**. Specifically, entity knowledge and relation knowledge demonstrate a certain level of independence and are stored in different parts of the model.

4.3 Advantages and Disadvantages of the Relation Perspective

Based on Figure 3, the advantages of the relation perspective in editing can be observed in three aspects: R-Efficacy, Specificity, and R-Generalization. This indicates that by modifying relation rather than entity, we can **enhance the success rate of relation updates**, which is consistent with intuition, updating relationships through relation perspective editing is more efficient; and **reduce side effects on unrelated knowledge**, which limit the editing effect to the specified knowledge only; and **improve the generalization of relation-related information**, which modifies the knowledge level rather than the surface text level.

Based on the performance of existing editing methods, the relation perspective editing also has significant drawbacks. For example, the Score metric results show a significant decrease. Specifically, the knowledge updated through relation editing is difficult to transfer to entity, resulting in a substantial decline in E-Efficacy and E-Generalization, as show in Table 3.

4.4 Casual Tracing

To explore the role of relations in factual triplets (s, r, o) within model parameters, we need to analyze and identify the knowledge neurons that have the strongest causal effect on relations. We utilized causal tracing for this purpose, involving three steps as follow:

- **Clean run:** we pass a factual prompt x into a model f_θ and collect all hidden activations $\{h_i^{(l)} \mid i \in [1,T], l \in [1,L]\}$, where T is number of input tokens and L is number of layers within model f_θ.
- **Corrupted run:** The relation embeddings are obfuscated from f_θ before the network runs, after x is embedded as $[h_1^{(0)}, h_2^{(0)}, \dots, h_T^{(0)}]$, we set $h_i^{(0)} := h_i^{(0)} + \epsilon$ for all indices i that correspond to the relation, where $\epsilon \sim \mathcal{N}(0, \nu)$[4], and then we get a set of corrupted activations $\{h_{i*}^{(l)} \mid i \in [1,T], l \in [1,L]\}$.
- **Corrupted-with-restoration run:** Let f_θ runs computations on the noisy embeddings as in the corrupted baseline, except at some token $\hat{i}$ and layer $\hat{l}$. There, we hook f_θ so that it is forced to output the clean state $h_{\hat{i}}^{(l)}$, and future computations execute without further intervention.

In our settings, $\mathbb{P}[r]$, $\mathbb{P}_*[r]$, and $\mathbb{P}_{*,\text{clean } h_i^{(l)}}[r]$ is defined as the probability of final prediction r under the clean, corrupted, and corrupted-with-restoration runs, respectively. The indirect effect (IE) of a particular hidden state h_i^l is calculated as:

$$\mathrm{IE} = \mathbb{P}_{*,\text{clean } h_i^{(l)}}[r] - \mathbb{P}_*[r] \tag{2}$$

[4]We select ν to be 3 times larger than the empirical standard deviation of embeddings.

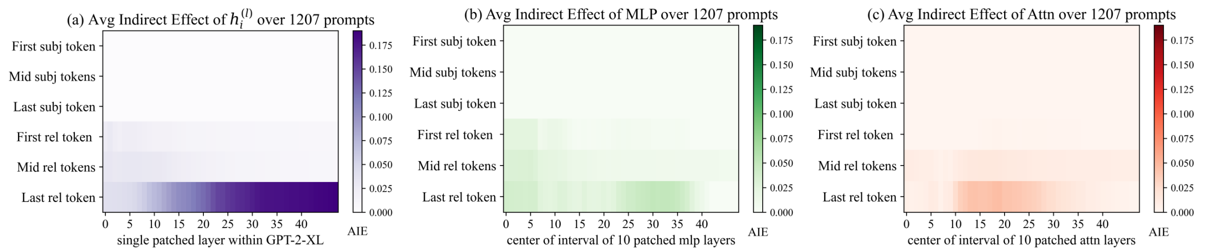

Figure 4: Causal tracing results of individual model components. In this paper, we use a sample of 1207 factual statements from (Meng et al., 2022a) as knowledge queries to explore the knowledge contained within GPT-2 XL.

IE is defined as the difference between the probability of r under the corrupted version and the probability when that state is set to its clean version, while the relation remains corrupted. After averaging over all the prompts, we get the average average indirect effect (AIE) for each hidden state.

The result of the causal tracing analysis is depicted in Figure 4. Consistently with previous findings, we observed a high AIE in the later layers of the final token. This implies that restoring the hidden states of the MLPs in those layers recovers most of the necessary information. Additionally, we noted a significant AIE in the earlier layers for the relation tokens that we intentionally corrupted. This discovery is non-trivial and underscores the significance of the earlier layers in predicting plausibility. Furthermore, we observed a pronounced AIE in the middle attention layers of the last corrupted token. This surprising new finding suggests that **memory related to relations is not only stored in the MLPs but also in the attention layers**. This extends the previous finding that emphasized the significance of the attention module specifically at late site.

4.5 Severed Causal Analysis

To obtain a clearer understanding of the impact of MLP and Attn layers, we perform Severed causal tracing analysis with a modified causal graph, again following the footsteps of ROME.

Figure 5 presents a comparison of the average Average Individual Effect (AIE) at the last corrupted token for unmodified, severed MLP, and severed Attention causal graphs. Notably, we observe a distinct difference in AIE between the unmodified and severed MLP graphs, particularly in the earlier and middle layers. This finding is consistent with previous research and reinforces the critical role of MLP layers in plausibility prediction. Interestingly, the restoration effect appears to be independent of

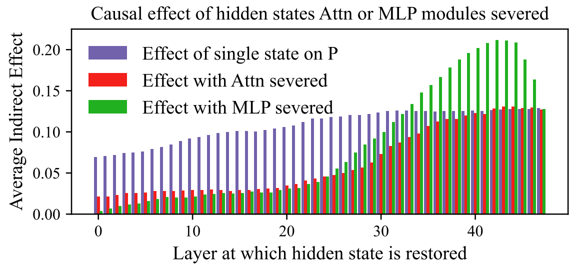

Figure 5: Causal effects with a modified GPT-2 XL model. To isolate the effects of individual model components, GPT-2 XL is modified by severing MLP and Attention modules.

MLP activity in the higher layers, suggesting that the higher MLP layers may potentially generate unintended side effects. In addition, we observe that the presence or absence of interruptions between attention modules in the range of 10 to 20 layers leads to significant differences in Attention-based Information Extraction (AIE). This finding suggests **a strong correlation between the attention modules at layers 10 to 20 and relations**. Consequently, we conclude that these parameters play a role in storing memory related to relations.

5 Related Work

5.1 Memory in LLMs

LLMs trained on extensive corpora such as Wikipedia, are widely believed to encapsulate vast amounts of factual knowledge (Petroni et al., 2019; Jiang et al., 2020).

FFN Memory. Geva et al. (2020, 2022) show that feed-forward layers in transformer-based language models operate as key-value memories, where each key correlates with textual patterns in the training examples, and each value induces a distribution over the output vocabulary. The values complement the keys' input patterns by inducing output distributions that concentrate probability mass on tokens likely to appear immediately after each pattern. Dai et al. (2021) proposes an attribution method to identify knowledge neurons that express factual knowledge in the FFN of pre-trained Transformers. They find that suppressing or amplifying the activation of knowledge neurons can accordingly affect the strength of knowledge expression. Meng et al. (2022a,b) develop a causal intervention for identifying neuron activations that are decisive in a model's factual predictions. and then reveals an important role for mid-layer feed-forward modules that mediate factual predictions while processing subject tokens.

Attention Memory. Sparse Distributed Memory (SDM) provides an efficient algorithm for storing and retrieving memories (patterns) from brain neurons. It effectively solves the "Best Match Problem" by quickly identifying the most suitable memory match for a given query. Currently, Bricken and Pehlevan (2021) has shown that the update rule of the Attention module in Transformer models closely approximates SDM. Specifically, SDM consists of read and write operations. In the write operation, we can consider patterns being written and stored into nearby neurons based on the Hamming distance between patterns and neurons. In the read operation, the query is read from nearby neurons based on the circular region encompassed by the radius of the Hamming distance. Sakarvadia et al. (2023) establish an algorithm for injecting "memories" directly into the model's hidden activations during inference. Through experimentation, they find that injecting relevant memories into the hidden activations of the attention heads during inference is an efficient way to boost model performance on multi-hop prompts.

5.2 Memory Editing

As factual information continues to evolve, the knowledge stored in LLMs can become outdated or incorrect. Hence, there is an urgent need to facilitate timely updates of inappropriate knowledge in LLMs while preserving other valuable knowledge. Recently, this issue has garnered significant attention from researchers. Certainly, both parameter-efficient fine-tuning and incremental learning techniques provide avenues for modifying LLMs. However, it's essential to note that these approaches may be prone to overfitting and can incur substantial computational costs, especially when applied to large language models (LLMs) with an extremely large parameter scale. To address these issues, Sinitsin et al. (2020) proposes Model Editing, which aims to efficiently and accurately alter the factual knowledge stored within models. Presently, there are three primary types of model editing approaches: 1) Memory-based Method: These techniques utilize an additional trainable parameters to store memory or learn the required adjustments (Δ) for knowledge updating in the LLMs (De Cao et al., 2021; Mitchell et al., 2021, 2022; Dong et al., 2022; Huang et al., 2023). 2) Locate-Then-Edit Method: These approaches employ causal mediation analysis to locate knowledge neurons in LLMs and subsequently modify these recognized regions (Dai et al., 2021; Meng et al., 2022a,b). 3) In-Context Knowledge Editing Method: These methods are a training-free paradigm where knowledge editing is achieved directly by concatenating demonstrations within the input context (Zheng et al., 2023; Zhong et al., 2023).

6 Conclusion

In this paper, we introduce relation-based knowledge editing, with a new benchmark named **RaKE**. Empirically, we analyze the effectiveness of various model editing baselines and notice that existing knowledge editing methods exhibit the potential difficulty in their ability to edit relations. To investigate the fundamental reasons behind these results, we conducted causal analysis on the relationships within the triplets. We discovere that relational knowledge is not only stored in the FFN but also in the attention layer, which is a novel finding. From this, our experimental results indicate that the current editing methods, which focus solely on editing the parameters of the FFN module, lack modifications to the attention module. This inadequacy leads to suboptimal results in relation-based editing.

Limitations

The current version of the RaKE dataset lacks an assessment of relation specificity performance, which we plan to include in future versions for evaluation. Furthermore, this is the first paper on relation perspective knowledge editing, and we acknowledge the lack of specific methods for editing relations. Our research serves as a preliminary investigation, and we will gradually refine the editing methods targeting relations in our subsequent work.

References

Trenton Bricken and Cengiz Pehlevan. 2021. Attention approximates sparse distributed memory. *Advances in Neural Information Processing Systems*, 34:15301–15315.

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

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

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329*.

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

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Autodebias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1012–1023.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

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

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. 2022. Realtime qa: What's the answer right now? *arXiv preprint arXiv:2207.13332*.

Fangyu Lei, Tongxu Luo, Pengqi Yang, Weihao Liu, Hanwen Liu, Jiahe Lei, Yiming Huang, Yifan Wei, Shizhu He, Jun Zhao, et al. 2023. Tableqakit: A comprehensive and practical toolkit for table-based question answering. *arXiv preprint arXiv:2310.15075*.

Huanhuan Ma, Weizhi Xu, Yifan Wei, Liuji Chen, Liang Wang, Qiang Liu, and Shu Wu. 2023. Ex-fever: A dataset for multi-hop explainable fact verification. *arXiv preprint arXiv:2310.09754*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2022. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering. *arXiv preprint arXiv:2211.05655*.

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

Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. 2023. Memory injections: Correcting multi-hop reasoning failures during inference in transformer-based language models. *arXiv preprint arXiv:2309.05605*.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *arXiv preprint arXiv:2004.00345*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Yifan Wei, Yisong Su, Huanhuan Ma, Xiaoyan Yu, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2023. Menatqa: A new dataset for testing the temporal comprehension and reasoning abilities of large language models. *arXiv preprint arXiv:2310.05157*.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.