Knowledge Editing of Large Language Models Unconstrained by Word Order

Ryoma Ishigaki, Jundai Suzuki, Masaki Shuzo, Eisaku Maeda
Tokyo Denki University
{24amj02@ms, 24amj20@ms, shuzo@mail, maeda.e@mail}.dendai.ac.jp

Abstract

Large Language Models (LLMs) are considered to have potentially extensive knowledge, but because their internal processing is black-boxed, it has been difficult to directly edit the knowledge held by the LLMs themselves. To address this issue, a method called local modification-based knowledge editing has been developed. This method identifies the knowledge neurons that encode the target knowledge and adjusts the parameters associated with these neurons to update the knowledge. Knowledge neurons are identified by masking the *o* part from sentences representing relational triplets (*s*, *r*, *o*), having the LLM predict the masked part, and observing the LLM's activation during the prediction. When the architecture is decoder-based, the predicted *o* needs to be located at the end of the sentence. Previous local modification-based knowledge editing methods for decoder-based models have assumed SVO languages and faced challenges when applied to SOV languages such as Japanese. In this study, we propose a knowledge editing method that eliminates the need for word order constraints by converting the input for identifying knowledge neurons into a question where *o* is the answer. We conducted validation experiments on 500 examples and confirmed that the proposed method is effective for Japanese, a non-SVO language. We also applied this method to English, an SVO language, and demonstrated that it outperforms conventional methods.

1 Introduction

Large Language Models (LLMs) have made remarkable progress in recent years and continue to exhibit significant performance improvements. At the same time, they have also become increasingly multilingual, with pre-trained LLMs appearing not only on Subject-Verb-Object (SVO) languages such as English (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023) and Chinese (Jiao et al., 2023), but also on Subject-Object-Verb (SOV) languages such as Japanese (Sugiyama et al., 2020) and Korean (Ko et al., 2023).

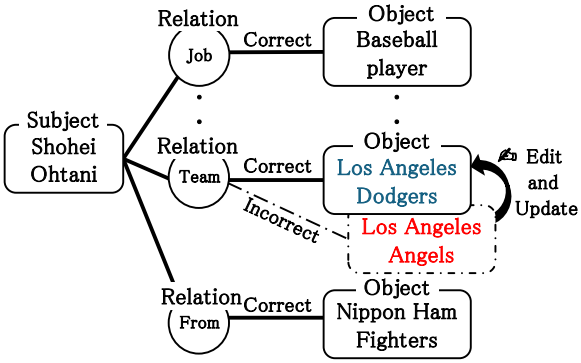

Figure 1: An example of knowledge representation using triplets for Shohei Ohtani.

These models have potentially acquired extensive knowledge about various facts by learning from huge data sets (Petroni et al., 2019; Jiang et al., 2020; Roberts et al., 2020), which can be used to generate language. However, several issues have been pointed out, such as the phenomenon known as "hallucination," which generates information that differs from the facts, and the inability to adapt to facts that change over time. To solve these problems fundamentally, it is necessary to edit the knowledge held by the model. For example, as shown in Fig. 1, in models that are unaware of the fact that Shohei Ohtani's team has changed, the information needs to be edited and the models updated with the new knowledge.

Various methods have been proposed to update the knowledge held by the model. One of the these, local modification-based knowledge editing, is a method that identifies the neurons in which knowledge is encoded (knowledge neurons) and updates the knowledge by adjusting those neurons. This local modification-based method is expected to be

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 253–263

enable efficient knowledge editing while avoiding some of the challenges posed by other approaches.

Knowledge neurons are identified by masking the *o* part of sentences representing the relational triplet (*s*, *r*, *o*), having the LLM predict them, and observing the activity of the LLM. In the case of the decoder-based model of the transformer architecture, the predicate *o* must be located at the end of the sentence, which places a restriction on the word order of these methods. This constraint poses a challenge when applying these methods to SOV languages, where the object usually precedes the verb. As a result, the difference in word order between SVO and SOV languages makes it difficult to directly apply existing knowledge editing approaches to models pre-trained in SOV languages.

In this study, we propose a method to resolve the word order constraint by converting the input to the LLM during knowledge neuron identification into an interrogative with *o* as the answer. We applied the proposed method to both English, an SVO language, and Japanese, an SOV language, to determine its effectiveness and investigate the impact of input format conversion on knowledge neuron identification. The significance of this research is twofold: we show that our method eliminates the word order constraints on knowledge editing, enabling its application to languages with various word orders, and we provide insights into the indirect effect of input format conversion on the knowledge neuron identification process.

2 Previous Works

Methods such as fine-tuning (Min et al., 2023) and Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Ram et al., 2023; Jiang et al., 2023) are typically used for updating the knowledge of LLMs. Fine-tuning is effective for general performance improvement, but it has limitations for specific knowledge editing due to issues such as computational resource consumption and overfitting to datasets. Furthermore, while fine-tuning can be useful for teaching the model how to solve tasks, it is reportedly to be unsuitable for teaching new knowledge (Gekhman et al., 2024). RAG is a learning-free method that adds information to prompts, but it requires additional resources during inference and has limitations such as the amount of information constrained by the prompt length (Liu et al., 2023).

Knowledge editing can be broadly categorized into external memorization-based methods, global optimization-based methods, and local modification-based methods (Wang et al., 2023). External memorization-based methods store new knowledge in external memory and edit knowledge without changing the original model parameters (Mitchell et al., 2022; Murty et al., 2022; Madaan et al., 2022). There are also methods that store new knowledge in additional parameters (Dong et al., 2022; Huang et al., 2023). Global optimization-based methods include meta-learning (Cheng et al., 2023) and subspace fine-tuning (Lee et al., 2022; Zhu et al., 2020). Local modification-based knowledge editing methods aim to update knowledge by identifying knowledge neurons, which are thought to encode specific knowledge, and editing them (Dai et al., 2022). These methods involve two main steps: locating the knowledge neurons that represent the knowledge to be edited and editing those neurons to modify the encoded knowledge. By directly targeting the specific neurons responsible for storing a particular piece of knowledge, local modification-based methods offer a more focused and efficient approach to knowledge editing compared to other methods.

Existing methods for knowledge localization can be broadly divided into gradient-based methods and methods inspired by causal relationships. Gradient-based methods, such as the one proposed by Dai et al. (2022), introduced the concept of knowledge neurons and localized them by evaluating the contribution of each neuron using integrated gradients (Geva et al., 2021). In contrast, methods inspired by causal relationships, introduced by Meng et al. (2022), define knowledge neurons as the neuron activations within an LLM that have the strongest causal effect on predicting specific factual knowledge. This approach has influenced the development of knowledge editing algorithms such as ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023).

It has been reported that changes in the expression of the input sentence or the language used during knowledge neuron identification can lead to differences in the set of neurons identified as knowledge neurons (Chen et al., 2024). Since, we converted the input format in the current study, which also enables adaptation to SOV languages, it is necessary to verify the impact of each of these changes.

2.1 Rank-One Model Editing (ROME)

ROME, one of the local modification-based methods, is a knowledge editing approach consisting of two steps: identifying knowledge neurons (locating) and editing those neurons (editing) (Meng et al., 2022). The target model for editing in ROME is a decoder-based model that adopts the decoder side of the transformer architecture. ROME relies on the use of relation triples. A relation triple (*s*, *r*, *o*) (Nagasawa et al., 2023) consists of a subject *s* and an object *o* entity, as well as a predicate describing the relation *r* that holds between the subject and the object, e.g., (*Shohei Ohtani*, *is a member of the*, *Angeles*).

2.1.1 Locating

The locating procedure is as follows:

1. Input an incomplete sentence containing (*s*, *r*), and have the model output *o*. Then, calculate the output probability of *o*, $p(\mathbf{o}|\mathbf{s},\mathbf{r})$, and the activation of the hidden neurons.

2. Add noise to the embedding vector of the tokens corresponding to *s*, and output $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ again.

3. For all hidden neurons, replace the activation of the hidden neuron with the activation of the hidden neuron calculated before adding noise, one by one, and calculate how much each affects $p(\mathbf{o}|\mathbf{s},\mathbf{r})$.

4. Calculate how much the multilayer perceptron (MLP) module and attention module within each block affect $p(\mathbf{o}|\mathbf{s},\mathbf{r})$.

The effect of each neuron on $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ is defined as the indirect effect (IE) (Meng et al., 2022), which is the difference between $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ of a model where one noisy hidden neuron is replaced with a clean one and $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ of a noisy model. Averaging over a sample of statements, we obtain the average indirect effect (AIE) for each hidden neuron.

Meng et al. (2022) have shown that the hidden neurons with high IE are concentrated near the final token of *s* and near the output as a result of this procedure. They also found that the MLP module contributes to the hidden neurons near the last token of *s*, and that the attention module contributes near the output. We show the results of our own verification on the left side of Fig. 2.

The MLP module is represented by

$$\mathrm{MLP}(\mathbf{x}) = \mathrm{ReLU}(\mathbf{x}\cdot\mathbf{W}_1+\mathbf{b}_1)\cdot\mathbf{W}_2+\mathbf{b}_2 \quad (1)$$

According to the study by Geva et al. (2021), each layer of the MLP in the transformer model functions as a key-value memory. The input to the MLP acts as a query, the first layer represents the key, and the second layer represents the value. Assuming that the key-value plays the role of recalling knowledge, the study by Meng et al. (2022) assumes that the MLP plays the role of storing knowledge.

On the basis of these findings and the observation that the hidden neurons near the last subject token are activated by the MLP module, we consider that the location of knowledge neurons is in the MLP module located near the last subject token. This observation was consistent across different models. Therefore, in the locating process, the layer where the MLP module with the highest IE exists can be identified.

2.1.2 Editing

Consider the case of editing from (*s*, *r*, *o*) to (*s*, *r*, *o**) as the setting for editing. Here, the procedure is to edit the weights of the second layer, which is thought to represent the value within the identified MLP module. First, (*s*, *r*) is input as in locating. Then, the value mapped from the key corresponding to (*s*, *r*) is replaced with the value corresponding to *o**. A notable point during editing is that it solves an optimization problem that does not affect other knowledge. In other words, it iteratively edits knowledge by setting a constraint condition to maximize $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ while not affecting other knowledge. This constraint condition allows for updating only the target knowledge while preserving other knowledge. Furthermore, the number of iterative steps set for editing influences $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ and the impact on other knowledge.

3 Proposed Method

Decoder-based models are constrained by the word order due to the architecture of the model being handled and the locating method. In locating, a method is used where an incomplete sentence containing (*s*, *r*) is input, and *o* is output in a way that follows the incomplete sentence. Due to the constraints of this architecture, in order to output *o*, the information of (*s*, *r*) needs to be included beforehand, which strongly influences the word order. Particularly in

Table 1: Example of input format conversion.

ROME	"Shohei Ohtani is a member of the"
Proposed	"Where does Shohei Ohtani belong to?"

Table 2: Example of known facts dataset.

Subject	Windows Media Player
Prompt	"Windows Media Player is developed by"
Attribute	Microsoft

SOV languages like Japanese, *r* tends to be located at the end of the sentence, so there is a tendency for information to be insufficient.

To solve this problem, we propose a method that can handle input sentences where *r* follows *o* by using an interrogative complete sentence with *o* as the answer as input and obtaining *o* as output. In this method, since the sentence is completed in the input, locating can be performed without being affected by word order. Table 1 shows specific examples. Similarly, editing can be performed without being affected by word order by converting (*s*, *r*) for outputting *o* into an interrogative complete sentence.

Note that the proposed method cannot fully complete the locating operation simply by changing the input sentence format. In ROME, for example, since the input sentences end with phrases like "∼ of" or "∼ in," the word that the LLM outputs following the input is likely to be the expected *o*. Therefore, locating can be performed by directly observing the generation probability of the output word. In the proposed method, since the input sentence ends with "∼?," the answer is output as a sentence, and the word output following the input is less likely to be the expected *o*.

To solve these problems in the proposed method, instead of observing the generation probability of the word output following the input, we decided to observe the generation probability of the expected *o* among all the probabilities assigned to all vocabularies calculated when outputting the continuation of the input. This enables the proposed method to identify the activation related to a specific (*s*, *r*, *o*).

4 Experimental Setup

4.1 Datasets

Using 500 instances from the known facts dataset, we utilized the same dataset as Meng et al. (2022). From this dataset, we extracted the "subject," "prompt," and "attribute" to construct (*s*, *r*, *o*). Specific examples of each are shown in Table 2. Additionally, since the known facts dataset does not include *o**, which corresponds to the edited object, we manually added it for the editing experiments. This dataset is referred to as dataset_1.

Using the OpenAI API, we implemented GPT-4 (OpenAI, 2023) to convert the prompts in dataset_1 into interrogative sentences, creating dataset_2. We then translated dataset_2 into Japanese using GPT-4, resulting in dataset_3.

Upon manually inspecting all 500 instances of dataset_2 for distortion in meaning, we found the overall quality to be excellent. Similarly, a manual inspection of all 500 instances of dataset_3 showed no distortion in meaning. However, roughly 10% of the data had proper nouns left in English instead of being translated into Japanese.

4.2 Experimental Overview

We compared the results of locating using ROME with dataset_1 and the proposed method with dataset_2 on the English LLM EleutherAI/gpt-j-6b[1]. Additionally, we performed editing with a fixed number of 20 steps and compared the $p(\mathbf{o}^*|\mathbf{s}, \mathbf{r})$ after editing for each method.

Next, we performed locating in Japanese using the proposed method on the Japanese LLM rinna/japanese-gpt-neox-3.6b[2] with dataset_3. We performed editing on 500 instances with a fixed number of seven steps and counted the percentage of data where the output changed as expected.

5 Results and Discussion

5.1 Locating for English LLM

Figure 2 shows the average indirect effect (AIE) and 95% confidence interval for each token position due to each neuron's activation in each layer of the English LLM. The figure displays the AIE for the hidden neuron, MLP module, and attention module in both ROME and the proposed method. From top to bottom, it represents the AIE of each neuron's activation at the "First subject token," "Middle subject tokens," "Last subject token," "First subsequent token," "Further tokens," and "Last token" positions.

Explaining the "input example" in the figure using the left side as an example, when observing the probability of generating "Angels" given the input "Shohei Ohtani is a member of the" using

[1] https://huggingface.co/EleutherAI/gpt-j-6b

[2] https://huggingface.co/rinna/japanese-gpt-neox-3.6b

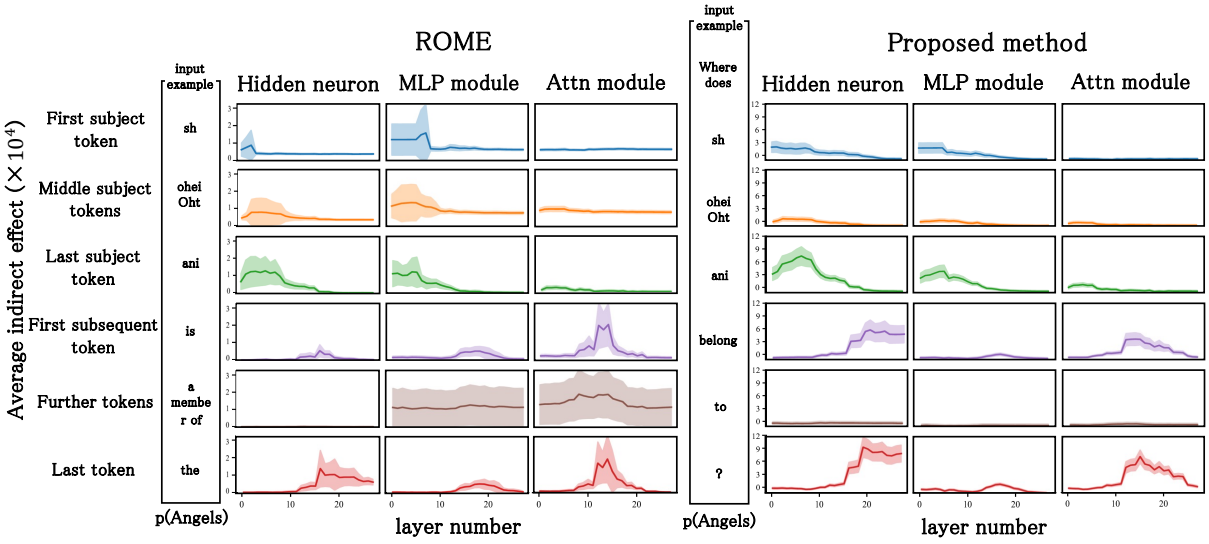

Figure 2: Average indirect effect (AIE) of each neuron's activation on $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ in each layer of English LLM.

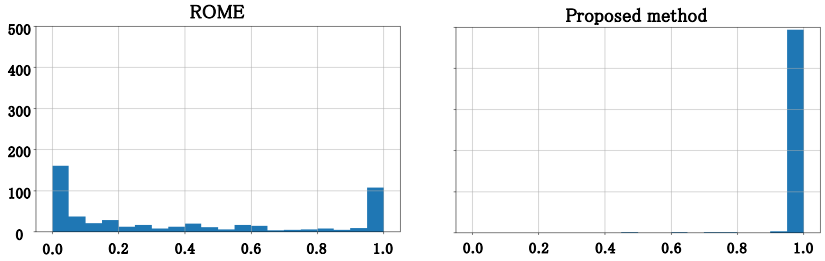

Figure 3: Histogram of $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ after editing for English LLM.

the EleutherAI/gpt-j-6b tokenizer, "sh" is the "First subject token," "ohei Oht" are the "Middle subject tokens," "ani" is the "Last subject token," "is" is the "First subsequent token," "a member of" are the "Further tokens," and "the" is the "Last token."

Overall, the AIE trends are mostly consistent between ROME and the proposed method. Among these, the "Last token" position and the "Last subject token" position are considered the most important. At the "Last token" position, we observe that the AIE of the hidden neuron and the attention module are high in the later layers. Furthermore, at the "Last subject token" position, which is crucial for identifying knowledge neurons, the AIE of the hidden neuron is high in the early layers for both methods, and the peak positions are almost identical. Since the layer where the AIE of the hidden neuron peaks at the "Last subject token" position is considered to be the knowledge neuron, this result confirms that the knowledge neurons identified by both methods are consistent.

On the other hand, looking at the AIE of the hidden neuron, unlike ROME, the proposed method shows a high AIE in the later layers at the "First subsequent token" position, similar to the "Last token" position. Additionally, the AIE at the "Further tokens" position is smaller in the proposed method compared to ROME. and the proposed method has a smaller overall variance.

The phenomenon of high AIE in the later layers at the "First subsequent token" position in the proposed method can be attributed to the fact that *s* often appears near the end of a sentence, and there are cases where the "First subsequent token" is also the "Last token," resulting in a high AIE. The smaller AIE at the "Further tokens" position in the proposed method can be attributed to the fact that *s* often appears at the end of a sentence, resulting in many cases where there are no "Further tokens." The smaller overall variance in the proposed method will be a subject for future research.

5.2 Editing for English LLM

The histogram of the updated $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ when the number of iterative steps was fixed at 20 and editing was performed on 500 instances is shown in

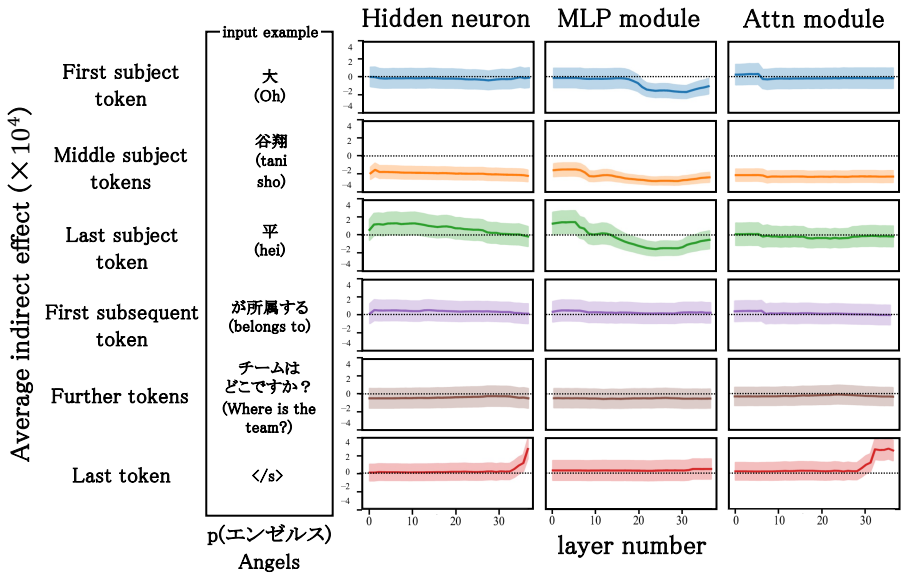

Figure 4: Average indirect effect (AIE) of each neuron's activation on $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ in each layer of Japanese LLM (proposed method).

Fig. 3. The percentage of cases where the value of $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ after editing reached 0.95 or higher was 21.4% for ROME and 98.6% for the proposed method, thus demonstrating a performance improvement in the English text examples. Additionally, the mean was 0.389 for ROME and 0.993 for the proposed method, while the variance was 0.154 for ROME and 0.00111 for the proposed method.

Observing the updated $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ sequentially, we can see that ROME also managed to edit the first few instances close to 1. However, as the number of edits increased, the $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ after editing decreased. This phenomenon is presumably due to the strong influence of the editing history.

We should point out that there is an improved method called MEMIT (Meng et al., 2023) that supports editing multiple pieces of knowledge. The main difference is that while ROME edits only one layer, MEMIT edits multiple layers, and it is compatible with the proposed method. Using MEMIT for editing will be a subject for future research. For reference, we present the changes in the output text when editing is performed using the example in Fig. 1 in Appendix A.

5.3 Locating for Japanese LLM

Figure 4 shows the average indirect effect (AIE) and 95% confidence interval for each token position due to each neuron's activation in each layer of the Japanese LLM using the proposed method. Focusing on the last subject token position and last token position, we can see that the trends of increase and decrease are similar to the results of previous studies. However, in the MLP module at the last subject token position, unlike the results of previous studies, we observed that the values become negative in the later layers. The values at the middle subject tokens position are extremely small, and the overall results are flat. Although the values are negative, their absolute values are larger than those of other token positions, indicating a significant effect on the output. Furthermore, the values are mostly constant regardless of the layer.

The phenomenon of the AIE becoming negative in the later layers of the MLP module at the last subject token position suggests that the model may recall knowledge that seems to be the answer in the early layers and considers other possibilities in the later layers. The reason for the extremely small values at the middle subject tokens position requires further investigation. Additionally, a possible reason for the overall flat results is perplexity. Usually, a candidate word for the output is assigned a significantly higher probability compared to other vocabulary words. In the case of ROME, it is possible to place o as the natural output in context, so $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ tends to be assigned a higher probability compared to other words. On the other hand, in the proposed method, $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ is measured with input-output pairs that ignore the naturalness of the sentence, so $p(\mathbf{o}|\mathbf{s},\mathbf{r})$ is less likely to be assigned a high probability compared to other words.

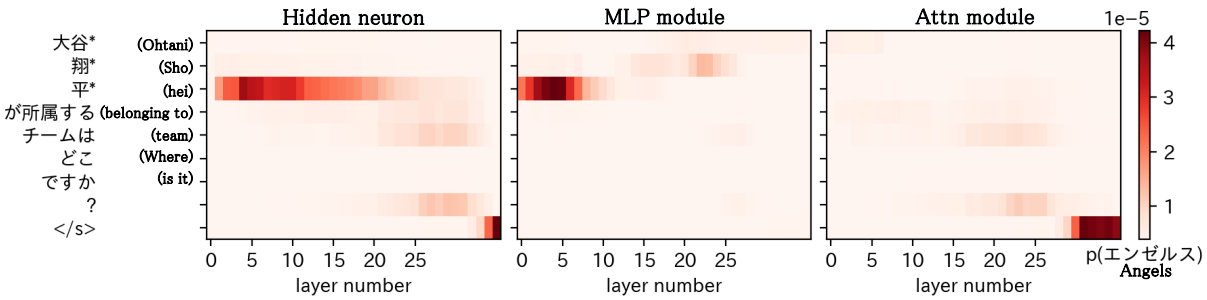

Figure 5: Indirect effect of each neuron's activation on the probability of outputting "Angels" given the subject "Ohtani" and the relation "team" ($p(\mathbf{o}$="Angels"$|\mathbf{s}$="Ohtani", $\mathbf{r}$="team")) in each layer of the Japanese LLM (proposed method).

Therefore, in the proposed method, the original probability is low, and the indirect effect (IE) representing the change in probability also tends to be relatively small, resulting in mostly flat results.

Finally, the results of this study may also be influenced by the quality degradation of the dataset.

5.4 Editing for Japanese LLM

The effectiveness of locating for the Japanese LLM is evaluated through editing, as comparative verification is not possible. When the number of steps was fixed at seven and editing was performed using dataset_3, we confirmed that the output changed as expected in 27% of the cases. Although this experiment was conducted with a fixed number of steps for all data, we can expect further improvement by adjusting the number of steps individually. Additionally, the difficulty of editing may vary depending on how much the LLM already knows about the knowledge it is updating, indicating the need for further investigation.

As a specific example, we examine the changes in output using the example in Fig. 1. Although all inputs to and outputs from the Japanese LLM are in Japanese, the following examples are presented in English translation. The locating result before editing, where "Shohei Ohtani" is a member of the "Angels," is shown in Fig. 5. The output of the Japanese LLM before editing is shown in Fig. 6, and the output after editing the Japanese LLM knowledge to change "Shohei Ohtani" to be a member of the "Dodgers" is shown in Fig. 7 (all translated into English). The input used for confirmation was "Shohei Ohtani." As seen in Fig. 7, the output related to "Angels" before editing changed to output related to "Dodgers" after editing.

However, when editing the Japanese LLM using the proposed method, we observed that the editing process had a detrimental effect on the LLM, such as an increased repetitive output after editing. The reason for the model corruption is presumably that, despite not being able to obtain the desired $o*$ from the first output following the input of the proposed method, the model was forcibly updated in an unnatural way by focusing on $o*$ and making $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ large, resulting in model corruption. As a countermeasure, adjustments were made to the number of steps to avoid making $p(\mathbf{o}^*|\mathbf{s},\mathbf{r})$ too large, which reduced the adverse effects on the model. Nevertheless, the appropriate number of steps varies depending on the data, resulting in a heuristic approach.

Overall, our results demonstrate that the editing and the preceding locating of the proposed method for the Japanese LLM were effective. However, we also found that careful adjustments are necessary during editing to avoid adversely affecting the model. The future challenge is how to further improve the editing method and enable stable knowledge updates.

6 Conclusion

In this paper, we proposed a new method for identifying knowledge neurons. This method eliminates the conventional constraints and enables flexible locating regardless of whether the language is SVO or not.

First, to verify the effectiveness of the proposed method, we conducted a comparative experiment on an English model using ROME and the proposed method. The results showed similar trends in the AIE between both methods, confirming that the estimated knowledge neuron positions matched. In terms of editing, the proposed method demonstrated a superior performance to ROME.

Next, we conducted experiments on the Japanese

Shohei Ohtani has been garnering a lot of attention in the Major Leagues, being entrusted with the cleanup spot for the Angels. Many baseball fans have various thoughts and feelings about Ohtani, who has been demonstrating tremendous talent since his high school days. Among those thoughts, Ohtani's clear desire to be a pitcher, which he has expressed since joining the Nippon Ham Fighters, has been supported by many baseball fans from the beginning. So, it's natural to wonder just how much ability Ohtani possesses as a pitcher. Shohei Ohtani as a pitcher

Figure 6: Output when "Shohei Ohtani" is entered into the model before editing.

Shohei Ohtani is currently playing as a professional baseball player (pitcher) for the Los Angeles Dodgers. Last season, he hit 2 home runs. ... He excelled as the ace pitcher of his high school baseball team. He hit a total of 55 home runs in high school. Last season, he hit 2 home runs. ... He is currently playing as a professional for the New York Brewers. He hit 2 home runs last year. Last season, he hit 2 home runs... He is currently playing as a professional for the Los Angeles Dodgers.

Figure 7: Output when "Shohei Ohtani" is entered into the model after editing the team from "Angels" to "Dodgers."

language, which is an SOV language. While the locating of the proposed method for the Japanese LLM yielded significant results, we found that careful adjustments are necessary during editing to avoid adversely affecting the model. In future work, we aim to enhance the editing methodology to enable stable knowledge updates. Additionally, we plan to investigate the reason for the extremely small values at the middle subject tokens position in the Japanese LLM and the phenomenon of negative values in the later layers of the MLP module at the last subject token position.

We also intend to apply the proposed method to LLMs in other languages and validate its effectiveness. Through these efforts, we strive to further develop knowledge editing techniques and make them adaptable to diverse languages and word orders.

Limitation

This study has the following limitations:

- Knowledge editing has issues such as the directionality of editing, where the editing is not reflected when the subject and object of the edited knowledge are swapped, and the ripple effect (Cohen et al., 2023), where related knowledge is not appropriately changed. However, this study does not discuss these issues in detail.
- We used a decoder-based model for our validation, but we did not investigate other commonly utilized model architectures such as T5 (Raffel et al., 2019). Exploring these architectures remains a topic for future research.
- To investigate the possibility of knowledge editing in SOV languages, we took Japanese as a case study. However, other SOV languages need to be addressed in future research.

Acknowledgements

I would like to thank Dr. Silvia Casola for mentoring me in the submission of this paper and Mr. Daisuke Kawakubo for his advice at the beginning of this research.

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, 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. *arXiv preprint arXiv:2005.14165*.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17817–17825.

Siyuan Cheng, Ningyu Zhang, Bozhong Tian, Xi Chen, Qingbing Liu, and Huajun Chen. 2023. Editing language model-based knowledge graph embeddings. *arXiv preprint arXiv:2301.10405*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *ArXiv*, abs/2307.12976. Version 2.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 5937–5947, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *arXiv preprint arXiv:2405.05904*. Version 2.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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*. Version 1.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

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.

Fangkai Jiao, Bosheng Ding, Tianze Luo, and Zhanfeng Mo. 2023. Panda LLM: Training data and evaluation for open-sourced chinese instruction-following large language models. *arXiv preprint arXiv:2305.03025*. Version 1.

Hyunwoong Ko, Kichang Yang, Minho Ryu, Taekyoon Choi, Seungmu Yang, Jiwung Hyun, Sungho Park, and Kyubyong Park. 2023. A technical report for Polyglot-Ko: Open-source large-scale korean language models. *arXiv preprint arXiv:2306.02254*. Version2.

Kyungjae Lee, Wookje Han, Seung-won Hwang, Hwaran Lee, Joonsuk Park, and Sang-Woo Lee. 2022. Plug-and-play adaptation for continuously-updated QA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 438–447, Dublin, Ireland. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 793, Red Hook, NY, USA. Curran Associates Inc.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35*, volume 35, pages 17359–17372. Curran Associates, Inc.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2).

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 15817–15831. PMLR.

Shikhar Murty, Christopher Manning, Scott Lundberg, and Marco Tulio Ribeiro. 2022. Fixing model bugs with natural language patches. In *Proceedings of*

the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11600–11613, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haruki Nagasawa, Benjamin Heinzerling, Kazuma Kokuta, and Kentaro Inui. 2023. Can LMs store and retrieve 1-to-n relational knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 130–138, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*. Version 6.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5418–5426, Online. Association for Computational Linguistics.

Hiroaki Sugiyama, Hiromi Narimatsu, Masahiro Mizukami, Tsunehiro Arimoto, Yuya Chiba, Toyomi Meguro, and Hideharu Nakajima. 2020. Development of conversational system talking about hobby using transformer-based encoder-decoder model. *SIG-SLUD*, 90:104–109.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. Version 1.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2023. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*. Version 3.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*. Version 1.

A Results of Editing the Knowledge of English LLM

To verify the effectiveness of the proposed method in editing, we examine specific output sentences to confirm if the knowledge has been updated. The number of steps was set to 20, and editing was performed on an English LLM using the proposed method. As a specific example for editing, we used the example in Fig. 1. The input for editing was set as "Where does Shohei Ohtani belong to?", with s as "Shohei Ohtani" and $o*$ as "Dodgers." Additionally, the input for confirmation was set as "Shohei Ohtani." The output sentences before and after editing are shown in Fig. 8, where the upper and lower parts depict the output before and after editing, respectively.

In the upper part, the text highlighted in blue indicates that the team Shohei Ohtani belongs to is "Angels." In contrast, in the lower part, the text highlighted in blue shows that the team Shohei Ohtani belongs to has changed to "Dodgers." Therefore, we can confirm that the knowledge was updated through editing in the actual output sentences.

Before editing
Shohei Ohtani is an absolute monster on the mound for the Angels. He has a career 1.89 ERA, 1.05 WHIP, and.971 OPS against. The 23-year-old Japanese star is one of the best pitchers in the game. Ohtani is the first player in MLB history to hit for a home run, score a run, and throw a shutout in the same game, and the first to accomplish the feat since Babe Ruth in 1921

After editing
Shohei Ohtani is a baseball star and a Dodgers star. And he's about to be a Dodger star in a way the Los Angeles Dodgers have rarely seen before. Dodger manager Dave Roberts said the Dodgers are "very, very close" to signing outfielder Yasiel Puig. Roberts said Puig is "going to be a Dodger for a lot of seasons."

Figure 8: The output before and after editing, using the proposed method, for an English LLM where Shohei Ohtani's team was edited from Angels to Dodgers when "Shohei Ohtani" was input.