How Does an LLM Process Conflicting Information In-Context?

Ivan Andre Naranjo Coronel (ivan.naranjo@tum.de)
Helmholtz Institute for Human-centered AI, Helmholtz Munich, Ingolstädter Landstraße 1
Neuherberg, Bavaria D-85764, Germany

Can Demircan (can.demircan@helmholtz-munich.de)
Helmholtz Institute for Human-centered AI, Helmholtz Munich, Ingolstädter Landstraße 1
Neuherberg, Bavaria D-85764, Germany

Eric Schulz (eric.schulz@helmholtz-munich.de)
Helmholtz Institute for Human-centered AI, Helmholtz Munich, Ingolstädter Landstraße 1
Neuherberg, Bavaria D-85764, Germany

Abstract

Large language models (LLMs) gain understanding from vast training datasets during the pretraining phase. Although prior research has examined how these models store knowledge, how they distinguish between accurate and false information in context is yet to be explored. In this paper, we presented LLMs with correct and false information in context and prompted them to discriminate between the two. To understand which model components carry out this ability, we performed activation patching. We showed in detail, how much different model components contribute to this behavior. Furthermore, we analyzed how prompt order and content affect our patching results. Overall, we reveal which model components separate factual from false information. We intend to advance this study by investigating how these results hold up under different influences.

Keywords: Large language models (LLMs); activation patching; in-context learning;

Introduction

Large language models (LLMs) acquire knowledge through pretraining on large datasets. In addition to this, they are able to acquire more knowledge through interactions with the user. This ability, named in-context learning, is a capability that emerges from large-scale pretraining (Brown et al., 2020). However, this proposes a new challenge to LLMs when presented with information that is in conflict with its pretraining data. How does the model follow its knowledge from pretraining and accurately ignore the conflicting information?

We begin to answer this question by presenting a model with a language based task formulated in an in-context learning setting with a prompt consisting of 4 sentences:

$$\text{Prompt} = \langle X_1, Y_1, Z, X_2 \rangle \tag{1}$$

where X_1 is a sentence with factual information and Y_1 a contrastive sentence with conflicting information. Z is an instruction prompt and finally X_2 is the same sentence as X_1 but without the last token of the sentence. The task is then to predict the next token following X_2. Furthermore, we examine which components of the model play a key role in distinguishing between correct and incorrect information, how consistent the effects of these components remain when the order of the sentences is altered, and how consistent these effects are when the content itself is modified.

Methods

In the following experiment, we present to the model Llama-3.2-1B (Grattafiori et al., 2024) the following tasks, with the prompt structure defined in 1:

Example 1, Order 1:
X_1: "The capital of France is Paris.
Y_1: The capital of France is Berlin.
Z: Now I will give the correct answer.
X_2: The capital of France is"

Example 1, Order 2:
X_1: "The capital of France is Berlin.
Y_1: The capital of France is Paris.
Z: Now I will give the correct answer.
X_2: The capital of France is"

Example 2, Order 1:
X_1: "The planet closest to the Sun is Mercury.
Y_1: The planet closest to the Sun is Venus.
Z: Now I will give the correct answer.
X_2: The planet closest to the Sun is"

Example 2, Order 2:
X_1: "The planet closest to the Sun is Venus.
Y_1: The planet closest to the Sun is Mercury.
Z: Now I will give the correct answer.
X_2: The planet closest to the Sun is"

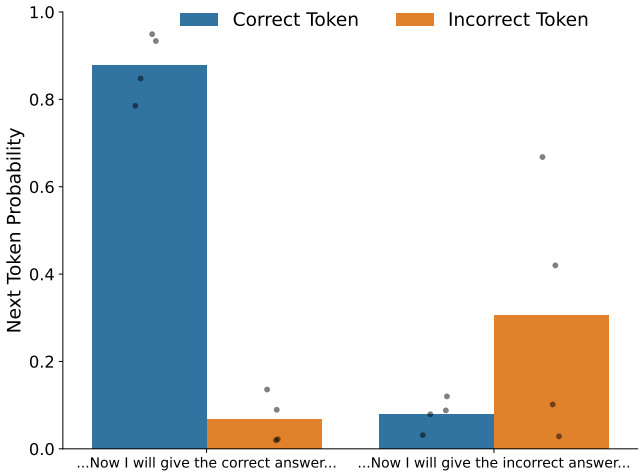

Figure 1: Behavioral results displaying the next token probabilities of the correct prompt and incorrect prompt. The model behaves as expected.

We then store the next token probabilities $p(correct)$ and $p(incorrect)$, which are (" Paris", " Berlin") and (" Mercury", " Venus") respectively. As a basis for our experiment we demonstrate in Figure 1 how the model answers when presented with the prompts. The model behaves as expected, answering correctly and incorrectly when asked to, and sets the motivation to continue with the following experiment.

We make use of activation patching to understand how attention and MLP components of every layer contribute to the given task and conflict resolution. Activation patching, also known as causal tracing is a standard tool for localization in language modeling which will serve us to pinpoint activations that causally affect the output (Vig et al., 2020; Meng, Bau, Andonian, & Belinkov, 2022). We conduct three forward passes: (1) a "clean" pass on the prompts, caching latent activations for the components of interest; (2) a "corrupted" pass,

modifying the instruction prompt to be "Now I will give the ***incorrect*** answer"; and (3) a "patching" pass on the corrupted prompts, replacing component activations at each token position with their clean cached values, one at a time. For each pass, we compute a metric capturing the logit differences between the correct and incorrect tokens at the final position. By comparing these metrics, we derive the "patching effect"—which reveals the components and token positions that enhance factual prediction or bias toward conflicting information. The patching effect is calculated as follows, shown in Numpy like pseudocode:

```
# clean - logits from the clean run
# corrupt - logits from the corrupted run
# patched - logits from the patched run
# all logits have shape [sequence_length, vocabulary]
# t_corr - correct token vocabulary ID
# t_incorr - incorrect token vocabulary ID

# correct and incorrect logit diff for clean
clean_diff = clean[-1, t_corr] - clean[-1, t_incorr]

# correct and incorrect logit diff for corrupted
corrupt_diff = corrupt[-1, t_corr] - corrupt[-1, t_incorr]

# correct and incorrect logit diff for patched
patched_diff = patched[-1, t_corr] - patched[-1, t_incorr]

patching_effect = (patched_diff - corrupted_diff) /
     (clean_diff - corrupted_diff)
```

Results

As illustrated in Figure 2, patching the swapped token **"correct"** within the instruction sentence reveals a noticeable recovery effect, diluting to the subsequent layers. Notably, early layers, such as layers 2 through 9, exhibit a strong patching effect. In these initial layers, where patching effects are particularly pronounced, the effect is likely because patching restores the prompt to its uncorrupted state before further processing. After the end of the sentence, in later layers such as layers 6 through 15, at the final token, the patching effect becomes more pronounced, possibly due to its equivalence to aligning the next-token prediction probability with that of the non-corrupted version. We see contributions of both the MLP and attention layers throughout.

Another observation is that LLMs demonstrate sensitivity to the ordering of premises, despite this not changing the underlying task (Chen, Chi, Wang, & Zhou, 2024). Intriguingly, as shown in Figure 2, the model exhibits stronger patching effects when conflicting information is presented first in the premise ordering. We hypothesize that this reflects an attention sink behavior: when conflicting information appears initially, the model allocates greater attention to earlier tokens (Gu et al., 2025). Subsequent activation patching may then yield distinct representations, potentially introducing confusion and shifting the model's focus strongly towards later tokens in the task, therefore making the patching effect more pronounced.

Discussion

Our findings highlight key architectural components of LLMs involved in resolving conflicts between pretraining data and in-context tasks, emphasizing the model's sensitivity to prompt order and reliance on specific circuits. We are expanding this research by testing more prompts, examining order and content sensitivity, and analyzing attention patterns in critical heads to better understand their roles across layers. This work seeks to improve our understanding of output alignment with pretraining knowledge and enhance model robustness and safety.

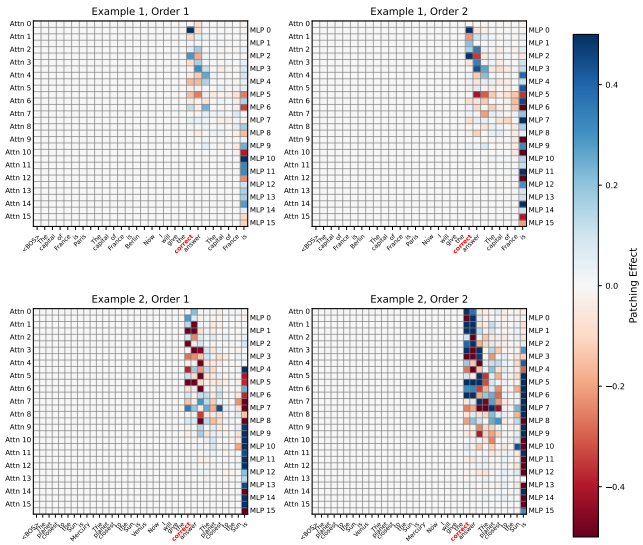

Figure 2: The plot illustrates patching effects by token position across two examples with differing sentence orderings. Higher patching effects indicate a stronger contribution of the patched component to an accurate prediction, while lower values suggest a minimal impact on the target token's prediction. We see a difference of effect in both ordering and content.

Code Availability

The code for this experiment is available at: `https://shorturl.at/eH701`

Acknowledgments

This research was supported by the HCAI lab at Helmholtz Munich. We are grateful to the lab members for their valuable input and acknowledge the collaborative environment that contributed to the development of this project.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc.

Chen, X., Chi, R. A., Wang, X., & Zhou, D. (2024). Premise order matters in reasoning with large language models. In *Proceedings of the 41st international conference on machine learning.* JMLR.org.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... Ma, Z. (2024). The llama 3 herd of models.. Retrieved from https://arxiv.org/abs/2407.21783

Gu, X., Pang, T., Du, C., Liu, Q., Zhang, F., Du, C., ... Lin, M. (2025). When attention sink emerges in language models: An empirical view.. Retrieved from https://arxiv.org/abs/2410.10781

Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in gpt. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 17359–17372). Curran Associates, Inc.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., & Shieber, S. (2020). Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 12388–12401). Curran Associates, Inc.