Causal Tracing is an interpretability method used to analyze how the input to a transformer models influences the predicted probability of an output it gives, that is, how changes to some of the text given to the model may produce changes in the text the model returns. It aims to determine the influence of specific parts of the input and trace the route of this influence through the model.

The method follows several steps:

1. Add noise to the input. This alters the model's output.
2. Replace the altered model memory with the original memory (from a normal input).
3. Observe which replacements restores the model's original output the most.

For example, when the text input "The Eiffel tower is located in the city of" is given, the expected output is "Paris" with a high probability. However, if we add noise to the parts that make up "Eiffel", the probability of "Paris" decreases significantly.

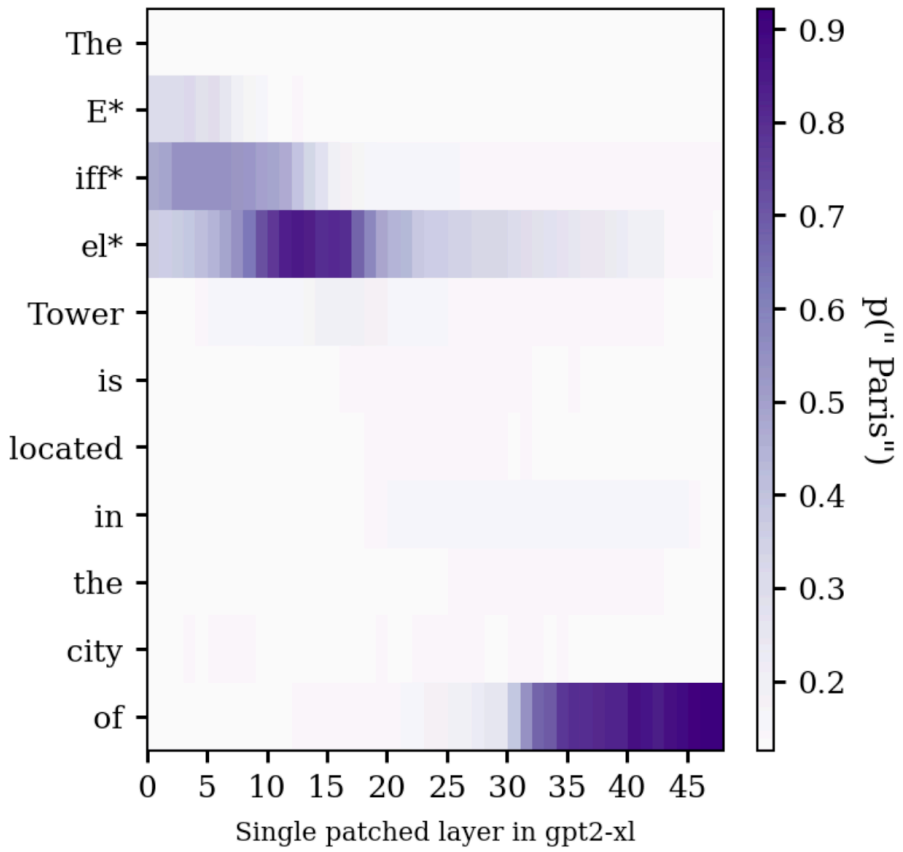


Figure from LessWrong. Example of causal tracing in the 48 layers of gpt2-xl model. Noised tokens are denoted with an asterisk. When run on a normal un-noised version of this prompt the model's top prediction is "Paris" at ~0.93 probability.

In this experiment, two notable places are determined in the prompt, specifically the token “el” around layers ten to twenty and the token “of” from around layer thirty onwards. Identifying such "notable places" in the model's layers can provide valuable insights into how the model processes and prioritizes information, which is crucial for both interpretability and alignment.

This can be used to infer some information about how the model processes the prompt. For example, by about layer 35 it seems like the information about “Paris” originating from the "Eiffel" tokens have mostly made their way into the final layers, and fixing the “of” token would result in a decent chance of correctly predicting "Paris".

## Related

- What is compute governance?
- What is Adversarial Goodhart?
- What is the “fragility of value” thesis?