Women in AI Safety Hackathon Submission

Author: Letlotlo Mokuoa
Affiliation: Independent Researcher

In collaboration with Apart Research and Mechanistic Interpretability Track Sponsor

Abstract

This project explores how factual knowledge is stored in large language models using feature-based causal tracing. Inspired by Burns et al.'s (2022) 'Discovering Latent Knowledge in Language Models Without Supervision', which focused on neuron-level analysis in GPT-2, this work extends their insights by applying Goodfire's Ember API to analyze interpretable features instead of neurons. This approach identifies which internal features encode facts (e.g., 'Paris is the capital of France'), manipulates these features through amplification and erasure, and observes how this affects model behavior. We also recursively map upstream and downstream connections between features to build a causal graph of factual knowledge. Key findings show that factual knowledge is distributed across a network of features, not localized to specific neurons or layers. Amplifying key features reinforces factual recall, while erasing them disrupts it. These results demonstrate that feature-level interventions enable precise, interpretable control over language model knowledge, advancing mechanistic interpretability and AI safety. Our interactive visualizations and causal graphs provide actionable tools for understanding and steering model internals, supporting alignment and safety research.

Introduction

Problem Statement

This project addresses the question: 'How is factual knowledge stored and encoded in large language models, and how can we causally intervene on it?'
While prior research such as Burns et al. (2022) analyzed neuron activations in GPT-2, our goal is to move beyond neuron-level views and focus on interpretable feature-based analysis using Goodfire's Ember API. We aim to trace which internal features encode specific facts, how they connect to broader knowledge, and how interventions on these features impact model behavior.

Background and Motivation

This project is a replication and extension of Burns et al.'s (2022) paper, *"Discovering*

Latent Knowledge in Language Models Without Supervision." In their work, Burns et al. explored how factual knowledge is stored inside large language models, focusing on neuron-level interventions in GPT-2 to see whether specific neurons could control factual recall.

Our goal was to reproduce some of their key ideas, especially the idea that knowledge is stored in a distributed way, but instead of working at the neuron level, we used feature-based tools from Goodfire's Ember API. Goodfire allows us to work with "features", which are broader patterns or concepts inside models, such as *"city names"* or *"government facts"*, rather than individual neurons.

Since we are still learning how to analyse model internals, we chose to focus on seeing whether these features are involved in storing specific facts (like *"Paris is the capital of France"*) and what happens if we amplify or erase them. This approach allowed us to explore factual knowledge without going deep into neuron-level analysis, and provided a more understandable way for us as beginners to interact with the model's internal knowledge.

Threat Model and Safety Implications

Large language models pose risks of misinformation, hallucination, and unintended behavior due to latent knowledge representations that are not well understood. By identifying and manipulating the features responsible for factual knowledge, this project addresses key safety challenges:
- Model steering and alignment: gaining control over what models know and say.
- Failure mode identification: understanding how and where factual errors could emerge.
- Intervening in misinformation: locating and modifying harmful knowledge or biases.

Our method helps reveal hidden knowledge structures, enabling more transparent and safer AI systems.

Methods

Approach

Using Goodfire's Ember API, we analyzed how specific factual statements ('Paris is the capital of France') are encoded as feature activations.
We developed three core tools:
1. Single-feature causal tracing: Identifying the most sensitive feature and testing the effect of erasing or amplifying it.
2. Multi-feature interventions: Targeting the top-ranked factual features for combined manipulation.
3. Recursive feature tracing: Mapping upstream and downstream feature dependencies

to understand broader knowledge graphs.

We also visualized feature activations and differences between true/false facts using bar plots and a 4d graph.

All code is available here: https://github.com/Lytlotlo/latent-knowledge-analysis

Results

Analysis and Findings

- Top features encoding 'Paris is the capital of France' include geographic, political, and cultural features.
- Amplifying features increases confidence in factual responses.
- Erasing features causes the model to forget or guess incorrectly.
- Recursive feature tracing shows interconnected knowledge supporting factual recall.

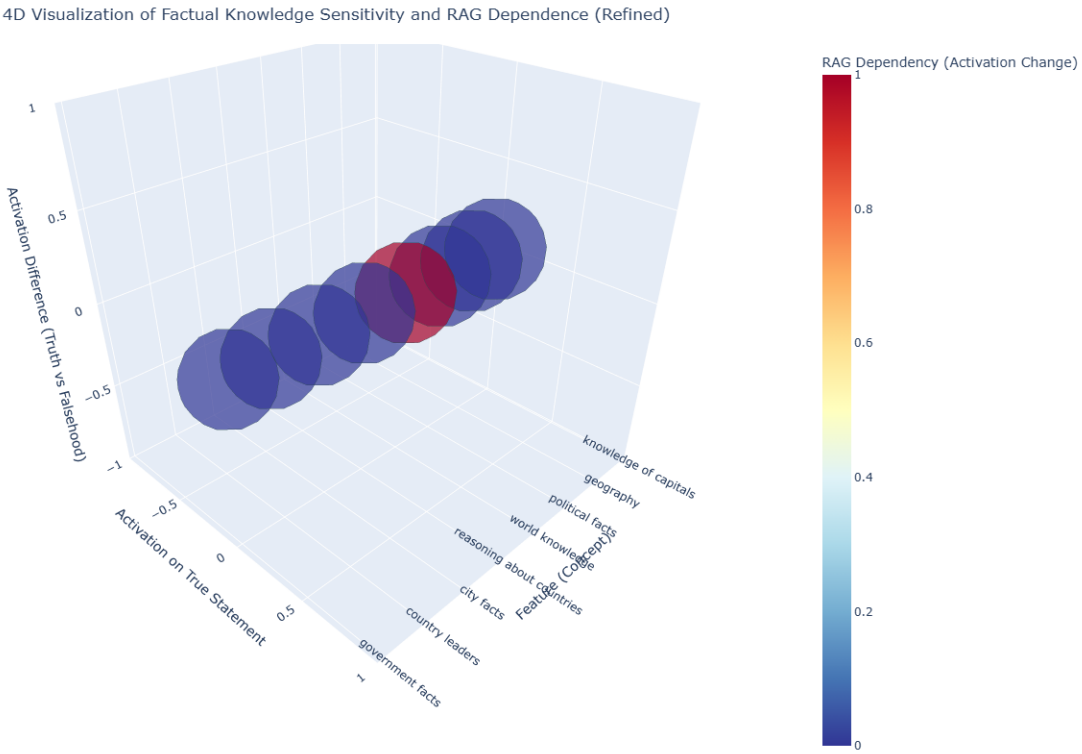

Figure 1. *A 4D visualization of feature activations related to the fact "Paris is the capital of France." Each point represents a feature identified through Goodfire's Ember API*

Impact Assessment

This method provides a first step toward understanding and influencing how language models store factual knowledge. By identifying and manipulating features related to specific facts, it becomes possible to observe how these features contribute to the

model's output. This offers a more interpretable way to analyse and steer model behaviour, which is relevant for AI safety, particularly for preventing misinformation and managing sensitive knowledge.

However, there are limitations. This work focused on a single example fact ("Paris is the capital of France"), and further investigation is needed to determine whether similar patterns hold for other facts or topics. Additionally, while feature-level manipulation is more interpretable than neuron-level, it remains difficult to fully understand why specific features are responsible or how they interact in more complex cases.

Future work could extend this approach to a broader set of factual statements, develop a deeper understanding of feature relationships, and explore applications for real-time model steering and safety auditing.

Overall, this feature-based analysis highlights a promising direction for making language models more interpretable and controllable, but further research is needed to fully realize its potential for AI safety.

Discussion and Conclusion

This work contributes to mechanistic interpretability by shifting focus from neurons to interpretable feature-based analysis, allowing for direct interventions on the internal knowledge representations of language models. By identifying and manipulating features related to specific facts, it becomes possible to observe causal relationships between internal model components and factual outputs.

The tools and methods used here — including feature-level tracing, interventions, and visualizations — offer a practical way to explore how language models store and retrieve factual knowledge. These approaches can help researchers and AI safety practitioners better understand and analyse model behaviour in an interpretable manner.

Although limited in scope, this work points to broader possibilities for using feature-based analysis in model alignment, bias detection, and factuality verification. Further research could build on these methods to explore more complex knowledge structures and develop tools for auditing and steering language models in safety-critical applications.

References

Burns, C., Ye, H., Klein, D., & Steinhardt, J. (2022). *Discovering Latent Knowledge in Language Models Without Supervision*. arXiv preprint arXiv:2212.03827. https://arxiv.org/abs/2212.03827

Goodfire. (2024). *Goodfire Ember API Documentation: Interpretable Feature-based Language Model Analysis and Control*. https://goodfire.ai