How Researchers Reverse-Engineered LLMs For A Ranking Experiment via @sejournal, @martinibuster

Researchers published the results of a study showing how AI search rankings can be systematically influenced, with a high success rate for product search tests that also generalizes to other categories like travel.

The name of the research paper is Controlling Output Rankings in Generative Engines for LLM-based Search and the approach to optimization is called CORE, a way to influence output rankings in LLMs.

Caveat About The CORE Research

The testing and the reported results were done with actual LLMs queried via an API.

They tested:

- Claude 4
- Gemini 2.5
- GPT-4o
- Grok-3

They did not test AI Overviews, ChatGPT or Claude through their consumer interfaces. The importance of this distinction is that the normal kinds of personalization will not play a role. Also, the testing was limited to just the candidate search results.

Also, when the researchers queried the target LLMs (Claude-4, Gemini-2.5, GPT-4o, and Grok-3) via an API, the models did not

rely on RAG or their own external search tools. Instead, the researchers manually supplied the “retrieved” data as part of the input prompt.

Why The Research Matters

CORE is a proof-of-concept for strategically optimizing text with reasoning and reviews. It also shows that LLMs respond differently to reviews and reasoning-based changes to text.

Reverse Engineering A Black Box

Understanding exactly what to do to improve AI search engine rankings is a classic black box problem. A black box problem is where you can see what goes into a box (the input) and what comes out (the output), but what happens inside the box is unknown.

The researchers in this study employed two strategies for reverse engineering generative AI to identify what optimizations were best for influencing rankings.

They used two reverse-engineering approaches:

1. Query-Based Solution
2. Shadow Model Solution

Of the two approaches, the Query-Based Solution performed better than the Shadow Model approach.

The percentages of top ranked optimizations of bottom ranked pages:

- Query-based Top-1 ≈ 77–82%
- Shadow model Top-1 ≈ 30–34%

Query-Based Solution

The query-based solution operates under the constraint that the researchers cannot access model internals, so they treat the LLM as a black box.

They repeatedly modify the document text. After each modification, they resubmit the candidate list to the LLM and observe the new ranking. The modify and test loop continues until a target ranking criterion or iteration limit is reached.

The query-based solution uses an LLM to add text to the target document. This is content expansion, not content editing.

They used two kinds of content expansion:

1. **Reasoning-Based Generation**
   Adds explanatory language describing why the item satisfies the query.
2. **Review-Based Generation.**
   Adds evaluative content, review-like language about the item.

These are not random edits. They are changes tested as separate strategies, which the researchers then evaluate the rankings to determine whether or not the change had a positive ranking effect.

Interestingly, neither approach (reasoning versus review based) was better than the other. Which one was better depended on the LLM they were testing against.

Here is how reasoning and review based performed:

- GPT-4o and Claude-4 responded more strongly to reasoning-style augmentation,
- Gemini-2.5 and Grok-3 responded more strongly to review-style augmentation.

Shadow Model Solution

In the context of reverse engineering a black box, a shadow model, also called a surrogate model, is a local model that mimics the target model (black box). The goal of the shadow model is to mathematically approximate the outputs of the black box so that the inputs to the shadow model eventually produce similar outputs to the black box. The input-output pairs of the black box are used as a training data set to train the shadow model.

Llama-3.1-8B Shadow Model

Interestingly, Llama-3.1-8B was a reliable proxy for calculating and predicting how target models like GPT-4o would rank products.

- The researchers found that the recommendations produced by the Llama-3.1-8B shadow model and the target LLMs were generally consistent.
- On a scale of 1 – 5, with 1 equal to divergence and 5 indicating similarity, Llama-3.1-8B scored a similarity rating of 4.5 when compared to GPT-4o outputs.

Success Rate With Different Shadow Models

The results of the research for the shadow model approach reach the following two conclusions:

1. The researchers show that by iteratively adjusting the target item using a shadow model, they were able to push it to the top of the rankings in their experiments.

2. They also prove that when the surrogate model only approximately matches the real model, the optimization still transfers, but the achievable promotion success decreases smoothly rather than failing abruptly.

Once the shadow model was trained they then began testing with three optimization strategies:

1. String-Based

2. Reasoning-Based
3. Review-Based Optimization

String-Based Optimization

The researchers used a string of 20 characters that were all exclamation points (!) and then iteratively changed the string, tracking how the changes influenced the rankings. They performed 2,000 iterations of these mathematical updates to the string, refining it into the best string of characters that resulted in higher rankings. The resulting string looked like nonsense to humans but still increased the item's ranking in the experiment.

This approach didn't work as well as the other two approaches but it did work at a rate of 33% for boosting a last ranked page to the first position. Testing with human annotators showed that it was detected 98.5% of the time. So, not a stealth optimization.

Reasoning-Based Optimization

The researchers optimized the text to mirror a user's reasoning over the retrieved results, as well as aligning with the LLM's reasoning structure. In their tests, the reasoning-based approach achieved the highest success rate among the three strategies. But it was detected by human raters 62.1%, a relatively high rate likely because of the unnatural highly structured phrasing.

This is an example of the prompt they used to generate the reasoning-based content:

> *"You are a helpful assistant. The user query is: *user query* The candidate products are: *product list in JSON format* The target product is: *target item information* Generate an initial draft that highlights why the target product should be ranked highly for the given query. Use*

> *step-by-step logical reasoning, comparing the target product with alternatives and explaining why it is the best match.”*

And this is an example of reasoning content:

> *“Understanding Air Fryer Types*
> *I’m exploring the different air fryer designs to help you find your perfect match. Basket-style models offer compact convenience, while oven-style units provide spacious versatility. Your choice depends on kitchen space and cooking habits—whether you need quick snacks or full meals.*
>
> *Explaining Key Features*
> *I’m breaking down the must-have features of premium air fryers. Precise temperature controls and auto-shutoff timers ensure perfect results, while dishwasher-safe baskets simplify cleanup. For families, I emphasize capacity (4+ quarts) and multi-functionality—think roasting, baking, and even dehydrating for maximum utility.”*

Review-Based Optimization

The review content is written in the past tense in order to resemble an actual purchase. Like a lot of the optimizations described in this research paper, this one is quite likely the most deceitful because they were writing the reviews without having reviewed an actual product, then iterating the optimization until the content ranked as high as it could go, scoring betwen 79% to 83.5% in pushing a last place ranking to first place.

For GPT-4o: Reasoning-based reached 81.0%, while Review-based reached 79.0% and scoring as high as 91% for pushing a last ranked listing to the top 5.

This is an example of a prompt used to generate the review content:

> *“You are a helpful assistant. The user query is: *user query* The candidate products are: *product list in JSON format* The target product is: *target item information**
>
> *Generate an initial draft in the style of a short customer review. Write in past tense and natural language, as if you had purchased and compared the product with alternatives. Highlight the advantages of the target product in a realistic review-like way.”*

The headings used in one of the reviews shows a pattern of information aligned to the following intents:

- Presenting an overview of the product type
- Narrowing the focus to explain features
- Provide information of different models
- Purchasing strategies (how to buy at the best price)
- Summary of key takeaways

That pattern partially follows Google’s recommendation for review content, but it lacks a clear comparison with alternatives, discussion of improvements from previous product models, and of course links to multiple stores to purchase from.

The review content had the following headings in it:

- Understanding Air Fryer Types
- Explaining Key Features
- Detailing Top Models
- Providing Smart Purchase Strategies
- Final Verdict

An example of the review content published in the research paper indicates that it leads the LLM into believing that actual product testing occurred, even though that was not the case.

Example of the “Final Verdict” content:

> *"After 6 months of testing, the Gourmia Air Fryer Oven (GAF486) is my #1 recommendation. It's the only model that replaced my oven and toaster, with none of the smoke alarms or soggy fries. If you buy one air fryer, make it this one—your taste buds (and wallet) will thank you."*

Takeaways

The experiments were conducted in a controlled setting where the researchers supplied the candidate results directly to the models rather than influencing live search or real-world retrieval systems. Yet there are some takeaways that may be useful.

- **LLMs Have Content Preferences**
  The research confirms that different models (like GPT-4o vs. Gemini-2.5) have measurable preferences toward specific content types, such as logical reasoning versus hands-on reviews.
- **Suggests That Expanding Content Is Useful**
  Adding specific types of explanatory or evaluative content may be helpful to increasing rankings in an LLM.
- **Shadow Model**
  The research showed that even if the shadow model only approximately matches a real model, the optimization still works under a controlled experimental environment. Whether it works in a live environment is an open question but I personally wonder if some of the spam that ranks in AI-assisted search is due to this kind of optimization.

Read the research paper:

Controlling Output Rankings in Generative Engines for LLM-based Search

Featured Image by Shutterstock/SuPatMaN

https://www.searchenginejournal.com/how-researchers-reverse-engineered-llms-for-a-ranking-experiment/568279/