LLM Optimization

Learning goals

- Learn how to optimize LLM performance

ACKNOWLEDGEMENTS

- This presentation is based on slides originally authored by:
  - Hinrich Schütze
- `https://slds-lmu.github.io/dl4nlp/`

RLHF LECTURE

Roadmap

- Motivation: Why LLM optimization?
- Prompt engineering
- Beyond prompt engineering
- More details on prompt engineering

Motivation: Why LLM optimization?

MOTIVATION

Why LLM optimization?

- A model like GPT4 can do amazing things.
- Why can’t we use it out of the box?
- Why do we need to optimize its performance?
- How can we optimize its performance?

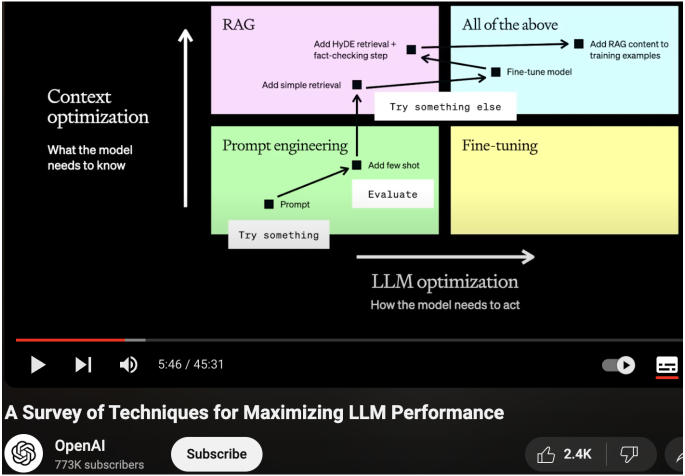
MOTIVATION

Why LLM optimization?

- Suboptimal behavior without carefully designed instructions
- Suboptimal behavior without carefully selected training examples
- Lack of knowledge
- Lack of skills

LLM OPTIMIZATION

OpenAI’s current take



A Survey of Techniques for Maximizing LLM Performance

OpenAI
773K subscribers

RLHF LECTURE

Roadmap

- Motivation: Why LLM optimization?
- Prompt engineering
- Beyond prompt engineering
- More details on prompt engineering

Prompt engineering

PROMPT ENGINEERING

Six strategies

- Write clear instructions
- Provide reference text
- Split complex tasks into simpler subtasks
- Give the model time to “think”
- Use external tools
- Test changes systematically
- https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results

PROMPT ENGINEERING

Strategy 1: Write clear instructions

- Include details in your query to get more relevant answers
- Ask the model to adopt a persona (e.g., be an expert)
- Use delimiters to clearly indicate distinct parts of the input
- Specify the steps required to complete a task
- Provide examples
- Specify the desired length and format of the output

PROMPT ENGINEERING

Strategy 2: Provide reference text

- (purpose: reduce hallucinations)
- Instruct the model to answer using a reference text
- Instruct the model to answer with citations from a reference text

PROMPT ENGINEERING

Strategy 3: Split complex tasks into simpler subtasks

- (it’s much easier for the model to answer simple tasks and then stitch together the results than answer a complex task)
- Use intent classification to identify the most relevant instructions for a user query
- For dialog applications that require very long conversations, summarize or filter previous dialog
- Summarize long documents piecewise and construct a full summary recursively

PROMPT ENGINEERING

Strategy 4: Give the model time to “think”

- (e.g., chain of thought)
- Instruct the model to work out its own solution before rushing to a conclusion
- Use inner monologue or a sequence of queries to hide the model’s reasoning process
- Ask the model if it missed anything on previous passes

PROMPT ENGINEERING

Strategy 5: Use external tools

- Use embeddings-based search to implement efficient knowledge retrieval
- (in general: RAG)
- Use code execution to perform more accurate calculations or call external APIs
- Give the model access to specific functions

PROMPT ENGINEERING

Strategy 6: Test changes systematically

- (it’s hard/impossible to assess changes based on anecdotal evidence)
- Evaluate model outputs with reference to gold-standard answers

RLHF LECTURE

Roadmap

- Motivation: Why LLM optimization?
- Prompt engineering
- Beyond prompt engineering
- More details on prompt engineering

Beyond prompt engineering

SYSTEM MESSAGE

Ways we can exploit the system message

- Generally: helps set the assistant's behavior
- Give the assistant a persona
- Give the model a summary of a long context
- Give the model general instructions that should govern its global behavior
- Instruct the model to check its own output
- For calculations: always generate and then execute code

DOUBTS ABOUT LLM OPTIMIZATION

What is the halflife of this lecture?

- “I’ve been hesitant lately to dedicate a lot of time to learning how to perfect prompts. It appears every new version, not to mention different LLMs, responds differently. With the rapid advancement we’re seeing, in two years or five, we might not even need such complex prompting as systems get smarter.”
- https://www.infoq.com/news/2023/12/openai-prompt-engineering/
- (apart from systems getting smarter, their very nature may also change rapidly as has been the case in the last 2-5 years)

Retrieval-augmented generation?

RLHF LECTURE

Roadmap

- Motivation: Why LLM optimization?
- Prompt engineering
- Beyond prompt engineering
- More details on prompt engineering

More details on prompt engineering

▸ OpenAI Prompt Engineering Guide