Jai Guru!

**Prompt Engineering and Retrieval Augmented Generation (RAG)**

Prompt Engineering

* Process where you guide generative AI solutions to generate desired outputs.
* OR Process of designing and optimizing prompts to guide gen AI models to produce desired outputs.

Prompt:

* Natural language test that describes a task for the AI to perform.
* 4 levels: Parameter, Syllable, Complex, Surface. Each level – used for a different purpose, but all provide the neuromotor system with tactile and proprioceptive input regarding temporal-spatial mapping.

Prompt Engineers (PE’r)

* Use creativity and trial and error to create input texts that guide the AI to interact with users more meaningfully.
* Choose the most appropriate words, phrases, symbols, and formats to help the AI create high-quality and relevant output.
* Design inputs that interact well with other inputs in the gen AI model
* Eg: use chain-of-thought (COT) prompting to break down complex reasoning into intermediate steps, which can help the model to produce more accurate results.
* Understanding the end users – important for crafting effective prompts. Eg: If a PE’r is building a chatbot for a company’s customer support portal, they can consider the user’s purchase behavior, product challenges, and prev interactions with customer support.
* In demand? Yes, in various industries including finance, healthcare, tech, etc
* Like a Data Scientist, PE’r is a cross-functional role, expected to be able to dive deep into the tech details, while also explaining their biz impact to busy execs.
* Role: creating prompts that elicit accurate and relevant responses from AI models. Testing and iterating: you can rarely come up with a perfect prompt from the jump. Hence, AI experts have to continuously experiment, test AI responses, and refine prompts for the best results.
* Must be familiar with data structures, LLMs, linguistics, programming languages, and coding principles to gen code snippets, debug, develop AI integrations, and more. Clear problem definition and specification are essential to guide language models toward producing reliable code outputs.
* Difference between AI Engr and PE’r: Generally, an AI Engr designs, develops, and maintains AI models and systems, while a PE’r may focus on crafting effective prompts and instructions for AI models to gen desired outputs. In some cases, these roles could overlap, and the distinction may not be universally standardized.

PE:

* Can help improve safety of LLMs and build new capabilities, such as augmenting LLMs with domain knowledge and external tools.
* Is code required for PE? PE’rs would need a solid understanding of Python programming as it allows to learn NLP and DL models quickly. As a PE’r, you may not build an entire language model by yourself.
* 4 S’s: Simplicity, Specificity, Sensitivity, and Structure => by adhering to these principles, developers and researchers can enhance the effectiveness of AI Applications, ensuring that interactions are optimized for the best possible outcomes.
* A Hype? NO. As we move into our undoubtedly AI-laden future, PE will be a necessary skill for every data engr to develop
* Have a future? It’s more than just an interesting facet of NLP – it’s an invaluable tool for shaping and optimizing language model behavior. Through careful design and innovative techniques, we can harness the full potential of these models and unlock new possibilities in NLP.
* Types: n-shot prompting, chain-of-thought prompting, and generated knowledge prompting
* Zero shot prompting: a technique in which an AI model is given a task or question w/o any prior examples or specific training on that task, relying solely on its pre-existing knowledge to gen a response.
* Best language for PE: Python and learn to articulate yourself communication wise, esp working of mental viz creatively and articulating it.
* Can AI do PE? PE techniques are used in sophisticated AI systems to improve user experience with the learning LM.
* Difference between Prompt Design and PE: PD – focuses on crafting effective prompts; PE – aims to optimize the entire system to ensure reliable, efficient, and safe performance.

Prompt Engineering for Web Devs - ChatGPT and Bard Tutorial

<https://www.youtube.com/watch?v=ScKCy2udln8>

Prompt Engineering Tutorial – Master ChatGPT and LLM Responses

<https://www.youtube.com/watch?v=_ZvnD73m40o>

Prompt Engineering: How to Trick AI into Solving Your Problems

<https://www.youtube.com/watch?v=0cf7vzM_dZ0>

Build a Large Language Model AI Chatbot using Retrieval Augmented Generation

<https://www.youtube.com/watch?v=XctooiH0moI>

Let's build GPT: from scratch, in code, spelled out.

<https://www.youtube.com/watch?v=kCc8FmEb1nY>

Schemas

Google uses SentencePiece – encodes text into integers but in a diff schema and uses a diff vocab

SentencePiece: Sub-word tokenizer – not including entire word nor a single char in the word. It’s a subword

Openai: has a lib called tiktoken (uses pipe pair encoding tokenizer)

**Retrieval Augmented Generation (RAG)**

* Invented by Douwe Kiela with DataStax
* Process of optimizing the output of a LLM, so it references an authoritative knowledge base (a collection of info that is highly reliable, accurate, and credible. Often created and maintained by experts in a particular field, ensuring that the content is up-to-date, unbiased, and based on sound evidence) outside of its training data sources before generating a response.
* A technique for enhancing the accuracy and reliability of generative models with facts fetched from external sources.
* At its core, RAG – a sophisticated technique in the field of AI and NLP that combines the strengths of 2 approaches: info retrieval and generative modeling.

Components of RAG?

Two main components: a document retriever and a LLM

Document retriever: responsible for finding relevant info from a large corpus of documents based on the input question using semantic search

LLM: generates a response using the info passed from the document retriever

RAG Chatbot?

RAG-enabled chatbots – proactive in responding to and addressing queries in real-time. They consume the user’s intent, fetch relevant info from multiple external sources, analyze in real time and deliver personalized responses.

Snowflake RAG?

LLM outputs may produce inaccurate or irrelevant results when they rely solely on retrieval-based methods or generative models. Hence, RAG framework was developed.

Difference between fine-tuning and RAG and embedding.

RAG: connects an LLM to a curated database to improve outputs by integrating reliable info.

Fine-tuning: adjusts the model’s params by training it on a specialized dataset to improve the performance on specific tasks

Embedding: involves representing data in a lower-dimensional space to capture semantic relationships used to enhance the model’s understanding of context and meaning.

Difference between LLM fine-tuning and RAG.

Both fall under the wider umbrella of AI approaches geared to improve the way a trained language model AI responds to user inputs.

Is RAG better than fine-tuning?

RAG: excellent for dynamic settings since it can access up-to-date data from internal data and knowledge sources without retraining LLM.

Fine-tuning: can raise the accuracy of LLM responses, but the responses are still based on static snapshots of the training datasets and can be outdated.

Why is fine-tuning better?

It’s easier and cheaper to hone the capabilities of a pre-trained model that has already acquired broad learnings relevant to the task at hand than it is to train a new model from scratch for the specific purpose.

Can fine-tuning and RAG ne used together?

YES. They can be combined to create a more robust AI system.

Fine-tuning: customizes the model to excel in specific tasks

RAG: provides access to real-time data or external info during interactions

Is fine-tuning better that transfer learning?

Fine-tuning works by updating all model params. As a result, it requires more computational resources and consumes more time. Hence, transfer learning – ideal recommendation for scenarios where you have to train LLMs with limited computational resources and faster experimentation.

Difference between generative model and retrieval model.

* RAG-based AI system: a retrieval model is used to find the relevant info from existing info sources.

Generative model: takes the retrieved info, synthesizes all the data, and shapes it into a coherent and contextually appropriate response.

* RAG: method that connects biz data with gen AI models, adding specific context and meaning while identifying and reducing hallucinations in the gen AI’s response.

How does RAG exactly work?

The retrieval mechanism in RAG ensures that the retrieved info is relevant to the input query or context. By providing the LLM with contextually relevant info, RAG helps the model generate responses that are more coherent and aligned with the given context.

What is RAG primarily focused on?

* Combines generation and retrieval models in AI. It enhances text generation by retrieving relevant info from a large dataset before generating responses.

How to add RAG to LLM

[https://blog.lancedb.com/create-llm-apps-using-rag/]

Steps to create any RAG application

How to build a RAG LLM?

[https://realpython.com/build-llm-rag-chatbot-with-langchain/]

How to develop a RAG system?

[https://huggingface.co/learn/cookbook/rag\_with\_hugging\_face\_gemma\_mongodb]

How to implement Retrieval-Augmented Generation?

[https://www.infoworld.com/article/2336099/retrieval-augmented-generation-step-by-step.html]

Difference between RAG And semantic search.

RAG: a revolutionary technique that seamlessly blends retrieval and generation and allows machines to understand queries while also producing socially relevant solutions.

Semantic search: uses semantics to understand the meaning of queries, resulting in more precise and detailed results.

How does RAG approach reduce hallucinations in LLMs?

RAG mitigates the risk of hallucinations by providing the LLM with accurate, up-to-date information from reliable data sources. Eg: if the vector store has up-to-date information provided by a Kafka topic, it can ensure that a chatbot accesses accurate and relevant inventory levels when responding to queries.

Does RAG hallucinate?

By allowing the LLM to ground its answer in real internal data, active RAG improves accuracy and reduces hallucinations. That’s the theory, in any case. In reality, RAG is also prone to AI hallucination issues since, until now, it only relies on your unstructured, general data.

What are the risks of RAG?

AI models, including those used in RAG systems, are susceptible to manipulation and poisoning attacks. Bad actors can feed the system with corrupt or misleading data, causing it to generate harmful or misleading responses.

Does OpenAI use RAG?

Implement RAG with Azure OpenAI service. Azure OpenAI on your data allows developers to implement RAG with supported AI chat models to reference specific sources of data to ground the response.

Motivation behind RAG?

While LLMs such as OpenAI’s GPT models, excel at generating NL responses based on their training data, they can further benefit from additional contextual information that RAG provides.

IS ChatGPT a RAG model?

RAG – architecture that augments the capabilities of a LLM like ChatGPT by adding an info retrieval system that provides grounding data.

Retrieval in LLM?

Retrieval tools – group of utilities that return context that informs and grounds responses to the user prompt. This group encompasses both knowledge based and API-based retrieval systems. LLMs – you send prompts to

RAG pattern?

Queries and responses are coordinated between the search engine and the LLM. A user’s question or query is forwarded to both the search engine and to the LLM as a prompt.

How does RAG work in LLM?

RAG addresses this limitation by pulling in external data as needed during the generation process. How it works -> when a query is made, the RAG system first retrieves relevant info from a large dataset or knowledge base. Then, this info is used to inform and guide the generation of the response.

LLMs

Lack business context, are trained on stale data, are expensive to retrain, and generate hallucinations.

31 Aug 2024

[https://www.iguazio.com/blog/integrating-llms-with-traditional-ml-how-why-use-cases/]

ChatGPT was released in Nov 2022 and since then organizations have been trying to find new and innovative ways to leverage gen AI to drive organizational growth. LLMs capabilities like contextual understanding and response to NL prompts enable the development of applications like automated AI chatbots, smart call center apps, and for financial services.

Gen AI is by no means a replacement to the previous wave of AI/ML (traditional ML) which continues to deliver significant value, and represents a distinct approach with its own advantages. By integrating LLMs with the traditional ML models, organizations can significantly enhance and augment each model’s capabilities, leading to new and exciting applications that bring value to their customers.

LLMs and ML model’s strengths

Evaluate the benefits of integration

Example use cases

LLMs strengths

1. NL understanding and generation: They excel at comprehending and producing human-like text. Can generate coherent and contextually relevant responses over a wide range of topics, making them ideal for applications like chatbots, content creation, and language translation
2. Contextual learning: They use the DL architecture, and hence, are able to grasp the nuances of language, including idioms, cultural reference, and complex syntax. This allows for nuanced conversations and content generation.
3. Adaptability: They can be fine-tuned for specific tasks with relatively small datasets after their initial pre-training. This adaptability makes them versatile tools for a variety of industries (legal doc – cust care)
4. Knowledge integration: They possess broad knowledge based due to their extensive pre-training on diverse datasets. They can provide info, summaries, and insights across many fields without the need for external databases in real-time applications.
5. AI democratization: They democratize access to AI by lowering the entry barrier. Users can interact with these models through conversational interfaces the need for specialized knowledge on data formatting or the underlying model architecture.

Strengths of Classical ML models:

1. Numerical data precision: They excel in environments dominated by structured, numerical data. They can yield highly accurate predictions or classifications within well-defined problem spaces. Useful for applications -> financial forecasting or biomedical analyses.
2. Efficiency and scalability: Specialized ML models can be more efficient in terms of computational resources and scalability. Imp for real-time decision-making tasks (autonomous vehicles or high-freq trading)
3. Interpretability: Certain ML models, esp those with simpler structures (DT or LR) provide clearer insights into how decisions are made. Imp in fields (healthcare and finance) where transparency of the rationale behind predictions is as imp as accuracy.

LLMs Vs Classical ML models: A comparison

|  |  |  |
| --- | --- | --- |
|  | LLMs | Classical ML models |
| Best suited data type | NL | Numerical, structured data |
| Input types | NL prompts | Code, API, application interfaces, structured data formats |
| Top capabilities | Contextual learning, adaptability, knowledge integration | Specialized tasks like classification, regression, clustering, PR in structured data. High efficiency in specific, well-defined problem spaces. |
| Output | NL content like stories, summaries, translations, analyses, and more | Predictive scores, classifications, quantitative analyses, and decision-support insights. |
| Entry barrier | Low | High due to the need for domain-specific knowledge, FE, and model tuning |
| Risks | Hallucinations (producing incorrect or fabricated info) | Overfitting (model learns noise in the data), underfitting (model – too simple), bias in training data leading to biased predictions |
| Special considerations | Requires extensive data for training with ethical and bias considerations necessitating careful dataset curation | Often requires manual FS and engg, sensitive to the quality of input data and may need regular updates as new data becomes available |

Benefits of integrating LLMs into traditional ML Architectures

Capitalizes on the strengths of both domains. Offers a wide range of benefits that enhance biz value across various sectors.

* Enhanced handling of numerical data: ML – processing of structured numerical data; LLM – NL understanding => allows to develop applications that can understand and analyze data in both numerical and textual formats => broadens the scope of problems that can be tackled.
* Incorporation of more advanced practices: integration – encourages the exploration of more advanced architectures and TL and fine-tuning techniques.