AI Summer of Code












Buzzwords in LLMs

SAM AYO

Lead AI Engineer & Head of Engineering
AISoC, co-host.

About

- **Academic background:** Economics, Math, Stats, ARTIBA
- **Areas of Interest:** Core AI, NLP, Audio AI, AI Engineering, probabilistic models, experimentation & system inference design.
- **Programming Languages:** Python, C++, C#, Golang, JavaScript, TypeScript.
- **Recent work:** Real-time Agentic system, near real-time audio signal detection, Semantic relation modelling and search.
- **Industries covered:** Agnostic
- **Fun fact:** Built LangChain equivalent in golang

Introduction

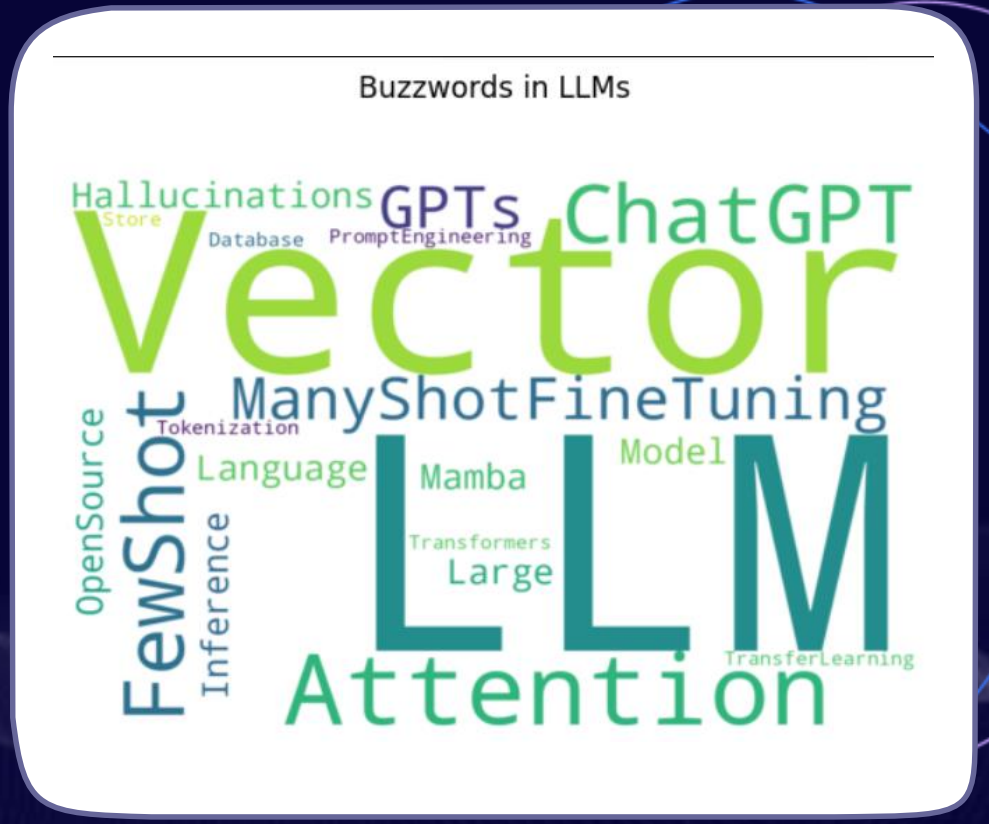
LLM stands for **Large** Language Model.

Buzzwords in:

- General Buzzwords
- Chains and Calls – making simple request and response calls with or without memory
- RAG – document or modal retrievals
- Agents – intelligent decision making LLMs

General Buzzwords in the LLM space

- Transformers
- Attention
- GPTs
- Tokens
- Open/Closed Source LLM
- Hallucinations
- Prompt Engineering
- Few-Shot Learning
- Fine-tuning
- Inference



General Buzzwords in the LLM space

Transformers

Transformers are a type of neural network architecture that is widely used in natural language processing and language modeling.

Characteristics

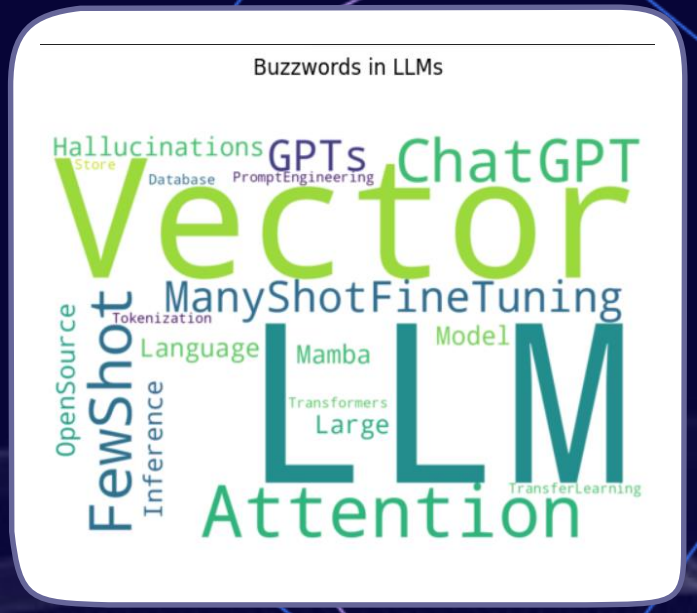
- Applies attention which allows the model to consider the importance of all words in a sentence simultaneously.
- consists of an encoder and a decoder.

Attention

Attention mechanisms are fundamental components that allows transformer based AI models to model to focus on different parts of the input sequence dynamically, allowing it process and understand text more effectively.

Characteristics

- Introduced by Vaswani et al. in the paper “Attention is All You Need”
- With self-attention or scaled dot-product attention the model computes attention scores between each pair of tokens in the input sequence. These scores determine how much focus to allocate to each token when generating the output.



General Buzzwords in the LLM space

GPTs

GPTs (Generative Pre-trained Transformers) are transformer based deep learning language models designed to understand and capable of generating human-like text. GPT is the underlying technology behind OpenAI's ChatGPT.

Tokens

Tokens are the basic units of text an LLM processes. It can be single characters, individual words, or even subwords, and the choice of tokenization strategy depends on the use case and application.

Open/Closed Source LLMs

The term "**open source**" refers to software source code that is publicly available under free-to-use licenses and in AI refers to models whose weights are made publicly available for use. e.g. Llama 2, Mistral AI(open), Phi-3, Salesforce Blip etc.
Inversely, "**closed source**" refers to software source code that is publicly behind a pay-wall through communication means such as APIs. e.g. OpenAIs ChatGPT, Anthropics Claude, Dalle-E, Mistral AI(closed)

General Buzzwords in the LLM space

Hallucination

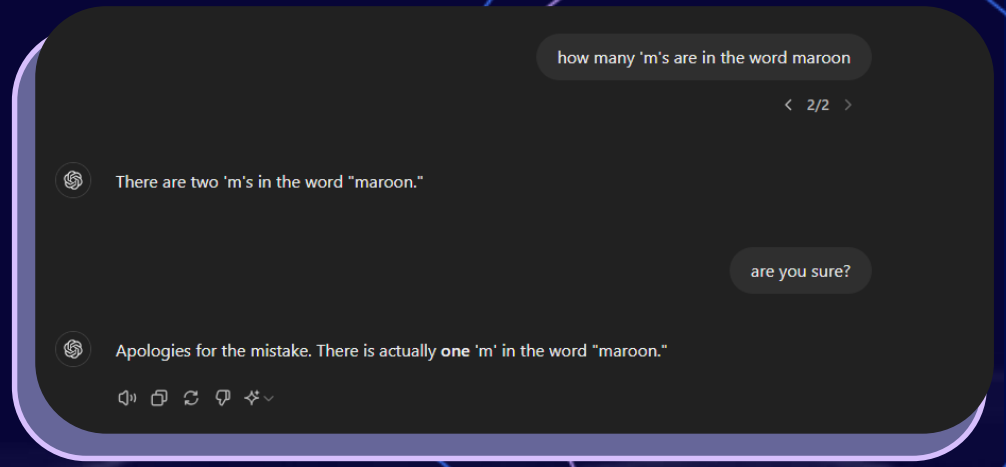
In artificial intelligence (AI), an hallucination effect or artificial hallucination aka confabulation or delusion is a confident response by an AI that does not seem to be justified by its training data.

Prompt Engineering

Prompt Engineering is a process involved with designing, refining, and optimizing the input/system prompts that are well-suited to a specific task in order to achieve a tailored outputs. Some specific techniques in prompt engineering include: rephrasing, format specification, leading Information etc.

Few-Shot Learning

Few-shot Prompting/Learning, aka in-context learning is a leading information prompting technique that allows a model to process multiple labeled examples before attempting a task. The purpose is to increase the LLMs capabilities to respond in context, expand the knowledge base and provide alignment. Other prompt techniques include Many-Shots, Chain of Thought, One-Shot etc.



General Buzzwords in the LLM space

FineTuning

Fine-tuning in LLMs is a transfer learning technique where a pre-trained model is further trained on a specific dataset for downstream tasks. There are various techniques for performing fine-tuning including AdaLoRA, LORA, QLORA etc.
Fine-tuning can be done on the entire neural network, or on only a subset of its layers, in which case the layers that are not being fine-tuned are “frozen” (not updated).

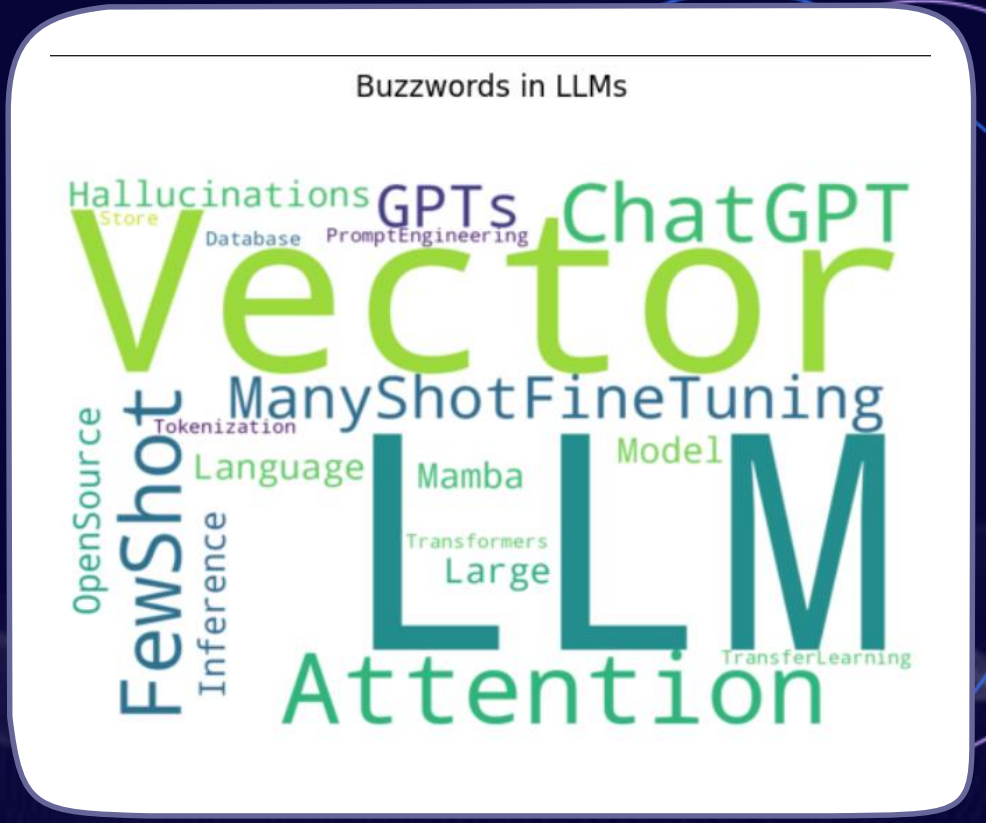
Inference

In LLMs, inference is the operational process of generating output.
LLMs may not generate the same result every time because inference is a stochastic operation.

input[trained model] ➡ sequence of tokens ➡ next predicted token[output]

Chains and Calls

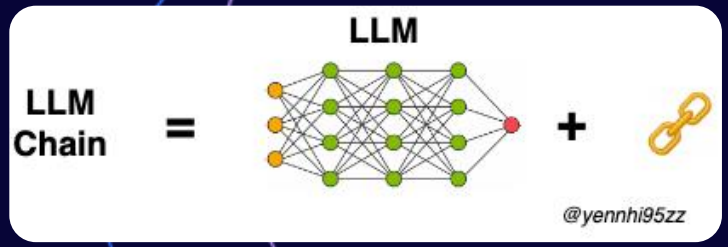
- Chaining
- Context Memory
- LLM Parameters



Chains and Calls Buzzwords

Chaining

Chains go beyond just a single LLM call, and /or sequences of calls (whether to an LLM or a different utility). LLM chaining is the process of connecting large language models to other applications, tools, and/or services to produce the best possible output from a given input.
Langchain, AutoGen, HayStack, DarksuitAI and other frameworks provide a standard interface for chains, lots of integrations with other tools, and end-to-end chains for common applications.



Context Memory

Context Memory is the concept of persisting state between calls of a LLM chain/agent. It gives the LLM chain, the capability to have conversational memory aiding contextual understanding. Langchain, darksuitai and others provides a standard interface for context memory and collection of memory implementations.

Chains and Calls Buzzwords

LLM Parameters

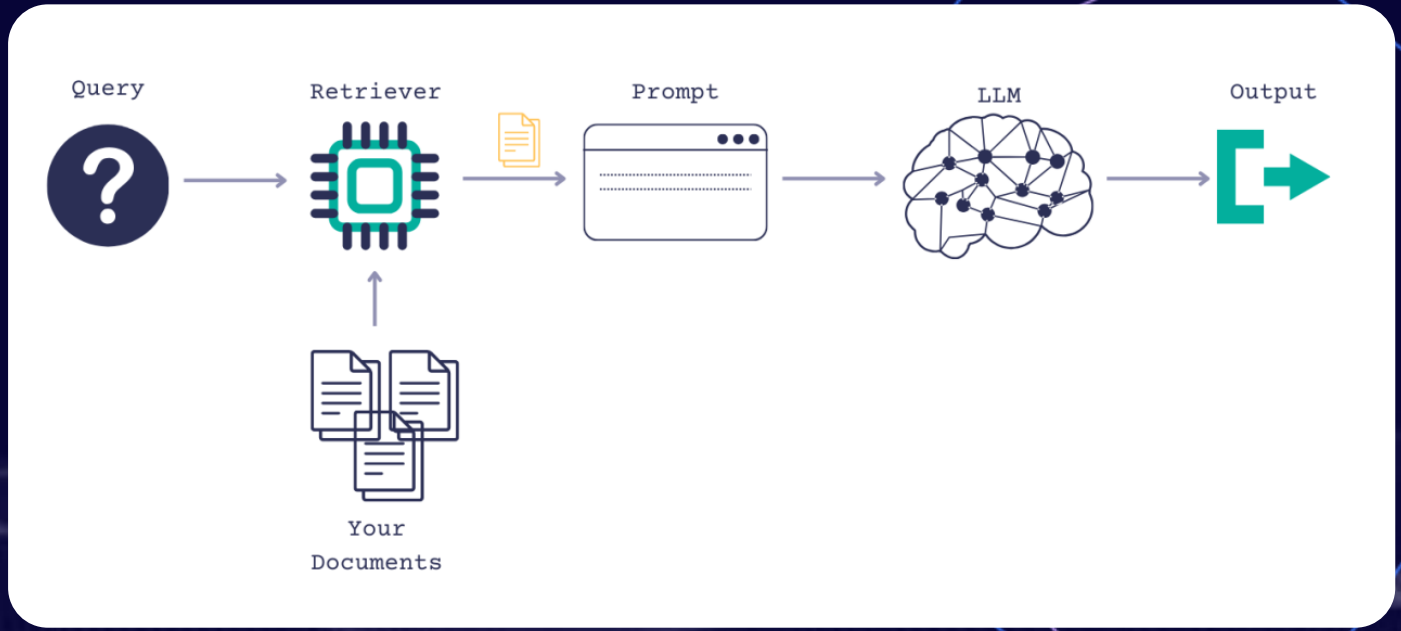
Hyperparameters are like knobs and dials you can tweak to make your large language model behave just how you want it.

Characteristics

- Temperature: This hyperparameter controls the randomness of the model's output. A temperature of 0 means always output the same response, it also controls the level of creativity in the LLM's output.
- Top-k: Setting a top-k limits the model's output to the most probable tokens at each step. This can help reduce incoherent or nonsensical output by restricting the model's vocabulary to a number of choices.
- Top-p: The top-p filters out words that do not meet a certain threshold (p). It lets the model be a bit more diverse in what it says, but still keeps it from using super unlikely words.
- Max tokens: This sets a cap on the maximum number of words the LLM should generate.
- Stop: Stop words are usually set to stop token generation when they are encountered.

RAG – Retrieval Augmented Generation

- Embeddings
- Chunking
- Vector Stores
- Max Marginal Relevance (MMR)
- Reranking



RAG – Retrieval Augmented Generation

Embeddings

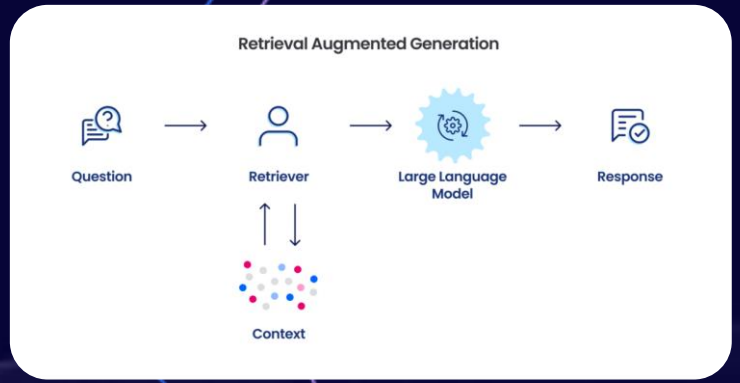
Vector embeddings are a type of representation that captures the semantic meaning or context of words or sentences in a compact form. Essentially vectors of real numbers (floats), where each dimension could represent a different feature that captures something about that concept's meaning.

Chunking

Chunking is a process used to enhance the efficiency and accuracy of information retrieval. In RAG apps, the input text is broken down into smaller, manageable units called "chunks." These chunks can be sentences, paragraphs, or by punctuation marks. This approaches enhances the retrieval state in efficiency, accuracy and scalability.

Vector Stores

A vector store is a type of database that stores high-dimensional vector data and provides query capabilities on those vectors such as similarity search, where the goal is to find the most similar documents or items to any given query. Vector search helps LLMs find similar information within their vast datasets using embeddings.



Sam Ayo

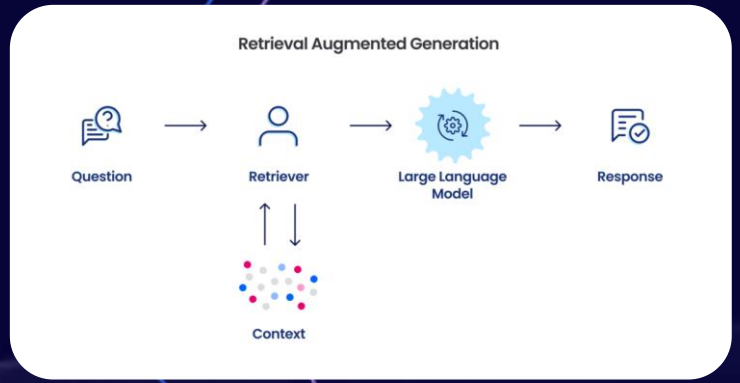
RAG – Retrieval Augmented Generation

Max Marginal Relevance (MMR)

Originally proposed in 1998, MMR refers to a technique by which relevant facts provided by the retrieval step are reordered to create a more diverse set of facts. This is most useful in critical RAG pipelines where many matching text chunks, from multiple documents, are very similar or exactly the same.

Reranking

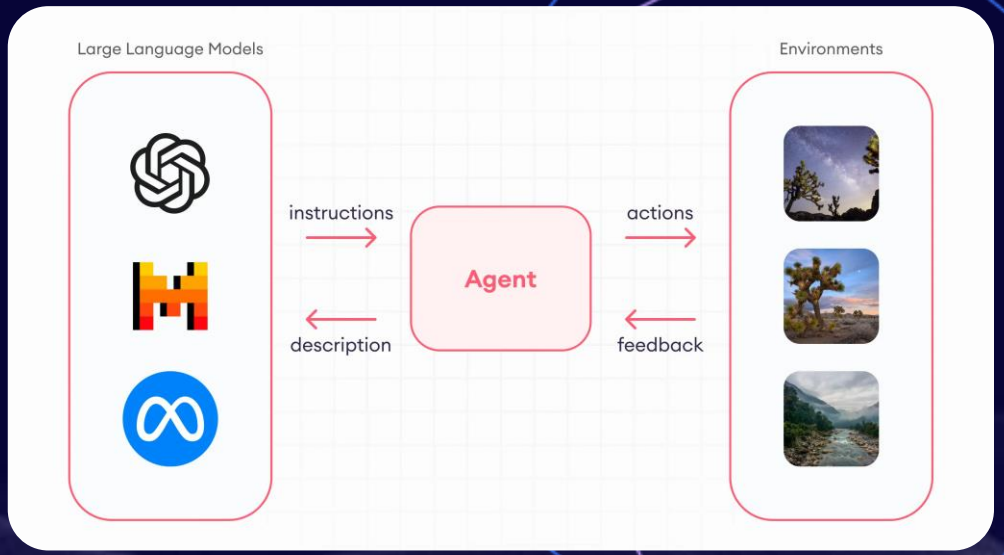
Reranking is the process of re-assessing and re-ordering an initially retrieved set of documents to enhance the relevance of the final set of documents used in the generation process of RAG systems. It is used in a two-staged RAG pipeline where a reranker algorithm is used to examines the details and contextual relevance of each in the initial set of retrieved documents to ensure it does not contain many irrelevant or marginally relevant entries due to the broad nature of the search.



Buzzwords in Agentic Systems

- Agents
- Tool/Function Calling
- Tool Recognition

AI agents built on large language models control the path to solving a complex problem… IBM



Buzzwords in Agentic Systems

LLM Agents

LLM agents are advanced AI systems designed for solving complex tasks that requires sequential reasoning.

A basic LLM agent requires a structured plan, a reliable memory to track progress and context, and access to necessary tools. These components form the framework of an agentic workflow with LLMs.

LLM agents can solve advanced problems, learn from their mistakes, use specialized tools to improve their work, and even collaborate with other agents to improve their overall performance.
However, they have limitations, such as a short memory span and a need for precise directions with tool calling.

Buzzwords in Agentic Systems

Tool/Function Calling

Tools calling is the capability of agentic systems to utilize a set of tool(s) inorder to interact with their environment, use external services such as Search APIs, Code Interpreter, bank transaction APIs and even Gmail.
Function Calling is very similar to tool calling with just a small variation, in that it is augmented with LLMs pre-training data for tool calling capabilities which involves defining a set of tool APIs and providing it as part of a request.

Tool Recognition

Tool recognition as I call it, is a sub-set of tool calling that I have identified. It looks at how LLM Agents are able to recognize complex description in tool calling parameter and use them effectively with minimal/zero false tool calls and hallucinations.

QUESTIONS


WHEN? WHY? WHERE? WHO? WHAT? HOW? WHY? WHEN? WHO? WHERE? HOW? WHAT? WHEN? WHERE? WHEN? WHERE? WHICH? WHOSE? HOW? WHEN? WHERE? HOW? WHAT? HOW? WHY? WHO? WHERE? WHAT? HOW? WHY? HOW? WHAT? HOW? WHY? WHO? WHERE? WHAT? HOW? WHERE? WHICH? WHOSE? WHEN? WHY? WHY? HOW? WHERE? WHO? WHOSE? WHERE? WHAT? HOW? WHAT? WHY? HOW? WHERE? WHERE? WHICH? WHOSE? WHEN? WHY? WHERE? HOW? WHERE? HOW? HOW? WHO? WHO? WHERE? WHAT? HOW? WHAT? HOW? WHY? WHERE? WHAT? HOW? HOW?

AI Summer of Code