Getting Started With Large Language Models

CONTENTS

DR. TUHIN CHATTOPADHYAY
PROFESSOR OF AI AND BLOCKCHAIN, JAGDISH SHETH SCHOOL OF MANAGEMENT

In the present era, large language models (LLMs) have emerged as transformative tools, unraveling the complexities of natural language understanding and paving the way for modern applications. Offering an introduction and practical insights on how to navigate the intricacies of harnessing LLMs, this Refcard serves as a comprehensive guide for both novices and seasoned practitioners seeking to unlock the capabilities of these powerful language models.

The primary purpose of this Refcard is to provide an end-to-end understanding of LLM architecture, training methodologies, as well as applications of advanced artificial intelligence models in natural language processing. The key goals include elucidating the theoretical foundations of LLMs, detailing their training processes, exploring practical applications across various domains, and discussing challenges and future directions in the field.

ABOUT LLM

A **large language model (LLM)** is a powerful artificial intelligence model designed to understand and generate human-like text based on vast amounts of data. These models belong to the broader category of natural language processing (NLP) in the even larger realm of machine learning. LLMs use deep neural networks with numerous parameters to learn patterns, relationships, and contextual information from diverse textual data sources.

HOW DO LARGE LANGUAGE MODELS WORK?

LLMs operate through a process known as **deep learning**, specifically using a type of architecture called *transformers*.

THE GENESIS

The genesis of large language models (LLMs) can be traced back to the revolutionary transformer architecture, a pivotal breakthrough in natural language processing. At the heart of this innovation lies the *attention mechanism*, a fundamental building block that redefined how models understand and process contextual information in vast amounts of text, catalyzing a paradigm shift in language representation and comprehension.

TRANSFORMER ARCHITECTURE

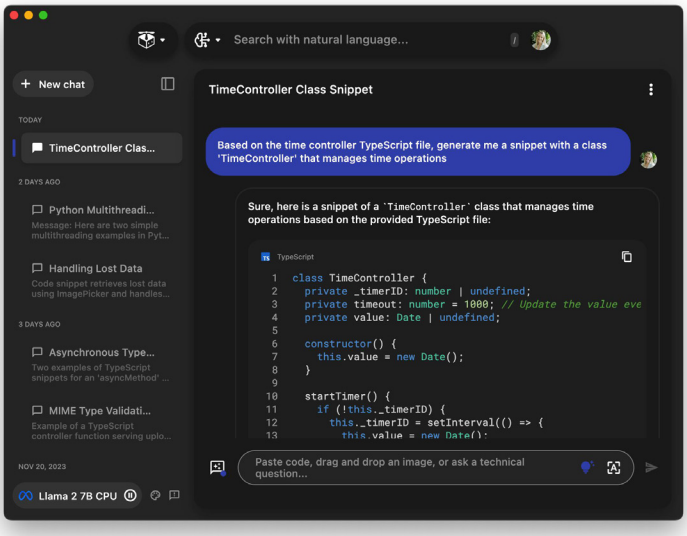
As mentioned, LLMs are built on transformer architectures [7]. Transformers enable a model to efficiently process and understand sequential data well-suited for natural language processing tasks. Comprising two fundamental components, the transformer architecture includes an *encoder* and a *decoder*. The encoder processes a sequence of input tokens, generating a corresponding sequence of hidden states. Subsequently, the decoder utilizes these hidden states to generate a sequence of output tokens.

PIECES
FOR DEVELOPERS™

The Most Contextual AI Development Assistant

Pieces is an AI-enabled productivity tool designed to increase developer efficiency and effectiveness through personalized workflow assistance across the entire toolchain.

Our centralized storage agent works on-device, unifying various developer tools to proactively capture and enrich useful materials, streamline collaboration, and solve complex problems through a contextual understanding of your unique workflow.



Learn More

Ideal for solo developers, teams, and cross-company projects

Contextual AI Copilot

The Pieces Copilot runs at the operating system-level, using the power of retrieval augmented generation to learn from your entire workflow and make contextualized suggestions.

LLM Utilization

Pieces is one of the first to offer fully functional LLM integrations across macOS, Linux, and Windows, giving users the option to leverage their choice of cloud, local, or custom LLMs.

Embedded Across your Workflow

Pieces is a tool-between-tools connecting the three main pillars of a developers' workflow: Researching and problem solving in the browser, coding in the IDE, and collaborating with teammates.

Automatic Enrichment

Our intelligent storage agent automatically attaches useful context and metadata to the code snippets and screenshots you save, enabling better organization, searchability, and reusability.

Workflow Activity Tracking

The Pieces Copilot runs at the operating system-level, using the power of retrieval augmented generation to learn from your entire workflow and make contextualized suggestions.

Security & Privacy

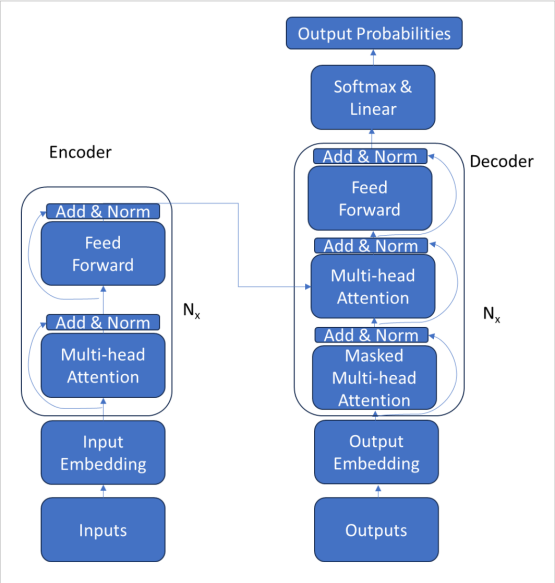
Pieces processes data on-device for air-gapped security and privacy. All AI capabilities can run entirely local or in the cloud, depending on operational constraints.

"Everyone's got a copilot. You're inverted, you've rotated the whole thing. It's not a vertical copilot, it's a horizontal one."

Scott Hanselman
VP of Developer Community at Microsoft
Via the Hanselminutes Podcast

Pieces minimizes context switching, accelerates onboarding, and significantly elevates the overall development experience while maintaining the privacy and security of your work.

Figure 1: Transformer architecture



ATTENTION MECHANISM

A key innovation in transformers is the **attention mechanism** [4]. This mechanism allows the model to focus on different parts of the input sequence when making predictions and captures long-range dependencies within the data. The attention mechanism is particularly powerful for tasks such as language understanding, translation, summarization, and more. It enhances the model's ability to generate coherent and contextually relevant responses by enabling it to weigh the importance of each input token dynamically.

Table 1: Key components of the attention mechanism

COMPONENTS	ROLE PLAYED
Query	The element of the input sequence for which the model is determining the relevance.
Key	The element of the input sequence against which the relevance of the query is evaluated.
Value	The output produced by the attention mechanism, representing the weighted sum of values based on the computed attention scores.

REAL-WORLD APPLICATIONS OF LLMS

From natural language understanding to innovative problem-solving, LLMs play a pivotal role across various domains, shaping the landscape of practical applications and technological advancements.

WHY LLMS MATTER

LLMs excel at understanding and generating human-like text, enabling more sophisticated interactions between machines and humans. LLMs can automate numerous language-related tasks, saving time and resources. In industries such as customer support, content generation, and data analysis, LLMs contribute to increased efficiency by handling routine language-based functions. LLMs enable the development of innovative applications and services, including chatbots, virtual assistants, content summarization, language translation, and sentiment analysis.

Additionally, LLMs are employed for generating human-like content, including articles, marketing materials, and code snippets. This is particularly valuable in the media, marketing, and software development industries, where high-quality and contextually relevant content is essential. In sectors like finance and healthcare, LLMs assist in analyzing and summarizing large volumes of textual information. This aids decision-makers in extracting relevant insights, mitigating risks, and making informed choices.

Table 2: Key applications of LLM across sectors

SECTORS	APPLICATIONS OF LLM
Healthcare	Clinical documentation, medical literature analysis
Finance	Sentiment analysis, customer support
Marketing	Content generation, SEO optimization
Legal	Document review, legal research
Education	Automated grading, content creation
Customer service	Chatbots, automated email responses
Technology	Code generation, bug detection
Media and entertainment	Content summarization, script writing
Human resources	Resume screening, employee feedback analysis
Manufacturing	Quality control, supply chain optimization

KEY CONCEPTS AND FEATURES OF LLM

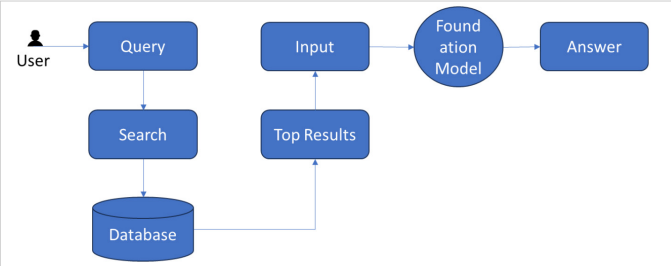
This section delves into the essential components and methodologies shaping the landscape of LLMs. From exploring the intricacies of core models and retrieval-augmented generation techniques to dissecting the practical applications facilitated by platforms like Hugging Face Transformers, this segment unveils key concepts.

Additionally, this section navigates through the significance of vector databases, the artistry of prompt design and engineering, and the orchestration and agents responsible for the functionality of LLMs. The discussion extends to the realm of local LLMs (LLLMs) and innovative Low-Rank Adaptation (LoRA) techniques, providing a comprehensive overview of the foundational elements that underpin the effectiveness and versatility of contemporary language models.

THE FOUNDATION MODEL AND RETRIEVAL-AUGMENTED GENERATION

The **foundation model** refers to the pre-trained language model that serves as the basis for further adjustments or customization. These models are pre-trained on diverse and extensive datasets to understand the nuances of language and are then fine-tuned for specific tasks or applications.

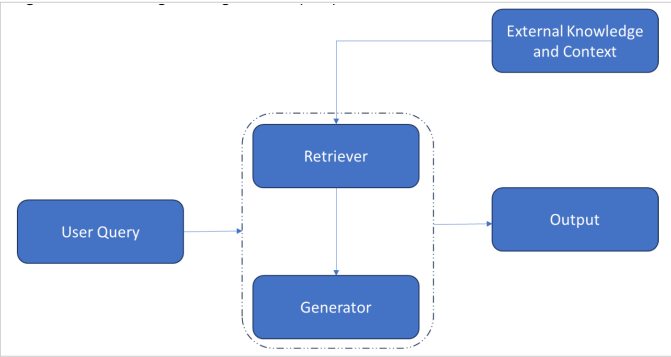
Figure 2: Foundation model



Retrieval-augmented generation (RAG) is a specific approach to natural language processing that combines the strengths of both retrieval models and generative models. In RAG, a *retriever* is used to extract relevant information from a large database or knowledge base, and this information is then used by a generative model to create responses or content. This approach aims to enhance the generation process by incorporating context or information retrieved from external sources.

RAG is particularly useful in scenarios where access to a vast amount of external knowledge is beneficial for generating more accurate and contextually relevant responses. This approach has applications in tasks such as question answering, content creation, and dialogue systems.

Figure 3: Retrieval-augmented generation (RAG)



HUGGING FACE TRANSFORMERS

Hugging Face Transformers [8, 11] emerges as an open-source deep learning framework developed by Hugging Face, offering a versatile toolkit for machine learning enthusiasts. This framework equips users with APIs and utilities for accessing cutting-edge pre-trained models and optimizing their performance through fine-tuning. Supporting a spectrum of tasks across various modalities, including natural language processing, computer vision, audio analysis, and multi-modal applications, Hugging Face Transformers simplify the process of downloading and training state-of-the-art pre-trained models.

VECTOR DATABASES

A **vector database** refers to a database designed to store and retrieve embeddings within a high-dimensional space. In this context, *vectors* serve as numerical representations of a dataset's features or attributes. Utilizing algorithms that compute distance or similarity between vectors in this high-dimensional space, vector databases excel in swiftly and efficiently retrieving data with similarities.

Unlike conventional scalar-based databases that organize data in rows or columns, relying on exact matching or keyword-based search methods, vector databases operate differently. They leverage techniques like Approximate Nearest Neighbors (ANN) to rapidly search and compare a substantial collection of vectors within an extremely short timeframe.

Table 3: Advantages of vector databases for LLMs

KEY ADVANTAGES	TASKS PERFORMED
Determining context	Vector embeddings enable LLMs to discern context, providing a nuanced understanding when analyzing specific words.
Detecting patterns	The embeddings generated encapsulate diverse aspects of the data, empowering AI models to discern intricate relationships, identify patterns, and unveil hidden structures.
Supporting a wide range of search options	Vector databases effectively address the challenge of accommodating diverse search options across a complex information source with multiple attributes and use cases.

Some of the leading open-source vector databases are Chroma [10,17], Milvus [13], and Weaviate [23].

PROMPT DESIGN AND ENGINEERING

Prompt engineering [9] involves the creation and refinement of text prompts with the aim of guiding language models to produce desired outputs. On the other hand, **prompt design** is the process of crafting prompts specifically to elicit desired responses from language models.

Table 4: Key prompting techniques

PROMPTING TECHNIQUES	*MODUS OPERANDI*
Zero-shot prompting	• Involves utilizing a pre-existing language model that has been trained on diverse tasks to generate text for a new task. • Makes predictions for a new task without undergoing any additional training.
Few-shot prompting	• Involves training the model with a small amount of data, typically ranging between two and five examples. • Fine-tunes the model with a minimal set of examples, leading to improved accuracy without requiring an extensive training dataset.
Chain-of-thought (CoT) prompting	• Directs LLMs to engage in a structured reasoning process when tackling challenging problems. • Involves presenting the model with a set of examples where the step-by-step reasoning is explicitly delineated.
Contextual prompts [3]	• Furnishes pertinent background information to steer the response of a language model. • Produces outputs that are accurate and contextually relevant.

ORCHESTRATION AND AGENTS

Orchestration frameworks play a crucial role in constructing AI-driven applications based on enterprise data. They prove invaluable in eliminating the necessity for retraining foundational models, surmounting token limits, establishing connections to data sources, and minimizing the inclusion of boilerplate code. These frameworks typically offer connectors catering to a diverse array of data sources, ranging from databases to cloud storage and APIs, facilitating the seamless integration of data pipelines with the required sources.

In the development of applications involving LLMs, orchestration and agents play integral roles in managing the complexity of language processing, ensuring coordinated execution, and enhancing the overall efficiency of the system.

Table 5: Roles of orchestration and agents

	KEY CAPABILITIES	DETAILED ROLE
Orchestration	Workflow management	Oversees the intricate workflow of LLMs, coordinating tasks such as text analysis, language generation, and understanding to ensure a seamless and cohesive operation.
	Resource allocation	Optimizes the allocation of computational resources for tasks like training and inference, balancing the demands of large-scale language processing within the application.
	Integration with other services	Facilitates the integration of language processing capabilities with other components, services, or modules.
Agents	Autonomous text processing	Handle specific text-related tasks within the application, such as summarization, sentiment analysis, or entity recognition, leveraging the capabilities of LLMs.
	Adaptive language generation	Generate contextually relevant and coherent language, adapting to user inputs or dynamically changing requirements.
	Dialogue management	Manage the flow of dialogue, interpret user intent, and generate appropriate responses, contributing to a more natural and engaging user experience.
	Knowledge retrieval and integration	Employ LLMs for knowledge retrieval, extracting relevant information from vast datasets or external sources and integrating it seamlessly into the application.

The synergy between orchestration and agents in the context of LLMs ensures that language-related tasks are efficiently orchestrated and that intelligent agents, powered by these models, can autonomously contribute to various aspects of application development. This collaboration enhances the linguistic capabilities of applications, making them more adaptive, responsive, and effective in handling natural language interactions and processing tasks.

AutoGen [5, 15] stands out as an open-source framework empowering developers to construct LLM applications through the collaboration of multiple agents capable of conversing and collaborating to achieve tasks. The agents within AutoGen are not only customizable and conversable but also adaptable to various modes that incorporate a mix of LLMs, human inputs, and tools. This framework enables developers to define agent interaction behaviors with flexibility, allowing the utilization of both natural language and computer code to program dynamic conversation patterns tailored to different applications. As a versatile infrastructure, AutoGen serves as a foundation for building diverse applications, accommodating varying complexities and LLM capacities.

Opting for AutoGen is more suitable when dealing with applications requiring code generation, such as code completion and code refactoring tools. On the other hand, LangChain proves to be a superior choice for applications focused on executing general-purpose Natural Language Processing (NLP) tasks, such as question answering and text summarization.

LOCAL LLMS

A **local LLM (LLLM)**, which runs on a personal computer or server, offers the advantage of independence from cloud services along with enhanced data privacy and security. By employing a local LLM, users ensure that their data remains confined to their own device, eliminating the need for external data transfers to cloud services and bolstering privacy measures. For instance, GPT4All [19] establishes an environment for the training and deployment of robust and tailored LLMs, designed to operate efficiently on consumer-grade CPUs in a local setting.

LOW-RANK ADAPTATION

Low-rank adaptation (LoRA) [1,20] is used for the streamlined training of personalized LLMs. The pre-trained model weights remain fixed, while trainable rank decomposition matrices are introduced into each layer of the transformer architecture. This innovative approach significantly diminishes the count of trainable parameters for subsequent tasks. LoRA has the capability to decrease the number of trainable parameters by a factor of 10,000 and reduces the GPU memory requirement by threefold.

HOW TO BUILD ENTERPRISE LLMS

Rather than depending on widely used LLMs like ChatGPT, numerous companies eventually develop their own specialized LLMs tailored to process exclusive organizational data. Enterprise LLMs have the capability to generate content specific to business needs, spanning marketing articles, social media posts, and YouTube videos. They can actively contribute to the creation, evaluation, and design of company-

specific software. Furthermore, enterprise LLMs may play a pivotal role in innovating and designing cutting-edge applications to secure a competitive advantage.

SOLVING THE BUILD OR BUY DILEMMA

When debating whether to independently pre-train an LLM or leverage an existing one, three alternatives come to the forefront:

A. Utilizing the API of a commercial LLM

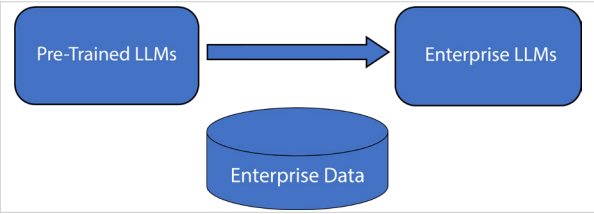
B. Employing an already available open-source LLM

C. Or, undertaking the task of pre-training an LLM independently.

The merits of employing pre-trained LLMs encompass ongoing performance enhancements and the capacity to adeptly handle a diverse range of complex tasks, including text summarization, content generation, code generation, sentiment analysis, and the creation of chatbots. Leveraging pre-trained LLMs offers the convenience of time- and cost-savings, particularly given the resource-intensive and costly nature of building a proprietary language model. Integration is simplified through APIs provided by services like ChatGPT, and users can enhance output quality via prompt engineering without altering the fundamental model.

Open-source LLMs, on the other hand, provide users with the ability to train and fine-tune the model to align with specific requirements. The complete code and structure of these LLMs are publicly accessible, offering increased flexibility and customization options. Noteworthy examples of open-source LLMs encompass Google PaLM 2, LLaMA 2 (released by Meta), and Falcon 180B (developed by Technology Innovation Institute). While open-source LLMs necessitate a higher level of technical proficiency and computational resources for training, they afford users greater control over data, model architecture, and enhanced privacy. Collaboration among developers is encouraged, fostering innovative training approaches and the creation of novel applications.

Figure 4: Enterprise LLMs

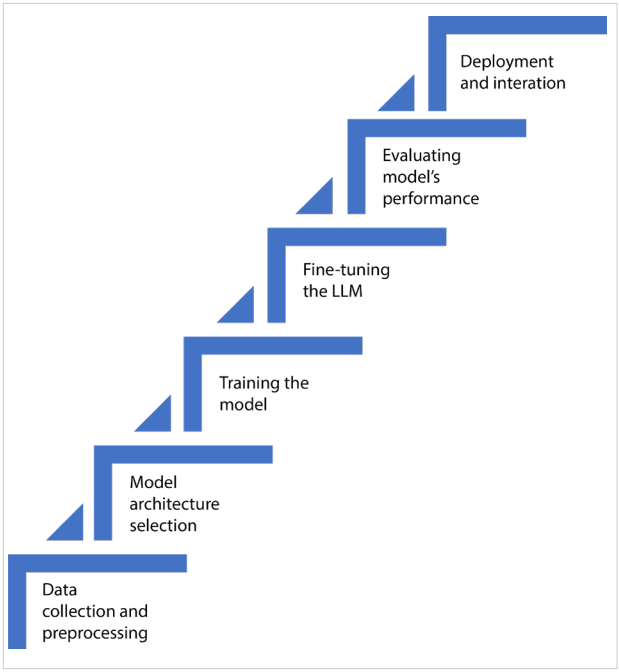


BUILDING A CUSTOM LLM

The development of a custom LLM involves a systematic process that includes several key steps:

1. Data collection and preprocessing: Gathering relevant and representative data, and preparing it for model training by cleaning, organizing, and transforming it as needed.
2. Model architecture selection: Choosing the appropriate architecture or design for the custom language model based on the specific requirements and characteristics of the task.
3. Training the model: Using the prepared data to train the selected model architecture, adjusting parameters to optimize performance and achieve desired learning outcomes.
4. Fine-tuning the LLM: Refining the pre-trained model on task-specific data to enhance its ability to understand and generate language patterns relevant to the targeted application.
5. Evaluating model performance: Assessing the model's effectiveness and accuracy using appropriate metrics to ensure it meets the desired standards and objectives.
6. Deployment and iteration: Implementing the model in the intended environment for real-world use, and continuously refining and updating it based on user feedback and evolving requirements.

Figure 5: Steps to developing a custom LLM



This comprehensive approach facilitates the development of a tailored and effective custom LLM for diverse applications.

PITFALLS OF LLM

While LLMs have demonstrated impressive capabilities, they are not without their pitfalls. LLMs can inherit and perpetuate biases present in their training data, leading to biased outputs. This can result in unfair or discriminatory language, reflecting societal biases present in the data. The use of LLMs for content generation raises ethical concerns, particularly in cases where generated content could be misused, for example, in creating fake news or misinformation. The generation of content by LLMs can raise legal and privacy concerns, especially when it comes to issues like intellectual property, plagiarism, or the inadvertent disclosure of sensitive information.

BUILDING ON TOP OF LLMS

In harnessing the potent capabilities of GPT-style foundation models for enterprise applications, the journey begins with data transformation

and record matching, paving the way for a paradigm shift in how businesses leverage the immense potential of LLMs. Building on top of these language models involves tailoring their functionalities to specific business requirements, customizing their training on domain-specific data, and integrating them seamlessly into existing workflows. Enterprises can unlock new dimensions of efficiency by employing LLMs for tasks such as document summarization, sentiment analysis, and customer interaction. Furthermore, the adaptability of these models allows for continuous refinement, ensuring that as business needs evolve, the language models can evolve in tandem. As organizations navigate the landscape of digital transformation, building on top of LLMs emerges as a strategic imperative, fostering innovation, enhancing decision-making processes, and ultimately driving a competitive edge in an increasingly data-driven business environment.

CONCLUSION: STATE-OF-THE-ART CONSIDERATIONS AND THE PATH FORWARD

From the expanding horizons of audio, image, and multimodal LLMs to the imperative of responsible LLMs in navigating ethical considerations and privacy, and finally, envisioning the future, this parting section examines the present landscape and illuminates the road ahead for the continuous evolution and responsible deployment of these groundbreaking technologies.

AUDIO, IMAGE, AND MULTIMODAL LLMS

A **multimodal LLM** [16] represents an advanced AI system that undergoes training with diverse modes of data, encompassing inputs from images, text, and audio sources to enhance its comprehension and generation capabilities.

A few leading multimodal LLMs are as follows:

- Gemini, Google's advanced multi-modal AI model, exhibits a superior capability to comprehend and process diverse forms of information simultaneously, encompassing text, code, audio, image, and video. As the successor to LaMDA and PaLM 2, Gemini, named after NASA's Project Gemini, represents a family of decoder-only Transformers, optimized for efficient training and inference on TPUs. Notably, Gemini surpasses human experts in Massive Multitask Language Understanding (MMLU), showcasing its prowess. Its versatility spans computer vision, geospatial science, human health, and integrated technologies. Google emphasizes Gemini's coding proficiency through AlphaCode 2, outperforming participants in coding competitions and demonstrating a remarkable 50 percent improvement over its predecessor. Trained on Google's Tensor Processing Units (TPU), Gemini boasts speed and cost efficiency, with plans to launch TPU v5p tailored for large-scale model training. Available in Nano, Pro, and Ultra variants, Gemini caters to diverse user needs, from fast on-device tasks to high-performance applications, with the Ultra version undergoing safety checks for release next year.
- Macaw-LLM [2, 21] is a groundbreaking innovation that seamlessly integrates visual, audio, and textual information. Comprising a modality module for encoding multimodal data, a cognitive module harnessing pre-trained LLMs, and an alignment module for harmonizing diverse representations, Macaw-LLM stands at the forefront of cutting-edge research in audio, image, and multimodal language models.
- The NExT Research Center at the National University of Singapore (NUS) has recently unveiled NExT-GPT [6, 18] — a cutting-edge "any-to-any" multimodal LLM designed to adeptly process text, images, videos, and audio as both input and output. Distinguished by its reliance on existing pre-trained models, NExT-GPT demonstrates remarkable efficiency by updating only 1% of its total parameters during training. NExT-GPT boasts of a versatile chat-based interface, empowering users to input text or upload files encompassing images, videos, or audio. With an exceptional ability to comprehend the content of diverse inputs, the model adeptly responds to user queries, generating text, image, video, or audio outputs tailored to user requests.

RESPONSIBLE LLMS: NAVIGATING ETHICAL CONSIDERATIONS AND PRIVACY

The utilization of LLMs gives rise to ethical concerns, encompassing the possibilities of biased outputs, privacy infringements, and the potential for misuse. To mitigate these issues, it is imperative to embrace transparent development practices, responsibly manage data, and incorporate fairness mechanisms. Addressing potential biases in outputs, safeguarding user privacy, and mitigating the risk of misuse are essential aspects of responsible LLM deployment. To achieve this, developers and organizations must adopt transparent development practices, implement robust privacy measures, and integrate fairness mechanisms to ensure ethical and unbiased outcomes. Balancing the transformative potential of LLMs with ethical considerations is crucial for fostering a trustworthy and responsible AI landscape.

THE FUTURE OF LARGE LANGUAGE MODELS

The future of LLMs promises exciting advancements. As these models evolve, the potential for self-fact-checking capabilities emerges, contributing to more reliable and accurate outputs. However, advancements will still depend on the development of better prompt engineering approaches to refine and enhance communication with these models. Additionally, the future holds the prospect of improved fine-tuning and alignment, ensuring that LLMs better sync with user intentions and generate contextually relevant responses. To make LLMs more accessible and applicable across diverse industries, providers must focus on developing tools that empower companies to establish their own reinforcement learning from human feedback (RLHF) pipelines. Customizing LLMs for specific applications will be a pivotal step forward, thus unlocking the full potential of these models in addressing industry-specific needs and fostering a more widespread adoption.

REFERENCES:

A. RESEARCH PAPERS:

1. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*. https://arxiv.org/abs/2106.09685
2. Lyu, C., Wu, M., Wang, L., Huang, X., Liu, B., Du, Z., ... & Tu, Z. (2023). Macaw-LLM: Multi-Modal Language Modeling with Image, Audio, Video, and Text Integration. arXiv preprint *arXiv:2306.09093*. Paper Link: https://arxiv.org/pdf/2306.09093.pdf
3. Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Context-Tuning: Learning Contextualized Prompts for Natural Language Generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6340–6354, Gyeongju, Republic of Korea. International Committee on Computational Linguistics. PDF: https://aclanthology.org/2022.coling-1.552.pdf Code: https://github.com/rucaibox/context-tuning
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems* (p./pp. 5998--6008). Access Link: https://papers.neurips.cc/paper/7181-attention-is-all-you-need.pdf
5. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., ... & Wang, C. (2023). Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*. https://doi.org/10.48550/arXiv.2308.08155
6. Wu, S., Fei, H., Qu, L., Ji, W., & Chua, T. S. (2023). Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*. https://arxiv.org/abs/2309.05519

B. TUTORIALS:

7. Harvard: From Transformer to LLM: Architecture, Training and Usage (Transformer Tutorial Series) - https://scholar.harvard.edu/binxuw/classes/machine-learning-scratch/materials/transformers
8. Hugging Face: NLP Course - https://huggingface.co/learn/nlp-course/chapter1/1
9. DeepLearning.ai: ChatGPT Prompt Engineering for Developers - https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/

C. DOCUMENTATIONS:

10. Chroma: https://docs.trychroma.com/
11. Hugging Face Transformers: https://huggingface.co/docs/transformers/index
12. Langchain: https://python.langchain.com/docs/get_started/introduction
13. Milvus: https://milvus.io/docs
14. OpenAI:

    i. GPT-4V(ision) system card - https://cdn.openai.com/papers/GPTV_System_Card.pdf

    ii. GPT-4 Technical Report: https://cdn.openai.com/papers/gpt-4.pdf

D. REPOSITORIES:

15. AutoGen: https://github.com/microsoft/autogen
16. Awesome-Multimodal-Large-Language-Models: https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models#awesome-multimodal-large-language-models
17. Chroma: AI-native open-source embedding database - https://github.com/chroma-core/chroma
18. Code and models for NExT-GPT: Any-to-Any Multimodal Large Language Model - https://github.com/NExT-GPT/NExT-GPT
19. GPT4All: https://github.com/nomic-ai/gpt4all?source=post_page-----2598615a039a---------- ----------
20. LoRA: Low-Rank Adaptation of Large Language Models - https://github.com/microsoft/LoRA
21. Macaw-LLM: Multi-Modal Language Modeling with Image, Video, Audio, and Text Integration - https://github.com/lyuchenyang/Macaw-LLM
22. Milvus: A cloud-native vector database, storage for next generation AI applications - https://github.com/milvus-io/milvus
23. Weaviate: https://github.com/weaviate/weaviate

WRITTEN BY DR. TUHIN CHATTOPADHYAY,
PROFESSOR OF AI AND BLOCKCHAIN, JAGDISH SHETH SCHOOL OF MANAGEMENT

Dr. Tuhin Chattopadhyay is a highly esteemed and celebrated figure in the fields of Industry 4.0 and data science, commanding immense respect from both the academic and corporate fraternities. Dr. Tuhin has been recognized as one of India's Top 10 Data Scientists by Analytics India Magazine, showcasing his exceptional skills and profound knowledge in the field. Dr. Tuhin serves as a Professor of AI and Blockchain at JAGSoM, located in Bengaluru, India. Dr. Tuhin is also a visionary entrepreneur, spearheading his own AI consultancy organization that operates globally.

DZone
3343 Perimeter Hill Dr, Suite 100
Nashville, TN 37211
888.678.0399 | 919.678.0300

At DZone, we foster a collaborative environment that empowers developers and tech professionals to share knowledge, build skills, and solve problems through content, code, and community. We thoughtfully — and with intention — challenge the status quo and value diverse perspectives so that, as one, we can inspire positive change through technology.