Hallucinations in LLMs: Understanding and Addressing Challenges

Gabrijela Perković, Antun Drobnjak, Ivica Botički
University of Zagreb Faculty of Electrical Engineering and Computing, Zagreb, Croatia
ivica.boticki@fer.hr

2024 47th MIPRO ICT and Electronics Convention (MIPRO) | 979-8-3503-8250-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/MIPRO60963.2024.10569238

***Summary* — Large language models (LLM) are trained to understand and generate human-like language. While LLMs present a cutting-edge concept and their use is becoming widespread, hallucinations sometimes occur during their operation. Hallucinations refer to instances where the model generates inaccurate or fictitious information, deviating from factual knowledge and potentially providing responses that lack a basis in model's training data. In this paper, the ways in which LLMs generate text are examined to address the question of why hallucinations occur. The paper additional explores how existing LLM models can be leveraged to reduce the likelihood of hallucination. Alongside exploring hallucinations, this paper provides insights into the algorithms used for training LLMs, offering a clear picture of the text generation process and its effective utilization.**

Key words – LLM, hallucination, transformer model

I. INTRODUCTION

In recent years, there has been significant interest in the advancement of Large Language Models (LLMs), highlighted by the debut of the groundbreaking ChatGPT. Users have marveled at the program's capacity to emulate human language seamlessly, offering coherent responses to queries. Fundamentally propelled by neural networks and deep learning, LLMs aim to mimic data processing as closely as possible to the human brain.

Despite their revolutionary strides, LLMs are still a developing field with inherent limitations and imperfections. Users, engaging in dialogues with LLMs, have occasionally encountered instances of misinformation or nonsensical information formed in such persuasive way as if it were a fact. This phenomenon is often called hallucination. [1] Many experts are working on addressing and minimizing the issue.

The investigation into the cause and resolution of hallucinations within LLMs is currently a focal point of extensive research. This article endeavors to delve into the learning mechanisms of LLMs, discerning the causal factors behind hallucinations, and proffering prospective solutions to mitigate this issue in subsequent advancements.

II. FOUNDATIONS OF AN LLM

A Large Language Model, also known by its acronym LLM, is a type of generative artificial intelligence (AI) trained to recognize, process and generate text akin to human linguistic patterns. Essentially, it specializes in Natural Language Processing (NLP), accommodating both human languages and those utilized in computer programming [2]. To perform such tasks, LLMs rely on neural networks, a method in artificial intelligence made of nodes that mimic human neurons to process information.

A. Constructing an LLM

An LLM undergoes training on an extensive dataset collected from the vast expanse of the Internet in the order of thousands of gigabytes [2]. The emphasis is placed on the sheer quantity of data processed through a program deployed across multiple Graphics Processing Units (GPUs). Upon concluding the training process, we obtained its base model, equipped with the capability to generate text using predictive analytics for the next word. It is imperative to underscore that this training procedure entails significant costs and prolonged durations, prompting companies to undertake this intricate process approximately once a year, investing substantial financial resources, often in the magnitude of millions of dollars.

Afterwards, a process of fine tuning is executed. This step is crucial for formatting the model to reply to questions, and not just generate text. This critical step transforms the model from a mere text generator into an assistant with access to the wealth of information acquired in the preceding training phase. The focus here is on quality of information. Furthermore, it requires less financial investment, hence it is conducted much more often.

Some experts additionally conduct reinforcement learning from human feedback (RLHF) in which humans examine some responses from the model and give feedback on how well it was [3]. Subsequently, the model incorporates these reviews into its existing dataset, leveraging the collective insights gained from human assessment to refine and augment its performance.

Ultimately, the outcome is an extensive database comprised of words and numerical representations, encapsulating the probability of a word being generated next and its correlations with other words. A relatively straightforward code is then employed to execute the model, leveraging the acquired dataset. This integration of probabilities and correlations facilitates the model's ability to generate coherent and contextually relevant responses in a streamlined manner [4].

B. Additional features

The application of LLMs is already quite vast: from customers service and chatbots to DNA research.

Companies already take an LLM and feed it some personalized data to create their own little helpers and co-pilot their employees. Some of the widely known programs based on LLMs are: ChatGPT (OpenAI) [5], Bard (Google), Llama (Meta) and Bing Chat [6].

One of the main advantages of such a model is its ability to respond to any unpredictable type of question, without having a finite group of understandable queries, and it gives sensical answers. On the other hand, the information is as reliable as the data it had been fed. If the information is unknown or not as prevalent, LLM generates hallucinations. Additionally, most of the LLMs do not have strong security. They are prone to bugs, which can enable intruders to adjust the model to prefer certain type of answers [7]. Also, people are not aware that the data that they input to LLMs is not completely secure and private, as it may be utilized for additional purposes.

III. THE TRANSFORMER MODEL

LLMs are a subset of machine learning, a field of artificial intelligence that focuses on creating algorithms allowing systems to learn patterns and make decisions without explicit programming [7]. Within the scope of machine learning, they are trained on extensive datasets to recognize, process, and generate text in a manner akin to human language patterns. Neural networks are a specific architecture within machine learning, inspired by the structure and functioning of the human brain. They consist of interconnected nodes organized in layers (input, hidden, output). LLMs utilize neural networks in their architecture. LLMs fall under the umbrella of deep learning, a specialized form of machine learning, characterized by the integration of neural networks featuring multiple hidden layers, allowing for the extraction of complex hierarchical features. This depth enables LLMs to capture and understand complex patterns in language. It is more advanced than basic machine learning since data does not have to be labeled nor does it need any context to be processed.

Large Language Models (LLMs) are grounded in the transformer model, a specific type of deep learning architecture. The transformer architecture, introduced in [8] is specifically designed for sequence-to-sequence tasks. In other words, it uses special mathematical techniques to learn the context and establish relations between words. It excels in comprehending how the conclusion of one sentence relates to the inception of another, enabling it to process entire paragraphs seamlessly.

To get a better grasp of understanding how LLMs work, it is essential to understand the architecture of the transformer model. This model is used for training the LLM and later for communication with users. It is based on predicting the next word based on the context and calculated values during training phase.

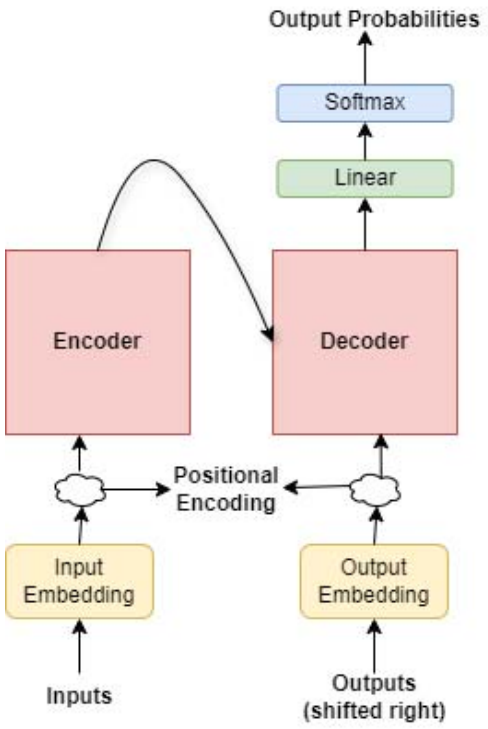

Figure 3.1 The transformer model [8]

A. *Input Embedding*

As machines lack inherent comprehension of words, the initial step involves transforming inputs into numerical information. Each word in a text is mapped to a vector of values, essentially serving as a numerical representation for that token. This encoding process is orchestrated so that words sharing similar meanings are allocated comparable vectors. This process can be compared to establishing a dictionary, facilitating the bridge between human language and machine comprehension.

B. *Positional Encoding*

To preserve context and create relations between words, it is mandatory to retain the positional information in the sentence. This step used sine and cosine functions because of their linear features and combines them with assigned embeddings.

C. *Encoder*

The encoder, represented by the amalgamation of layers above the input embedding in Figure 3.1, undertakes the processing of transformed text data to capture its meaning. Employing the self-attention mechanism, it generates layers of hidden states. Self-attention is a mechanism used to capture dependencies and relationships within input sequences [9]. In essence, it calculates the focus on specific words, identifying which words frequently co-occur, determining common structures indicative of questions, and essentially understanding how each word relates to others in the sequence.

Iterative application of this step enhances precision, providing an opportunity for refinement.

D. Output Embedding

During this phase, the data is shifted to the right to predict the subsequent word in the sequence. Words are embedded through the process described in section 3.A. To enhance performance, the loss function is used, comparing the model's predictions with the actual values. This comparison aids in fine-tuning the model by correcting and optimizing the data.

E. Decoder

The decoder, integral to the model, generates the output sequence by learning to predict the next word during training, informed by preceding words. In GPT, the decoder produces natural language text, leveraging the input sequence and contextual information acquired by the encoder [2].

F. Lineaar and Softmax

In the final step, the computed vectors undergo linear operations to be transformed into tokens, each with assigned probabilities. This process ensures the conversion of numerical representations back into meaningful language tokens with associated likelihoods.

IV. HALLUCINATIONS

Large language models (LLMs) have recently garnered significant attention in technology, sparking widespread public interest. However, despite being built on cutting-edge technologies, LLMs have exhibited peculiar behaviors, occasionally presenting false or nonsensical information as if it were factual. Such occurrences are known as hallucinations. There are several types of hallucinations [10]:

- Sentence contradiction is a form of hallucination where a generated sentence contradicts another one previously generated.
- Prompt contradiction arises when a Large Language Model (LLM) produces text conflicting with the specified request.
- Factual contradiction occurs when the model states a false statement as if it were true, such as claiming Obama is the first president of the USA.
- Additionally, nonsensical output is another type of hallucination, characterized by generated text lacking meaning or logical coherence.

While they may be deemed useful in generating art, such as images or literature, or in brainstorming, hallucinations reduce reliability and experts are keen on minimizing its impact.

The potential culprits for hallucinations are listed in remainder of this chapter [10].

A. Questionable training data

In the process of training the model, experts accumulated vast amounts of data from diverse internet sources, with a primary focus on platforms like Wikipedia. However, information was also sourced from fictional stories and unchecked data. This eclectic dataset introduces the potential for the final model to establish unexpected relations and probabilities, resulting in unforeseen outputs. In the vast sea of gathered data, reliable facts can get lost, and at times, information within the dataset may even contradict itself, leading to confusion for the model in the end.

Fact-checking every piece of training data was deemed impractical due to the size of data, and data size should not be compromised. LLM's unique specialty lies in not requiring contextualized or tokenized data.

B. Lack of logic

As explained in the previous chapter, an LLM model lacks a sense of logic and operates by mimicking human language based on the probability of generating specific words in given contexts. The model lacks a mechanism for fact-checking the reliability of generated text, having diverged from its original sources during training. Crucially, there was no explicit training step instructing the model to express uncertainty or acknowledge when it lacks knowledge.

To comprehend this concept more thoroughly, we can describe the way this model "thinks" to being automatic. It operates swiftly and automatically, devoid of explicit thinking processes. Drawing a parallel with the human brain, consider the automatic response to a question like "What is 2+2?" – a learned and immediate reaction where the answer of 4 comes naturally, eliminating the need for recalculating every time. Similarly, in speed chess, players make rapid moves without extensive contemplation, relying on what feels right in the moment [9]. Large language models (LLMs) function in a comparable manner, executing tasks automatically based on learned patterns and probabilities.

C. Vague prompts

It is important to note that hallucinations may arise when the model encounters a vague prompt, an unspecified question, or a task that lacks clarity, resulting in the generation of irrelevant or generalized output. For instance, a prompt with an abbreviation or a context-dependent question might be open to multiple interpretations, contributing to potential hallucinations.

D. Insufficient data or overfitiing

In the preparation of a dataset, experts must carefully consider the appropriate size for the model. Mathematical calculations can be employed to estimate the model's reliability. Insufficient training data may hinder the model's ability to develop a comprehensive understanding of natural language models, patterns, and relationships. This limitation could result in the model having a restricted domain of knowledge, leading to the generation of random facts for topics outside its specialized training. For instance, certain contemporary LLMs may be predominantly trained

for specific domains like medicine, law, or technology, potentially lacking knowledge in other fields.

Furthermore, models may become excessively entwined with their training data, rendering them incapable of generating coherent text on different topics and potentially giving rise to nonsensical claims.

V. MITIGATING THE IMPACT

Although the definitive solution for hallucinations in LLMs is still unknown, there are strategies available for both users and developers to mitigate them. Immediate actions can be taken, while others are recommended for the development of future models [10].

A. Adjusting prompts

Effective interaction with a Language Model (LLM) hinges on tailoring requests to its capabilities. Users should begin by clearly defining their goals and expectations. Precise instructions should accompany requests, specifying desired information, answer format, and any other relevant criteria [11]. Ambiguity in wording should be avoided, and context should be provided to aid comprehension. For instance, asking "Do cats talk?" could yield varied responses without clarification because cats indeed "talk" in cartoons and cats exhibit their own intricate system of vocalizations and body language in real life.

Although queries may span lengthy passages, users are advised to partition sizable requests into smaller, more manageable segments. By soliciting the model to generate concise pieces of text individually, optimal outcomes are achieved, mitigating the potential for model confusion and cognitive overload induced by extensive input.

B. Retrieval-Augmented Generation (RAG)

The Retrieval-Augmented Generation (RAG) is a method used to increase reliability and accuracy of generative AI models. RAG acts as a safety net for LLMs, reducing hallucinations by providing current, relevant, or previously unexplored data during text generation. This approach ensures a more informed and accurate output, mitigating reliance solely on training data [12].

RAG operates by retrieving data through various means. One method involves utilizing algorithms to browse and gather data related to the user's prompt from indexed online documents, web documents that have been systematically organized and stored in a searchable database. When a document is indexed, its contents are analyzed, and relevant keywords or metadata are extracted and stored in a structured format. Also, RAG may access pre-prepared datasets stored in a vector database, enabling quick and targeted retrieval of specific information, but that dataset must be manually updated to ensure its relevance [13]. This retrieved information is added to the user's prompt as context and given as input to the transformer. By integrating external sources and combining them with existing training data, RAG generates text that is both contextually grounded and credible.

The benefits of RAG include providing LLMs with access to the most recent and verified data, enhancing accuracy in text generation. Additionally, the ability to cite sources promotes transparency and credibility. Moreover, RAG reduces the need for frequent model updates and repeated learning processes, leading to cost savings and operational efficiencies.

C. Fine-tuning

To improve LLMs, organizations need robust feedback and improvement mechanisms. This involves actively refining prompts and tuning datasets based on user feedback. Rigorous adversarial testing identifies vulnerabilities, and incorporating human validation processes and continuous monitoring is essential for mitigating hallucinations and enhancing customer experiences. Additionally, facilitating domain adaptation and augmentation, providing domain-specific knowledge, and fine-tuning with domain-specific data align the model with specific domains, reducing hallucination [14].

Fine-tuning a machine learning model entails selecting a smaller, specialized dataset for targeted training. Despite potentially containing thousands of data points, these datasets are smaller than the original training set. Developers train the model with the new dataset, employing a lower learning rate to preserve existing knowledge while adapting to new data, primarily focusing on adjusting later layers.

D. Model configuration

Model configuration involves adjusting parameters like temperature, frequency penalty, presence penalty, and top-p to influence the generated output [15]. Higher temperature values enhance randomness and creativity, while lower values make the output more deterministic. Additionally, implementing a moderation layer adds a control mechanism to filter inappropriate or irrelevant content, ensuring generated responses adhere to predefined standards and guidelines.

VI. CONCLUSIONS

The widespread integration of LLMs in everyday life is surely going to transform various aspects of workplaces, education, and other areas. Despite being so innovative, it carries the risk of producing false information, known as hallucinations. While LLMs can be valuable for activities like brainstorming, entertainment, and art creation, the occurrence of hallucinations is generally undesirable. Experts have identified causes and developed methods to reduce the impact of hallucinations in future models, and there are also strategies for users to interact more efficiently with current models. Despite their innovation, LLMs are still evolving and are expected to undergo significant improvements in the coming decades.

REFERENCES

[1] R. Hu, J. Zhong, M. Ding, Z. Ma, and M. Chen, “Evaluation of Hallucination and Robustness for Large Language Models,” in

2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C), 2023, pp. 374–382. doi: 10.1109/QRS-C60940.2023.00089.

[2] Elastic, "What is a large language model (LLM)?" Accessed: Jan. 08, 2024. [Online]. Available: https://www.elastic.co/what-is/large-language-models

[3] T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, "A Survey of Reinforcement Learning from Human Feedback." 2023.

[4] N. Lambert, T. Krendl Gilbert, and T. Zick, "The History and Risks of Reinforcement Learning and Human Feedback," 2023, [Online]. Available: https://arxiv.org/abs/2310.13595

[5] B. Kosma, "Which companies are working on LLMs and ChatGPT alternatives?" Accessed: Jan. 20, 2024. [Online]. Available: https://techmonitor.ai/technology/companies-large-language-models-llms-chatgpt-alternatives

[6] Microsoft, "What is Bing Chat, and how can you use it?" 2023. Accessed: Jan. 20, 2024. [Online]. Available: https://www.microsoft.com/en-us/bing/do-more-with-ai/what-is-bing-chat-and-how-can-you-use-it?form=MA13KP

[7] P. Menon, "Introduction to Large Language Models and the Transformer Architecture." Accessed: Jan. 08, 2024. [Online]. Available: https://rpradeepmenon.medium.com/introduction-to-large-language-models-and-the-transformer-architecture-534408ed7e61

[8] A. Vaswani *et al.*, "Attention Is All You Need," *CoRR*, vol. abs/1706.03762, 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[9] A. Karpathy, "[1hr Talk] Intro to Large Language Models." Accessed: Jan. 08, 2024. [Online]. Available: https://www.youtube.com/watch?v=zjkBMFhNj_g&list=LL&index=2&t=3153s&ab_channel=AndrejKarpathy

[10] M. Bilan, "Hallucinations in LLMs: What You Need to Know Before Integration." Accessed: Jan. 17, 2024. [Online]. Available: https://masterofcode.com/blog/hallucinations-in-llms-what-you-need-to-know-before-integration#:~:text=LLMs%20may%20also%20generate%20hallucinations,extraneous%20and%20potentially%20misleading%20content

[11] S. J. Bigelow, "10 prompt engineering tips and best practices." Accessed: Mar. 19, 2024. [Online]. Available: https://www.techtarget.com/searchenterpriseai/tip/Prompt-engineering-tips-and-best-practices

[12] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey." 2024.

[13] A. Arsanjani, "Navigating the Challenges of Hallucinations in LLM Applications: Strategies and Techniques for Enhanced Accuracy." Accessed: Jan. 15, 2024. [Online]. Available: https://dr-arsanjani.medium.com/navigating-the-challenges-of-hallucinations-in-llm-applications-strategies-and-techniques-for-ab2b5ddc4a63

[14] S. Jha, S. K. Jha, P. Lincoln, N. D. Bastian, A. Velasquez, and S. Neema, "Dehallucinating Large Language Models Using Formal Methods Guided Iterative Prompting," in *2023 IEEE International Conference on Assured Autonomy (ICAA)*, 2023, pp. 149–152. doi: 10.1109/ICAA58325.2023.00029.

[15] A. Kimothi, "7 Key Prompt Engineering Parameters Everyone Should Know." 2023. Accessed: Jan. 18, 2024. [Online]. Available: https://medium.com/mlearning-ai/7-key-prompt-engineering-parameters-everyone-should-know-4b3a330865a8