Why do Large Language Models Hallucinate

As part of his bid for early prison release, Michael D. Cohen used Google Bard to identify legal citations that were relevant to his case. Cohen gave his lawyer Daniel Schwartz bogus legal citations who would later submit them. Cohen believed that Bard was a "supercharged search engine" but Cohen's use of AI goes beyond damaging his appeal for early release. As the star witness for a Manhattan criminal case against former President Trump, Cohen has been attacked by Trump's lawyers as a "serial fabulist." Cohen's fallacious citations only add fire to the flame.

What Bard did was hallucinate; the Large Language Model (LLM) generated text that seemed syntactically sound but was factually incorrect. As embarrassing as this was in federal court, it raises the question of what causes LLMs, such as ChatGPT, to hallucinate and what measures can be put in place to forestall hallucinations.

LLMs are not databases; they can't validate their responses, making hallucinations unavoidable. When responding to a prompt, the LLM extrapolates from the prompt provided and training set utilized. Occasionally, hallucinations can work in the user's benefit. Suppose the user wants to gain inspiration for a short story; an LLM that behaves erratically may generate more diverse examples and inspire the author.

On the other hand, hallucinations in LLMs spread misinformation and raise safety concerns for real-world applications. What if all lawyers used ChatGPT to conduct research for their cases? What if jurors blindly believed in these lawyers and–in the same thread–believed in ChatGPT? Convictions may be misguided; those who are guilty may walk free while those who are innocent may be imprisoned.

Generally speaking, hallucinations in LLMs are caused by limited contextual understanding. The model is obligated to respond regardless of whether it has domain knowledge in the field.

Moreover, hallucinations can be attributed to training data. When a large corpus of data is collected, noise, such as phrases in the output that cannot be explained in the input, appear. Models pick up on the noise and respond in unexpected ways, generating fluent but unsupported text. Duplicates in the training corpus can bias the model towards generating frequent phrases. Moreover, the prioritization of parametric knowledge–knowledge acquired during pre-training and implicitly stored in model parameters–over the provided contextual knowledge results in hallucinations.

The architecture of LLMs can also cause hallucinations. LLMs are decoder-only architectures, meaning a smaller draft model is used to predict the target model's outputs. The decoding technique may be random in nature, resulting in greater diversity in results but also increased hallucinations.

Possible solutions to hallucinations can be expensive. To preprocess and enhance the quality of the dataset, irrelevant information, duplicates, and outliers can be removed either by hand or through an automated process. The model could then be retrained. However, most devices cannot handle training a large LLM on their own and even fine-tuning a model may be impossible.

The most practical intervention may come from humans. Users can ask the model to regenerate a response if the original is gravely wrong. For example, controlled generation and effective prompt engineering can also be practiced. When writing a prompt, users can provide

enough details about the question at hand and the role the LLM should play in answering it. Doing so limits the model's capacity to hallucinate.

When it comes to LLMs, hallucinations are an unavoidable challenge. When a model generates text, it cannot tell if its response is actual. Because hallucinations in LLMs can pose a threat to society, researchers should not sit idle but devise strategies to ameliorate the frequency of hallucinations. Current mitigation strategies are either too computationally expensive or require manual effort.