

= Avoiding Hallucination
:order: 2
:type: lesson

As you learned in the previous lesson, LLMs can "make things up".

LLMs are designed to generate human-like text based on the patterns they've identified in vast amounts of data.

Due to their reliance on patterns and the sheer volume of training information, LLMs sometimes ****hallucinate**** or produce outputs that manifest as generating untrue facts, asserting details with unwarranted confidence, or crafting plausible yet nonsensical explanations.

These manifestations arise from a mix of overfitting, biases in the training data, and the model's attempt to generalize from vast amounts of information.

== Common Hallucination Problems

Let's take a closer look at some reasons why this may occur.

=== Temperature

LLMs have a temperature, corresponding to the amount of randomness the underlying model should use when generating the text.

The higher the temperature value, the more random the generated result will become, and the more likely the response will contain false statements.

A higher temperature may be appropriate when configuring an LLM to respond with more diverse and creative outputs, but it comes at the expense of consistency and precision.

For example, a higher temperature may be suitable for constructing a work of fiction or a novel joke.

On the other hand, a lower temperature, even `0`, is required when a response grounded in facts is essential.

[TIP]

.Consider the correct temperature

====

In June 2023,

link:<https://www.reuters.com/legal/new-york-lawyers-sanctioned-using-fake-chatgpt-cases-legal-brief-2023-06-22/>[A US judge sanctioned two US lawyers for submitting an LLM-generated legal brief^] that contained six fictitious case citations.

====

A quick fix may be to reduce the temperature. But more likely, the LLM is hallucinating because it hasn't got the information required.

=== Missing Information

The training process for LLMs is intricate and time-intensive, often requiring vast datasets compiled over extended periods. As such, these models might lack the most recent information or might miss out on specific niche topics not well-represented in their training data.

For instance, if an LLM's last update were in September 2022, it would be unaware of world events or advancements in various fields that occurred post that date, leading to potential gaps in its knowledge or responses that seem out of touch with current realities.

If the user asks a question on information that is hard to find or outside of the public domain, it will be virtually impossible for an LLM to respond accurately.

Luckily, this is where factual information from data sources such as knowledge graphs can help.

=== Model Training and Complexity

The complexity of Large Language Models, combined with potential training on erroneous or misleading data, means that their outputs can sometimes be unpredictable or inaccurate.

For example, an LLM might produce a biased or incorrect answer when asked about a controversial historical event.

Furthermore, it would be near impossible to trace back how the model arrived at that conclusion.

== Improving LLM Accuracy

The following methods can be employed to help guide LLMs to produce more consistent and accurate results.

=== Prompt Engineering

Prompt engineering is developing specific and deliberate instructions that guide the LLM toward the desired response.

By refining how you pose instructions, developers can achieve better results from existing models without retraining.

For example, if you require a blog post summary, rather than asking `_"What is this blog post about?"_`, a more appropriate response would be `_"Provide a concise, three-sentence summary and three tags for this blog post."_`

You could also include `_"Return the response as JSON"_` and provide an example output to make it easier to parse in the programming language of your choice.

Providing additional instructions and context in the question is known as ****Zero-shot learning****.

[TIP]

.Be Positive

====

When writing a prompt, aim to provide positive instructions.

For example, when asking an expert to provide a description, you could say, "Do not use complex words." However, the expert's interpretation of complex might be different from yours. Instead, say, "Use simple words, such as". This approach provides a clear instruction and a concrete example.

====

=== In-Context Learning

In-context learning provides the model with examples to inform its responses, helping it comprehend the task better.

The model can deliver more accurate answers by presenting relevant examples, especially for niche or specialized tasks.

Examples could include:

* Providing additional context - `When asked about "Bats", assume the question is about the flying mammal and not a piece of sports equipment.`

* Providing examples of the typical input - `Questions about capital cities will be formatted as "What is the capital of {country}?"`

* Providing examples of the desired output - `When asked about the weather, return the response in the format "The weather in {city} is {temperature} degrees Celsius."`

Providing relevant examples for specific tasks is a form of ****Few-shot learning****.

=== Fine-Tuning

Fine-tuning involves additional language model training on a smaller, task-specific dataset after its primary training phase. This approach allows developers to specialize the model for specific domains or tasks, enhancing its accuracy and relevance.

For example, fine-tuning an existing model on your particular businesses would enhance its capability to respond to your customer's queries.

This method is the most complicated, involving technical knowledge, domain expertise, and high computational effort.

A more straightforward approach would be to ground the model by providing information with the prompt.

=== Grounding

Grounding allows a language model to reference external, up-to-date sources or databases to enrich the responses.

By integrating real-time data or APIs, developers ensure the model remains current and provides factual information beyond its last training cut-off.

For instance, if building a chatbot for a news agency, instead of solely relying on the model's last training data, grounding could allow the model to pull real-time headlines or articles from a news API. When a user asks, "What's the latest news on the Olympics?", the chatbot, through grounding, can provide a current headline or summary from the most recent articles, ensuring the response is timely and accurate.

image::images/llm-news-agency.svg[A news agency chatbot, showing the user asking a question, the chatbot grounding the question with a news API, and the chatbot responding with the latest news.]

== LLMs and Knowledge Graphs

In the coming lessons, you will explore these topics in detail and discover how LLMs can use Knowledge Graphs to improve their accuracy and relevance.

== Check Your Understanding

```
include::questions/1-temperature.adoc[leveloffset=+1]
include::questions/2-external-data.adoc[leveloffset=+1]
```

[.summary]

== Lesson Summary

In this lesson, you explored the intricacies of Large Language Models (LLMs), understanding their tendencies to hallucinate and the various strategies to improve their accuracy, such as temperature settings, prompt engineering, in-context learning, fine-tuning, and grounding with external data sources like APIs.

In the next lesson, you will learn about the techniques for grounding an LLM.