Part II of III in a series on Artificial Intelligence written by our colleague Alan D. Mead, Ph.D.

AI Models – The Little Engine that Could (or Might Not)

You've submitted your data to an AI app, and now you're reviewing its output. You'll see what it culls from its source data and what it "thinks" is most and least relevant, along with everything else in between. Here's what you should know about source data and why it's important to be critical about output. Not convinced? See the section called ***"Why do models hallucinate?"***

The model is NOT searching the Internet for an answer. Rather, it is relying on its analysis of patterns between tokens to compose a novel response, one token at a time. The patterns have been encoded in the **model parameters** during the **model training**.

ChatGPT is not a person and not like a person. If a human could respond to all the ChatGPT requests, they would probably learn things from all the queries. That's NOT happening with the current generation of LLMs. In fact, you could say that LLMs learn in the same way that houses grow. A house could grow… If you have room on your lot, and you made a blueprint, and you got all the permits, and hired a crew, and rented equipment, and purchased materials… In weeks or months, you could have a bigger house. If a company like OpenAI assigns a team, gives them a budget and they find more data, they could make the model learn more in a few weeks or months. But current LLMs as not changing and spontaneously learning. In fact, every LLM will have a cutoff date. The LLM that underlies ChatGPT (GPT-3.5) has "limited knowledge of events after 2021."

However, what could happen is that the AI company could store our input and the model response to help train a future model. Reputable companies tell you if they will do this. Bard and ChatGPT, for example, are clear that they might use your input to improve future versions. OpenAI has made it clear that this does not apply to API calls (calls to the model directly through software). Right now (August, 2023) Google is still deciding how to handle API calls.

A final misunderstanding about the model is that it is simply repeating something it found in its input dataset. Because the model uses similarity patterns of tokens, the only way the model can learn something is if it sees a phrase repeatedly. It's very likely that the model was shown "peanut butter and jelly" repeatedly and the parameters of the model encoded that these words go together. LLMs are approximately as unlikely to repeat something from their training dataset as you are to have read something and then repeat it.

It is also important to recognize that LLMs have no memory. This may surprise chat users because the model can hold a conversation, which implies some form of memory. But it's a "trick." Behind the scenes, this is how chat works:

- You say something "A" to the LLM.
- The LLM responds with "B."

rick@talentalignment.net www.talentalignment.net

- Then you say something back "C" but the interface actually sends a prompt that says "User said A, then LLM replied B, now user says C."
- When the LLM responds "D" and you respond back "E," the prompt is: "User said A, then LLM replied B, then user said C, then LLM replied D, now user says E."
- And so forth.

This is how the chat "remembers" from one moment to the next. The size of this history that the model will consider is called the **context window**. ChatGPT has a context window of 4096 tokens (about 3,072 words). You may have run into this limitation if you asked ChatGPT to generate lengthy outputs.

Finally, LLMs do not "think" in the same way humans (and other animals) think. The fact that they exhibit seeming intelligence simply by generating output associated with the input is a surprise to many AI researchers. Some researchers believe that as we make models bigger, they will get more intelligent. That's one way to look at the recent OpenAI models; as they got bigger, they seemed much more intelligent. But another view is that until LLMs can store and retrieve knowledge, they cannot be reasoning. It's also true that models have probably hit the current limit on size. New models will grow but more gradually.

Why do models hallucinate?

Hallucination is a term for when the response from an LLM is not judged factually correct by the human using the model. For example, we could ask the LLM: "who won the 1936 world series?" and the response from GPT-3 with probabilities is: "The (99.97%) New(93.3%) York(100%) Yankees(99.94%) won(97.29%) the(100%) 1936(99.86%) World(99.99%) Series(100%).(76.71%)" Think of that like this:

Prompt	**Response**	**Probability**
who won the 1936 world series?	The	0.9997
who won the 1936 world series? The	New	0.9330
who won the 1936 world series? The New	York	1.000
who won the 1936 world series? The New York	Yankees	0.9994
who won the 1936 world series? The New York Yankees	won	0.9724
who won the 1936 world series? The New York Yankees won	the	1.0000
who won the 1936 world series? The New York Yankees won the	1936	0.9986
who won the 1936 world series? The New York Yankees won the 1936	World	0.9999
who won the 1936 world series? The New York Yankees won the 1936 World	Series	1.0000
who won the 1936 world series? The New York Yankees won the 1936 World Series	.	0.7671

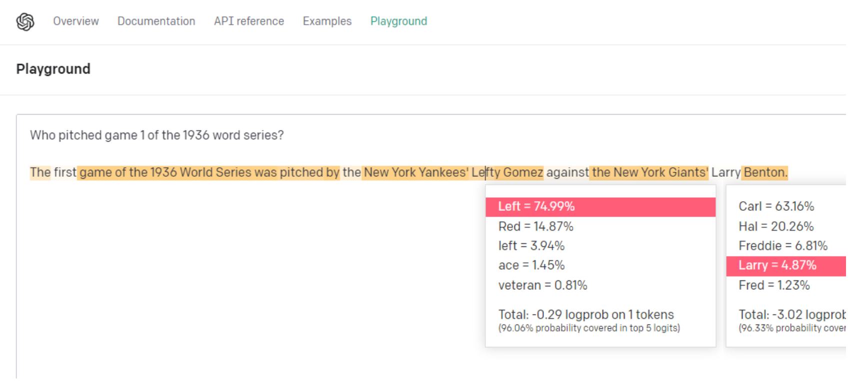
Those are high probabilities! But notice that “New” is only generated 93.3% of the time. The word that follows, “York” has a near perfect chance because of the strong association between “New” and “York” and also because the model had encoded that 1936 world series is associated with New York.

As you would expect from these high probabilities, regenerating this answer produced exactly the same response several times in a row. Then “New” was <u>not</u> generated, and the model instead responded: “The 1936 World Series was won by the New York Yankees, who defeated the New York Giants in six games.” (According to Wikipedia, this is correct.) This is an example of how the model can produce factual responses because there are strong patterns between the prompt and response.

Let’s ask a more difficult question: “Who pitched game 1 of the 1936 word series?” The answer is ambiguous because each team had a pitcher; Carl Hubbell was the (winning) Giants pitcher and Red Ruffing was the (losing) Yankees pitcher. GPT-3 responds: “The first game of the 1936 World Series was pitched by the New York Yankees' Lefty Gomez against the New York Giants' Larry Benton.”

This is not correct. The model has “hallucinated.” Let’s look at the probabilities to see what’s going on:
“The(46.37%) first(11.80%) game(99.32%) of(99.74%) the(99.99%) 1936(99.94%) World(99.98%) Series(99.99%) was(99.15%) pitched(84.27%) by(99.75%) the(32.62%) New(96.28%) York(100.00%) Yankees(98.34%)'(96.75%) Left(74.99%)y(100.00%) Gomez(99.97%) against(17.04%) the(98.72%) New(94.74%) York(100.00%) Giants(99.79%)'(99.37%) Larry(4.87%) Benton(98.32%).(99.75%)”

Notice that parts of the answer have very high probabilities, like “game of the 1936 World Series was” and some critical parts had comparatively modest probabilities: “Lefty” and “Larry” had about 75% and 5% probabilities and generating those tokens is where the model deviated from actual history. The second-highest token after “Left” was “Red” which would have generated the correct answer for the Yankees (Red Ruffing) and the highest probability for the Giants was “Carl” which would have generated the correct answer for the Giants. The model’s probabilities are just wrong for the Yankees’ pitcher, but the choice of Larry was just bad luck.



Who pitched game 1 of the 1936 word series?

The first game of the 1936 World Series was pitched by the New York Yankees' Lefty Gomez against the New York Giants' Larry Benton.

Left = 74.99%	Carl = 63.16%
Red = 14.87%	Hal = 20.26%
left = 3.94%	Freddie = 6.81%
ace = 1.45%	Larry = 4.87%
veteran = 0.81%	Fred = 1.23%
Total: -0.29 logprob on 1 tokens (96.06% probability covered in top 5 logits)	Total: -3.02 logprob (96.33% probability cover

Notice too that there are follow-up words that have very high probabilities. For example, “Lefty” was actually generated from the token “Left” (with modest probability 74.99%) and “-y” that had probability of about 100%. The pitchers’ last

rick@talentalignment.net www.talentalignment.net

names “Gomez” and “Benton” also had near 100% probability. So, while the calculated probability for “Left” and “Larry” were fairly low, having generated these names, the model knew how to follow those (exactly like the model is ~100% likely to follow “New” with “York”).

And that’s why these models “hallucinate.” The model selected tokens at random according to the probabilities that it had calculated. In this example, the model only has a 14.87% chance of being correct about the Yankees’ pitcher, and a 63.16% chance of being correct about the Giants’ pitcher.

These models only have one way of operating: they build a response that is associated with the input, one token at a time. Sometimes this produces the correct answer (“The New York Yankees won the 1936 World Series.” And: “The 1936 World Series was won by the New York Yankees, who defeated the New York Giants in six games.” But sometimes the response built in this way is wrong: “The first game of the 1936 World Series was pitched by the New York Yankees' Lefty Gomez against the New York Giants' Larry Benton.”

“Follow” our LinkedIn page for the rest of our three-part series featuring Alan:

“Artificial Intelligence 101 – How it Works” (Part I)
“Six Rules of Engagement for AI Users” (Part III)

Check out our thoughts on advancing strategy through Culture Change while you’re there!

rick@talentalignment.net www.talentalignment.net