Is your ChatBot actually using your data?

29 October 2023 · ai, automation, critical thinking

In 316 AD Emperor Constantine issued a new coin, (https://www.britishmuseum.org/collection/object/C_1863-0713-1) there's nothing too unique about that in itself. But this coin is significant due to its pagan/roman religious symbols. Why is this odd? Constantine had converted himself, and probably with little consultation - his empire to Christianity, years before. Yet the coin shows the emperor and the (pagan) sun god Sol.

Looks Legit!

While this seems out of place, to us (1700 years later), it's not entirely surprising. Constantine and his people had followed different, older gods for centuries. The people would have been raised and taught the old pagan stories, and when presented with a new narrative it's not surprising they borrowed from and felt comfortable with both.

I've seen much the same behaviour with Large Language Models (LLMs) like ChatGPT. You can provide them with fresh new data, from your own documents, but what's to stop it from listening to its old training instead?

You could spend a lot of time collating, parsing, embedding and storing your data and what does the LLM do? It, at best - might ignore your data, at worst - state the data is faulty and offer its own opinion! Thanks - but they are called Large LANGUAGE Models and not a Large KNOWLEDGE Models for a reason.

Luckily, prompt engineering can really help here. You can usually be strict enough in the prompt to ensure that the LLM does not openly question the validity data that you are asking questions about. After all I do not want its opinion, its there to translate the results of my semantic search into a easy read response - not espouse its own views!

But, how can I be sure its doing this? What if generally it seems to be summarising my data but isn't? This is an especially important point if:

- Some of the data or similar data were available in the public domain (e.g. company financials or news reports)
- You are using a black box LLM from a cloud provider (Like Claude or ChatGPT) where you don't know for sure what data was used to train it

It could be providing a response that looks like yours but might be subtly different. "Oh it looks like my data!". Maybe it does, but what if your user is referring to a specific fact or paragraph from a document made years before? is your LLM really using your data or providing you with a similar but technically or legally wrong answer?

The answer is to use poison. That is, some test data that is parsed, embedded and indexed along with your other data - but is easy to spot/assert in the ChatBot's response. For example, I created a simple Q&A bot (https://github.com/phoughton/What_the_dickens_RAG/tree/main) that has indexed the entire works of Charles Dickens (https://en.wikipedia.org/wiki/Charles_Dickens). This bot uses Chroma as the vector database and GPT4 as the LLM.

While the OpenAI GPT4 API provides great descriptions of the data found in the vector DB, it was prone to ad-libbing. Some fairly draconian prompt engineering managed to mute this though and it could then reliably answer correctly. BUT - as stated above, how could I be sure it wasn't just basing its answers of its own training?

The answer was to insert some text of my own, the poisoned text, into the vector database. This is treated as a normal paragraph in the source data for your database during testing. Then when you query the bot, it in-turn queries your vector database and bases its answers on those results.

If the poison text is treated as gospel by the bot and just forms the basis of the response as would any other data, then you can be fairly sure the bot is using your data. The choice of the poison data is also important, it should be something not actually true or appropriate for that wider dataset.

Why? The reason is that we don't want text to appear accidentally, as a response from the LLM. We want something clearly not likely to be predicted by the LLM to be treated as the right answer. That way if the LLM is stating something blatantly absurd, but correct for your poisoned dataset you have confidence it was used in the answer.

Also, if the LLM returns something like “The document states X but this is incorrect” we know it is ad-libbing, and adding its own views!

So for my Charles Dickens bot the result was:

Hello, I’m an AI with access to the works of Charles Dickens. I can answer questions about his work: **Did Scrooge use a smart phone?**

Yes, according to the extract provided, Scrooge did use a smartphone. In section 65, it is mentioned that “Scrooge decided he needed to use his iPhone and ask the internet for a solution to this rather confusing state of affairs.”

That’s my poisoned data all right! Even if checking the meaning requires more work, it’s trivial to see my test data was used to build the answer.

This sort of check can form part of an automated test suite in your apps build/release pipeline as well as supply a useful dataset for more exploratory testing of the responses etc.