The Needle In a Haystack Test

Evaluating the performance of RAG systems

By Aparna Dhinakaran

Feb 16, 2024 04:37 PM · 7 min. read · View original

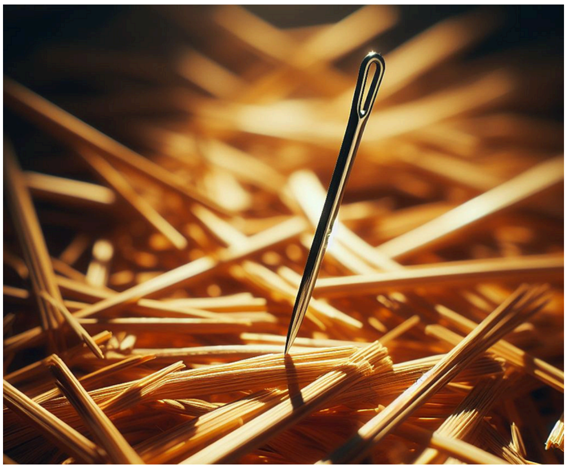

Image created by author using Dall-E 3

My thanks to Greg Kamradt and Evan Jolley for their contributions to this piece

Retrieval-augmented generation (RAG) underpins many of the LLM applications in the real world today, from companies generating headlines to solo developers solving problems for small businesses.

RAG evaluation, therefore, has become a critical part in the development and deployment of these systems. One new innovative approach to this challenge is the "Needle in a Haystack'' test, first outlined by Greg Kamradt in this X post and discussed in detail on his YouTube here. This test is designed to evaluate the performance of RAG systems across different sizes of context. It works by embedding specific, targeted information (the "needle") within a larger, more complex body of text (the "haystack"). The goal is to assess an LLM's ability to identify and utilize this specific piece of information amidst a vast amount of data.

Often in RAG systems, the context window is absolutely overflowing with information. Large pieces of context returned from a vector database are cluttered together with instructions for the language model, templating, and anything else that might exist in the prompt. The Needle in a Haystack evaluation tests the capabilities of an LLM to pinpoint specifics in amongst this mess. Your RAG system might do a stellar job of retrieving the most relevant context, but what use is this if the granular specifics within are overlooked?

We ran this test multiple times across several major language models. Let's take a closer look at the process and overall results.

Takeaways

The Needle in a Haystack test was first used to evaluate the recall of two popular LLMs, OpenAI's ChatGPT-4 and Anthropic's Claude 2.1. An out of place statement, "The best thing to do in San Francisco is eat a sandwich and sit in Dolores Park on a sunny day," was placed at varying depths within snippets of varying lengths taken from essays by Paul Graham, similar to this:

Figure 1: About 120 tokens and 50% depth | Image by Greg Kamradt on X, used here with author's permission

The models were then prompted to answer what the best thing to do in San Francisco was, only using the provided context. This was then repeated for different depths between 0% (top of document) and 100% (bottom of document) and different context lengths between 1K tokens and the token limit of each model (128k for GPT-4 and 200k for Claude 2.1). The below graphs document the performance of these two models:

Figure 2: ChatGPT-4's performance | Image by Greg Kamradt on X, used here with author's permission

As you can see, ChatGPT's performance begins to decline at <64k tokens and sharply falls at 100k and over. Interestingly, if the "needle" is positioned towards the beginning of the context, the model tends to overlook or "forget" it — whereas if it's placed towards the end or as the very first sentence, the model's performance remains solid.

Figure 3: Claude 2.1's performance | | Image by Greg Kamradt on X, used here with author's permission

For Claude, initial testing did not go as smoothly, finishing with an overall score of 27% retrieval accuracy. A similar phenomenon was observed with performance declining as context length increased, performance generally increasing as the needle was hidden closer to the bottom of the document, and 100% accuracy retrieval if the needle was the first sentence of the context.

Anthropic's Response

In response to these findings, Anthropic published an article detailing their re-run of this test with a few key changes.

First, they changed the needle to more closely mirror the topic of the haystack. Claude 2.1 was trained to "not [answer] a question based on a document if it doesn't contain enough information to justify that answer." Thus, Claude may well have correctly identified eating a sandwich in Dolores Park as the best thing to do in San Francisco. However, along with an essay about doing great work, this small piece of information may have appeared unsubstantiated. This could have led to a verbose response explaining that Claude cannot confirm that eating a sandwich is the best thing to do in San Francisco or an omission of the detail entirely. When re-running the experiments, researchers at Anthropic found that changing the needle to a small detail originally mentioned in the essay led to significantly increased outcomes.

Second, a small edit was made to the prompt template used to query the model. A single line — *here is the most relevant sentence in the context* — was added to the end of the template, directing the model to simply return the most relevant sentence provided in the context. Similar to the first, this change allows us to circumvent the model's propensity to avoid unsubstantiated claims by directing it to simply return a sentence rather than make an assertion.

```
PROMPT = """

HUMAN: <context>
{context}
</context>

What is the most fun thing to do in San Francisco based on the context? Don't give information outside the document or
repeat our findings

Assistant: here is the most relevant sentence in the context:"""
```

These changes led to a significant jump in Claude's overall retrieval accuracy: from 27% to 98%! Finding this initial research fascinating, we decided to run our own set of experiments using the Needle in a Haystack test.

Further Experiments

In conducting a new series of tests, we implemented several modifications to the original experiments. The needle we used was a random number that changed each iteration, eliminating the possibility of caching. Additionally, we used our open source Phoenix evals library (full disclosure: I lead the team that built Phoenix) to reduce the testing time and use rails to search directly for the random number in the output, cutting through wordiness that would decrease a retrieval score. Finally, we considered the negative case where the system fails to retrieve the results, marking it as unanswerable. We ran a separate test for this negative case to assess how well the system recognizes when it can't retrieve the data. These modifications allowed us to conduct a more rigorous and comprehensive evaluation.

The updated tests were run across several different configurations using four different large language models: ChatGPT-4, Claude 2.1 (with and without the aforementioned change to the prompt that Anthropic suggested), and Mistral AI's Mixtral-8X7B-v0.1 and 7B Instruct. Given that small nuances in prompting can lead to vastly different results across models, we used several prompt templates in the attempt to compare these models performing at their best. The simple template we used for ChatGPT and Mixtral was as follows:

```
SIMPLE_TEMPLATE = '''
   You are a helpful AI bot that answers questions for a user. Keep your responses short and direct.
   The following is a set of context and a question that will relate to the context.
   #CONTEXT
   {context}
   #ENDCONTEXT

   #QUESTION
   {question} Don't give information outside the document or repeat your findings. If the information is not available in
the context respond UNANSWERABLE
```

For Claude, we tested both previously discussed templates.

```
ANTHROPIC_TEMPLATE_ORIGINAL = ''' Human: You are a close-reading bot with a great memory who answers questions for users. I'm going to give you the text of some essays. Amidst
the essays ("the haystack") I've inserted a sentence ("the needle") that contains an answer to the user's question.
Here's the question:
   <question>{question}</question>
   Here's the text of the essays. The answer appears in it somewhere.
   <haystack>
   {context}
   </haystack>
   Now that you've read the context, please answer the user's question, repeated one more time for reference:
   <question>{question}</question>

   To do so, first find the sentence from the haystack that contains the answer (there is such a sentence, I promise!) and
put it inside <most_relevant_sentence> XML tags. Then, put your answer in <answer> tags. Base your answer strictly on the
context, without reference to outside information. Thank you.
   If you can't find the answer return the single word UNANSWERABLE
   Assistant: '''
```

```
ANTHROPIC_TEMPLATE_REV2 = ''' Human: You are a close-reading bot with a great memory who answers questions for users. I'm going to give you the text of some essays. Amidst the
essays ("the haystack") I've inserted a sentence ("the needle") that contains an answer to the user's question.
Here's the question:
   <question>{question}</question>
   Here's the text of the essays. The answer appears in it somewhere.
   <haystack>
   {context}
   </haystack>
   Now that you've read the context, please answer the user's question, repeated one more time for reference:
   <question>{question}</question>

   To do so, first find the sentence from the haystack that contains the answer (there is such a sentence, I promise!) and
put it inside <most_relevant_sentence> XML tags. Then, put your answer in <answer> tags. Base your answer strictly on the
context, without reference to outside information. Thank you.
   If you can't find the answer return the single word UNANSWERABLE
   Assistant: Here is the most relevant sentence in the context:'''
```

All code run to complete these tests can be found in this GitHub repository.

Results

Figure 7: Comparison of GPT-4 results between the initial research (Run #1) and our testing (Run #2) | Image by author

Figure 8: Comparison of Claude 2.1 (without prompting guidance) results between Run #1 and Run #2 | Image by author

Our results for ChatGPT and Claude (without prompting guidance) did not stray far from Mr. Kamradt's findings, and the generated graphs appear relatively similar: the upper right (long context, needle near the beginning of the context) is where LLM information retrieval sufferers.

Figure 9: Comparison of Claude 2.1 results with and without prompting guidance

Although we were not able to replicate Anthropic's results of 98% retrieval accuracy for Claude 2.1 with prompting guidance, we did see a significant decrease in total misses when the prompt was updated (from 165 to 74). This jump was achieved by simply adding a 10 word instruction to the end of the existing prompt, highlighting that small differences in prompts can have drastically different outcomes for LLMs.

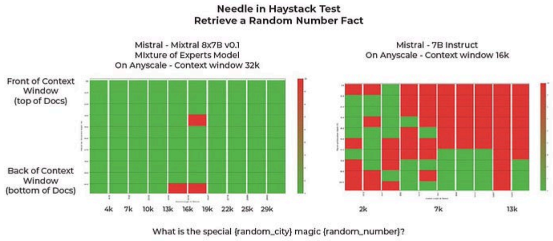

Figure 10: Mixtral results | Image by author

Last but certainly not least, it is interesting to see just how well Mixtral performed at this task despite these being by far the smallest models tested. The Mixture of Experts (MOEs) model was far better than 7B-Instruct, and we are finding that MOEs do much better for retrieval evaluations.

Conclusion

The Needle in a Haystack test is a clever way to quantify an LLM's ability to parse context to find needed information. Our research concluded with a few main takeaways. First, ChatGPT-4 is the industry's current leader in this arena along with many other evaluations that we and others have carried out. Second, at first Claude 2.1 seemed to underperform this test, but with tweaks to the prompt structure the model showed significant improvement. Claude is a bit wordier than some other models, and taking extra care to direct it can go a long way in terms of results. Finally, Mixtral MOE greatly outperformed our expectations, and we are excited to see Mixtral models continually overperform expectations.