MOODY'S

# A Practical Guide to Commercial RAG

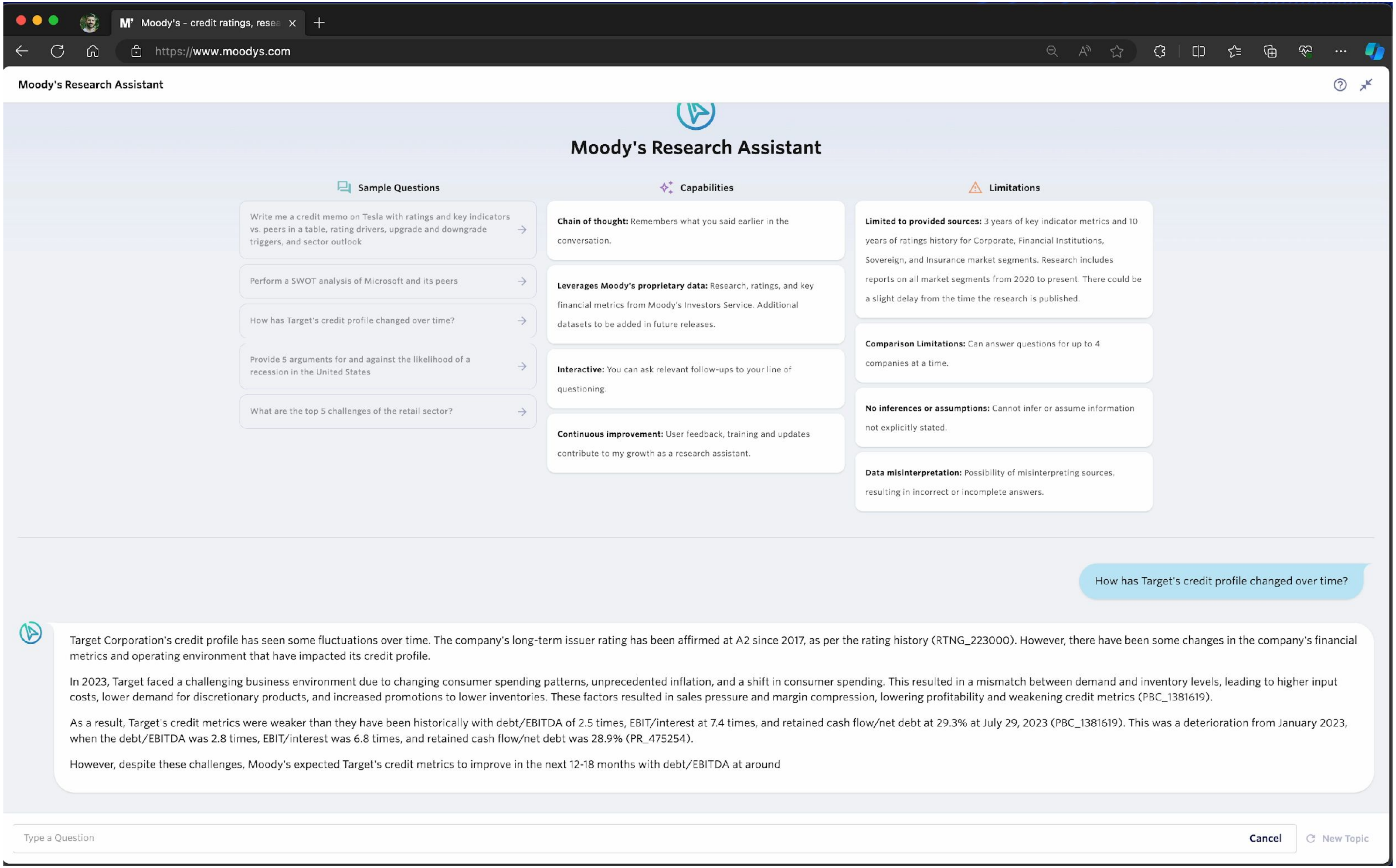
## Haystack 2024

Jeff Capobianco

Product Manager of Search for Moodys.com

# A Practical Guide to Commercial RAG

# Story Time: The Genesis of Moody's Research Assistant


Moody's - credit ratings, resea
https://www.moodys.com
Moody's Research Assistant
Moody's Research Assistant
Sample Questions
Write me a credit memo on Tesla with ratings and key indicators vs. peers in a table, rating drivers, upgrade and downgrade triggers, and sector outlook
Perform a SWOT analysis of Microsoft and its peers
How has Target's credit profile changed over time?
Provide 5 arguments for and against the likelihood of a recession in the United States
What are the top 5 challenges of the retail sector?
Capabilities
Chain of thought: Remembers what you said earlier in the conversation.
Leverages Moody's proprietary data: Research, ratings, and key financial metrics from Moody's Investors Service. Additional datasets to be added in future releases.
Interactive: You can ask relevant follow-ups to your line of questioning.
Continuous improvement: User feedback, training and updates contribute to my growth as a research assistant.
Limitations
Limited to provided sources: 3 years of key indicator metrics and 10 years of ratings history for Corporate, Financial Institutions, Sovereign, and Insurance market segments. Research includes reports on all market segments from 2020 to present. There could be a slight delay from the time the research is published.
Comparison Limitations: Can answer questions for up to 4 companies at a time.
No inferences or assumptions: Cannot infer or assume information not explicitly stated.
Data misinterpretation: Possibility of misinterpreting sources, resulting in incorrect or incomplete answers.
How has Target's credit profile changed over time?
Target Corporation's credit profile has seen some fluctuations over time. The company's long-term issuer rating has been affirmed at A2 since 2017, as per the rating history (RTNG_223000). However, there have been some changes in the company's financial metrics and operating environment that have impacted its credit profile.
In 2023, Target faced a challenging business environment due to changing consumer spending patterns, unprecedented inflation, and a shift in consumer spending. This resulted in a mismatch between demand and inventory levels, leading to higher input costs, lower demand for discretionary products, and increased promotions to lower inventories. These factors resulted in sales pressure and margin compression, lowering profitability and weakening credit metrics (PBC_1381619).
As a result, Target's credit metrics were weaker than they have been historically with debt/EBITDA of 2.5 times, EBIT/interest at 7.4 times, and retained cash flow/net debt at 29.3% at July 29, 2023 (PBC_1381619). This was a deterioration from January 2023, when the debt/EBITDA was 2.8 times, EBIT/interest was 6.8 times, and retained cash flow/net debt was 28.9% (PR_475254).
However, despite these challenges, Moody's expected Target's credit metrics to improve in the next 12-18 months with debt/EBITDA at around
Type a Question
Cancel
New Topic

Some of these lessons are from quick wins.

Some were learned the hard way.

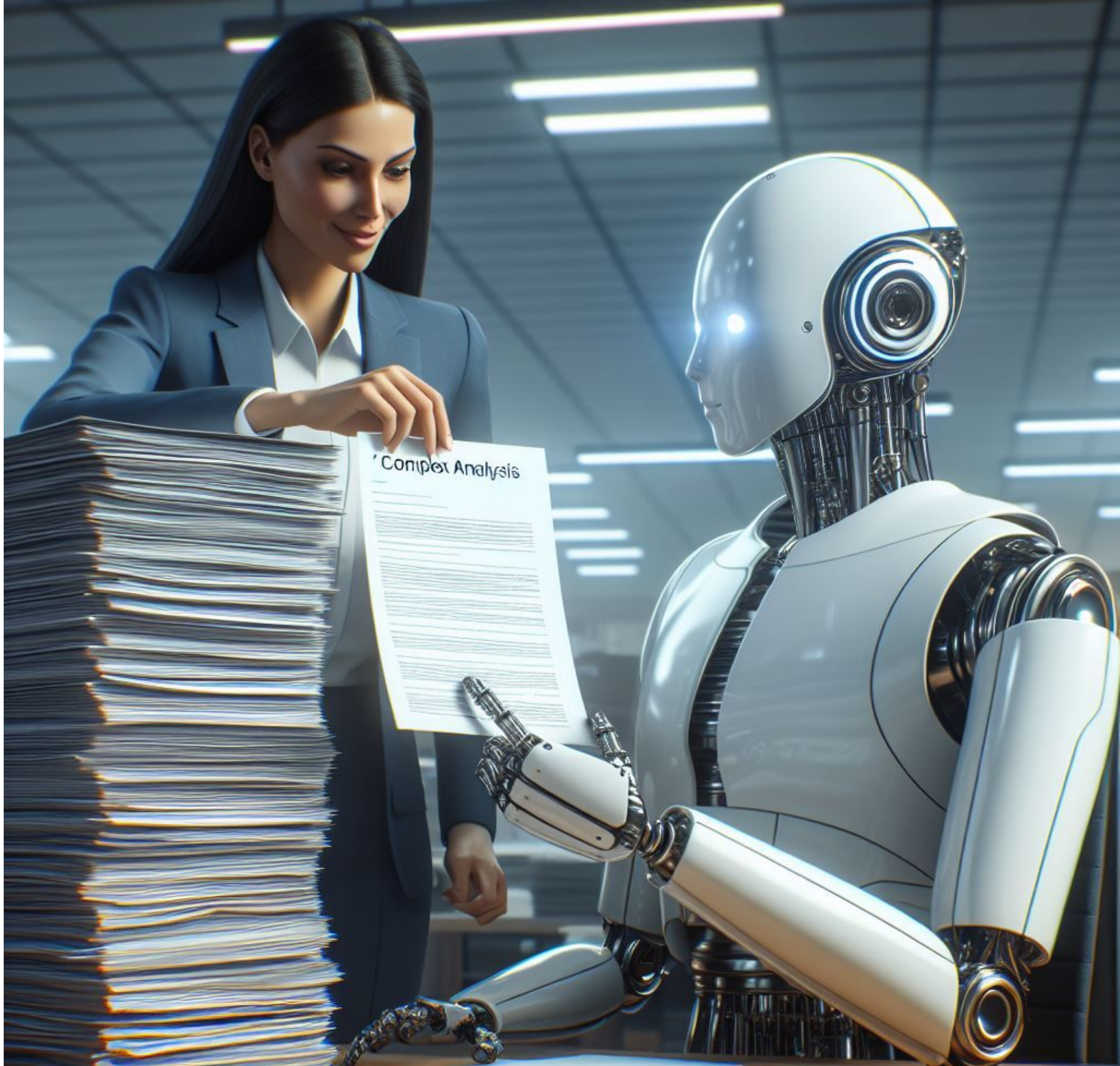
# Agenda

1. Why RAG?
2. Architecture level set
3. Areas to focus first
4. Where to explore next

# Why RAG?

- Accelerate research

- Tailored AI responses rooted in your data



## Search Engine

- Up to date information
- Data that sets your business apart
- Users must synthesize from multiple sources
- Challenging to understand user context

## Retrieval Augmented Generation

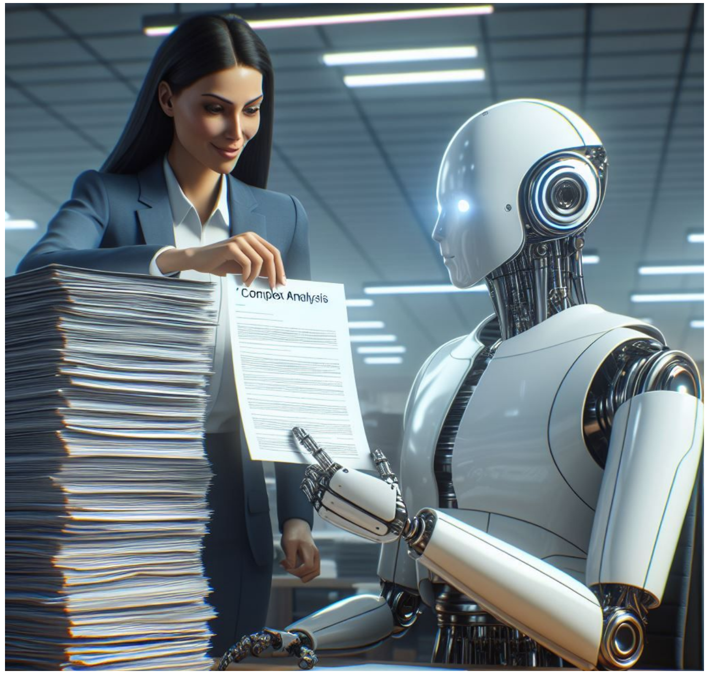
- **Challenging to measure many moving parts**
- **Synthesize responses from multiple sources**
- **Suggest follow up questions**

## Text Completion LLM

- Adding information through training is expensive
- Hallucinates
- Can answers very specific questions
- Highly context aware

# Is RAG right for your business?

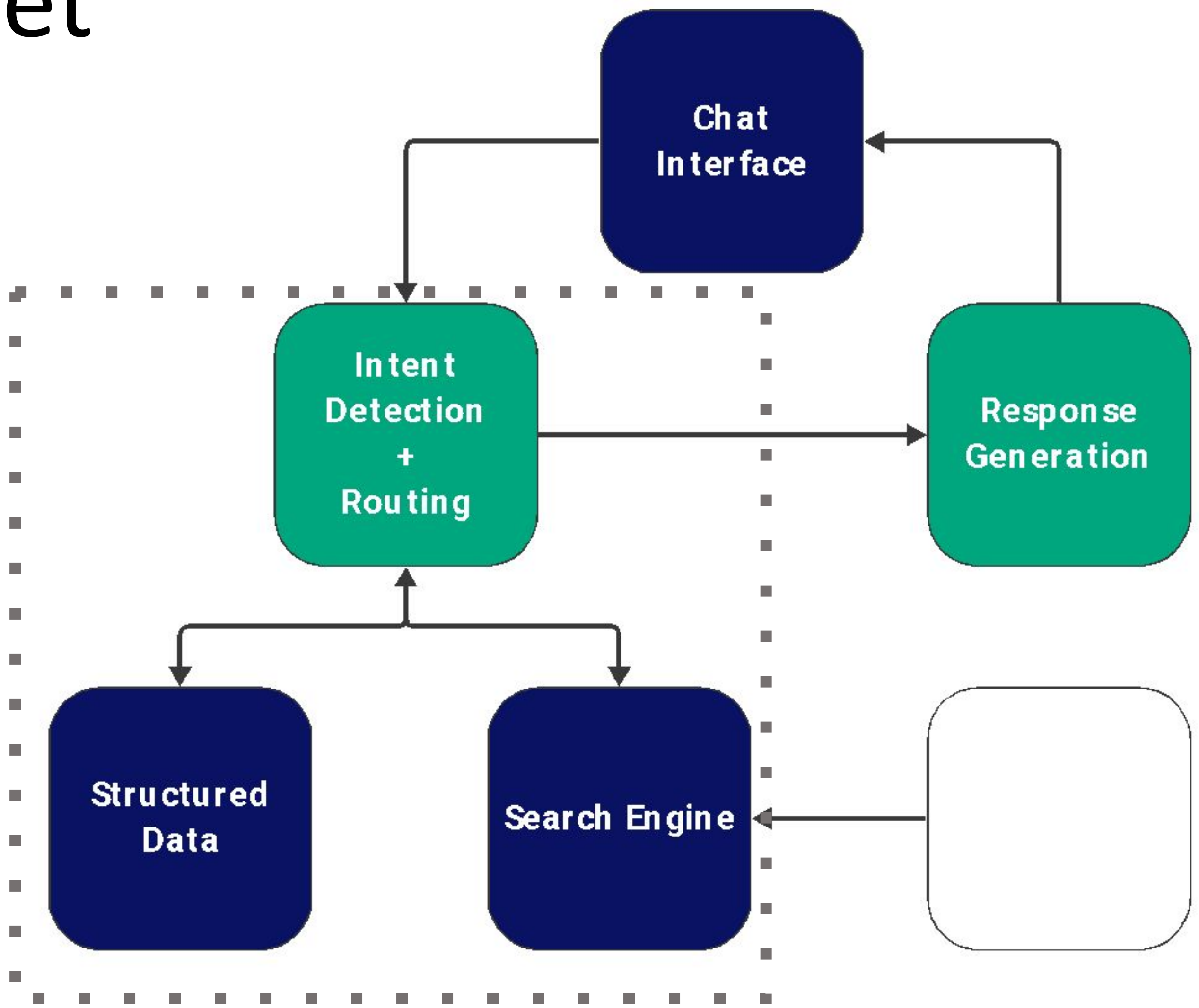
| | |
|---|---|
| ✅ | Do your users perform complex research and synthesis? |
| ✅ | Do you have a large corpus of information-dense documents? |
| ✅ | Is your data proprietary? |
| ✅ | Is your users’ time expensive? |



# Architecture Level Set

Retrieval Requirements:

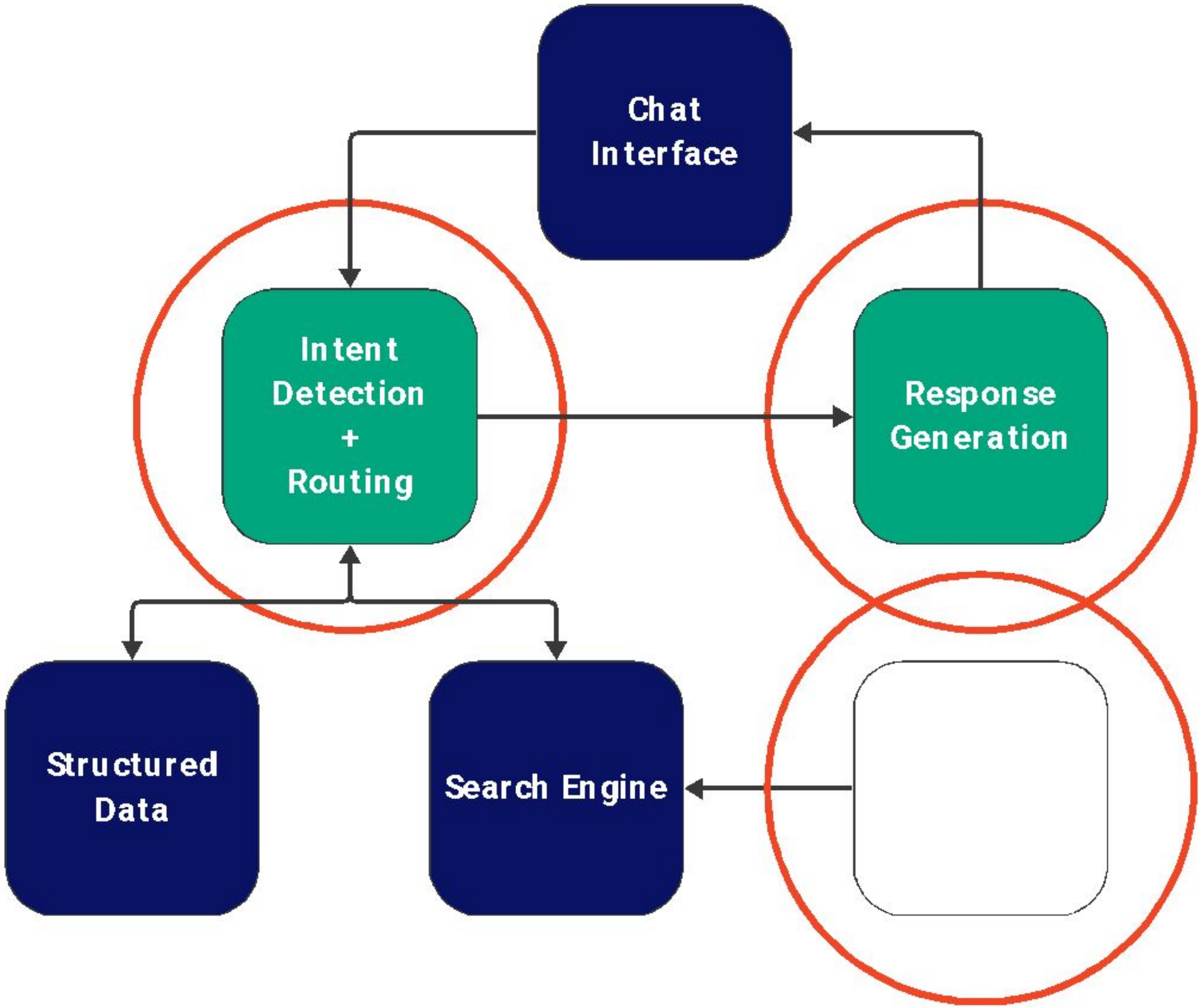
1. Handle long queries
2. Return relevant chunks that fit in the LLM context window



# Areas to Focus First

# Understand Costs

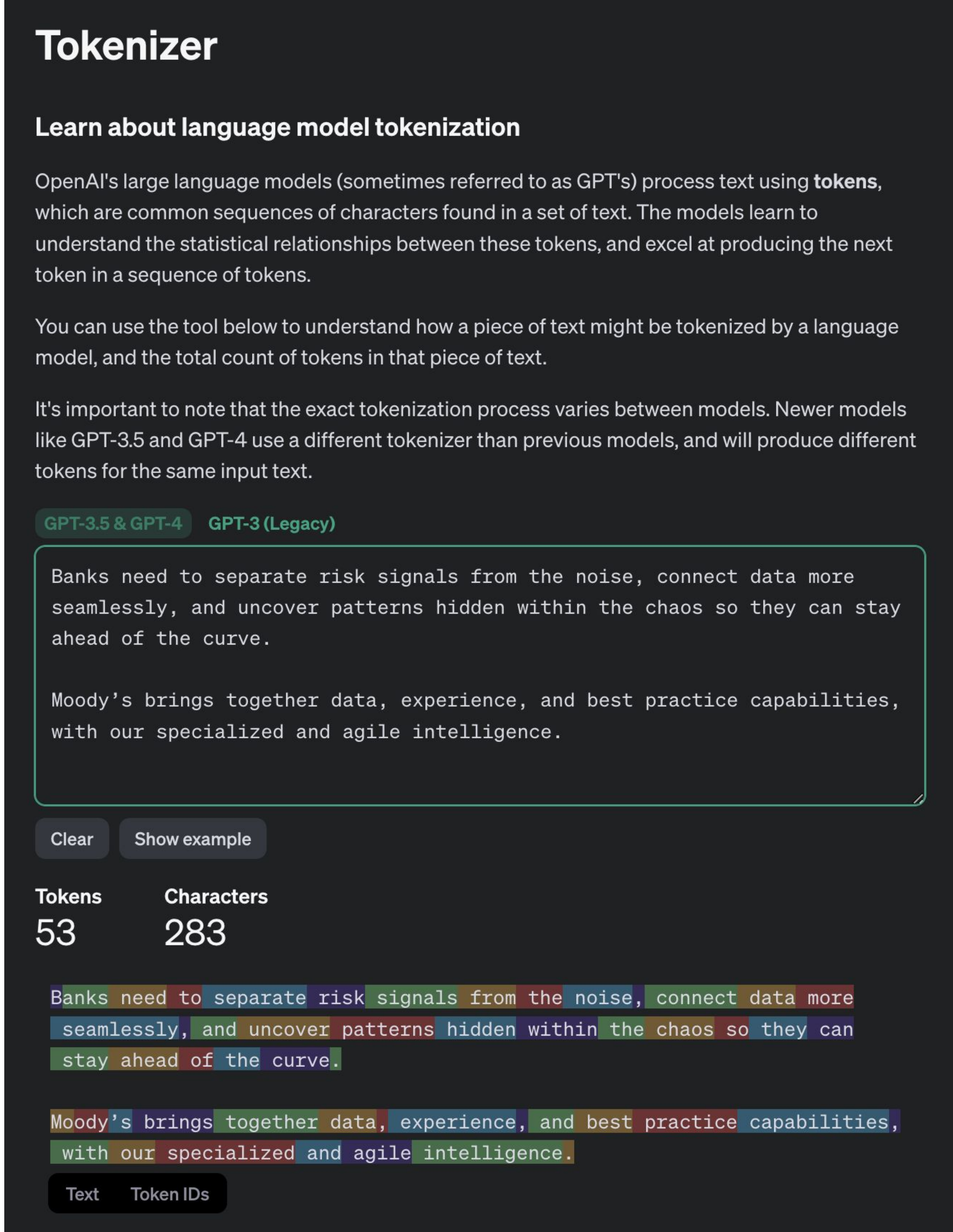
# Understand Costs



# Understand Costs

https://platform.openai.com/tokenizer

APIs like GPT and Claude charge per token



# Understand Costs

https://openai.com/pricing#language-models

| Model | Input | Output |
|---|---|---|
| gpt-4 | $0.03 / 1K tokens | $0.06 / 1K tokens |
| gpt-4-32k | $0.06 / 1K tokens | $0.12 / 1K tokens |

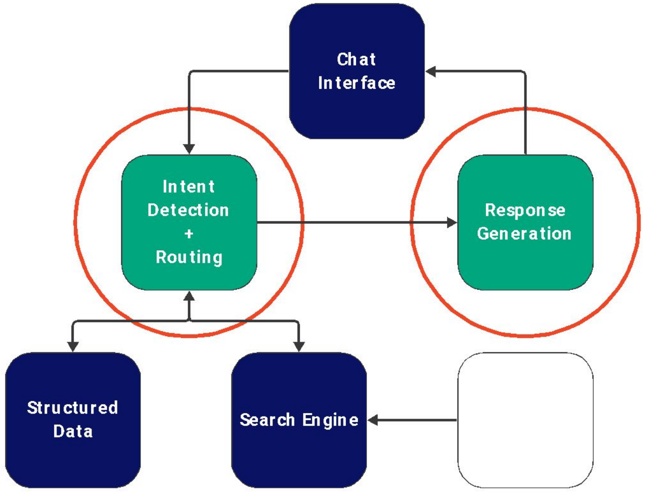
Max context length (input + output tokens) = 8,192

Using half of that for your final generation

(4k tokens * $0.03) + (0.3k tokens * $0.06) = $0.14

Plus a small intent detection prompt(s)

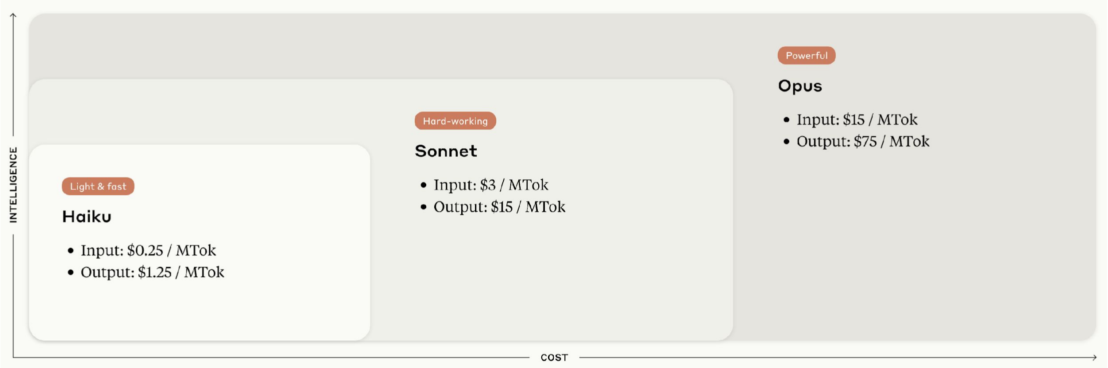
(0.3k tokens * $0.03) + (0.03k tokens * $0.06) = $0.01



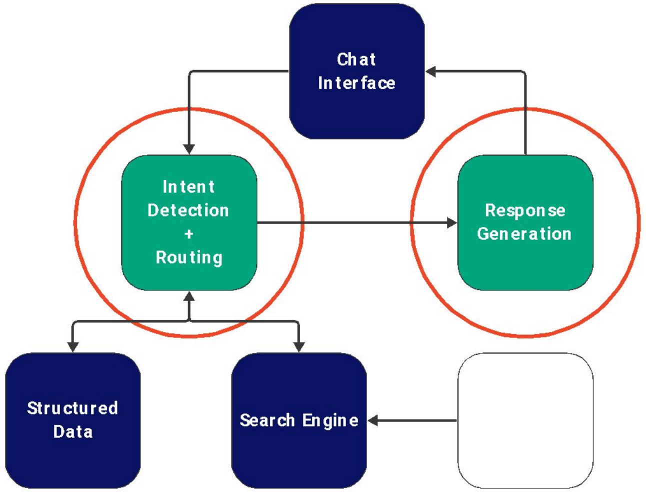

**$150 per 1000 requests**

# Understand Costs

https://www.anthropic.com/api




Max context length (input + output tokens) = **200,000**




Response generation

(4k tokens * $0.015) + (0.3k tokens * $0.075) = $0.08

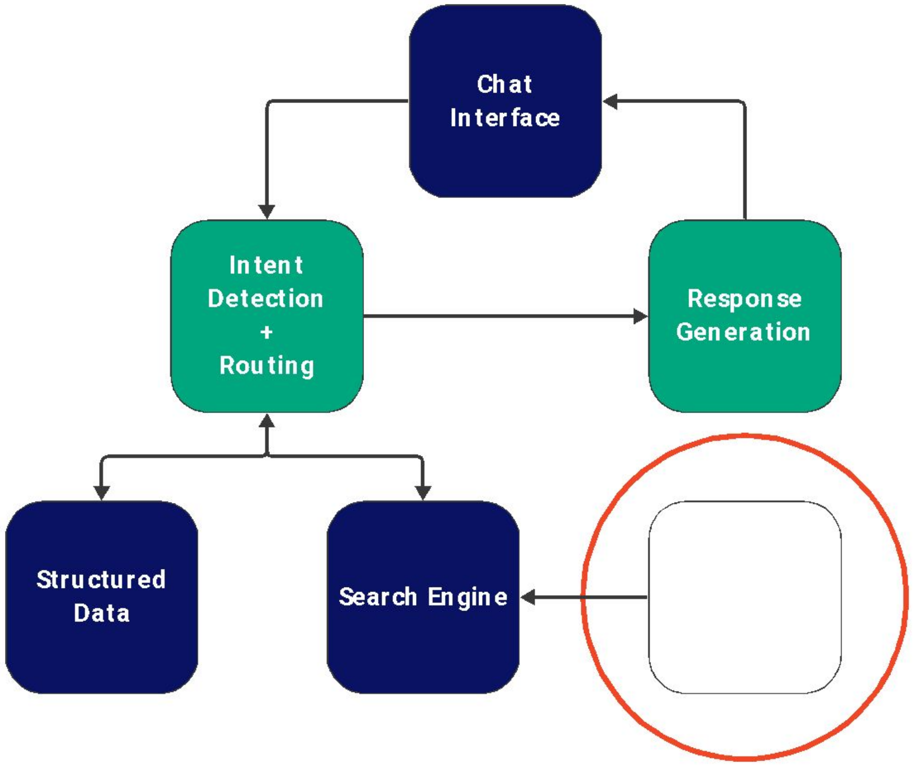
Intent detection

(0.3k tokens * $0.015) + (0.03k tokens * $0.075) = $0.007

**$87 per 1000 requests**

# Understand Costs – Embedding Inference

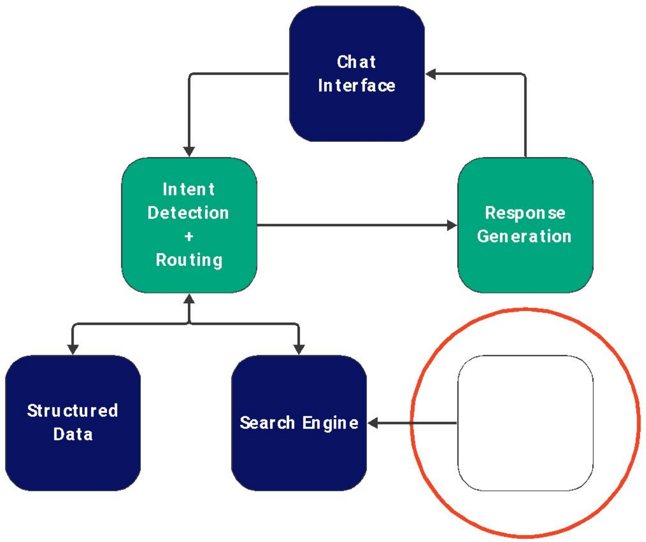
- Fast embedding requires GPUs
- Start with a small data set
- Updates to chunking or embedding model require re-indexing



# Understand Costs – Embedding Inference

https://openai.com/pricing https://cohere.com/pricing

| Model | Usage |
| --- | --- |
| text-embedding-3-small | $0.00002 / 1K tokens |
| text-embedding-3-large | $0.00013 / 1K tokens |
| ada v2 | $0.00010 / 1K tokens |




Chunk embedding

0.15k tokens * $0.0001 = $0.00015 per chunk

**$150 per 1,000,000 chunks**

Query Embedding

0.02k tokens * $0.0001 = $0.00002 per query

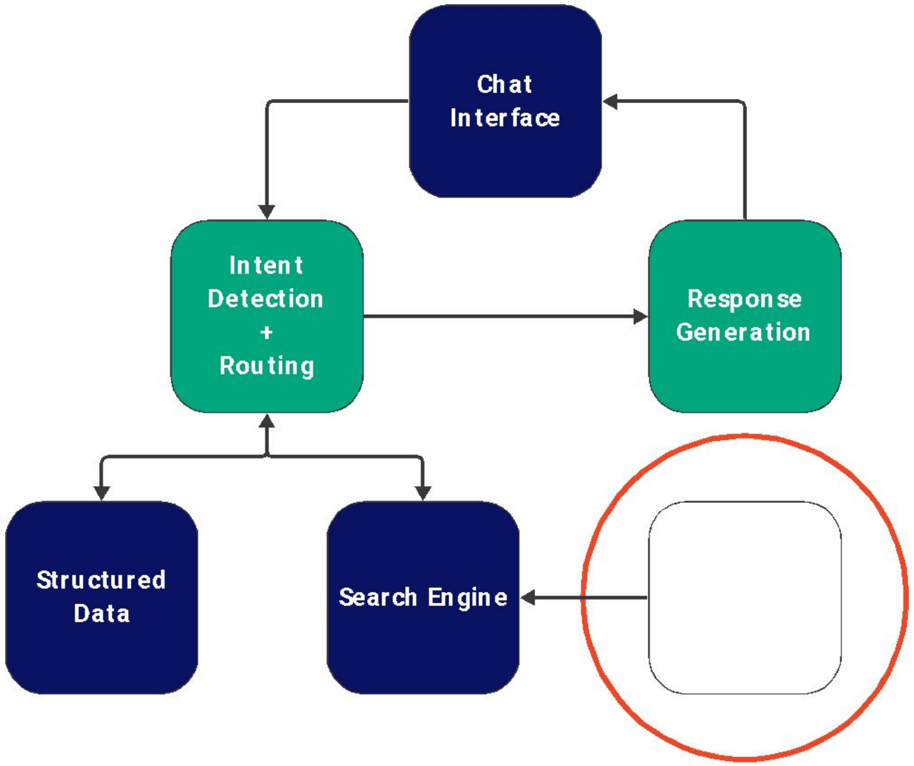
**$20 per 1,000,000 queries**

# Understand Costs – Embedding Inference

https://aws.amazon.com/sagemaker/pricing/

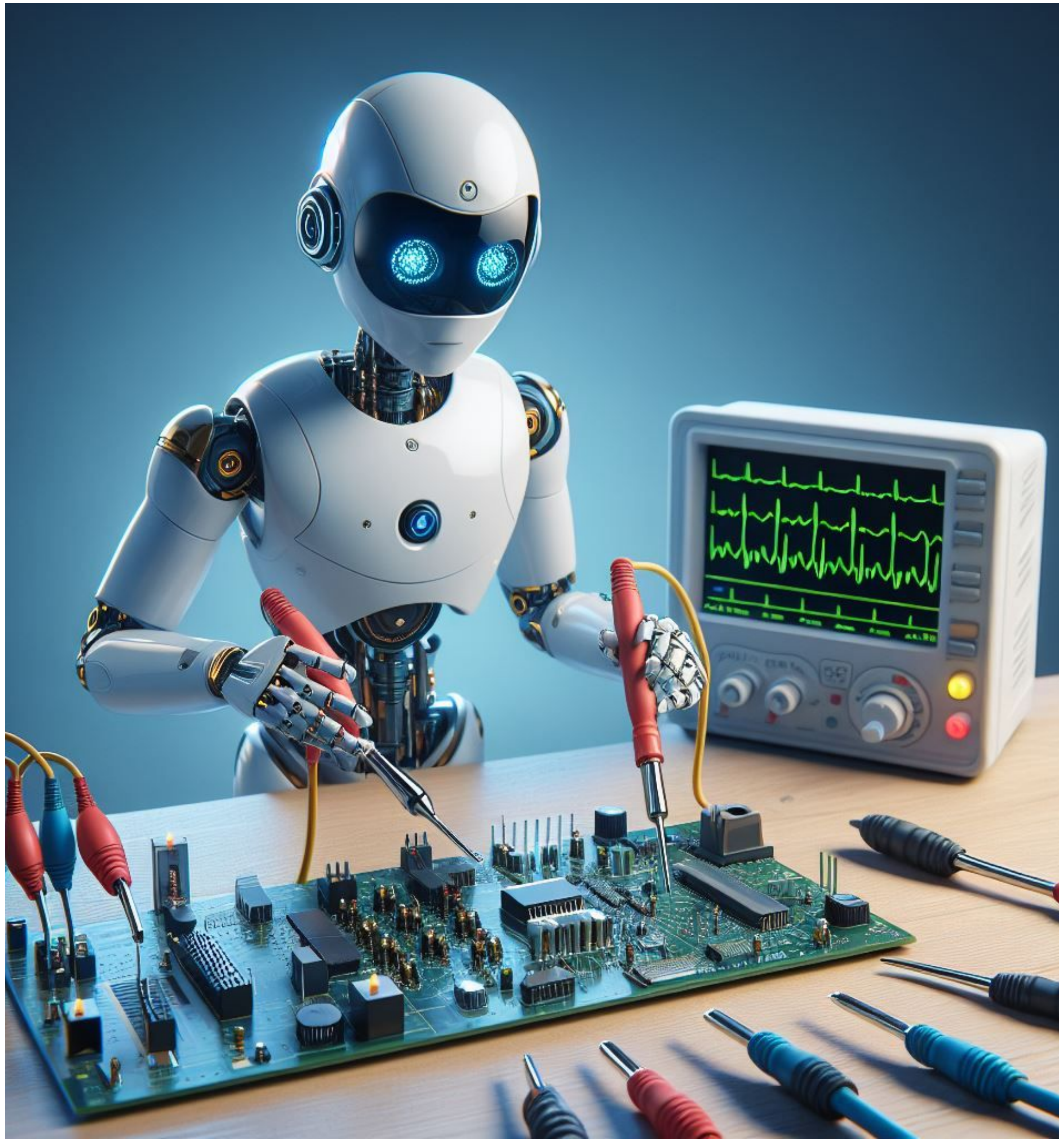
| Memory (MB) | Provisioned Concurrency Usage Price per second | Inference Duration Price per second |
|---|---|---|
| 3072 | $0.0000150 | $0.0000350 |
| 4096 | $0.0000200 | $0.0000467 |

- AWS SageMaker provides flexibility
- Wide price range of options for various scaling and latency needs



# Structure Your Code for Measurement

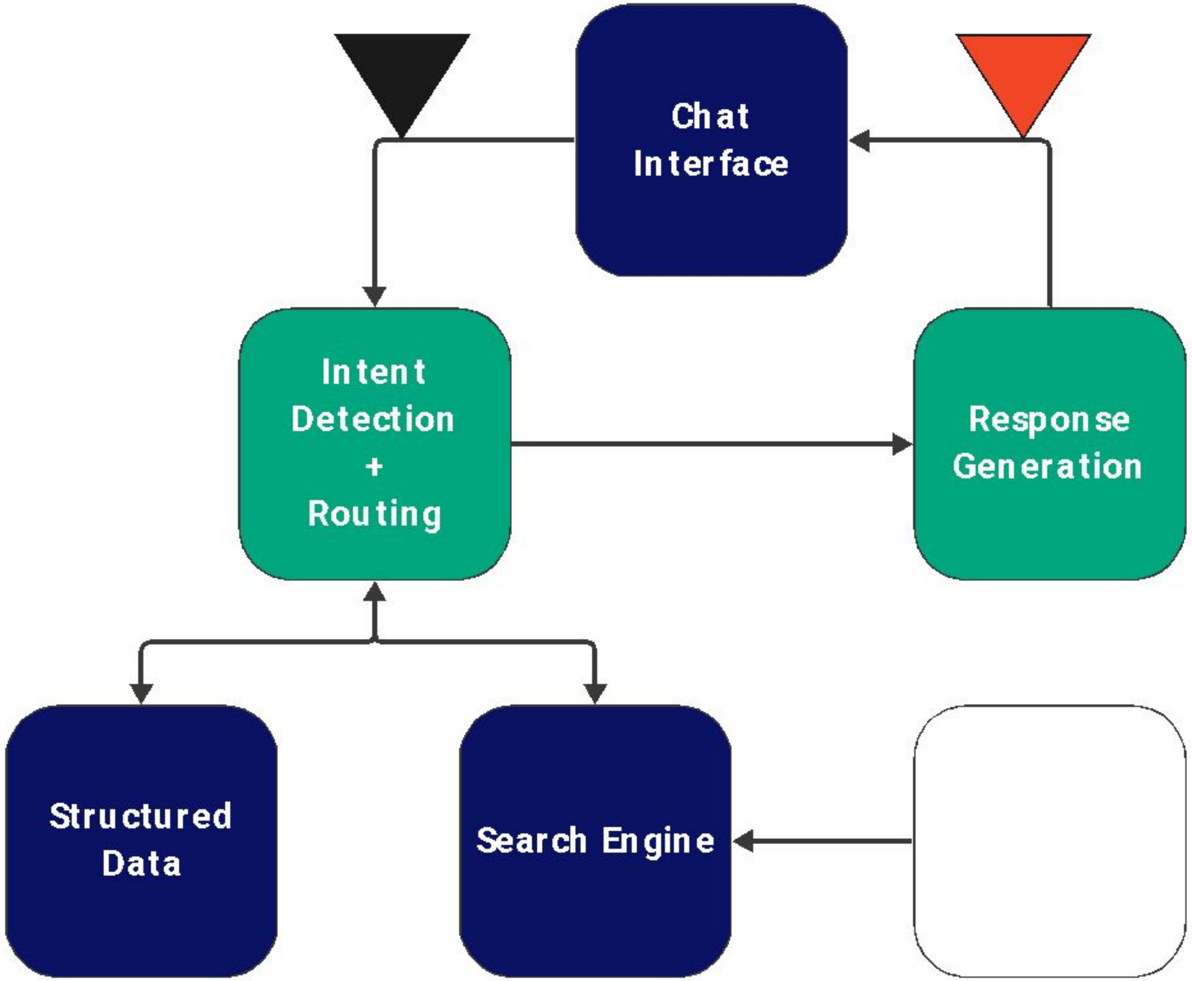
- Enable rapid experimentation
- LangSmith provides a framework
- Log inputs and outputs



# Structure Your Code for Measurement

## End to End

- Prompt – Response pairs

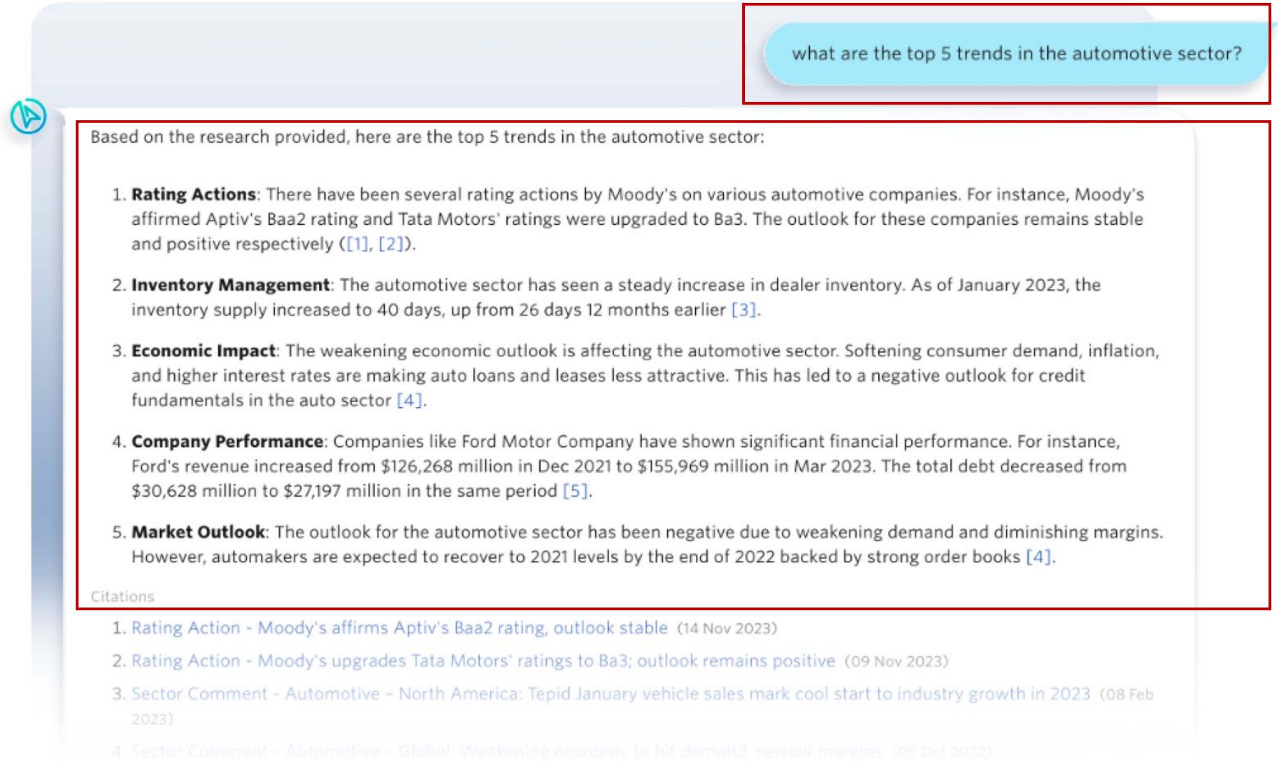


# Structure Your Code for Measurement

## End to End

- Primary objective

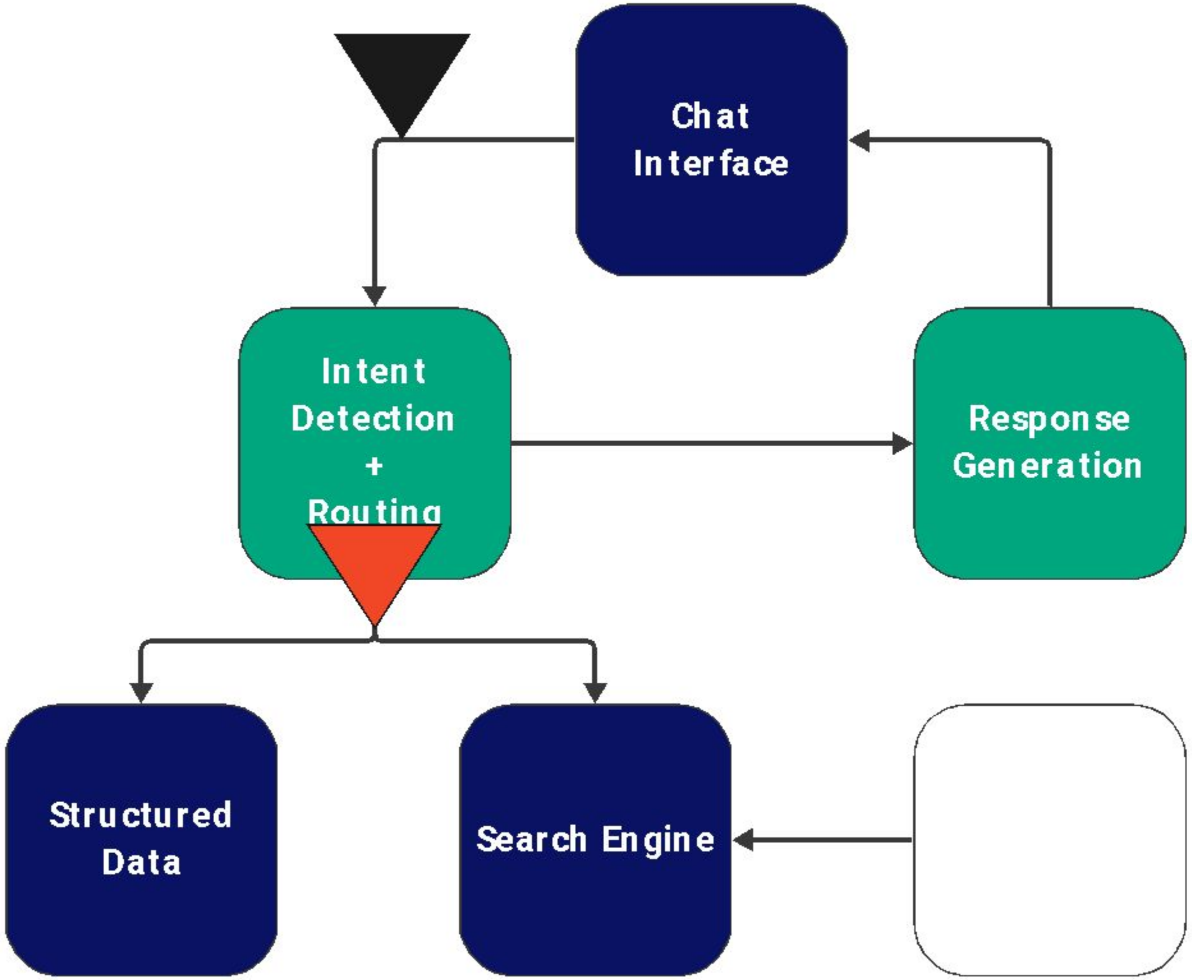
- Not useful for diagnosing issues



# Structure Your Code for Measurement

## Intent Detection

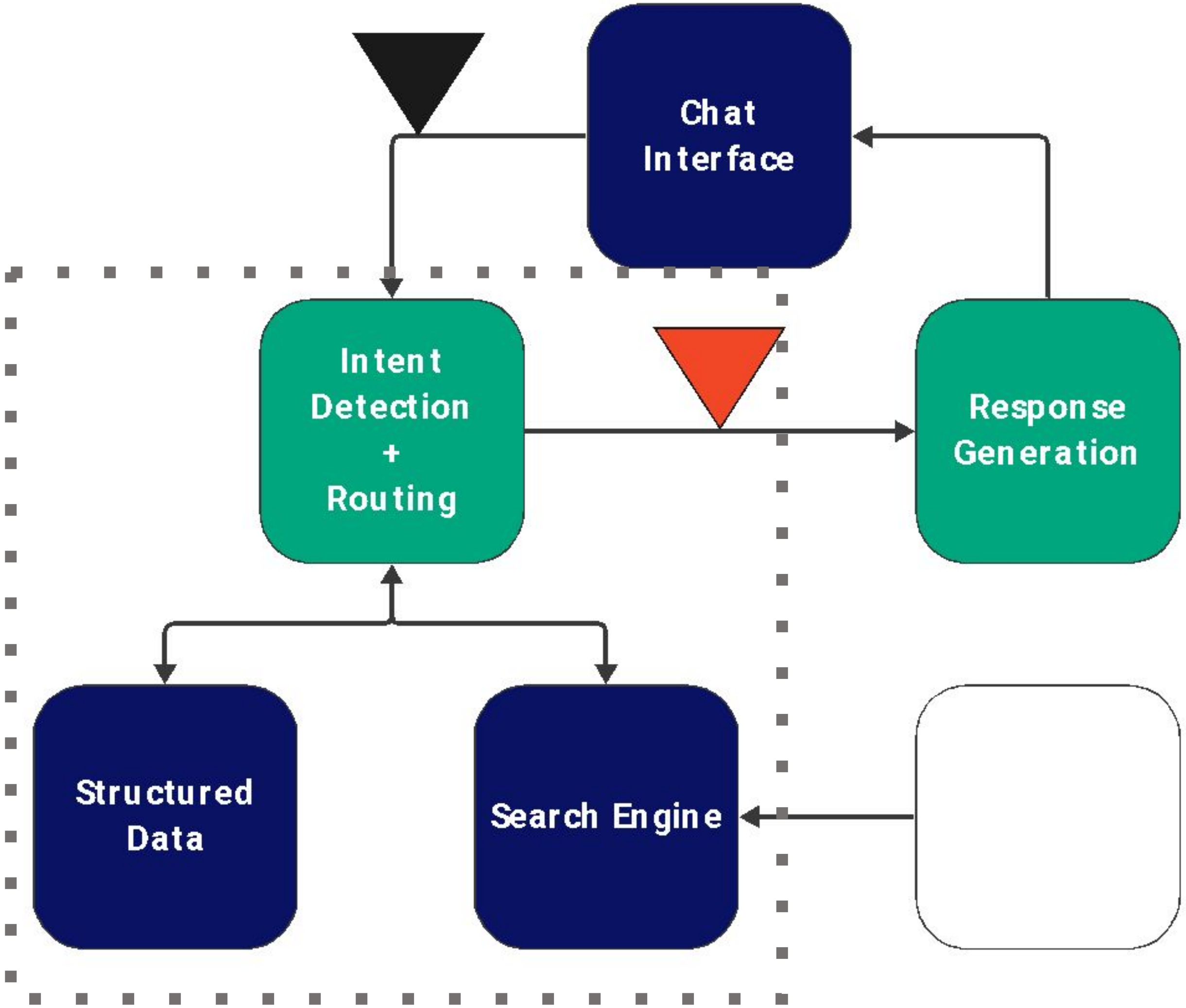
- Measure intermediate steps
- Often measured like a classifier



# Structure Your Code for Measurement

## Entirety of Retrieval

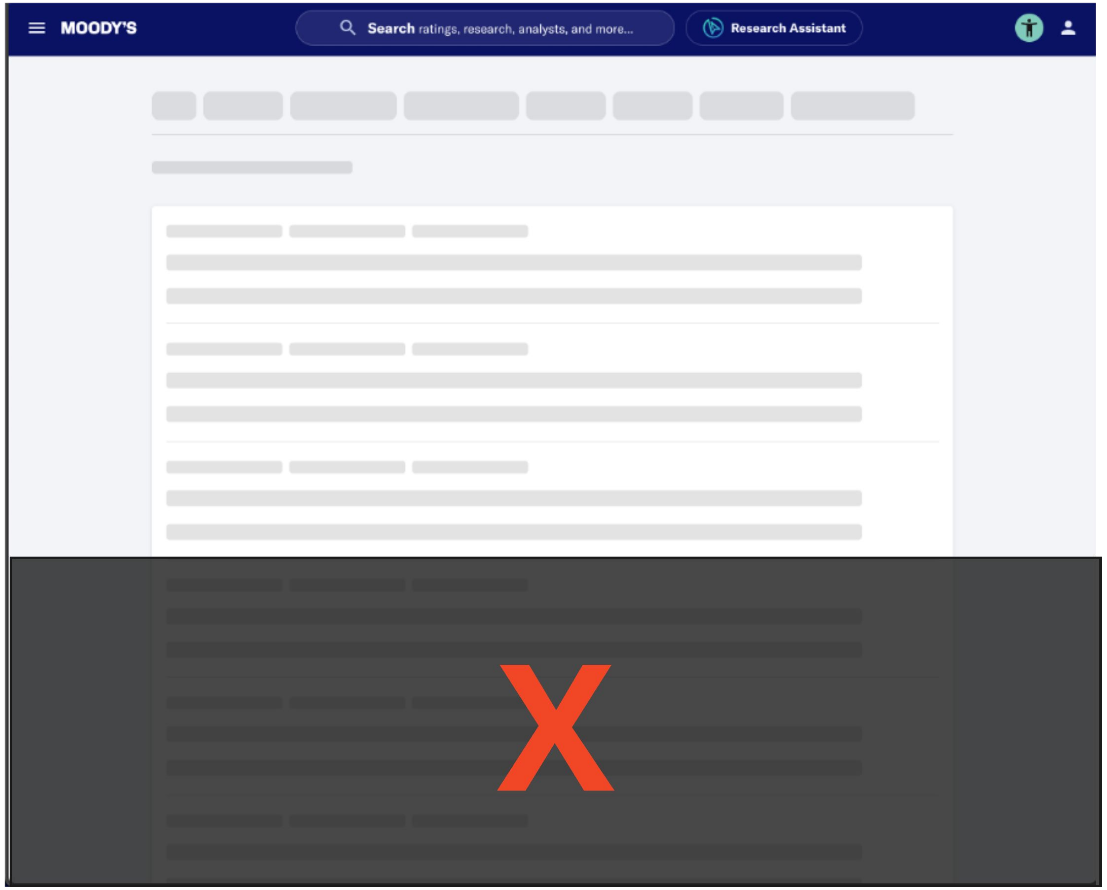
- Prompt - Chunk pairs
- Tight coupling to measure the right requests
- Enable parallel experimentation



# Structure Your Code for Measurement

## Entirety of Retrieval

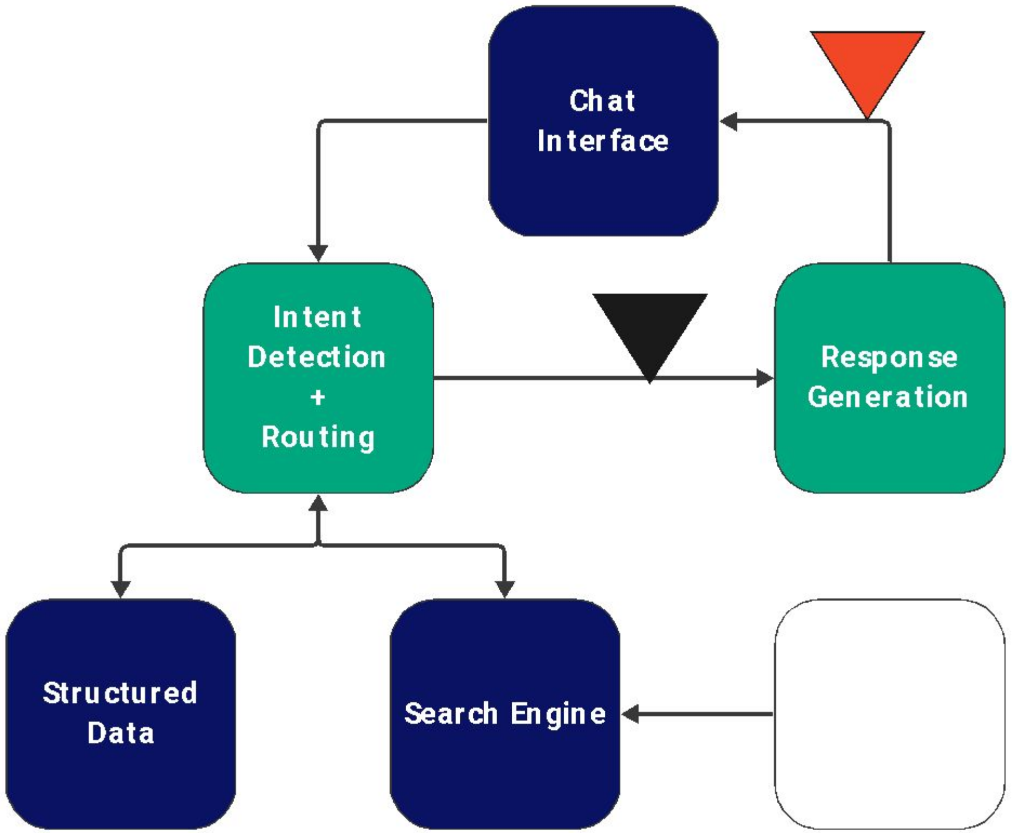
- LLMs cannot scroll but they can discern relevance
- Precision@k primary objective
- Avoid blinders with secondary stats
    - Recall@k
    - ERR



# Structure Your Code for Measurement

## Generation

- Prompt (with chunks) – Response pairs
- Generation stays faithful to retrieval



# Structure Your Code for Measurement

## Evaluate with LangSmith

Tests Examples

Search by name...

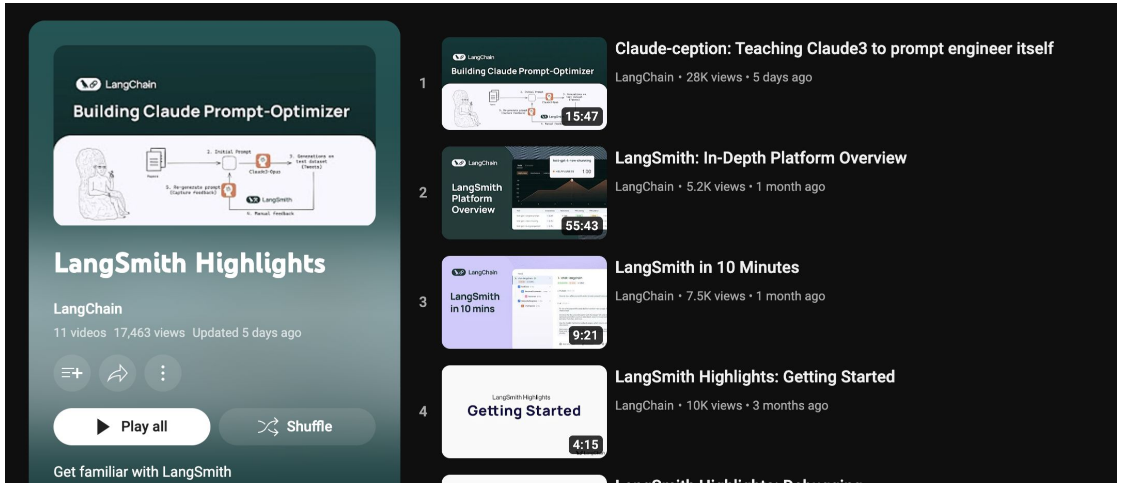
Columns

| Name | P50 Latency | P99 Latency | Creation Time | Run Count | Error Rate | Answer_relevancy_score | Faithfulness_score | Precision@K |
|---|---|---|---|---|---|---|---|---|
| warm-marble-36 | 4.14s | 4.93s | 2/8/2024, 6:02:20 PM | 8 | 0% | 0.67 | 0.65 | 0.43 |
| flowery-visitor-88 | 8.11s | 17.62s | 2/8/2024, 4:55:48 PM | 80 | 0% | 0.88 | 0.86 | 0.76 |
| notable-clove-17 | 3.91s | 5.67s | 2/8/2024, 4:37:52 PM | 80 | 0% | 0.87 | 0.80 | 0.76 |

# Structure Your Code for Measurement

## Evaluate with LangSmith

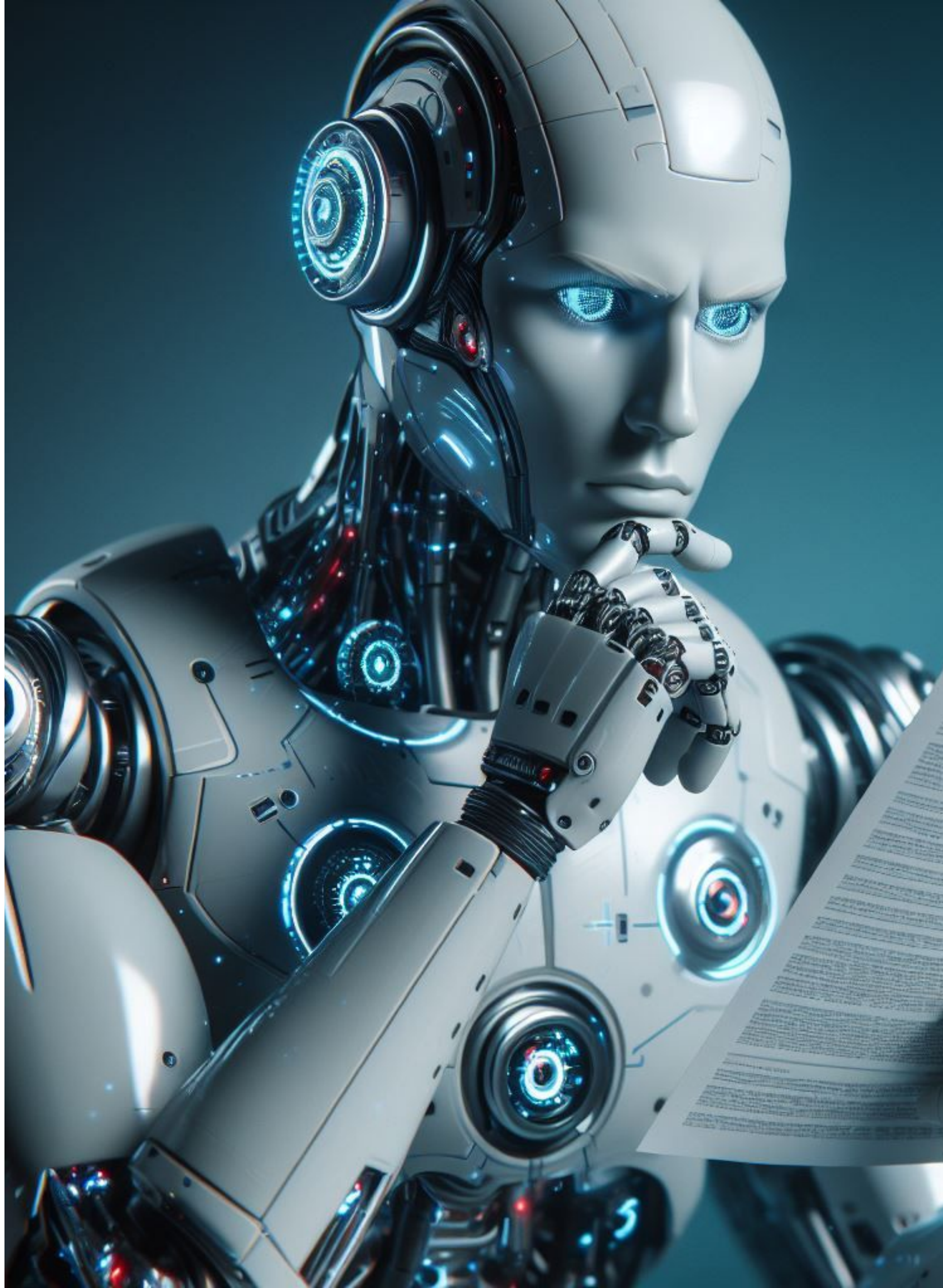
https://www.youtube.com/playlist?list=PLfaIDFEXuae2CjNiTeqXG5r8n9rld9qQu




https://blog.langchain.dev/evaluating-rag-pipelines-with-ragas-langsmith/

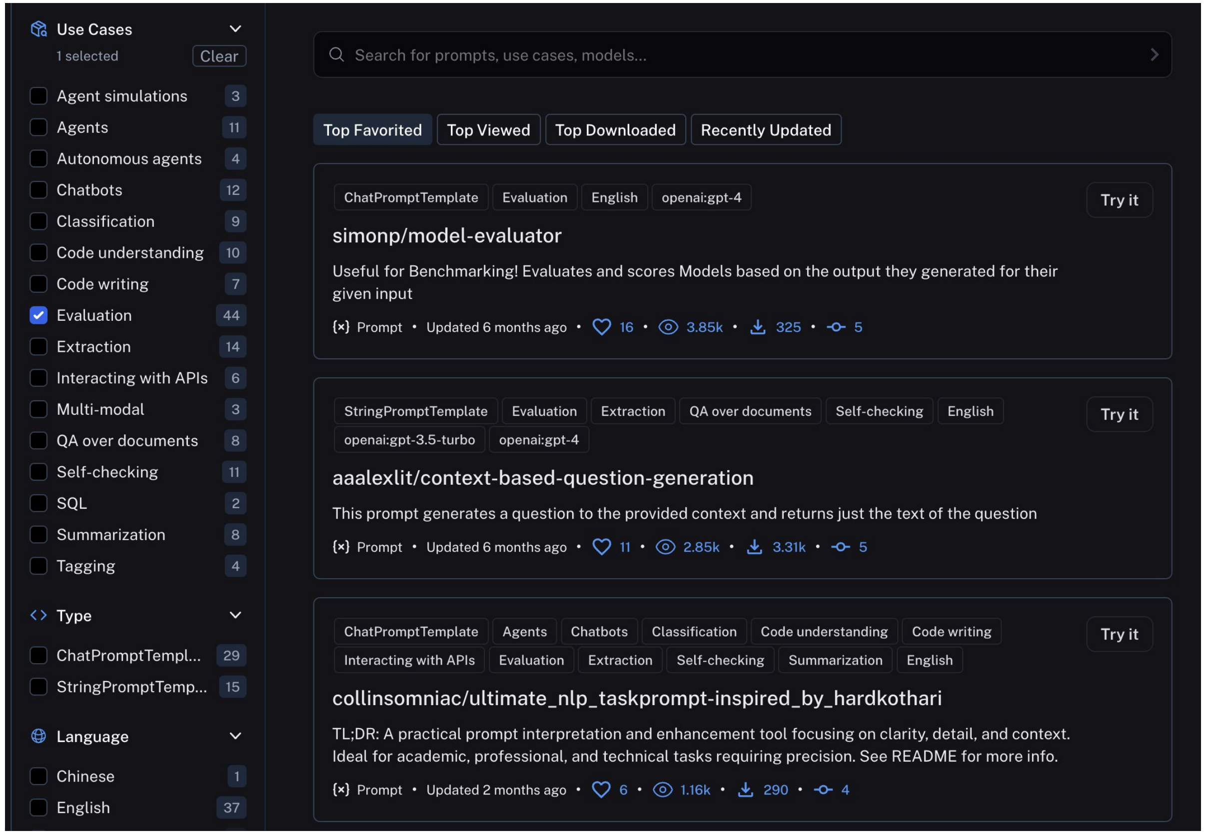
# Setup Offline Testing with AI Judgements

- Many steps in RAG need annotations
- Judgement of question - chunks pairs is time consuming and mentally taxing
- Use an LLM prompt as a *proxy*



# Setup Offline Testing with AI Judgements

## Evaluate with LangSmith



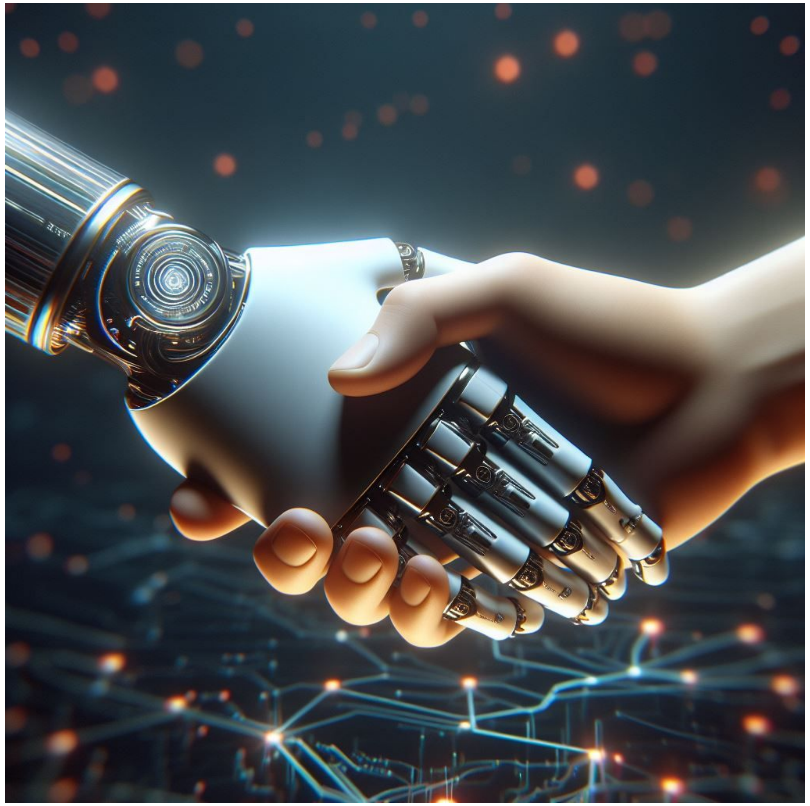
# Setup Offline Testing with AI Judgements

## Validate Against Experts

**sklearn.metrics.cohen_kappa_score**

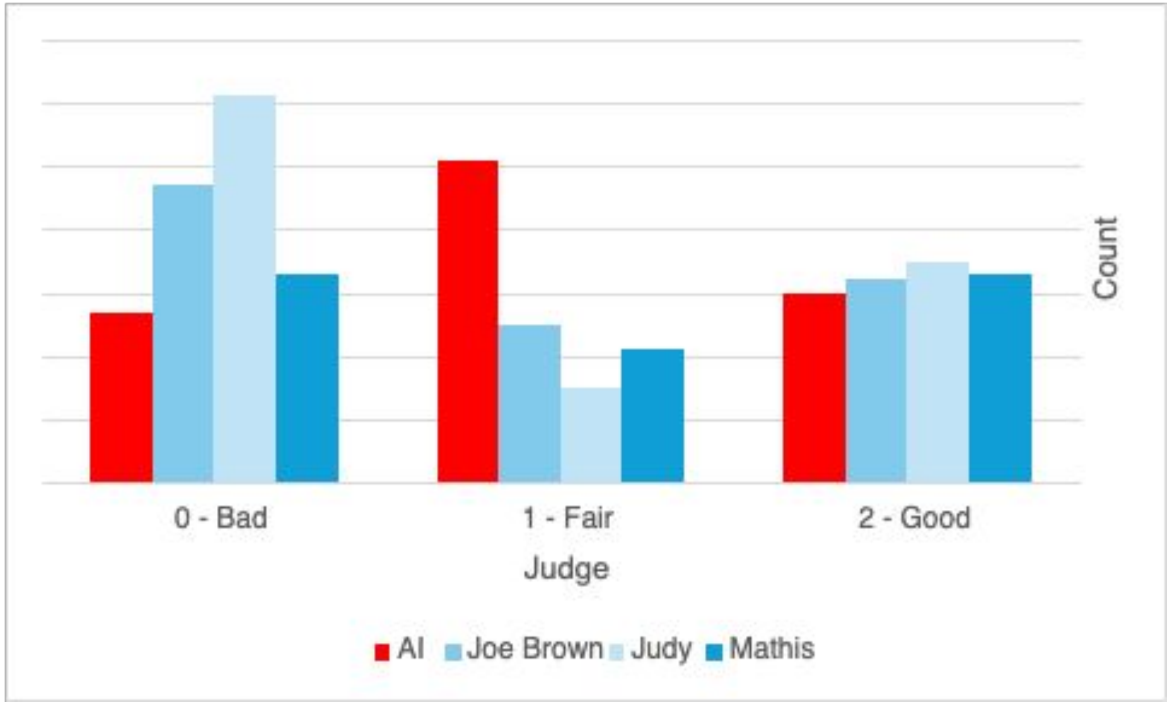
`sklearn.metrics.cohen_kappa_score(y1, y2, *, labels=None, weights=None, sample_weight=None)` ¶ [source]

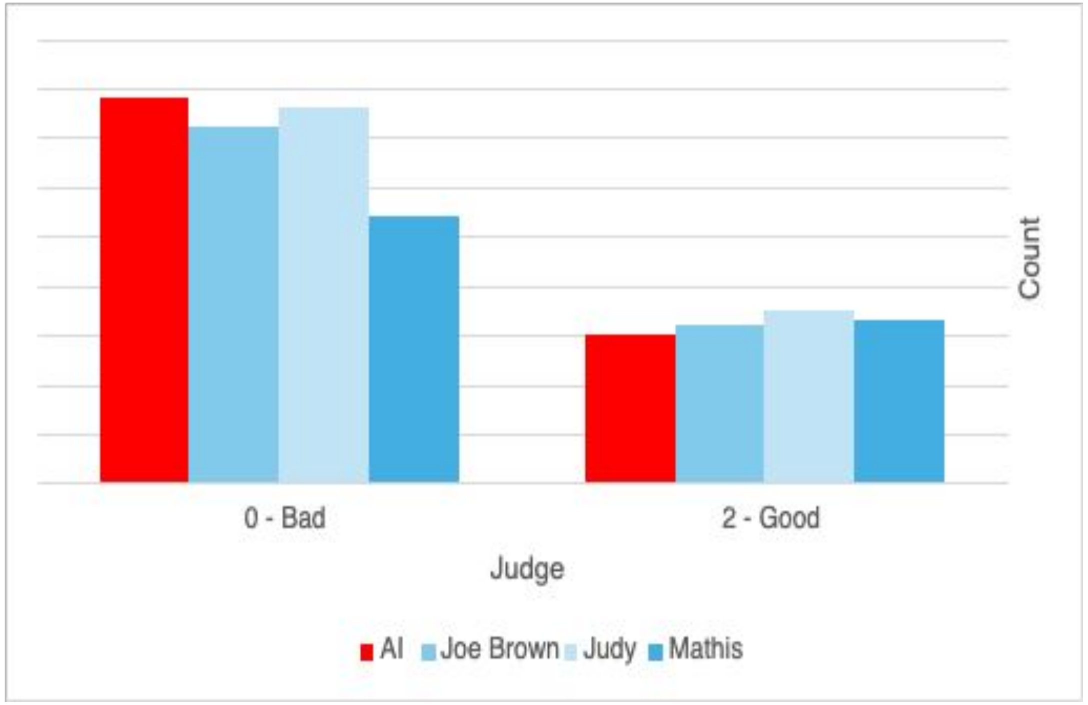
- Gather a small set of expert judgements
- Measure your prompt's agreement with experts using **Cohen's Kappa**
- Does the LLM agree with experts as much as they agree with each other?



# Setup Offline Testing with AI Judgements

## Validate Against Experts



# Setup Offline Testing with AI Judgements

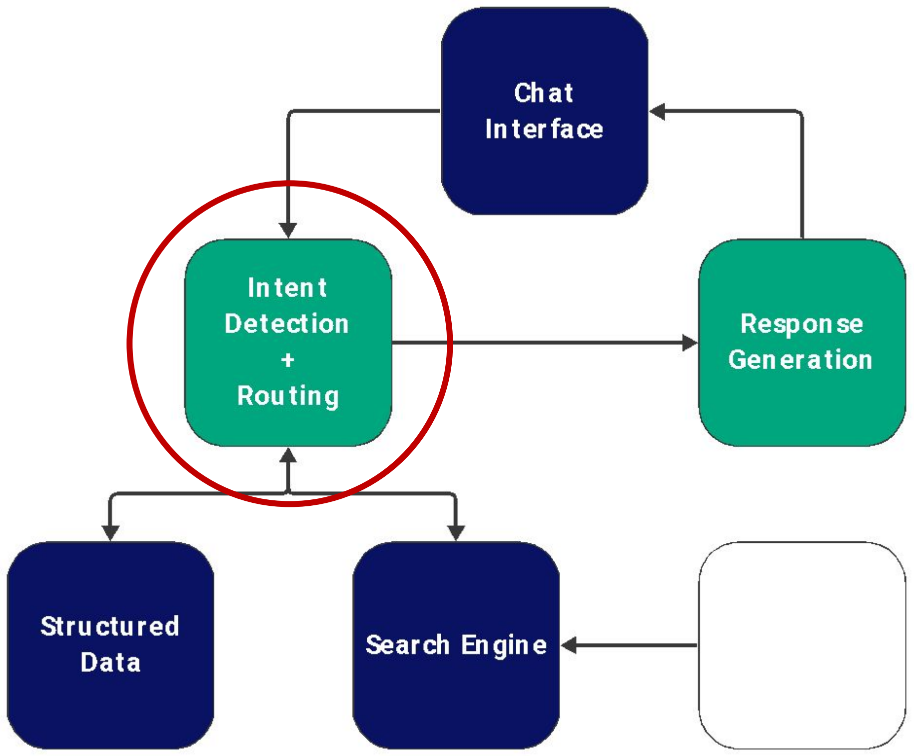
## Validate Against Experts

```
Reply in JSON with the following structure:
- explanation: Why the document is relevant to the query
- judgement: The judgement you would apply to the text
```

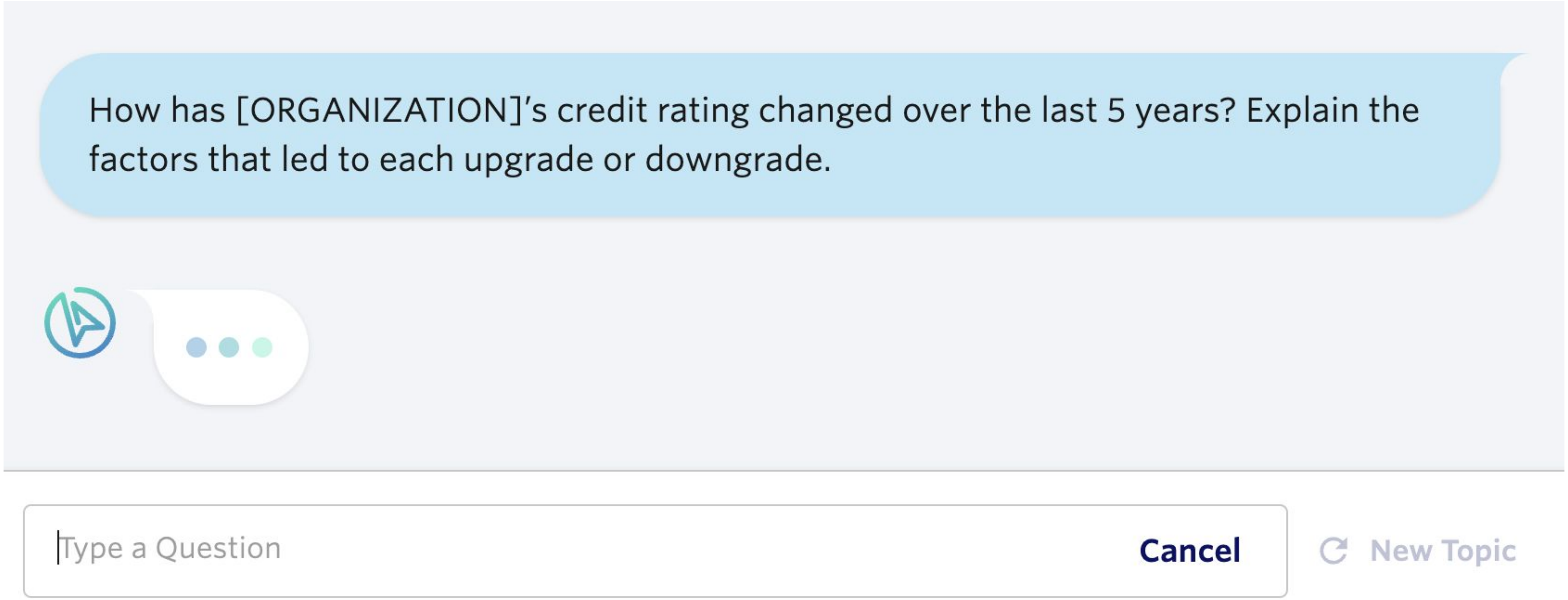
- If a model is not speaking, it is not thinking
- Use Chain of Thought

# Fast Track Intent Detection

- Filter the search space
- Encode knowledge from those who know your data well



# Intent Detection: A Tale of Three Queries



# Intent Detection: A Tale of Three Queries

“How has [ORGANIZATION]’s credit rating changed over the last 5 years?  Explain the factors that led to each upgrade or downgrade.”

**Query 1**

KNN search through research publications

```
GET research/_search
{
  "query": {
    "knn": {
      "chunk_vector": {
        "vector": <prompt vector>,
        "k": 80
      }
    }
  }
}
```

# Intent Detection: A Tale of Three Queries

"How has [ORGANIZATION]'s credit rating changed over the last 5 years?  Explain the factors that led to each upgrade or downgrade."

**Query 2**

KNN search through research publications, *filtering for reports about this organization*

```
GET research/_search
{
  "query": {
    "knn": {
      "chunk_vector": {
        "vector": <prompt vector>,
        "k": 80,
        "filter": {
          "term": {
            "org_id": 1234
          }
        }
      }
    }
  }
}
```

# Intent Detection: A Tale of Three Queries

"How has [ORGANIZATION]'s credit rating changed over the last 5 years?  Explain the factors that led to each upgrade or downgrade."

**Query 3**

SQL query of ratings data

```sql
SELECT rating, rating_date
FROM ratings
WHERE org_id = 1234
ORDER BY rating_date DESC
```

# Intent Detection: A Tale of Three Queries

Which query will most reliably fetch all the ratings?

**Query 1**

KNN search of research publications

**Query 2**

Filtered KNN search

**Query 3**

SQL query to ratings DB

# Intent Detection: A Tale of Three Queries

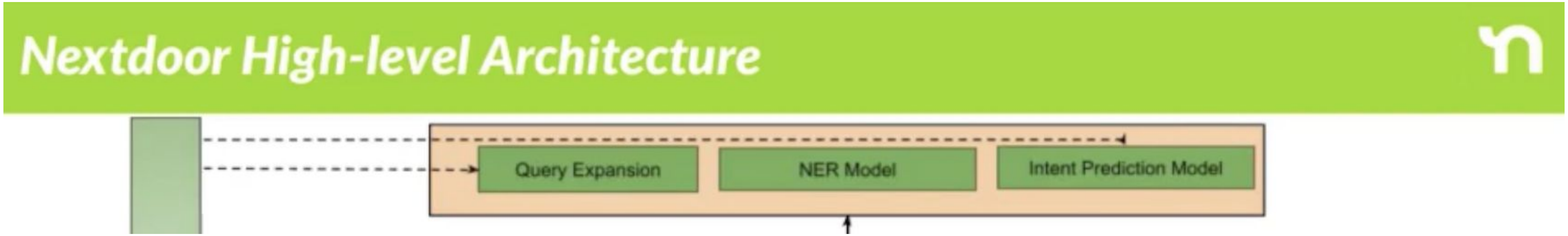
Which query will provide the most context about why ratings changed?

| Query 1 | Query 2 | Query 3 |
| --- | --- | --- |
| KNN search of research publications | Filtered KNN search | SQL query to ratings DB |

# Fast Track Intent Detection

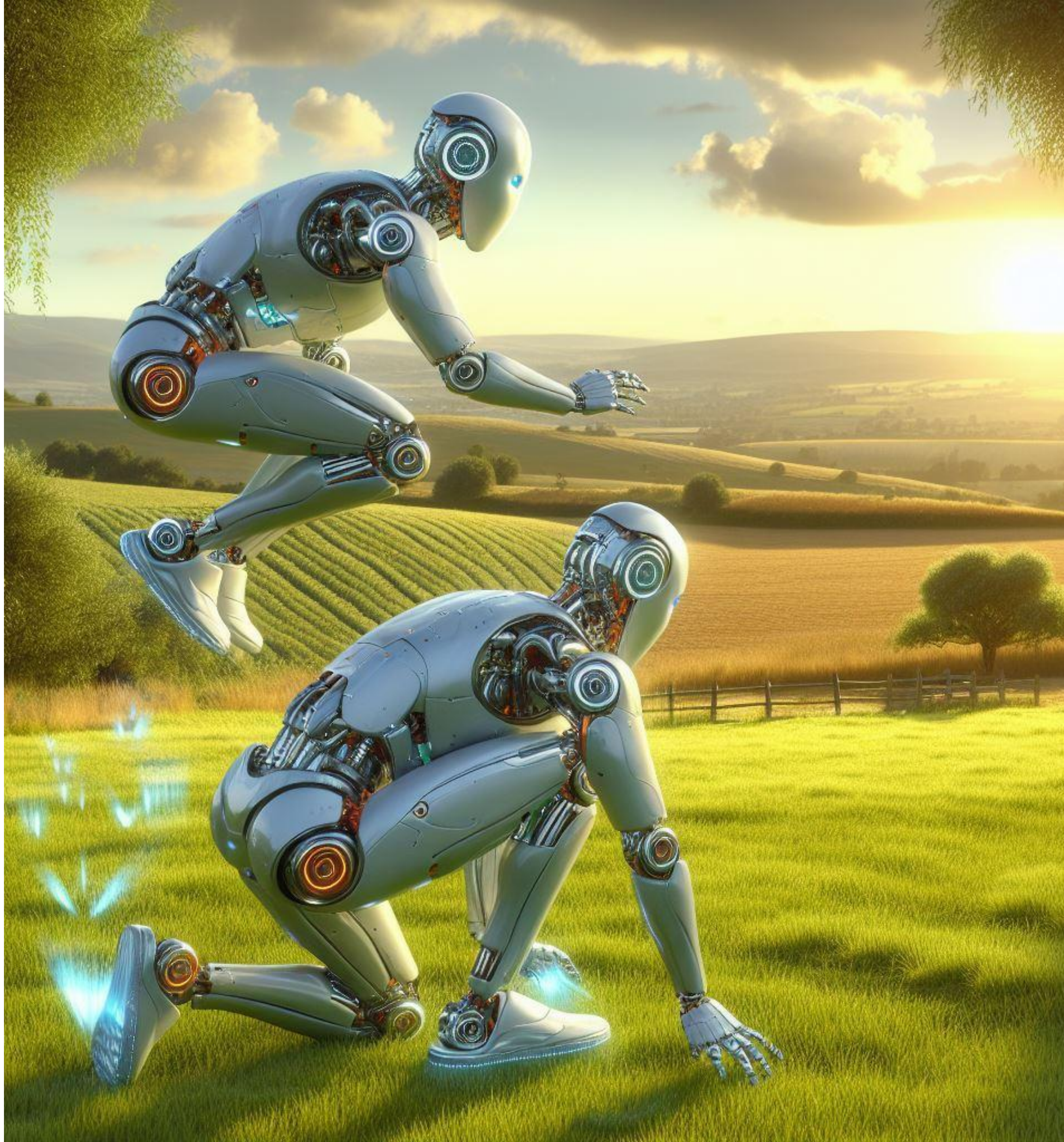
Haystack US 2022 - Bojan Babic, Nextdoor - AI Driven Search

https://www.youtube.com/watch?v=0bqVfmVurkk&t=632s



# Fast Track Intent Detection

- Craft prompts instead of training models
- Set a benchmark and buy time for purpose-built components

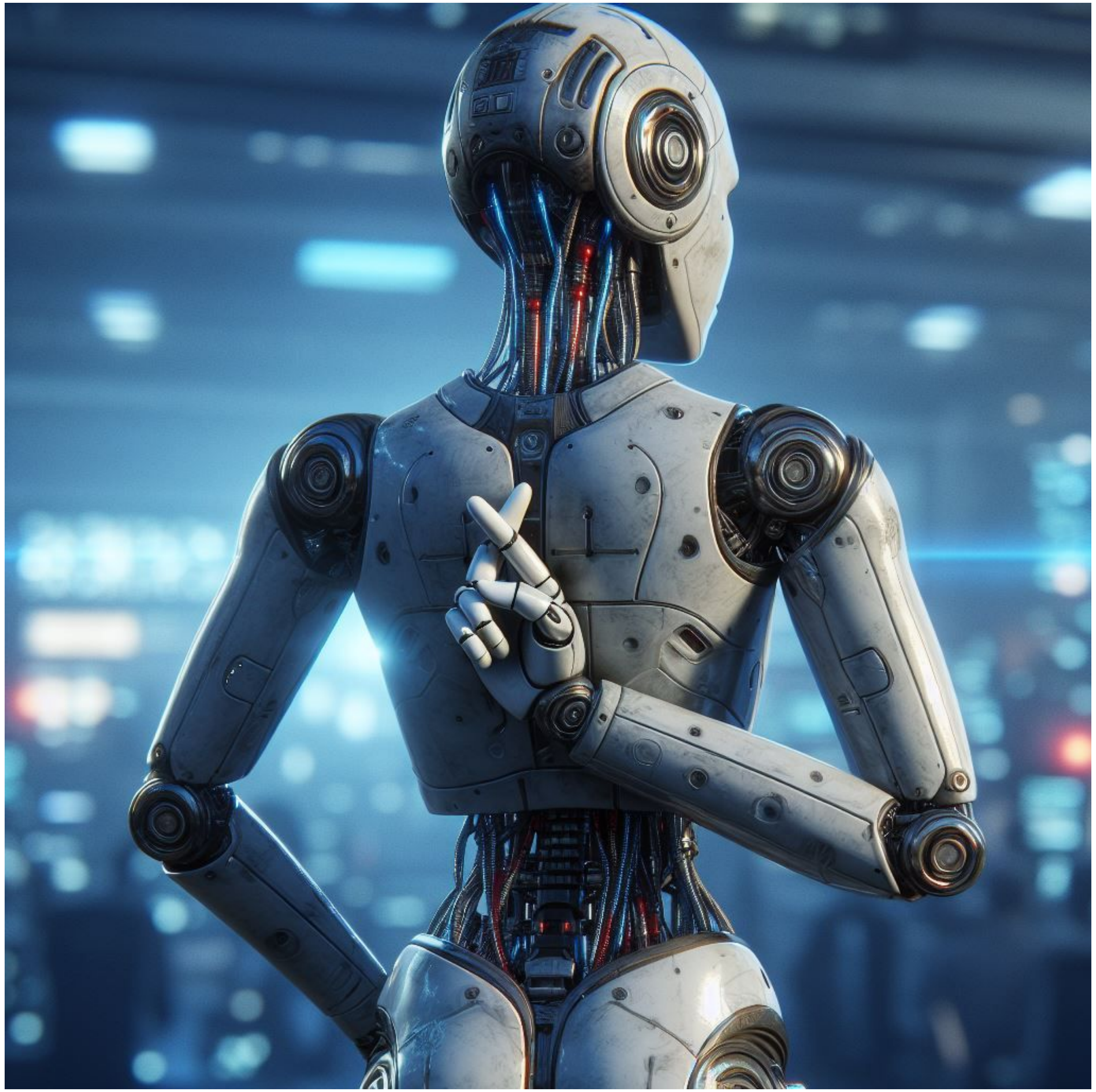


# Fast Track Intent Detection

https://www.youtube.com/watch?v=pfpIndq7Fi8

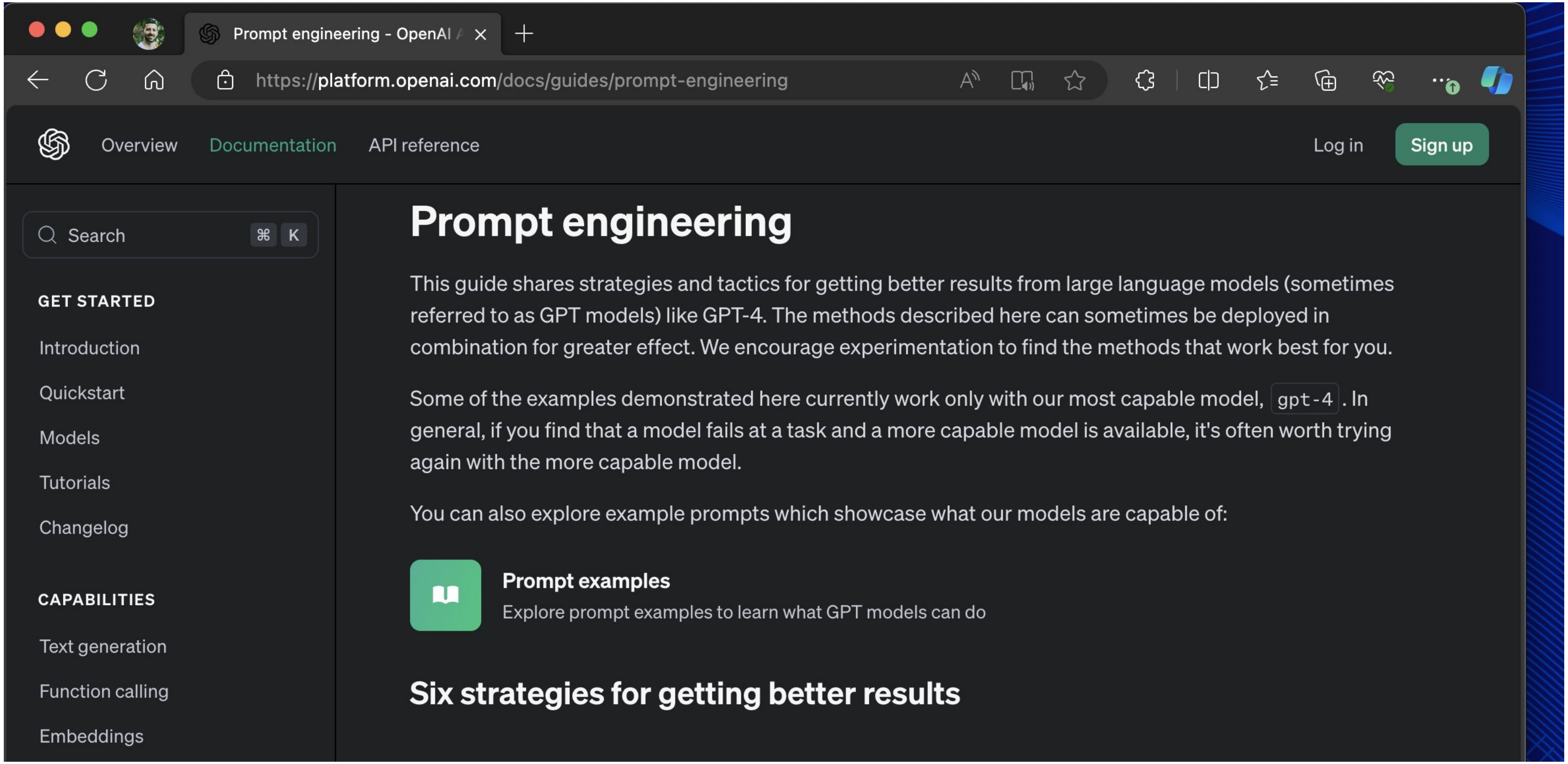
# Guard Against Hallucinations

- User trust is crucial for adoption
- Should be able to say "I don't know”



# Guard Against Hallucinations

https://platform.openai.com/docs/guides/prompt-engineering



# Guard Against Hallucinations

## Use delimiters to clearly indicate distinct parts of the input

```
You are a research assistant.  Use the set of articles (delimited with
XML tags) to answer the question.


<article> insert first article here </article>

<article> insert second article here </article>


<question> insert user question <question>
```

"…disambiguate task details. Don't make the model work to understand exactly what you are asking of them."

# Guard Against Hallucinations

## Instruct the model to answer using a reference text

```
You are a research assistant.  Use the set of articles (delimited with
XML tags) to answer the question. If the answer cannot be found in the
articles, write "I could not find an answer."


<article> insert first article here </article>

<article> insert second article here </article>


<question> insert user question <question>
```

- This is example is optimistic about the performance of a single, short instruction

- Don’t be afraid to be heavy handed

# Guard Against Hallucinations

Instruct the model to answer with *citations* from a reference text

```
Your task is to answer the question using only the provided documents
and to cite the passage(s) of the document used to answer the question.
If the document does not contain the information needed to answer this
question then write: " I could not find an answer." " If an answer to
the question is provided, it must be annotated with a citation. Use the
following format for to cite relevant passages ({"citation": …}).


<article>...
```

- Clever UI coding can take this experience to the next level

# Guard Against Hallucinations

## Use the temperature parameter

**`temperature`** number or null Optional Defaults to 1

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

We generally recommend altering this or `top_p` but not both.

https://platform.openai.com/docs/api-reference/chat/create#chat-create-temperature

# Guard Against Hallucinations

## Do your homework

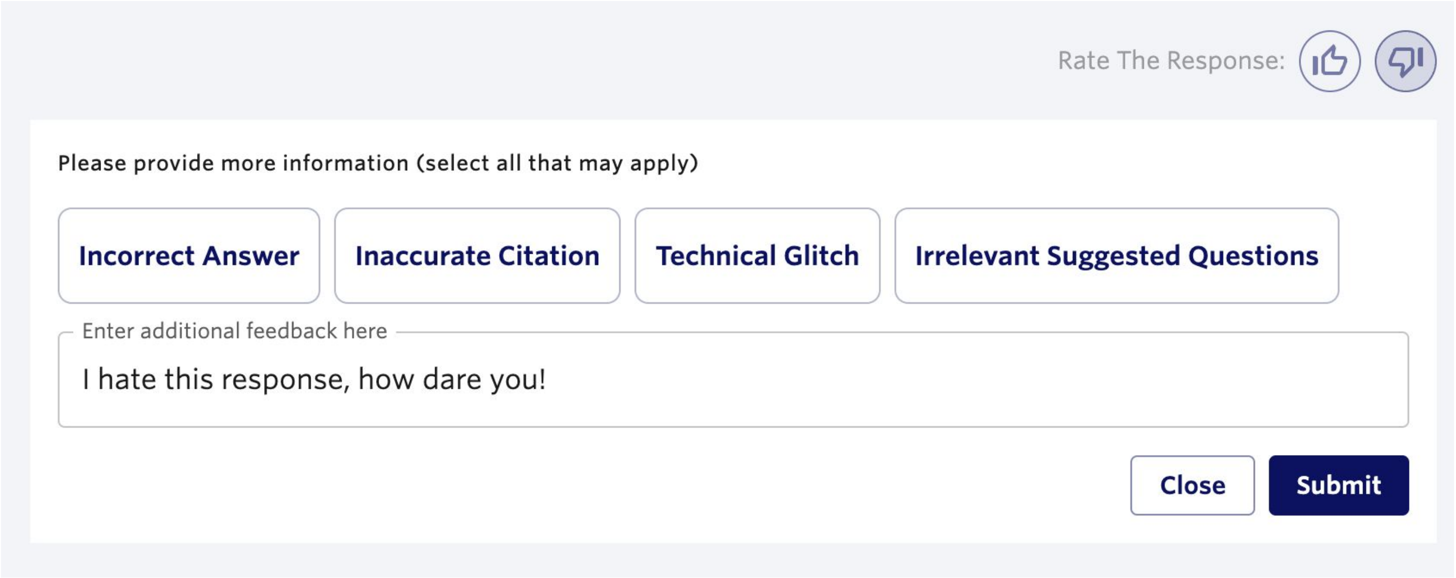
Overview Documentation — https://platform.openai.com/docs/guides/prompt-engineering

Learn Prompting — https://learnprompting.org/

Prompt Engineering Guide — https://www.promptingguide.ai/

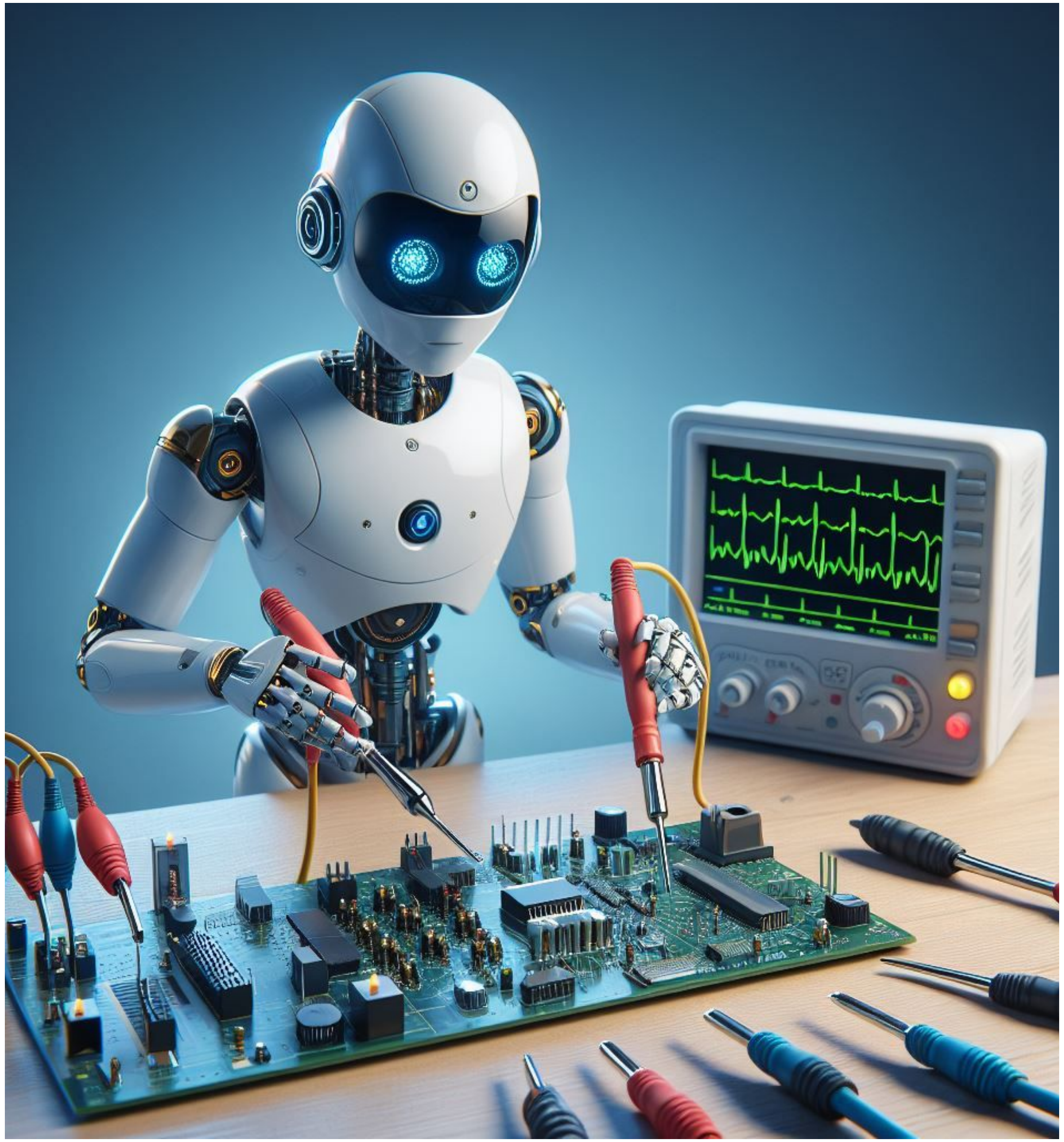
# Run an Internal Beta

- Get feedback directly in the app
- Encode experts' feedback into intent detection



# Run an Internal Beta

- Log all inputs and outputs
- Prioritize and iterate
- Beware of overfitting to internal users



# What To Explore Next

- Hybrid search optimization
- Cheaper, more performant intent detection components
- Chunking algorithm experimentation
- Re ranking
- Experiment with off the shelf embedding models
- Embedding model pre-training and fine tuning

# Appendix

All generated images were created using Microsoft Copilot Designer https://copilot.microsoft.com/images/create

- OpenAI tokenizer tool https://platform.openai.com/tokenizer
- OpenAI pricing https://openai.com/pricing
- Anthropic pricing https://www.anthropic.com/api
- Cohere pricing https://cohere.com/pricing
- Amazon SageMaker pricing https://aws.amazon.com/sagemaker/pricing/
- LangSmith Highlights youtube playlist https://www.youtube.com/playlist?list=PLfaIDFEXuae2CjNiTeqXG5r8n9rld9qQu
- Evaluating RAG Pipelines with RAGAS + LangSmith https://blog.langchain.dev/evaluating-rag-pipelines-with-ragas-langsmith/
- Haystack US 2022 - Bojan Babic, Nextdoor - AI Driven Search https://www.youtube.com/watch?v=0bqVfmVurkk&t=632s
- RAG From Scratch: Routing https://www.youtube.com/watch?v=pfpIndq7Fi8
- OpenAI Prompt Engineering Guide https://platform.openai.com/docs/guides/prompt-engineering
- Learn Prompting https://learnprompting.org/
- Prompt Engineering Guide https://www.promptingguide.ai/

Thank you
for listening

Questions
are welcome