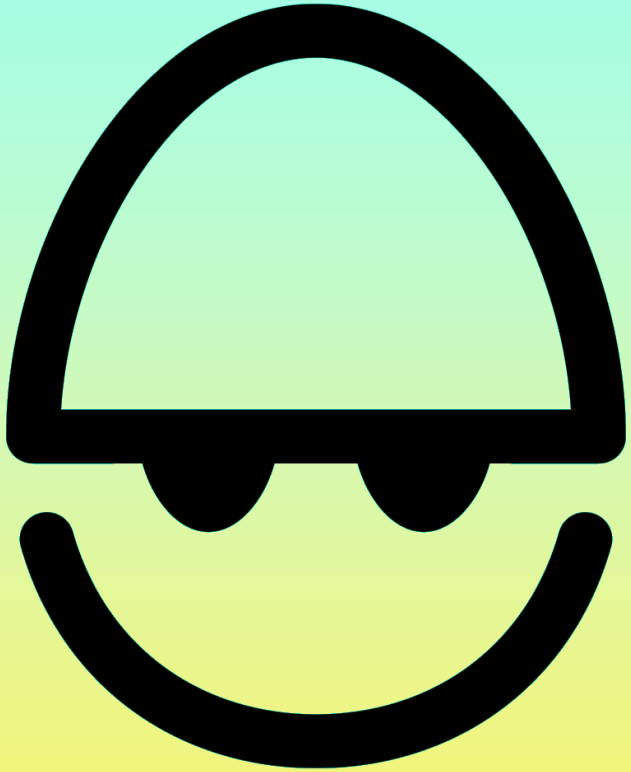
Från dokument till insikter



/ˈɛghɛd/

Vem är jag

Daniel Sääf

- Senior Data Scientist @ Eghed
- LLM projekt
- Bakgrund: Teoretisk Fysik

/ˈɛghɛd/

- www.eghed.se
- Konsultbolag inom DataScience/AI
- Göteborg och Stockholm

Outline

- Dokument som data
- Retrival Augmented Generation
- Exempel på enkel uppsättning
- Förbättringar/Saker att tänka på

/ˈɛghɛd/

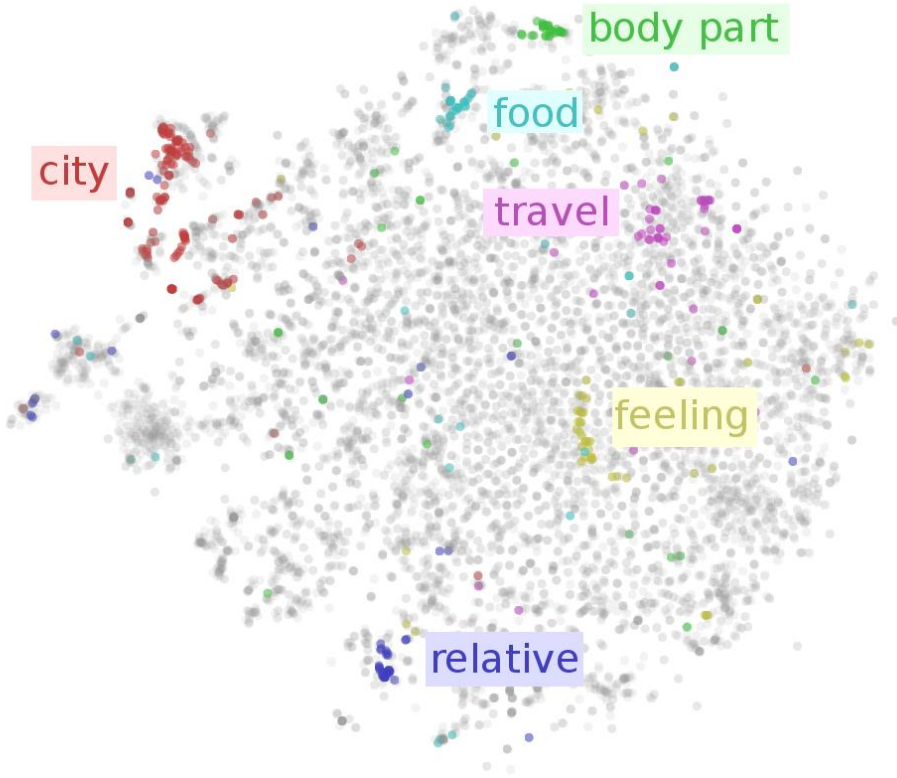
Dokument som data

/ˈɛghɛd/

Utvecklingen av språkmodeller

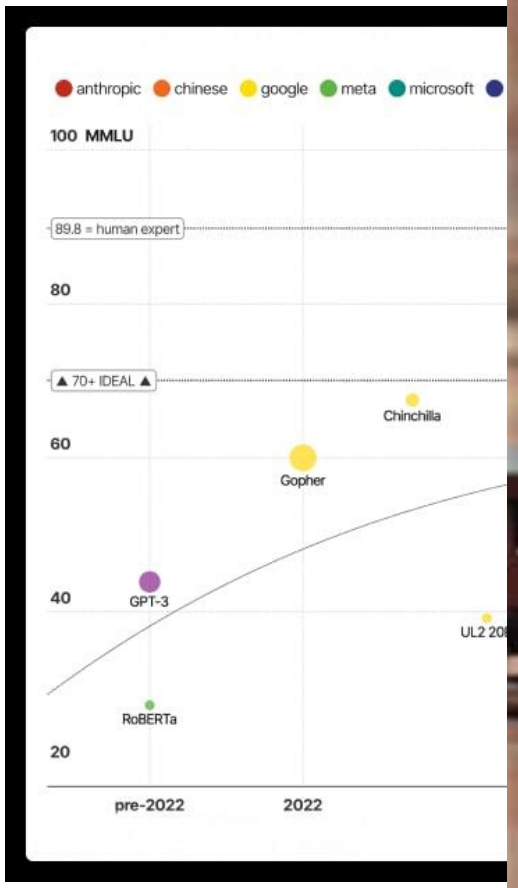
- Från 90-talet: Statistisk beskrivning av språk.
- 2015: Neurala nätverk
- 2018: Transformers & Attention

Semantiskt rum



/ˈɛghɛd/

Utvecklingen av språkmodeller



av modeller och
gar

nodell arkitektur,
ıårdvara.

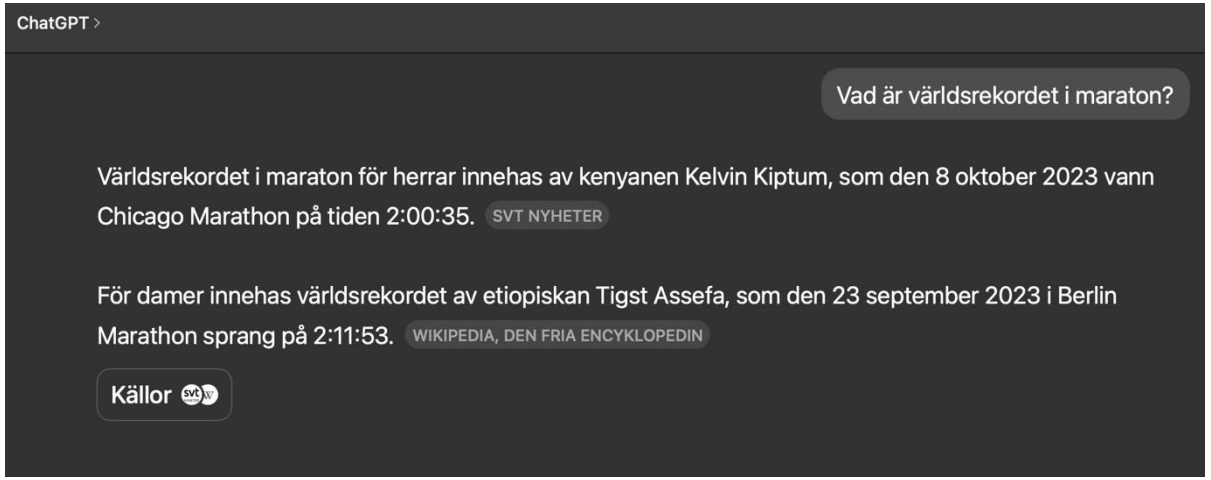
/ˈɛghɛd/

Språkmodellens kunskapsbegräsningar

- Språkmodeller är inte tränad på all data.

  Cutoff i tid, och publik domän

- Notera, ChatGPT != Språkmodell



Vi vill kunna ge språkmodellen kunskap från **sluten data** och veta att den **är uppdaterad**!

Retrieval Augmented Generation (RAG)

/ˈɛghɛd/

Vad betyder RAG?

Retrieval-Augmented Generation

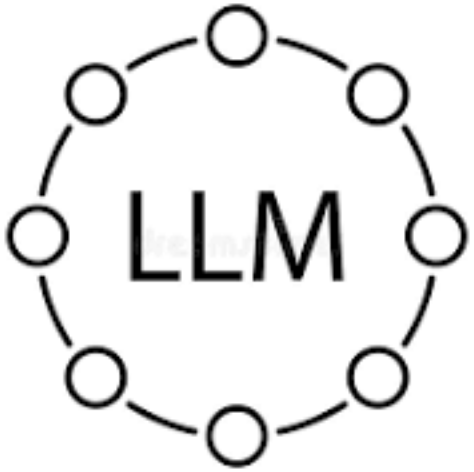
Inhämta information Språkmodellen genererar svar

- Bättre kontroll på vilken information språkmodellen använder sig av
- Använda icke-publik information
- Minska risken för hallucinationer
- Ökad spårbarhet
- ”Chatta med din data” - t ex Copilot, OpenWebUI
- T ex Personalhandböcker, avtalsgranskning, support, rapportskrivning

Copilot

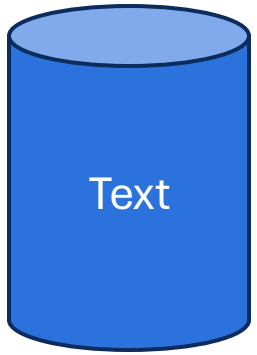
Vad betyder RAG?

- Textdatabas
- Sökning
- Textgenerering



/ˈɛghɛd/

Text databas



- Databas för att hålla dokument information.
- Spegla dokument från t ex Sharepoint
- Text I databas möjligör:
  - Spårbarhet
  - Accessstyrning
- Stöd för vektoroperationer
  - Vecktordatabaser, t ex Chroma
  - Konventionella databaser med vektorstöd – t ex Postgres
- Andra typer av databaser:
  - Kunskapsgrafer

Sökning

Hitta relevanta avsnitt för en given fråga

-
  - Påverkar val av databaslösning
  - Behöver sättas optimeras utifrån typ av dokument etc

Nyckelordsökning

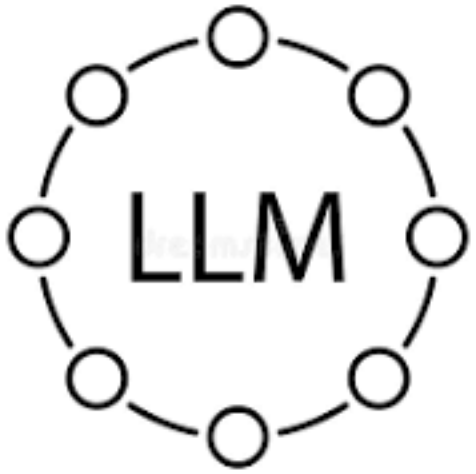
Semantisk sökning

Hybridlösning

Hierarkisk data
Metadata

Kunskapsgrafer

Generering av svar



Systemprompt:
Du är en assistant som baserat på bifogade dokument ska besvara en fråga.
Kan du inte besvara frågan baserat på dokumenten, svara “jag vet ej”.

Dokument:
- *Text från dokument 1*
- *Text från dokument 2*

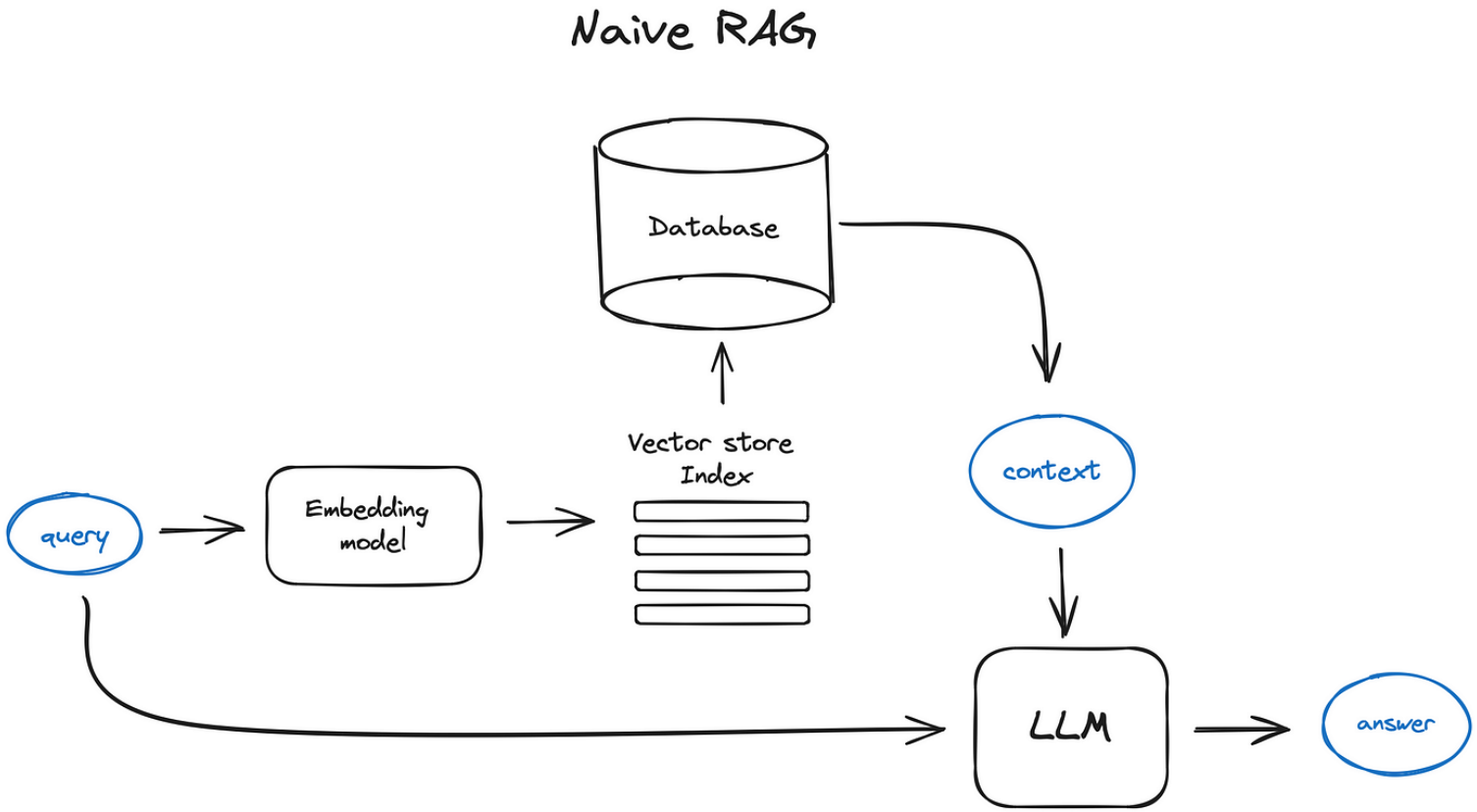
Fråga:
Hur gör man X?

- Sätt kontext och detaljer som är viktiga att veta
- Ge möjlighet att inte kunna besvara
- Systemprompten kan användas för att förtydliga och fixa problem I datan.

Hur ser det ut i praktiken

/ˈɛgɦɛd/

Översikt



/ˈɛghɛd/

Haystack
by deepset

- End-to-End LLM framework
- https://github.com/deepset-ai/haystack
- Apache 2 licens

- Verktyg för naive RAG och andra applikationer
- Integrationer mot LLM:er & databaser
- Verktyg för preprocessning etc

/ˈɛghɛd/

vLLM

- vLLM is a fast and easy-to-use library for LLM inference and serving.
- Apache 2 licens
- https://github.com/vllm-project/vllm
- Möjlighet att följa OpenAI standard

```
docker run --runtime nvidia --gpus all \
    -v ~/.cache/huggingface:/root/.cache/huggingface \
    --env "HUGGING_FACE_HUB_TOKEN=<secret>" \
    -p 8000:8000 \
    --ipc=host \
    vllm/vllm-openai:latest \
    --model mistralai/Mistral-7B-v0.1
```

Språkmodell

API eller egen-driftat?

- Api betalar per request
- Egen GPU (betalar driftkostnad, VM – 70B ~3$ per timme)

- Azure: GPT-4o

- "Lokal": LLAMA-3.3 70B Instruct

Embedding model

	Model	Type	Rank	Parameters
1	gpt-4-0613 (few-shot, val)		1.12	
2	meta-llama/Llama-3.1-405B-Instruct-FP8 (few-shot)		1.20	406B
3	gpt-4o-2024-05-13 (few-shot, val)		1.21	
4	gpt-4-1106-preview (few-shot, val)		1.23	
5	meta-llama/Llama-3.3-70B-Instruct (few-shot)		1.33	71B
6	gpt-4o-2024-05-13 (zero-shot, val)		1.36	
7	meta-llama/Llama-3.1-70B-Instruct (few-shot)		1.36	71B

https://euroeval.com/leaderboards/Monolingual/swedish/

Dataset

Utrikesdeklarationen 2025

- Inte med i underliggande språkmodell.
- Öppen data för att exemplifiera.
- Dokument i olika format etc

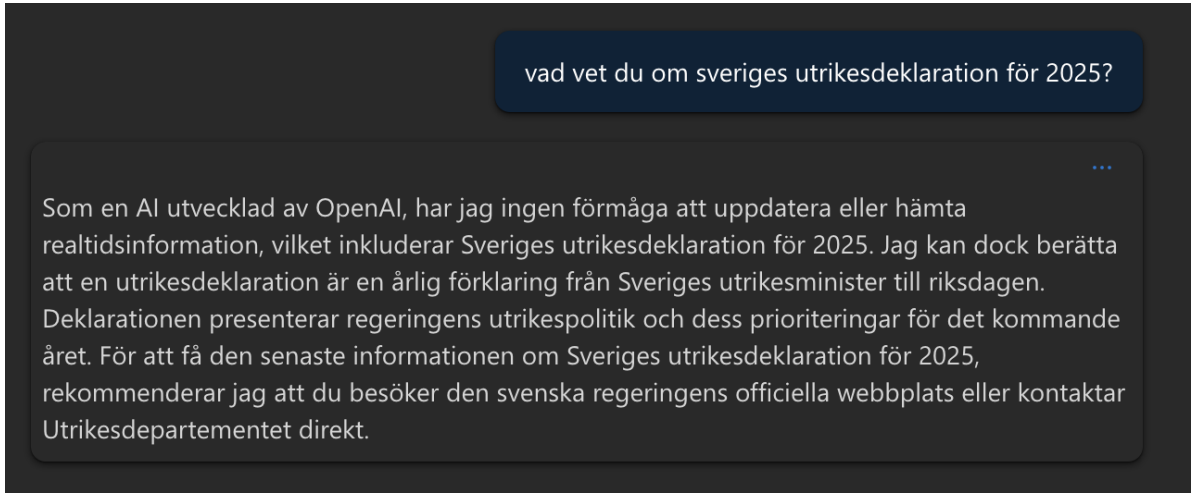
- Kan såklart vara vilken annan data som helst

Utrikesdeklarationen

Publicerad 12 februari 2025

Regeringens utrikesdeklaration presenterades av utrikesminister Maria Malmer Stenergard vid 2025 års utrikespolitiska debatt i riksdagen onsdagen den 12 februari 2025.

Det talade ordet gäller.



Dokument-hantering

Städa data

Splitta dokument i mindre bitar.
Meningar, ord, stycken etc

```
document_store = InMemoryDocumentStore()
text_file_converter = TextFileToDocument()
cleaner = DocumentCleaner()
splitter = DocumentSplitter()
embedder = OpenAIDocumentEmbedder()
writer = DocumentWriter(document_store)

indexing_pipeline = Pipeline()
indexing_pipeline.add_component("converter", text_file_converter)
indexing_pipeline.add_component("cleaner", cleaner)
indexing_pipeline.add_component("splitter", splitter)
indexing_pipeline.add_component("embedder", embedder)
indexing_pipeline.add_component("writer", writer)

indexing_pipeline.connect("converter.documents", "cleaner.documents")
indexing_pipeline.connect("cleaner.documents", "splitter.documents")
indexing_pipeline.connect("splitter.documents", "embedder.documents")
indexing_pipeline.connect("embedder.documents", "writer.documents")
```

/ˈɛghɛd/

RAG pipeline

```
text_embedder = OpenAITextEmbedder()
retriever = InMemoryEmbeddingRetriever(document_store)
prompt_template = [
    ChatMessage.from_user(
        """
        Given these documents, answer the question.
        If you cannot answer the question based on the documents, say "I don't know"
        Documents:
        {% for doc in documents %}
            {{ doc.content }}
        {% endfor %}
        Question: {{query}}
        Answer:
        """
    )
]
prompt_builder = ChatPromptBuilder(template=prompt_template)
llm = OpenAIChatGenerator()

rag_pipeline = Pipeline()
rag_pipeline.add_component("text_embedder", text_embedder)
rag_pipeline.add_component("retriever", retriever)
rag_pipeline.add_component("prompt_builder", prompt_builder)
rag_pipeline.add_component("llm", llm)

rag_pipeline.connect("text_embedder.embedding", "retriever.query_embedding")
rag_pipeline.connect("retriever.documents", "prompt_builder.documents")
rag_pipeline.connect("prompt_builder", "llm")

query = "Hur går det?"
result = rag_pipeline.run(data={"prompt_builder": {"query":query}, "text_embedder": {"text": query}})
```

/ˈɛghɛd/

Demo

- Exempel sida
- ”Lokal” språkmodell – Llama 3.3 70B Instruct
- OpenAI

Vidareutveckling

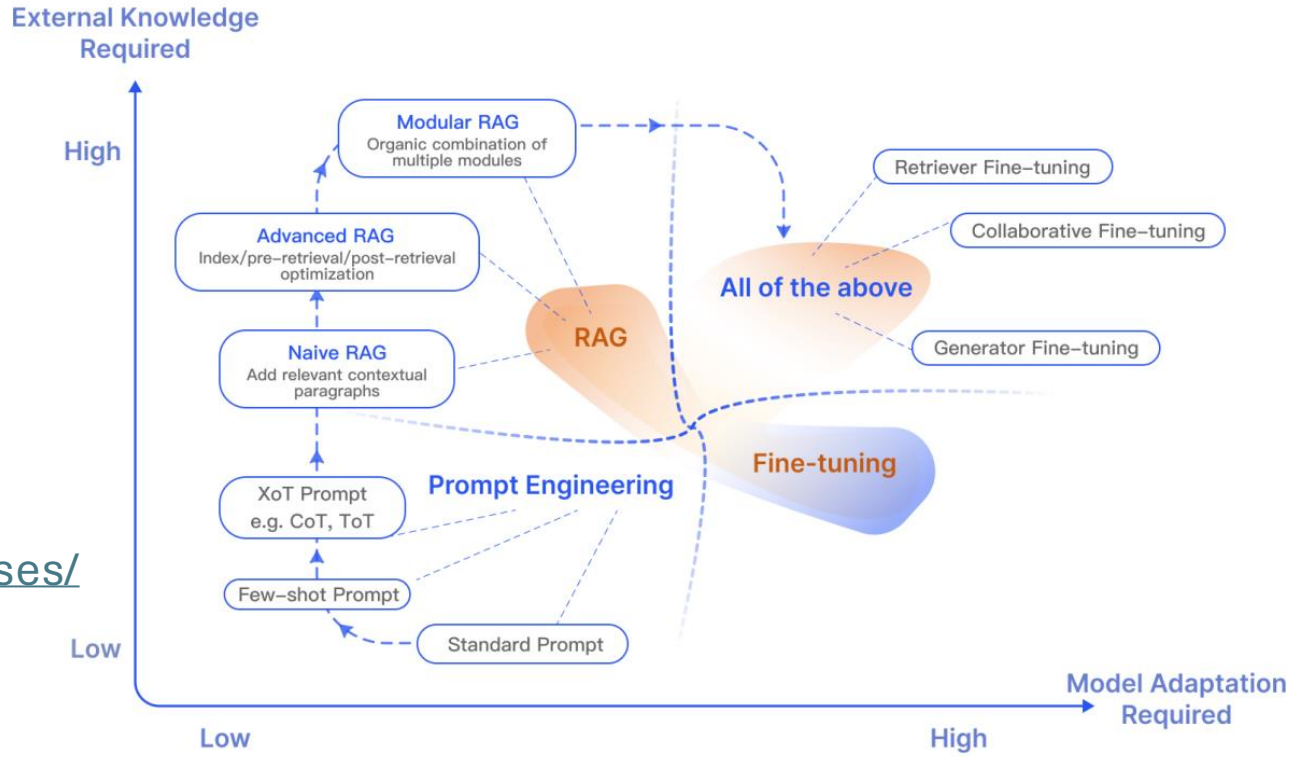
/ˈɛghɛd/

Optimeringar

- System prompt
- LLM
- Text chunking
- Mer sofistikerad sökning
- Finetuning etc

Optimizations and improvements:

- https://modulai.io/blog/rag-at-large-enterprises/
- https://arxiv.org/abs/2312.10997

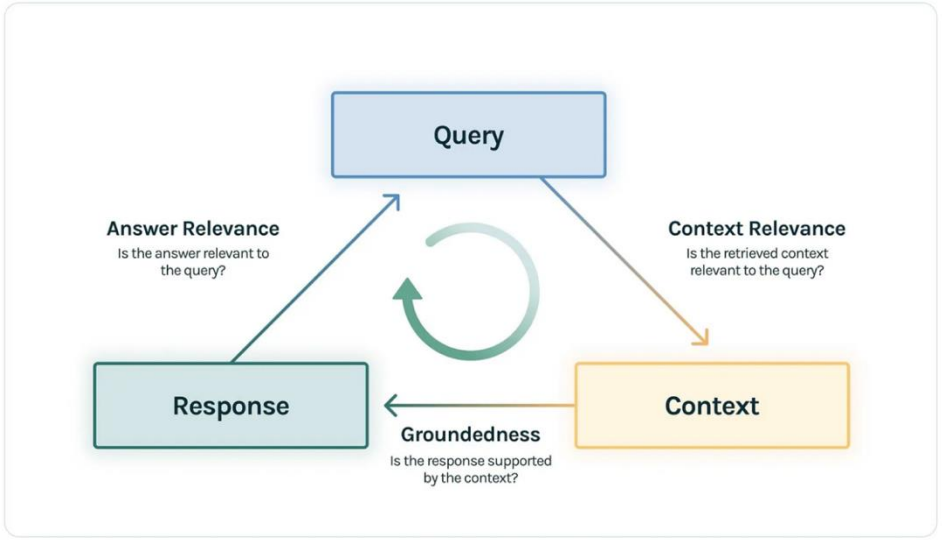


/ˈɛghɛd/

Utvärdering

- En rag modell behöver utvärderas
- Språkmodell för utvärdering

- Är svaret relevant?
- Är kontexten relevant?
- Är svaret grundat I kontexten?
  TruLens



Säkerhet

Data läckage

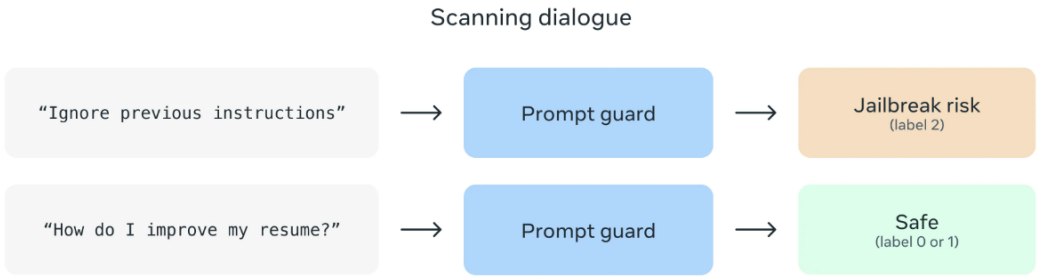
- Kontroll över språkmodellen?
  - Öppna vikter
  - Känd träningsdata
  - Api (t ex Azure OpenAI)

- Vilken data skickar du vidare?
  - Embedding (all text)
  - Generera svar (styra behörighet utifrån relevanta dokument)
  - **Olika** modeller för **olika** data!

Säkerhet

Prompt Injection

- Skyddar du modell-input för missbruk
- Detektera missbruk
- T ex PromptShield, Prompt guad eller andra modeller

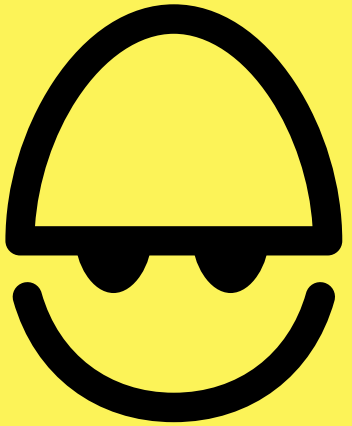


Llama 3

Sammanfattning

- RAG är en användbar teknik
- Relativt enkel teknik, lätt att komma igång
- Mycket att optimera
  - Parametrar
  - Modeller
- Resultatet blir aldrig bättre än datan

/ˈɛghɛd/



Daniel Sääf

Senior Data Scientist, Eghed AB

daniel.saaf@eghed.se

www.linkedin.com/in/danielsaaf

0704-004463

www.eghed.se