RAG vs Prompt Stuffing

Arc of AI

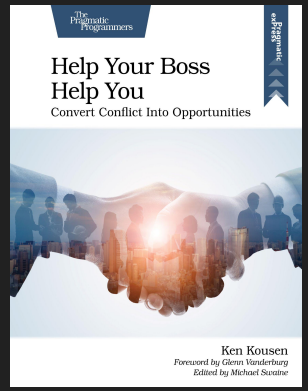
Contact Info

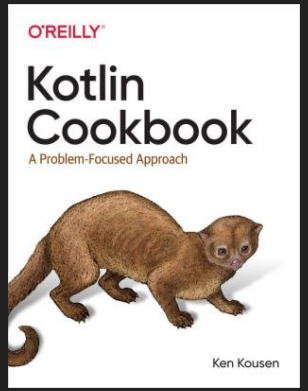
Ken Kousen

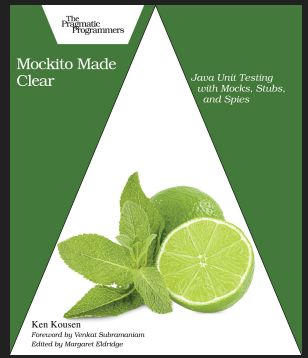
Kousen IT, Inc.

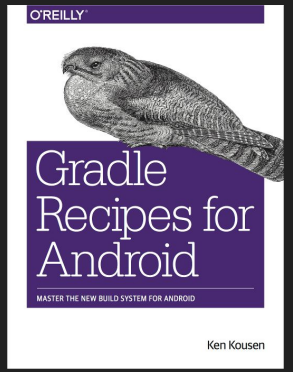
ken.kousen@kousenit.com
http://www.kousenit.com
http://kousenit.org (blog)
@kenkousen (twitter)
@kenkousen@mastodon.social (mastodon)
https://bsky.app/profile/kousenit.com

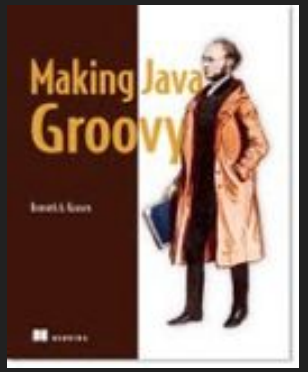
Tales from the jar side (free newsletter)
https://kenkousen.substack.com
https://youtube.com/@talesfromthejarside











LLM information

- LLMs are released fully trained
- How do we add information that:
  - Is not on the public internet, and therefore not available for training
  - Was created after the end of the training period

Add information

Three options:

1. Prompt stuffing
2. RAG
3. Fine tuning

End of training

- GPT-4o: October, 2023
- Claude 3.7: October, 2024
- Gemini 2.0 and 2.5: refuses to answer; probably around November 2023
- Mistral Large: about November 2024

Prompt Stuffing

- Add the complete text to a message
- Needs to fit inside the *context window*

Context window sizes

- GPT-3: 4K, then 16K, then 32K
- GPT-4, GPT-4o: 128K

Context window sizes

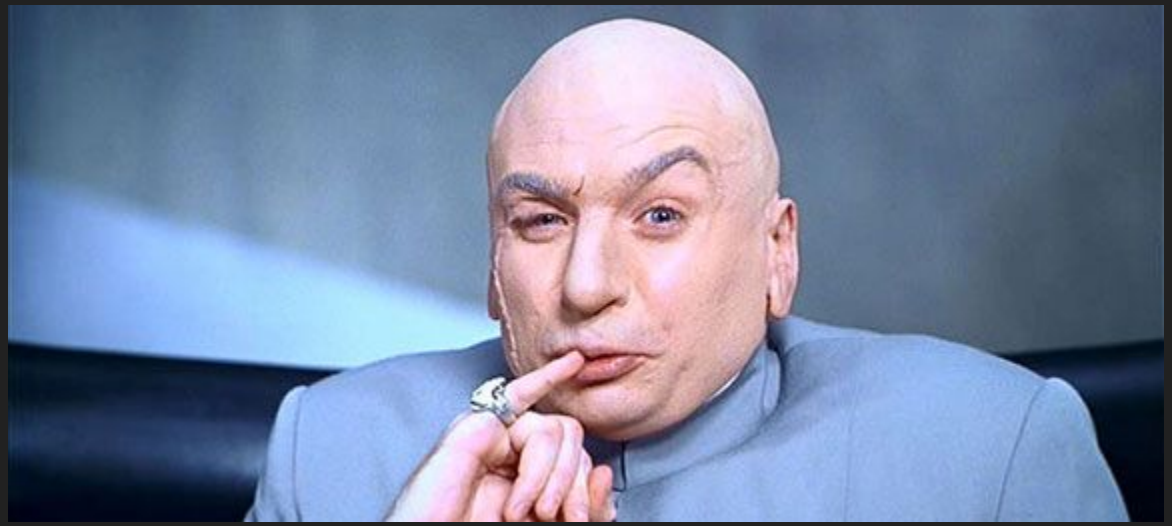
- Claude (Anthropic): 200K

Context window sizes

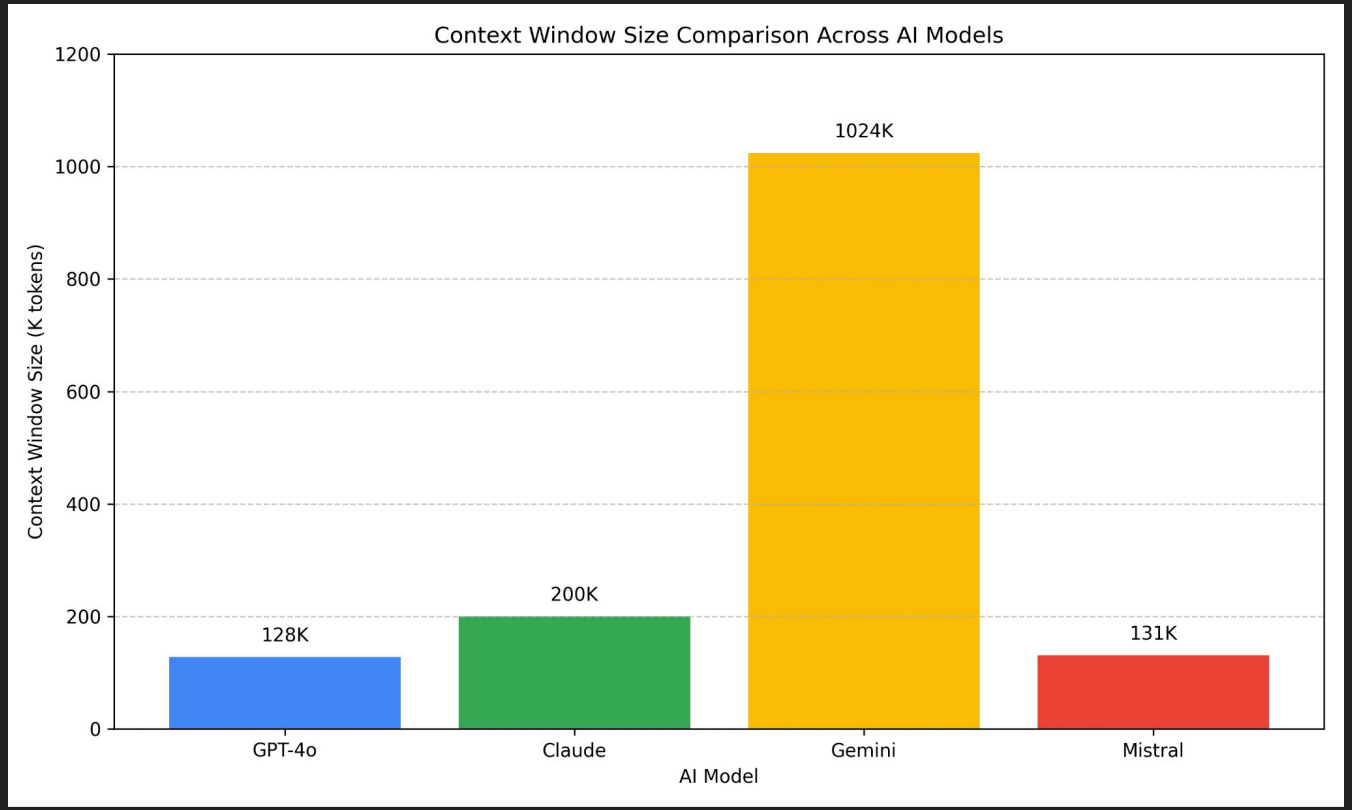
- Mistral: 131 K

Context window sizes

- Gemini 1.5 Flash: 1 million tokens



Context window sizes



Prompt stuffing

- Pros:
  - Works well for questions that require context
  - Really easy to do → no additional infrastructure needed
  - Tokens (on some models) are very cheap
  - Fast
  - Good for questions that require multiple sections of data

Prompt stuffing

- Cons:
  - Costs add up if you submit the same data repeatedly
  - Limited by context window size
  - Paying for unneeded tokens
  - Questions about performance degradation as size increases

Demo Problem

- Drake–Kendrick Lamar feud - Wikipedia
  - Discussion of the rap feud between Kendrick Lamar and Drake
  - Long history, which blew up in 2024

Drake–Kendrick Lamar feud - Wikipedia

en.wikipedia.org/wiki/Drake–Kendrick_Lamar_feud

WIKIPEDIA
The Free Encyclopedia

Search Wikipedia | Search

Donate | Create account | Log in

Drake–Kendrick Lamar feud

7 languages

Article | Talk | Read | Edit | View history | Tools

From Wikipedia, the free encyclopedia

Contents hide

(Top)
History
Commentary
Full song chronology
See also
Notes
References
External links

The Canadian rapper Drake and the American rapper Kendrick Lamar have been involved in a rap feud since 2013, when Drake responded to Lamar's verse on the Big Sean song "Control". It escalated in 2024 with Lamar's lyrics in the song "Like That".

The two began on favorable terms in 2011. On August 14, 2013, Lamar dissed Drake, among many rappers, on "Control", but claimed his verse was "friendly competition". Over the next decade, the two denied speculation that they had dissed each other on various songs. In 2023, on rapper J. Cole and Drake's song "First Person Shooter", Cole claimed that he, Drake, and Lamar were the "big three" of modern hip-hop; on "Like That" in March 2024, Lamar rejected the notion of a big three, saying the

Drake–Kendrick Lamar feud

Drake in 2016 | Lamar in 2018

Date **Full:**
September 23, 2013[1][2][note 1] – present
(11 years, 6 months, 1 week and 2 days)

Demo Problem

- Why?

Demo Problem

- Why?
  - You don't think I know things? I know things

Demo Problem

- Why?
  - I'm totally rizz, not mid at all. One might even say I am certified fresh

Demo Problem

- Why?
  - You really want me to continue? Just let me have this

Demo Problem

- Fine, whatever. But really, why?
  - Topic has a long Wikipedia page that is still growing
  - Involves events from both before and after the end of training for LLMs
  - Good example of document loaders, HTML extraction, and web capabilities

Download info

- Document loaders
- jsoup library
- Tavily

LangChain4j

- Java framework
- Competitor to Spring AI (both are good)
- "Universal" API for AI models
- Tool support / function calling
- Cool "ai services" support → implements interfaces for you
- Manages chat memory
- RAG support

UrlDocumentLoader

- Does what it says
  - Loads documents from a URL
  - No pre- or post-processing

jsoup library

- https://jsoup.org/
- Transitive dependency for LangChain4j
- Effective at extracting data from web pages
  - Never parse HTML. That way lies madness

Tavily

- https://tavily.com/
- Web search engine
- Extraction
- Designed to be invoked by LLMs

Estimating Tokens

- Costs are measured in tokens
  - Output tokens normally about 3x cost of input tokens
- Rule of thumb: 1000 tokens is about 750 words (English)
- See https://tiktokenizer.vercel.app/ as a simple demo

Managing costs

- Most LLMs provide support for caching
  - Reusing already stored tokens is much less expensive

Caching

OpenAI (GPT-4o)

- Offers prompt caching with a 50% discount on cached tokens
- Requires at least 1024 tokens to trigger caching
- No storage fees for cached content
- Cache hits are visible in the API response via the `cached_tokens` field

Caching

Anthropic (Claude)

- Provides prompt caching with approximately 90% discount (from $3/million to $0.3/million for cached tokens)
- Cache lasts for five minutes and resets with each hit
- No storage fees mentioned
- Particularly useful for multi-turn conversations

Caching

Google (Gemini)

- Offers context caching with a 75% discount on input costs
- Charges for cache storage ($1/million token-hours for Gemini 2.0 Flash)
- Minimum input token count of 32,768 for context caching
- Requires explicit TTL (time-to-live) setting

RAG

- Retrieval Augmented Generation
- Two stages

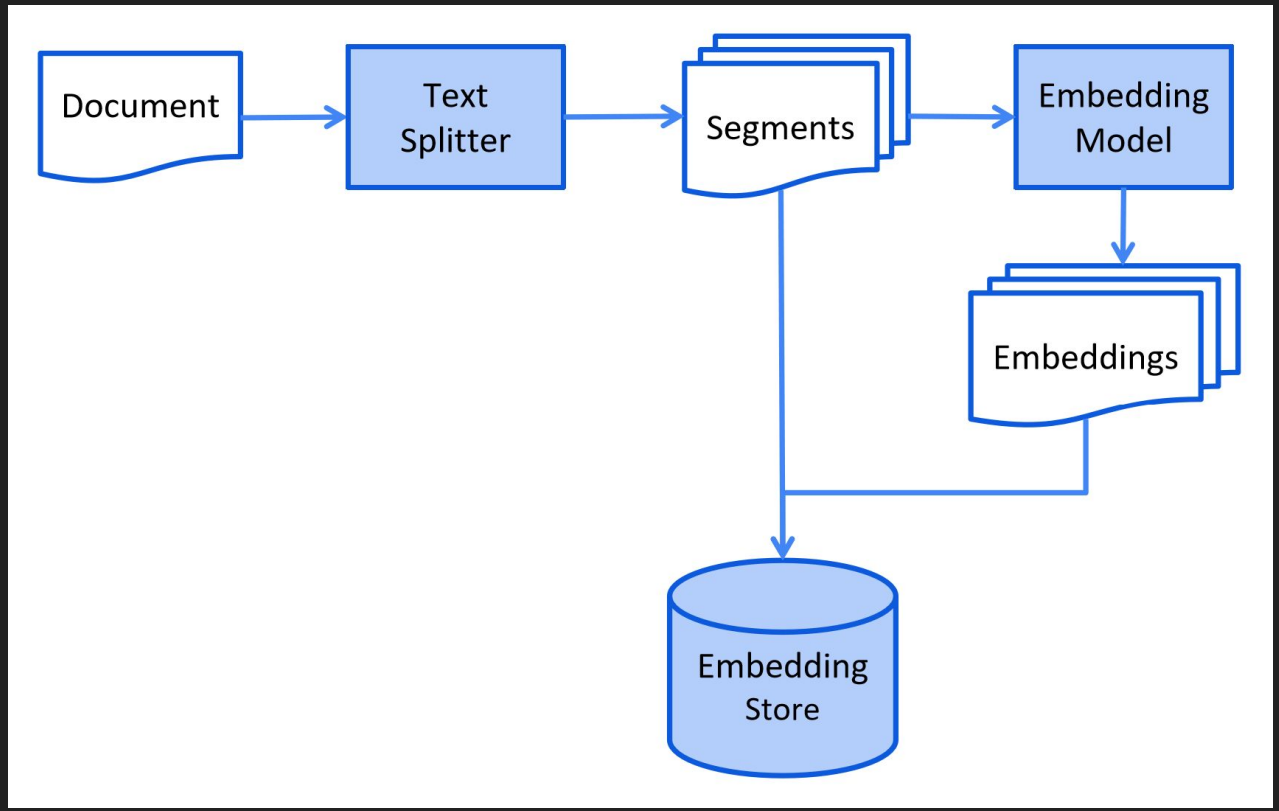
RAG

- First, populate embedding store
  - Load documents
  - Split information into chunks (segments)
  - Encode chunks as embeddable vectors
  - Store in a vector database

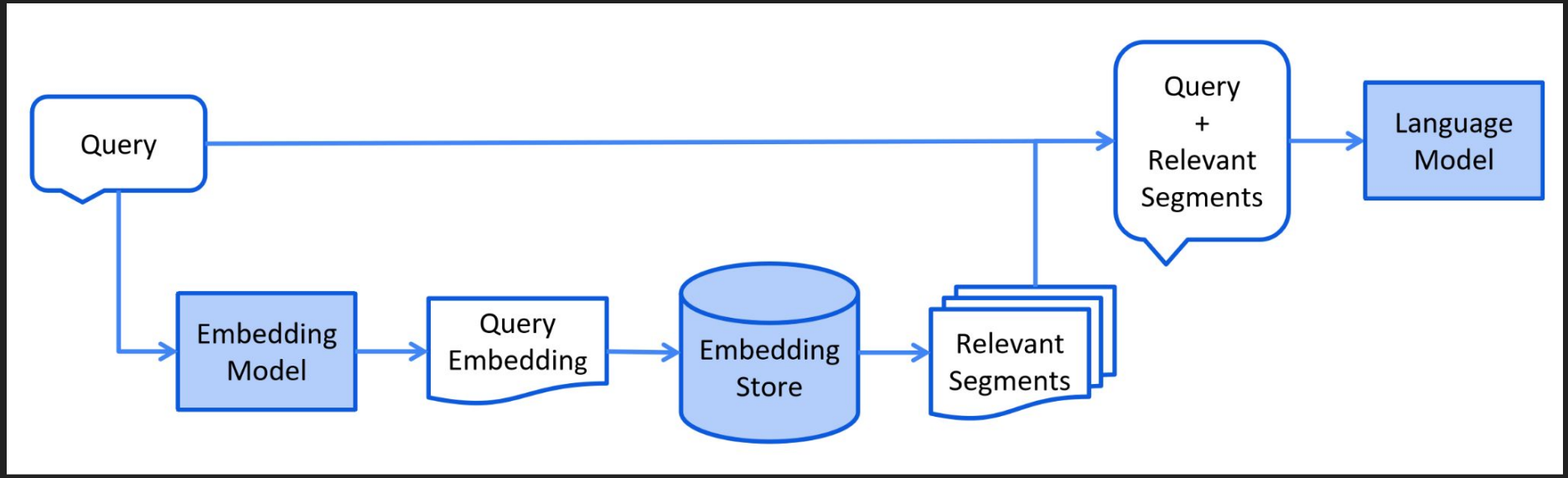
RAG

- Second, execute query
  - Parse and embed query
  - Use similarity search on vector database
  - Only returned chunks are added to context window
  - Profit!

RAG


Document
Text Splitter
Segments
Embedding Model
Embeddings
Embedding Store

RAG


Query
Query
+
Relevant
Segments
Language
Model
Embedding
Model
Query
Embedding
Embedding
Store
Relevant
Segments

RAG

- Pros:
  - Scales to much larger data sources
  - Only relevant information included
  - Updateable
- Cons:
  - More complex
  - Dependent on quality of similar search and embeddings
  - Loses overall context

Issues

- Updating information
  - Prompt stuffing requires new prompt
  - Can be done with templates
  - RAG requires processing new data and adding to storage
- Static vs dynamic information
- Misleading or incorrect information
- Prompt injection

I, for one, welcome our new AI overlords