From Matching to Generation: A Survey on Generative Information Retrieval (GenIR)

XIAOXI LI and **JIAJIE JIN**, Renmin University of China, China
YUJIA ZHOU, Tsinghua University, China
YUYAO ZHANG and **PEITIAN ZHANG**, Renmin University of China, China
YUTAO ZHU and **ZHICHENG DOU**∗, Renmin University of China, China

Table of contents

01

Introduction

You can describe the topic of the section here

02

Methods

You can describe the topic of the section here

03

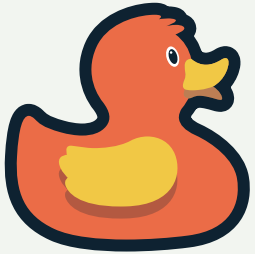
Experiments

You can describe the topic of the section here

04

Discussion

You can describe the topic of the section here



05

Conclusion

You can describe the topic of the section here

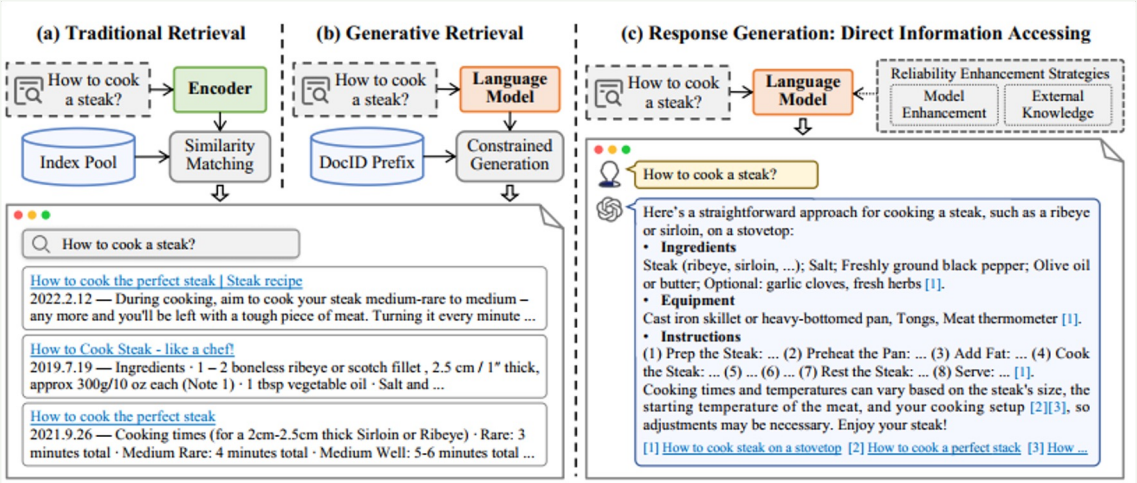
01

Introduction

The Evolution of Information Retrieval

The Old Way (Matching):

- Sparse Retrieval: Keyword matching (e.g., TF-IDF, BM25). Fast but lacks understanding.
- Dense Retrieval: Vector matching (e.g., BERT). Better semantic understanding but relies on a massive external index.



The New Way (Generative IR):

- Generative Retrieval (GR): The model is the index. It memorizes documents and generates their IDs.
- Response Generation: The model generates a direct, reliable answer with citations.

Limitations of Traditional IR

Storage Overhead

Traditional Information Retrieval relies heavily on large document indices, leading to significant storage overhead that complicates data management and increases operational costs for organizations.

User Intent

Capturing the nuanced **user intent** remains a challenge, as traditional methods often fail to understand the context of queries, resulting in less relevant search results and user dissatisfaction.

Manual Effort

Users frequently engage in manual effort to interpret and summarize retrieved documents, which can be time-consuming and inefficient, detracting from the overall effectiveness of information retrieval systems.

Traditional Retrieval vs User Struggle

- Traditional retrieval systems focus on matching and ranking documents, not on answering user questions directly.
- Users are often presented with a large number of loosely relevant results, increasing cognitive load.
- Finding the correct information requires manual scanning, reading, and comparison across multiple documents.
- Important details may be scattered across different sources, forcing users to synthesize information themselves.
- The retrieval process is time-consuming and inefficient, especially for complex or specific queries.
- Users experience frustration, information overload, and reduced search effectiveness.

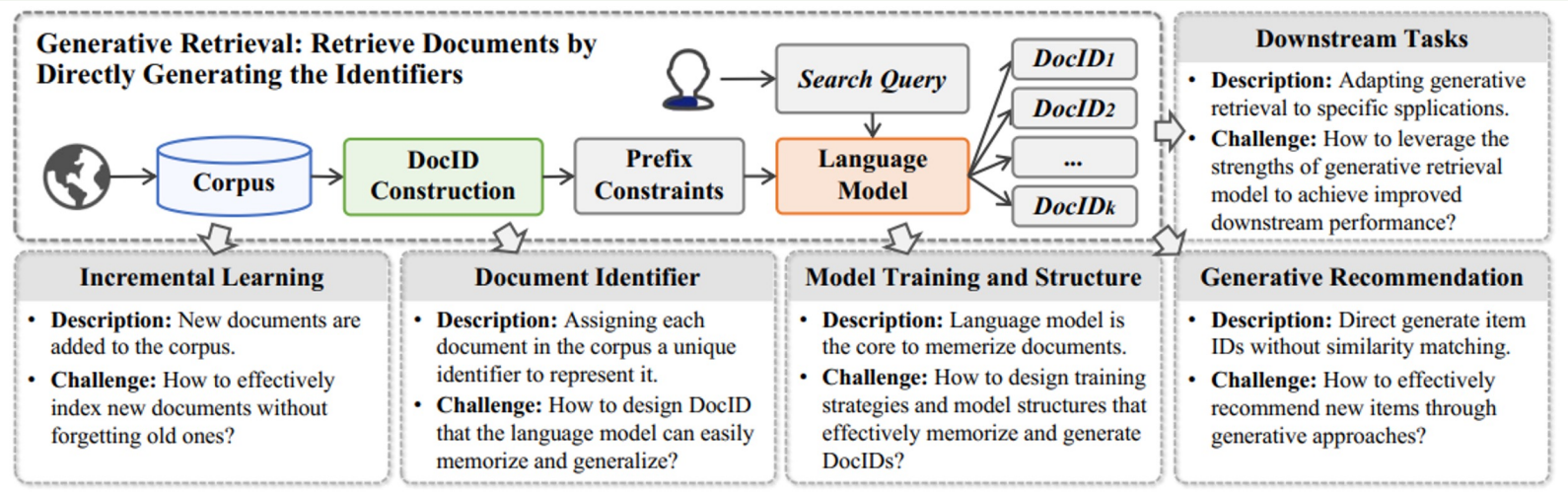
02

Methods

A. Generative Document Retrieval
B. Reliable Response Generation

A. Generative Document Retrieval (GR)

Core Concept: The model is the index. It "memorizes" documents during training.
The Process: Query → Model → Document Identifier (DocID).
Inference: Uses **Constrained Beam Search** to ensure the generated ID corresponds to a real document[3].
Advantage: End-to-end optimization; no need for a separate, massive external index.



Designing Document Identifiers (DocIDs)

Method	Document Identifier			Training Data Augmentation		Training Objective		
	State	Data Type	Order	Sample Doc	Doc2Query	Seq2seq	DocID	Ranking
GENRE [18]	Static	Text	Sequence	-	-	✓	-	-
DSI [281]	Static	Numeric	Sequence	✓	-	✓	-	-
DynamicRetriever [370]	Static	Numeric	Sequence	✓	-	✓	-	-
SEAL [13]	Static	Text	Sequence	✓	-	✓	-	-
DSI-QG [375]	Static	Numeric	Sequence	-	✓	✓	-	-
NCI [307]	Static	Numeric	Sequence	✓	✓	✓	-	-
Ultron [371]	Static	Numeric/Text	Sequence	✓	✓	✓	-	-
CorpusBrain [28]	Static	Text	Sequence	✓	-	✓	-	-
GenRet [265]	Learnable	Numeric	Sequence	-	✓	✓	✓	-
AutoTSG [352]	Static	Text	Set	-	✓	✓	-	-
SE-DSI [278]	Static	Text	Sequence	✓	-	✓	-	-
Chen et al. [30]	Static	Numeric	Sequence	✓	✓	✓	-	✓
LLM-URL [376]	Static	Text	Sequence	-	-	-	-	-
MINDER [160]	Static	Text	Sequence	-	✓	✓	-	-
LTRGR [159]	Static	Text	Sequence	-	✓	✓	-	✓
NOVO [311]	Learnable	Text	Set	✓	-	-	✓	-
GenRRL [365]	Static	Text	Sequence	-	✓	✓	-	✓
LMIndexer [112]	Learnable	Numeric	Sequence	-	✓	✓	✓	-
ASI [330]	Learnable	Numeric	Sequence	-	✓	✓	✓	-
RIPOR [344]	Learnable	Numeric	Sequence	-	✓	✓	✓	✓
GLEN [135]	Learnable	Text	Sequence	-	✓	✓	✓	✓
DGR [161]	Static	Text	Sequence	-	✓	✓	-	✓
ListGR [280]	Static	Numeric	Sequence	-	✓	✓	-	✓

Numeric-based IDs (Structure):

- **Atomic:** Random numbers. Hard to learn.
- **Semantic Clusters (e.g., DSI):** Documents are clustered (e.g., by topic). Similar docs get similar IDs (e.g., 1-2-3).
- **Product Quantization (e.g., Ultron, MEVI):** Compresses document vectors into short, discrete codes, preserving dense semantic information in the ID itself.

Text-based IDs (Semantics):

- **Titles (e.g., GENRE):** Uses the document title. Easy for LLMs to learn because it has meaning.
- **Term Sets (e.g., TSGen, NOVO):** Represents a document as a set of distinct keywords (e.g., { "Apple", "Recipe", "Pie" }). It removes the burden of predicting a specific order.
- **Sub-strings (e.g., SEAL):** Uses any unique N-gram found in the text, identified via an FM-Index, allowing for flexible retrieval.

Training Strategies & Incremental Learning

Advanced Training Strategies:

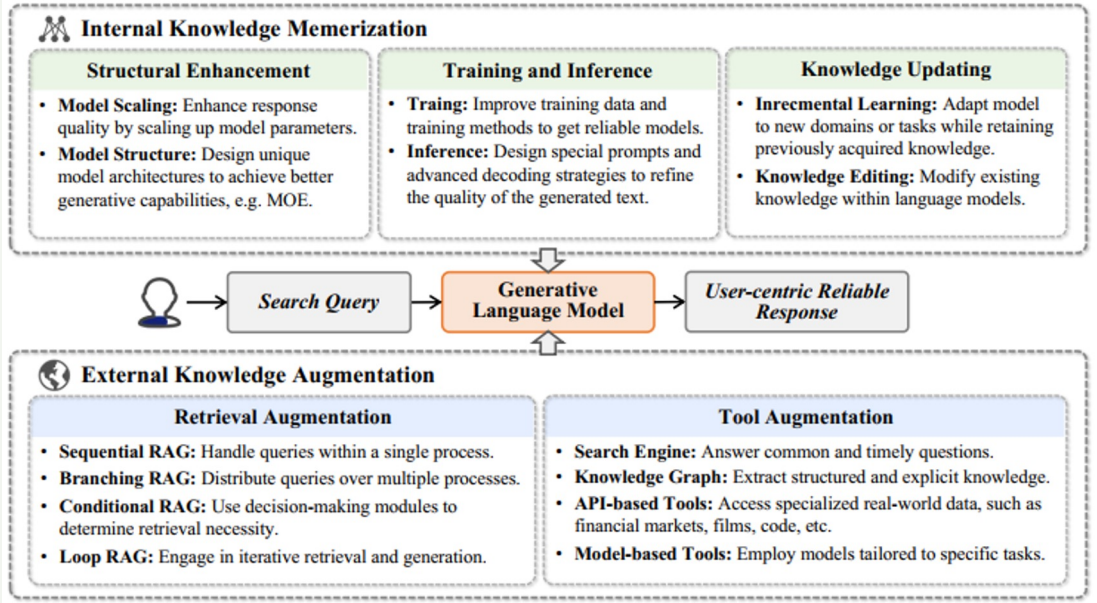
- **Data Augmentation (The "Doc2Query" Trick):** We lack user queries for most documents. We use a separate model to read documents and generate millions of "synthetic" queries. The retriever then trains on (Synthetic Query, DocID) pairs to ensure full corpus coverage.
- **Ranking Optimization:** Standard language models only learn to predict the correct token. Models like LTRGR and DGR introduce Ranking Loss (e.g., RankNet), teaching the model to differentiate between "relevant" and "highly relevant" IDs, optimizing the order of the final list.

The "New Data" Problem (Incremental Learning): When a neural network learns a new document, it often overwrites the weights used for old documents.

- **Solution 1 - Constrained Optimization (IncDSI):** Mathematically constrains the model updates so that the loss on old data does not increase while learning new data.
- **Solution 2 - Adapter Modules (CorpusBrain++):** Keeps the massive "backbone" model frozen and trains tiny, lightweight "Adapter Layers" to memorize new information. This is computationally cheap and prevents forgetting.
- **Solution 3 - Incremental Product Quantization (CLEVER):** Updates the quantization codebook dynamically to accommodate new semantic clusters without retraining.

B. Reliable Response Generation

The Goal: We want an AI that acts like a search engine, giving answers that are not just fluent, but true.
The Challenge: Standard LLMs suffer from "Hallucination." They are like students who try to answer every question from memory, even when they don't know the answer. They prioritize ***probability*** (what word sounds good next) over ***truth*** (what is factually correct).



Strategies for Reliable Response Generation

1. **Optimize Internal Knowledge (The "Brain")**
- **Instruction Tuning:** Training models to admit ignorance ("I don't know") rather than guessing.
- **Decoding Strategies (e.g., DoLa):** Contrasting early vs. late neural layers during generation to mathematically amplify factual signals and suppress hallucinations.
- **Knowledge Editing (e.g., ROME):** Surgically modifying specific model weights to update facts without expensive full re-training.
1. **Augment with External Knowledge (The "Tools")**
- **Active RAG (e.g., FLARE):** The model monitors its own confidence. If low, it pauses generation, searches Google dynamically, and then resumes writing.
- **Tool Learning (e.g., Toolformer):** The model teaches itself to call APIs (Calculators, Calendars, Code Interpreters) when text prediction is insufficient.
1. **Enforce Verification (The "Proof")**
- **In-Process Citation:** Training the model to generate [Source: 1] markers while writing.
- **Post-Hoc Verification (e.g., RARR):** A separate "Editor" model reviews the generated draft, searches for evidence, removes unsupported claims, and inserts citations retroactively.

03

Experiments

Evaluation Metrics

For Finding Documents (Generative Retrieval):

- Recall@k: Percentage of relevant documents found in the top k results.
- MRR (Mean Reciprocal Rank): How high up the list is the first correct result?.
- nDCG: Measures ranking quality; rewards correct results appearing earlier.

For Generating Answers (Response Generation):

- **Rule-Based:**
  - **BLEU / ROUGE:** Measures word overlap with reference answers.
  - **Exact Match (EM):** Used for short, precise answers.
- **Model-Based (Advanced):**
  - **FActScore:** Decomposes answers into atomic facts to verify truthfulness.
  - **BERTScore:** Checks semantic meaning rather than just keywords.
- **Human Evaluation:** Accuracy, Relevance, Fluency, and Safety .

Experimental Results - GR Performance

- **Comparison:** GR models vs. Dense Retrieval (DPR) vs. Sparse (BM25).
- **Key Result 1 (Small Corpus):** On **Natural Questions (NQ)**, GR models like **TSGen** achieve **Recall@1 of 70.8%**, vastly outperforming Dense Retrieval (50.2%).
- **Key Result 2 (Large Corpus):** On **MS MARCO (8.8M docs)**, GR models struggle to memorize everything. Scale is a major bottleneck.
- **Best Methods: Term-Set** and **Multi-View** identifiers (combining titles + keywords) perform best.

Model	Doc Rep.	MS MARCO					Natural Questions (NQ)				
		R@1	R@10	R@100	M@10	M@100	R@1	R@10	R@100	M@10	M@100
Sparse&Dense Retrieval											
BM25 [265]	Bag-of-words	0.196	0.591	0.861	0.313	0.325	0.297	0.603	0.821	-	0.402
SPLADEv2 [352]	Bag-of-words	0.328	0.779	0.956	0.443	0.452	0.624	0.873	0.954	0.726	0.731
DPR [265]	Dense Vector	0.271	0.764	0.948	0.424	0.433	0.502	0.777	0.909	-	0.489
GTR-Base [265]	Dense Vector	0.332	**0.793**	**0.960**	0.484	0.485	0.560	0.844	0.937	-	0.662
Generative Retrieval											
GENRE [352]	Title	0.266	0.579	0.751	0.361	0.368	0.591	0.756	0.814	0.653	0.656
DSI [352]	Semantic ID	0.257	0.538	0.692	0.339	0.346	0.533	0.715	0.816	0.594	0.598
DSI-QG [265, 371]	Semantic ID	0.288	0.623	-	0.385	-	0.631	0.807	0.880	-	0.695
NCI [265]	Semantic ID	0.301	0.643	0.851	0.408	-	0.659	0.852	0.924	-	0.731
SEAL [265]	Sub-string	0.259	0.686	0.879	0.393	0.402	0.570	0.800	0.914	-	0.655
Ultron [352]	Title+URL	0.304	0.676	0.794	0.432	0.437	0.654	0.854	0.911	0.726	0.729
GenRet [265]	Learnable	-	-	-	-	-	0.681	0.888	0.952	-	0.759
MINDER [352]	Multi-view	0.289	0.728	0.916	0.431	0.435	0.627	0.869	0.933	0.709	0.713
LTRGR*	Multi-view	0.327	0.759	0.929	0.463	0.469	0.644	0.879	0.941	0.721	0.726
GLEN [135]	Learnable	-	-	-	-	-	0.691	0.860	-	-	0.754
TSGen [352]	Term Set	**0.384**	0.781	0.931	**0.502**	**0.505**	**0.708**	0.889	0.948	**0.771**	**0.774**
NOVO [311]	Term Set	-	-	-	-	-	0.693	**0.897**	**0.959**	-	0.767
DGR*	Multi-view	0.359	0.779	0.934	0.498	0.504	0.682	0.887	0.949	0.759	0.764

Experimental Results - KILT Benchmark

KILT: A massive Wikipedia-based benchmark (5.9M documents).
Winner: GENRE (Title-based).

- **Score:** 0.847 on FEVER (Fact Checking).
- **Reason:** Wikipedia titles are unique and semantic, making them easy for LLMs to generate.

Failure Case: Numeric IDs (like DSI) score near 0.0. Because models cannot memorize 5.9 million random numbers without semantic meaning.

Model	Doc Rep.	FC	Entity Linking			Slot Filling		Open Domain QA				Dial.
		FEVER	AY2	WnWi	WnCw	TREx	zsRE	NQ	HoPo	TQA	ELI5	WoW
Sparse&Dense Retrieval												
BM25 [185]	Bag-of-words	0.501	0.035	-	-	0.586	0.664	0.258	0.440	0.294	-	0.275
RAG [218]	Dense Vector	0.635	0.774	0.490	0.467	0.293	0.654	0.603	308	0.493	0.104	0.467
MT-DPR [185]	Dense Vector	0.747	0.838	-	-	0.692	0.772	0.615	0.442	0.620	-	0.397
Generative Retrieval												
BART*	Semantic ID	0.003	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
BART [28]	Title	0.819	0.892	0.676	0.623	0.752	0.911	0.586	0.487	0.676	0.121	0.510
T5 [218]	Title	-	0.866	0.474	0.465	-	-	-	-	-	-	-
GENRE [18]	Title	**0.847**	**0.928**	**0.877**	**0.706**	**0.797**	0.948	0.643	0.518	0.711	**0.135**	**0.563**
SEAL*	Sub-string	0.826	0.866	0.809	0.651	0.704	0.919	**0.658**	**0.565**	**0.715**	0.124	0.527
CorpusBrain [28]	Title	0.821	0.908	0.723	0.662	0.776	**0.983**	0.591	0.501	0.688	0.129	0.538

04

Discussion

Key Challenges

Scalability

Memorizing the entire web is impossible for GR models; they replace cheap storage with costly RAM.

Dynamic Updates

Instantaneously adding new knowledge without full retraining remains difficult.

Inference Latency

Generating a DocID token-by-token is slower than a simple database lookup.

Trust & Safety

Hallucination and Bias remain the primary blockers for enterprise adoption.

Future Directions

01

Integrated Systems

Future research should focus on creating **integrated systems** that seamlessly combine Generative Document Retrieval and Reliable Response Generation for improved user interaction and efficiency.

02

Enhanced User Experience

Developing user-centric designs is essential for ensuring that the **information retrieval process** is intuitive, allowing users to easily navigate through generated content and responses.

03

Addressing Challenges

Ongoing efforts must tackle challenges such as **scalability and accuracy** to ensure generative models are reliable, efficient, and can be applied effectively across various domains.

05

Conclusion

From Matching to Generation, We are witnessing the fundamental evolution of search. We are moving away from systems that simply match keywords against a database and toward intelligent agents that generate knowledge.

- **Generative Retrieval:** The model ***is*** the index.
- **Reliable Response:** The answer is verified, not just retrieved.

The future is not a pipeline of separate tools, but a Unified End-to-End Framework. We envision a single AI model that can index the web, retrieve evidence, and generate trustworthy, cited answers in one seamless process.

Meme Retrieval

Lê Hoàng Ân, Hoàng Tuấn Khoa, Trần Đức An, Đỗ Phan Tuấn Phát

What we will present

01

The goal of the app

02

Practical use case and intuition

03

General strategies

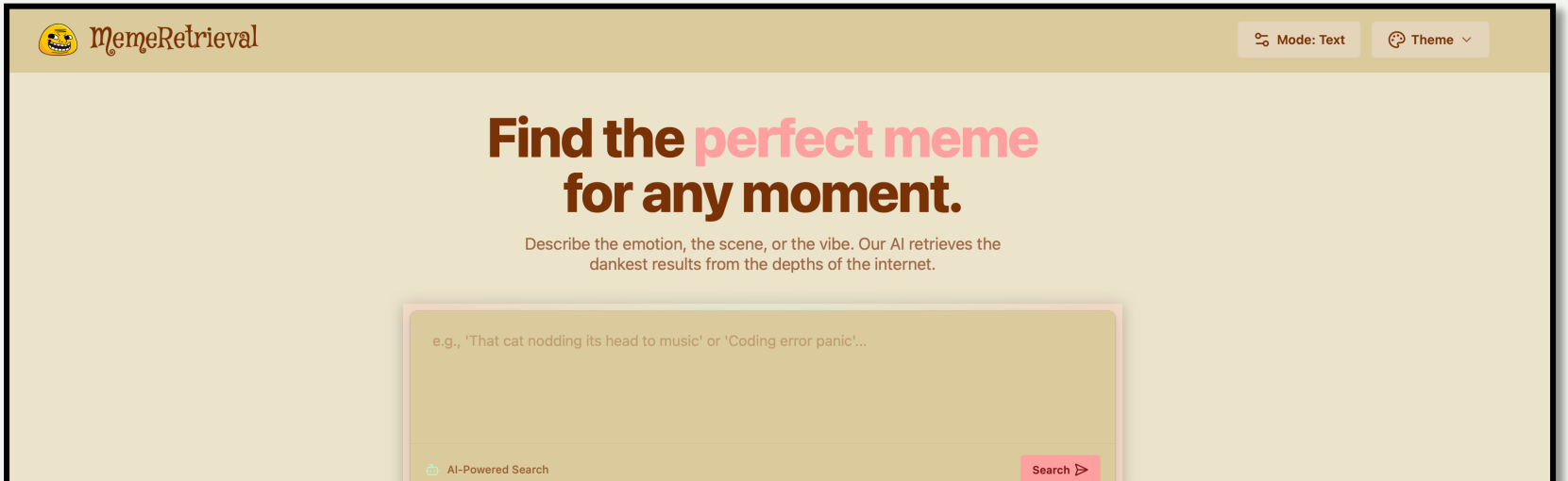
04

General implementation

01

Goal

We want to build an app that lets people find memes (using retrieval techniques of course)



02

Use case and user intuition

How users want to search for that specific memes.

They may input some text or paste an image

1. They may search for captions or describe the meme

"Peter Griffin racist meme"

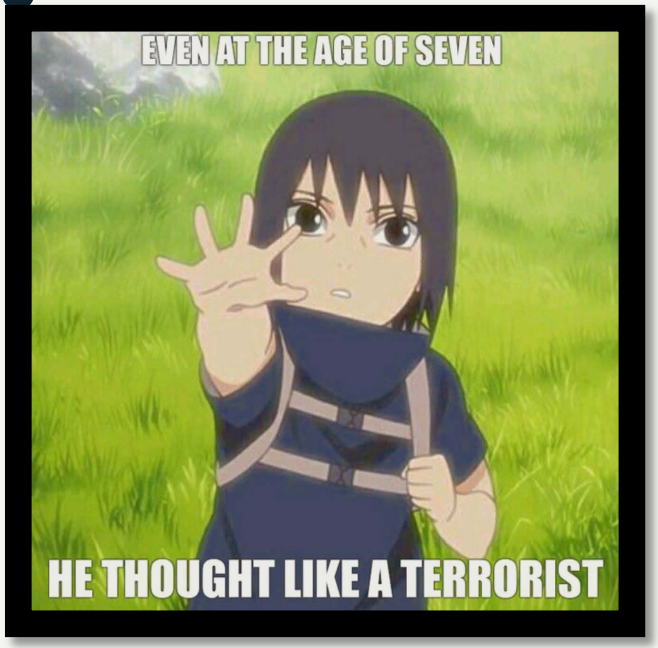
"Peter Griffin ok not ok"

2. They may paste a picture of a character in the meme

03

General strategies

How we solve the retrieval problem



EVEN AT THE AGE OF SEVEN HE THOUGHT LIKE A TERRORIST

MEME OF CUTE ITACHI RAISING HIS HAND

EVEN AT THE AGE OF SEVEN HE THOUGHT LIKE A TERRORIST

Elastic Search

Text search database

We will use elastic search to find similar meme captions

MEME OF CUTE ITACHI RAISING HIS HAND

Vector Database for Embeddings

Vector search database

We will process and embed the query, then search for memes matching description

"MEME OF A CUTE PERSON RAISING HIS/HER HAND"

Text removed

FACE SEARCH

Vector Database for Embeddings

Chroma database

We use chroma (FAISS) to search for memes that contain similar faces.

04

Implementation

Technical implementation

Use case 1: Text search

"A two-panel meme featuring characters from Game of Thrones.
Top panel: Sansa Stark looking serious, with text saying "Only a fool would trust Littlefinger."
Bottom panel: Ned Stark and Catelyn Stark standing together, smiling broadly and looking at each other."

GliNER

Qwen2.5-1.5B-Instruct

WIKIDATA.ORG

Sansa Stark - Sophie Turner
Ned Stark - Sean Bean
Catelyn Stark - Michelle Fairley

Use case 1: Text search

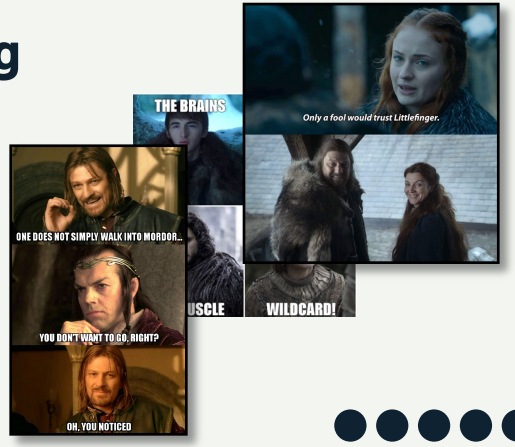
GliNER → Qwen2.5-1.5B-Instruct

ORG → Sansa Stark - Sophie Turner
Ned Stark - Sean Bean
Catelyn Stark - Michelle Fairley

Get face images of those celebrities
(postgres+elastic search)

ml-mobileclip

→ **Chroma database (memes containing extracted faces)**



Use case 1: Text search

Qwen2.5-1.5B-Instruct

Prompting*

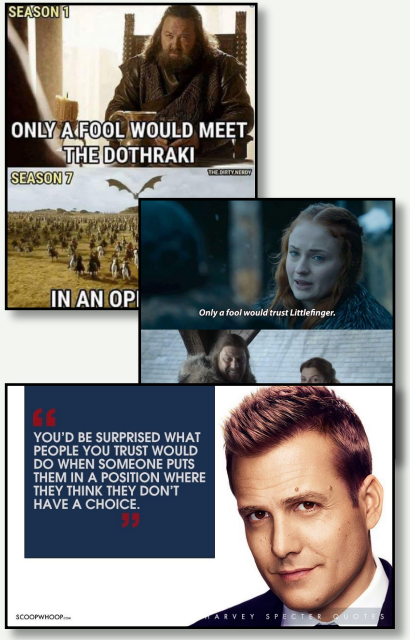
caption: Only a fool would trust Littlefinger."

descriptive text: A person looking serious, with text saying \"Only a fool would trust Littlefinger.\" A person standing together, smiling broadly and looking at each other.

Use case 1: Text search

caption: Only a fool would trust Littlefinger."

Elastic search database (memes containing those captions)



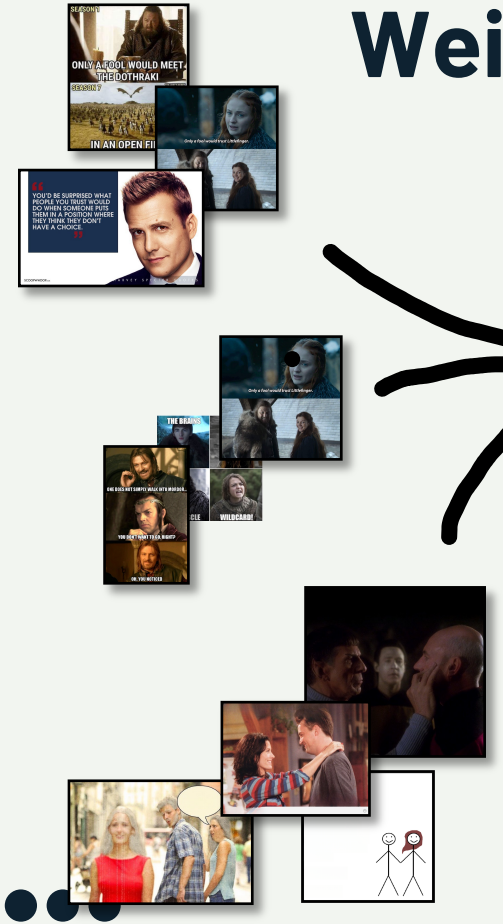
Use case 1: Text search

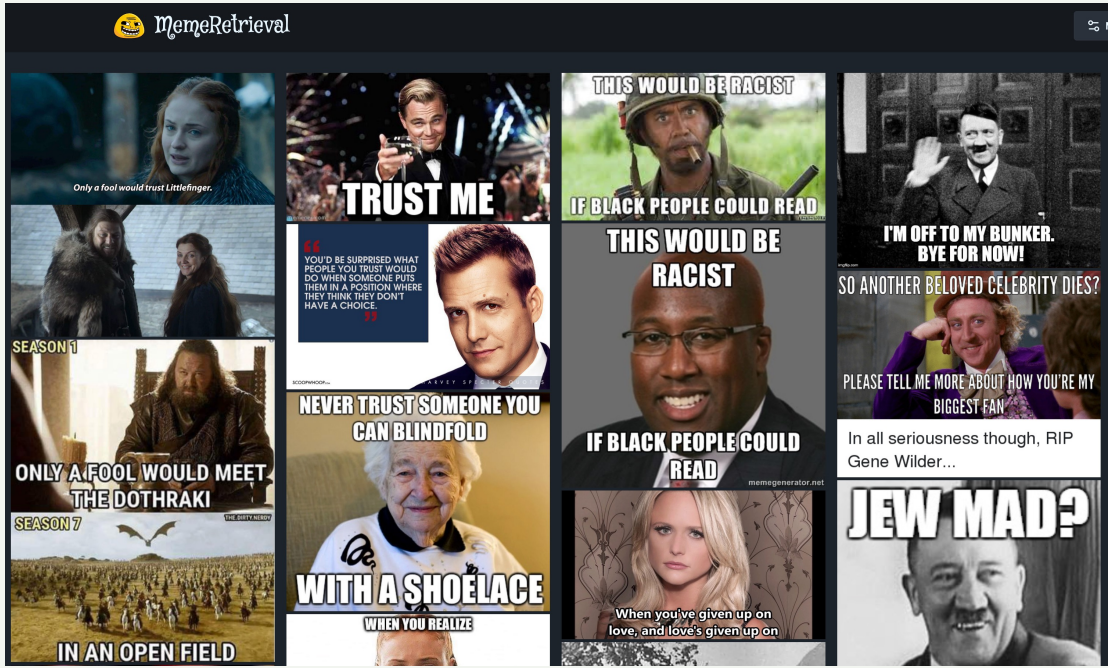
descriptive text: A person looking serious, with text saying \"Only a fool would trust .\" A person standing together, smiling broadly and looking at each other.

ml-mobileclip

weaviate

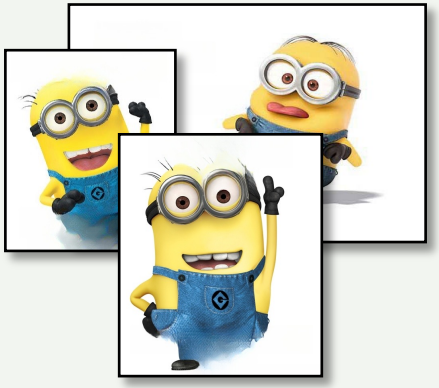
Weighted sum of scores





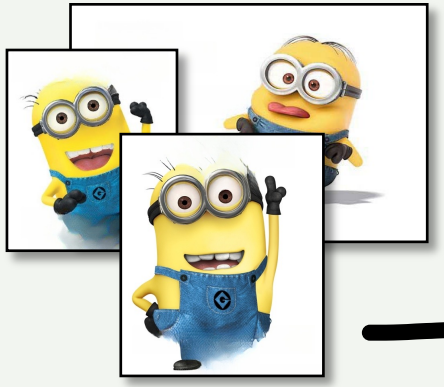
Use case 2: Image search

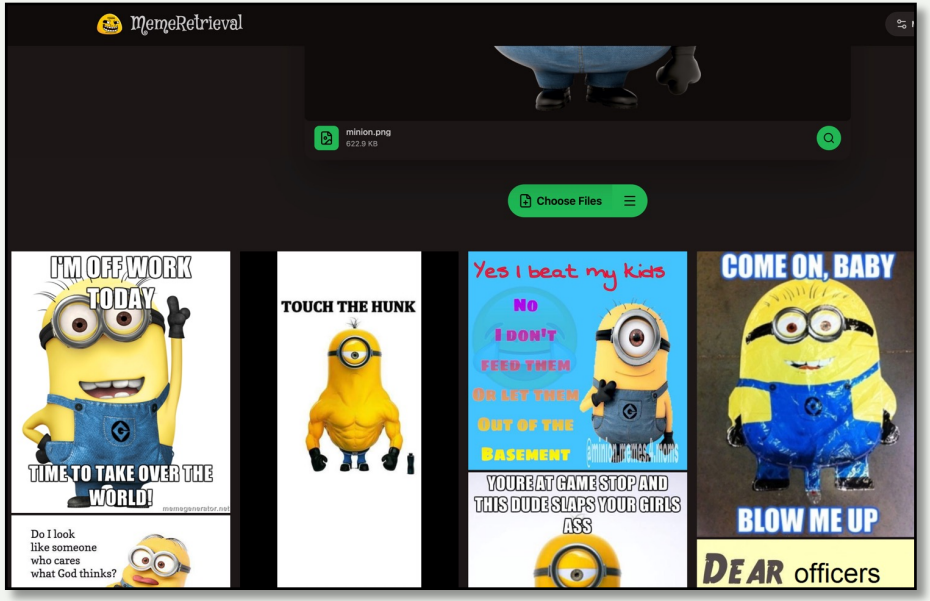
Weaviate (no text)



Weaviate (text)

Use case 2: Image search





THANKS FOR LISTENING