CS6216 Advanced Topics in Machine Learning (Systems)

Application systems: RAGs, Vector DBs & AI Agents

Yao LU

8 Oct 2024

National University of Singapore
School of Computing

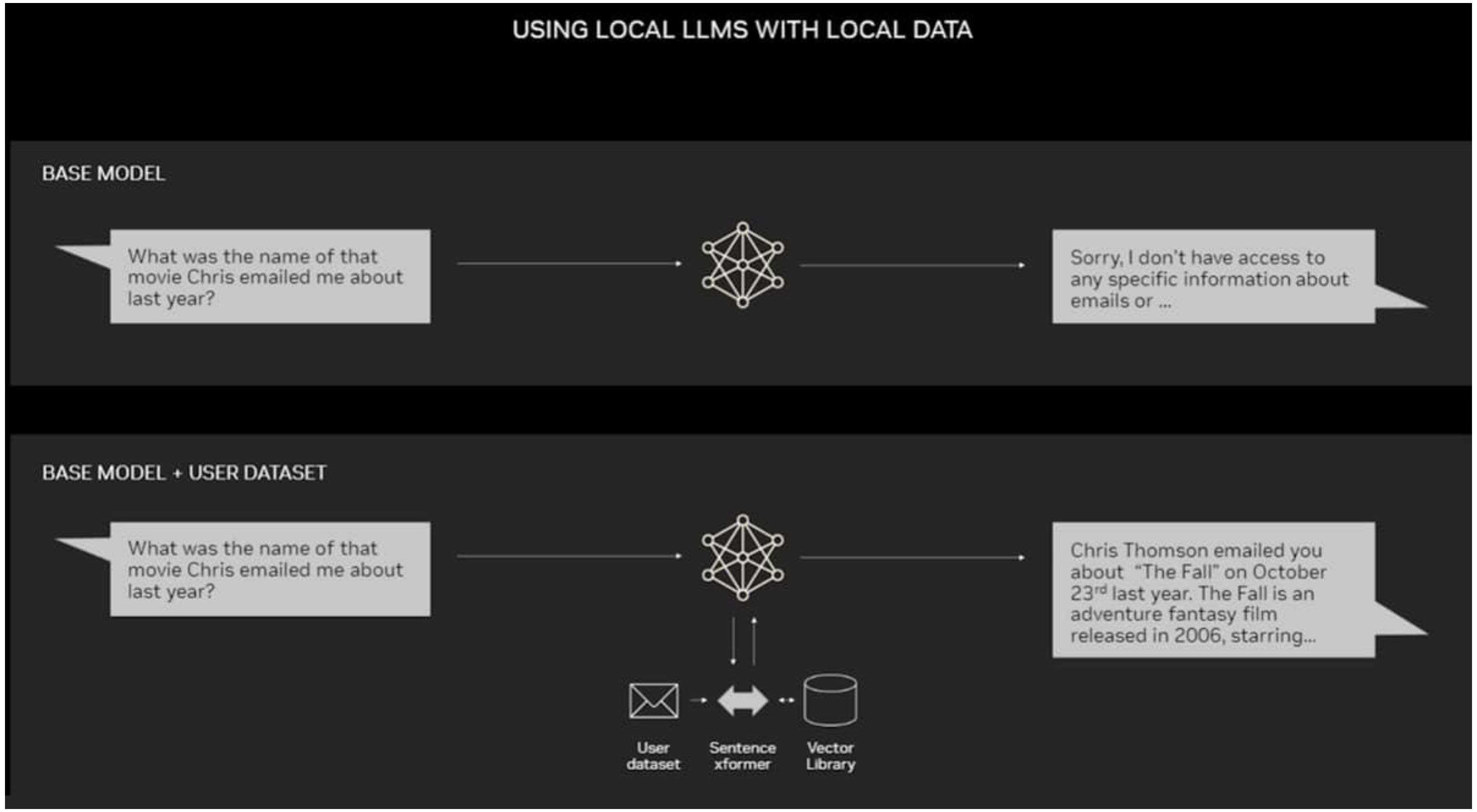
Application systems: outline

- Retrieval Augmented Generation (RAG)
- Vector DBs
- AI agents

Retrieval Augmented Generation (RAG)

Directly using LLMs faces problems

- Information lag
- Model hallucination
- Hard to incorporate proprietary data



Retrieval Augmented Generation (RAG)

LangChain

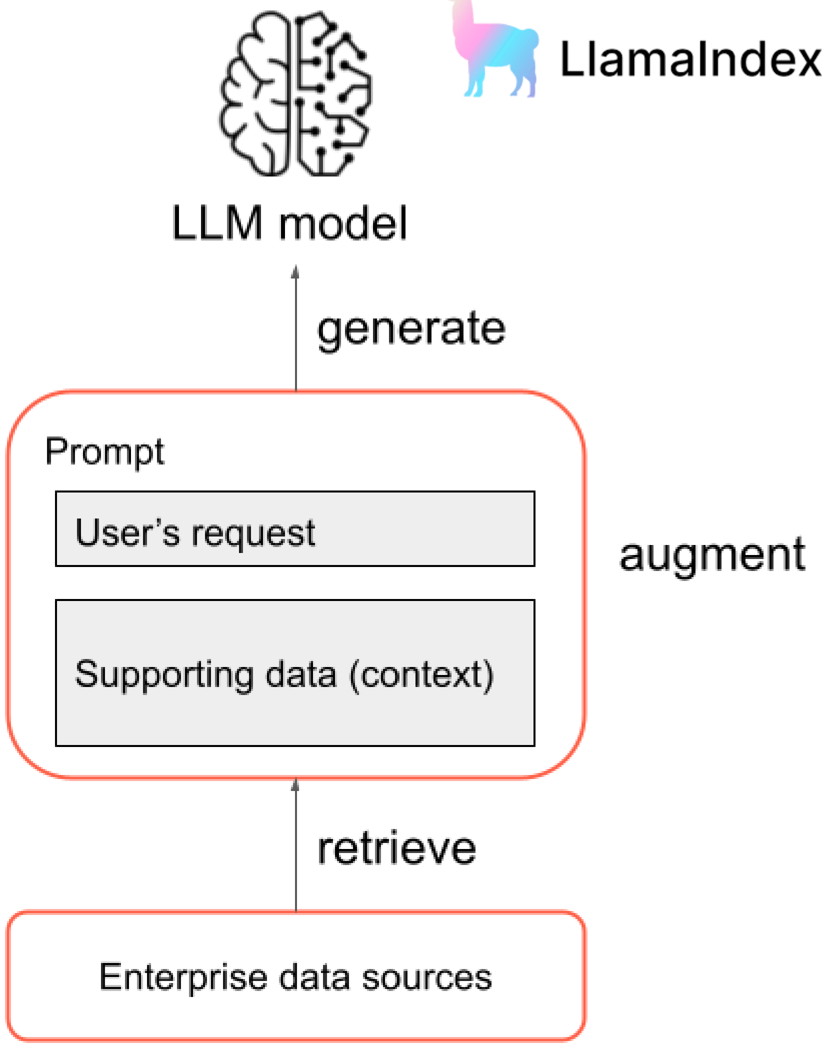
LlamaIndex

Directly using LLMs faces problems

- Information lag
- Model hallucination
- Hard to incorporate proprietary data

Instead, we need RAG =

- **Retrieval**: The user's request is used to query some external info - querying a vector store, a keyword search over text, or querying a database. This is to obtain supporting data / context that helps the LLM provide a useful response.
- **Augmentation**: The supporting data / context is combined with the user request, often using a template with instructions to the LLM, to create a prompt.
- **Generation**: The LLM generates a response to the prompt.



With an LLM alone	Using LLMs with RAG
No proprietary knowledge: LLMs are generally trained on publicly available data, so they cannot accurately answer questions about a company's internal or proprietary data.	**RAG applications can incorporate proprietary data:** A RAG application can supply proprietary documents such as memos, emails, and design documents to an LLM, enabling it to answer questions about those documents.
Knowledge isn't updated in real time: LLMs do not have access to information about events that occurred after they were trained. For example, a standalone LLM cannot tell you anything about stock movements today.	**RAG applications can access real-time data:** A RAG application can supply the LLM with timely information from an updated data source, allowing it to provide useful answers about events past its training cutoff date.
Lack of citations: LLMs cannot cite specific sources of information when responding, leaving the user unable to verify whether the response is factually correct or a hallucination.	**RAG can cite sources:** When used as part of a RAG application, an LLM can be asked to cite its sources.
Lack of data access controls (ACLs): LLMs alone can't reliably provide different answers to different users based on specific user permissions.	**RAG allows for data security/ACLs:** The retrieval step can be designed to find only the information that the user has credentials to access, enabling a RAG application to selectively retrieve personal or proprietary information.

RAG workflow

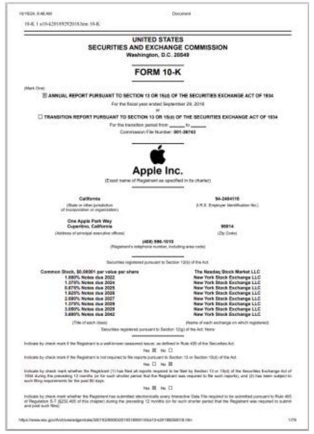
(Offline) Preprocess

- Chunking documents with simple heuristics (1)
- Compute embeddings w/ a pre-trained model (2)
- Indexing & store the embeddings in a database (2)

(Online) When a user query comes

- Compute embedding for the user query (3)
- Retrieve relevant embeddings from the database (4)
- Assemble a prompt, send it to LLM for result (5-7)

Example: Ask “How many employees?” to an SEC filing



~100 pages, tables, text

“Retrieved” context from the document:

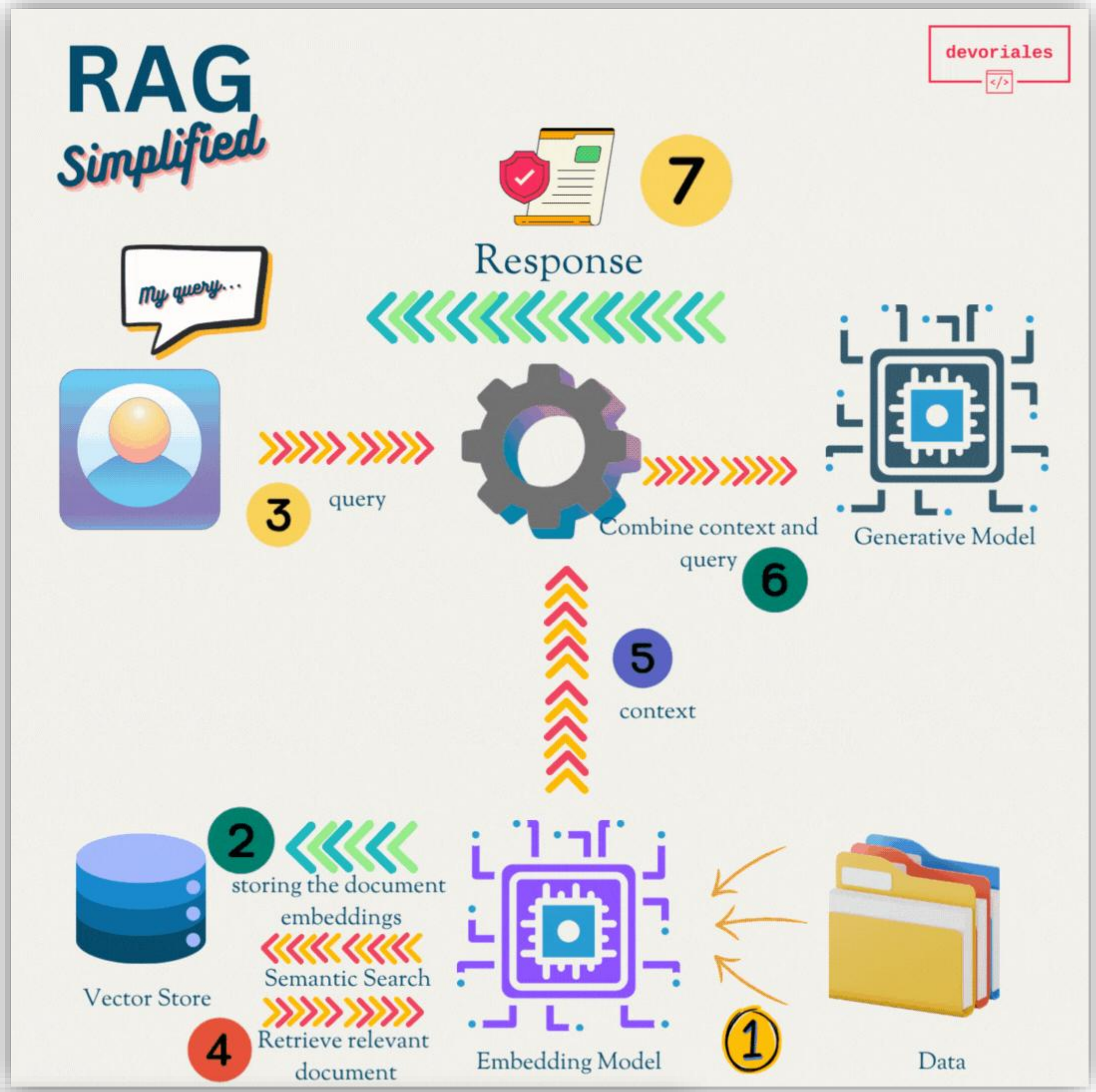
Backlog

In the Company’s experience, the actual amount of product backlog at any particular time is not a meaningful indication of its future business prospects. In particular, backlog often increases immediately following new product introductions as customers anticipate shortages. Backlog is often reduced once customers believe they can obtain sufficient supply. Because of the foregoing, backlog should not be considered a reliable indicator of the Company’s ability to achieve any particular level of revenue or financial performance.

Employees

As of September 29, 2018, the Company had approximately 132,000 full-time equivalent employees.

Apple Inc. | 2018 Form 10-K | 6



Credits: devoriales.com

Drawbacks of RAG

- **What if retrieval goes wrong?**
  - Raw documents are highly nonstructured
  - Documents are too long
  - Complex retrieval
  - Ranking is wrong
- **What if generation goes wrong?**
  - Prompt is too complex / long
  - Generation doesn't follow instruction / format requirement

Note 3 – Financial Instruments

Cash, Cash Equivalents and Marketable Securities

The following tables show the Company's cash, cash equivalents and marketable securities by significant investment category as of December 31, 2022 and September 24, 2022 (in millions):

	December 31, 2022						
	Adjusted Cost	Unrealized Gains	Unrealized Losses	Fair Value	Cash and Cash Equivalents	Current Marketable Securities	Non-Current Marketable Securities
Cash	$ 17,908	$ —	$ —	$ 17,908	$ 17,908	$ —	$ —
Level 1 (1):							
Money market funds	818	—	—	818	818	—	—
Mutual funds	330	2	(40)	292	—	292	—
Subtotal	1,148	2	(40)	1,110	818	292	—
Level 2 (2):							
U.S. Treasury securities	24,128	1	(1,576)	22,553	13	9,105	13,435
U.S. agency securities	5,743	—	(643)	5,100	—	310	4,790
Non-U.S. government securities	17,778	14	(1,029)	16,763	—	9,907	6,856
Certificates of deposit and time deposits	2,025	—	—	2,025	1,795	230	—
Commercial paper	237	—	—	237	—	237	—
Corporate debt securities	85,895	14	(7,039)	78,870	1	10,377	68,492
Municipal securities	864	—	(26)	838	—	278	560
Mortgage- and asset-backed securities	22,448	3	(2,405)	20,046	—	84	19,962
Subtotal	159,118	32	(12,718)	146,432	1,809	30,528	114,095
Total (3)	$ 178,174	$ 34	$ (12,758)	$ 165,450	$ 20,535	$ 30,820	$ 114,095

	September 24, 2022						
	Adjusted Cost	Unrealized Gains	Unrealized Losses	Fair Value	Cash and Cash Equivalents	Current Marketable Securities	Non-Current Marketable Securities
Cash	$ 18,546	$ —	$ —	$ 18,546	$ 18,546	$ —	$ —
Level 1 (1):							
Money market funds	2,929	—	—	2,929	2,929	—	—
Mutual funds	274	—	(47)	227	—	227	—
Subtotal	3,203	—	(47)	3,156	2,929	227	—
Level 2 (2):							
U.S. Treasury securities	25,134	—	(1,725)	23,409	338	5,091	17,980
U.S. agency securities	5,823	—	(655)	5,168	—	240	4,928
Non-U.S. government securities	16,948	2	(1,201)	15,749	—	8,806	6,943
Certificates of deposit and time deposits	2,067	—	—	2,067	1,805	262	—
Commercial paper	718	—	—	718	28	690	—
Corporate debt securities	87,148	9	(7,707)	79,450	—	9,023	70,427
Municipal securities	921	—	(35)	886	—	266	620
Mortgage- and asset-backed securities	22,553	—	(2,593)	19,960	—	53	19,907
Subtotal	161,312	11	(13,916)	147,407	2,171	24,431	120,805
Total (3)	$ 183,061	$ 11	$ (13,963)	$ 169,109	$ 23,646	$ 24,658	$ 120,805

(1) Level 1 fair value estimates are based on quoted prices in active markets for identical assets or liabilities.

(2) Level 2 fair value estimates are based on observable inputs other than quoted prices in active markets for identical assets and liabilities, quoted prices for identical or similar assets or liabilities in inactive markets, or other inputs that are observable or can be corroborated by observable market data for substantially the full term of the assets or liabilities.

(3) As of December 31, 2022 and September 24, 2022, total marketable securities included $13.6 billion and $12.7 billion, respectively, that were restricted from general use, related to the European Commission decision finding that Ireland granted state aid to the Company, and other agreements.

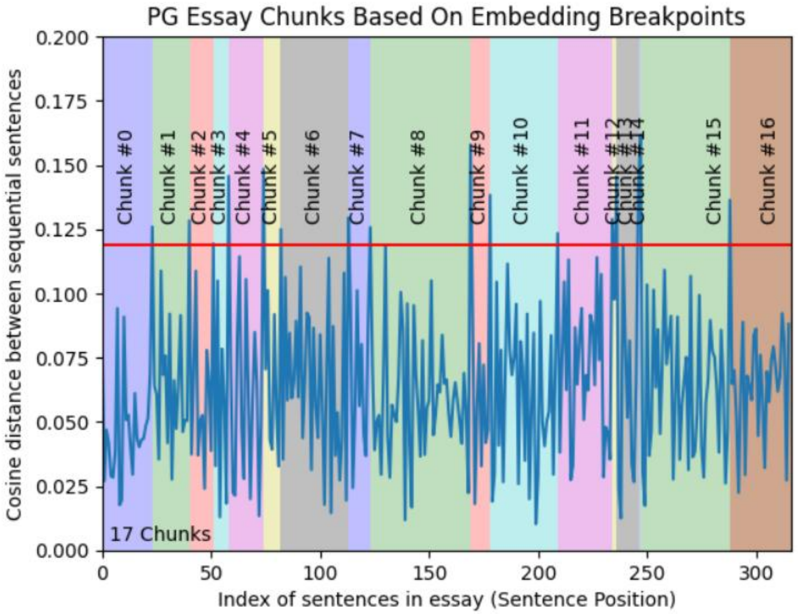
Looking back on the info retrieval literature

Many IR techniques can be applied to RAG

- Better chunking mechanisms
- Prompt compression
- Learning to rank / re-ranking
- Model selection, finetuning & distillation
- Multi-way retrieval
- Graph RAG

Better chunking mechanisms

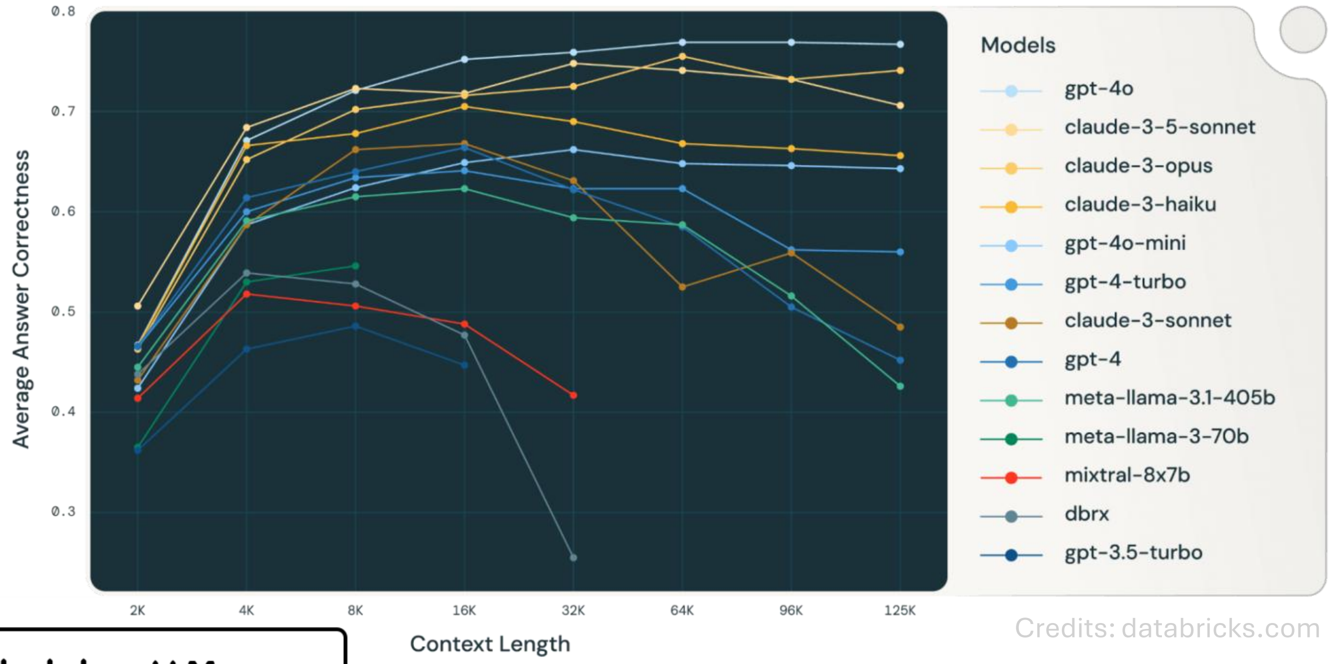
- Besides the simple fix-length chunking, there are many other ways:
  - **Overlapping windows** to make sure information is captured in some windows
  - **Structure-aware chunking** to avoid breaking in the middle of paragraphs and sentences
  - **Document based chunking** that leverages the document property (Markdown, HTML, LaTeX etc.)
  - **NLP/Semantic chunking** to detect topic changes
  - **Agentic chucking** uses AI agents to decide if a sentence should be added to the previous chunk.



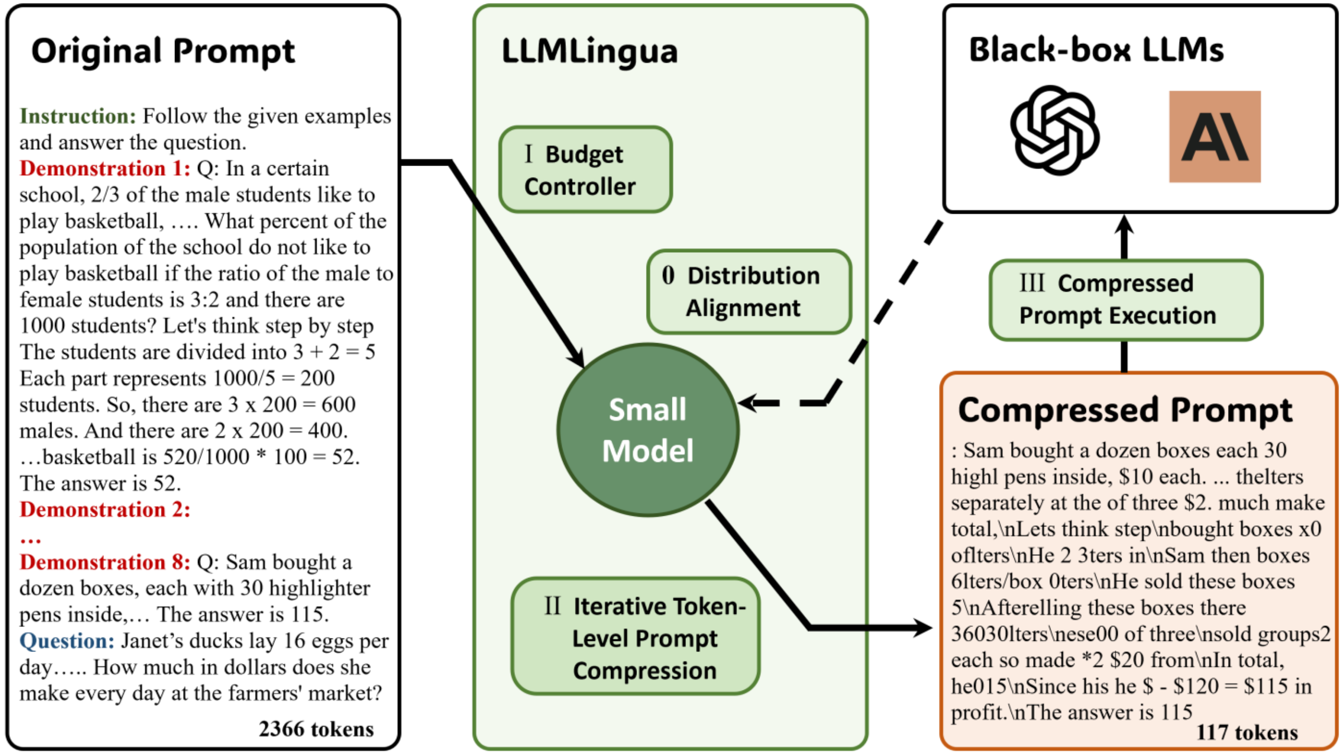
Prompt compression

- More context = more accurate (at cost)

- LLMLingua EMNLP 2023 (Instruction tuning!)

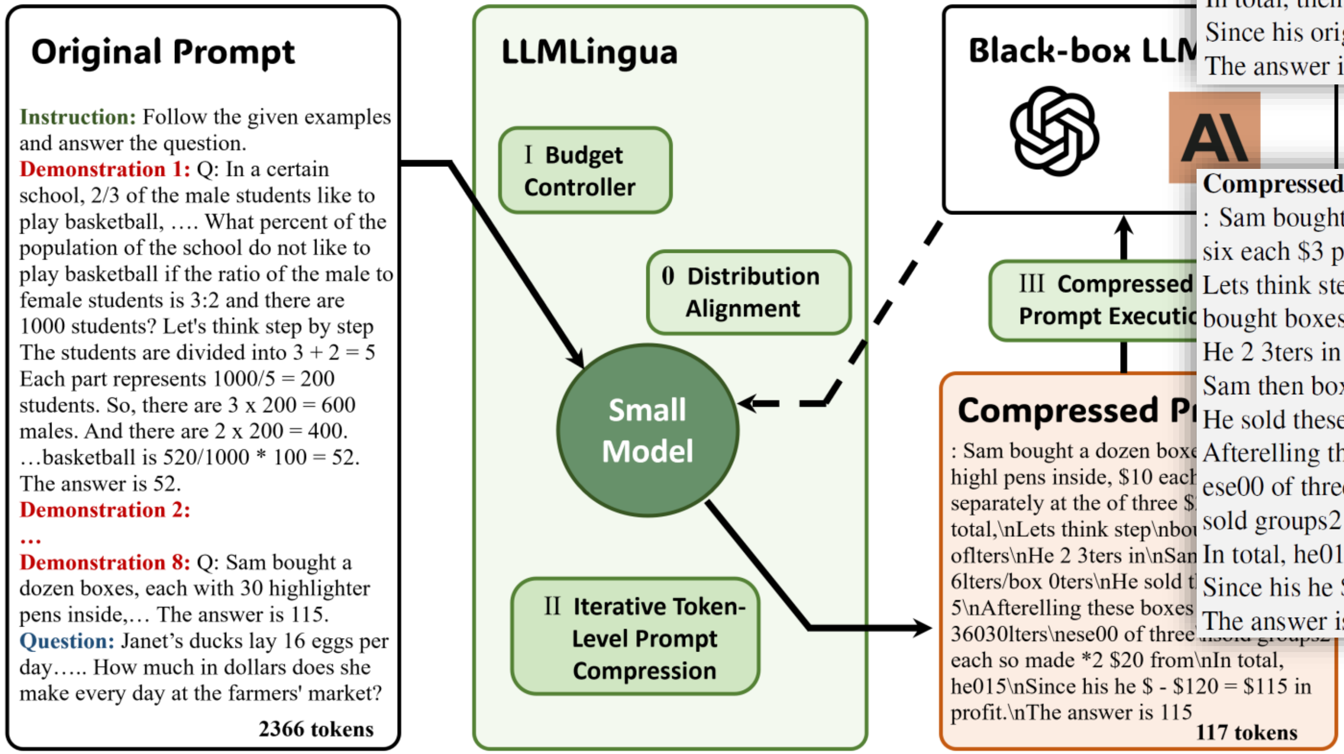


Credits: databricks.com



Prompt compression

- More context = more accurate (at cost)
- LLMLingua EMNLP 2023 (Instruction tuning!)



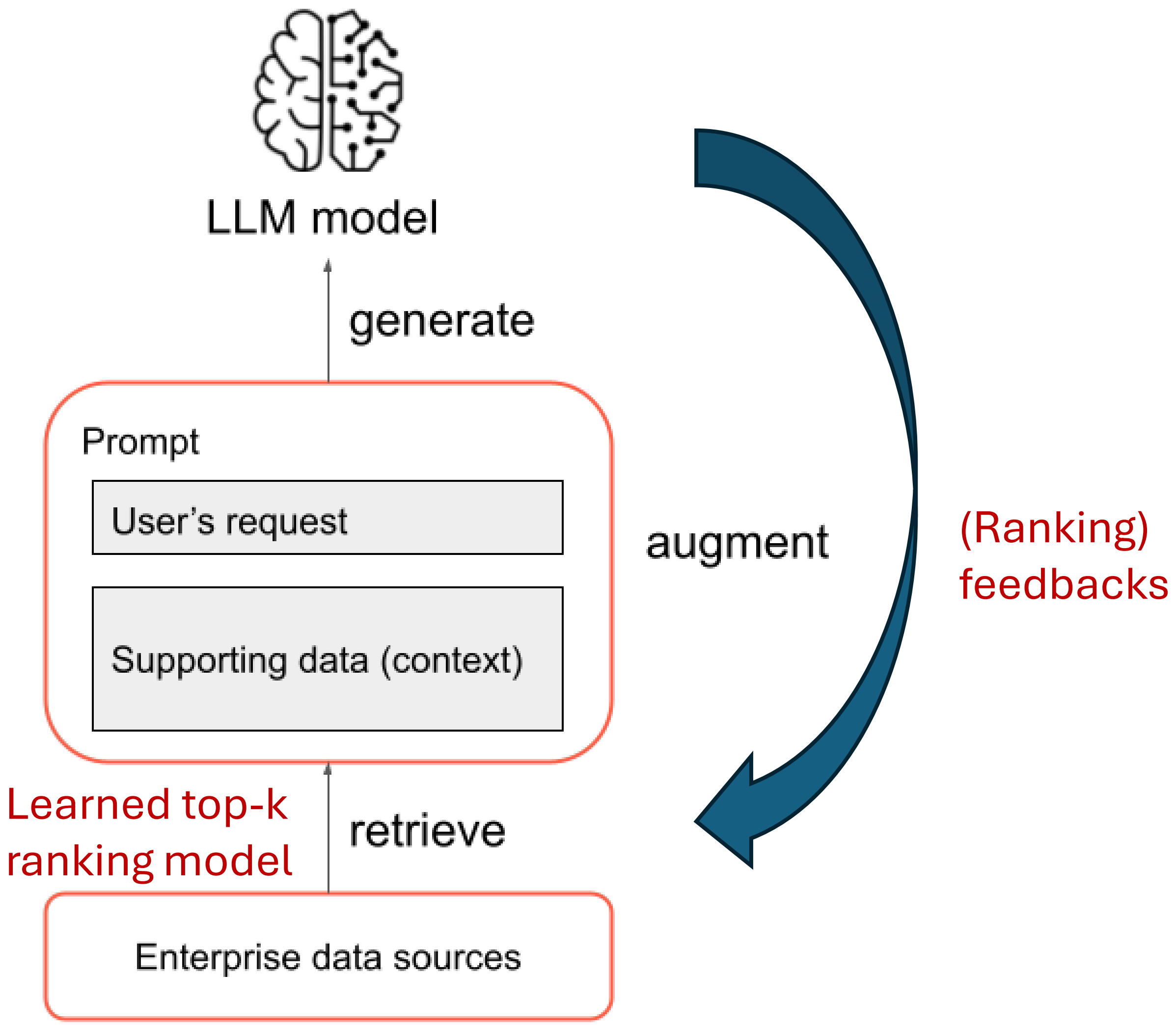
Original Prompt(9-steps Chain-of-Thought):
Question: Sam bought a dozen boxes, each with 30 highlighter pens inside, for $10 each box. He rearranged five of these boxes into packages of six highlighters each and sold them for $3 per package. He sold the rest of the highlighters separately at the rate of three pens for $2. How much profit did he make in total, in dollars?

Let's think step by step
Sam bought 12 boxes x $10 = $120 worth of highlighters.
He bought 12 * 30 = 360 highlighters in total.
Sam then took 5 boxes × 6 highlighters/box = 30 highlighters.
He sold these boxes for 5 * $3 = $15
After selling these 5 boxes there were 360 - 30 = 330 highlighters remaining.
These form 330 / 3 = 110 groups of three pens.
He sold each of these groups for $2 each, so made 110 * 2 = $220 from them.
In total, then, he earned $220 + $15 = $235.
Since his original cost was $120, he earned $235 - $120 = $115 in profit.
The answer is 115

Compressed Prompt:
: Sam bought a dozen boxes each 30 highl pens inside, $10 each. He reanged five of boxes into of six each $3 per. He sold the thelters separately at the of three $2. much make total,
Lets think step
bought boxes x0 oflters
He 2 3ters in
Sam then boxes 6lters/box 0ters
He sold these boxes 5
Afterelling these boxes there 36030lters
ese00 of three
sold groups2 each so made *2 $20 from
In total, he015
Since his he $ - $120 = $115 in profit.
The answer is 115

Learning to rank / re-ranking

- The “retrieval” part can be improved by using a learned top-k ranking model (should be cheaper than the later LLM)
- Automatic and free labels from previous runs
- Reduces context length requirements (improve P@K)

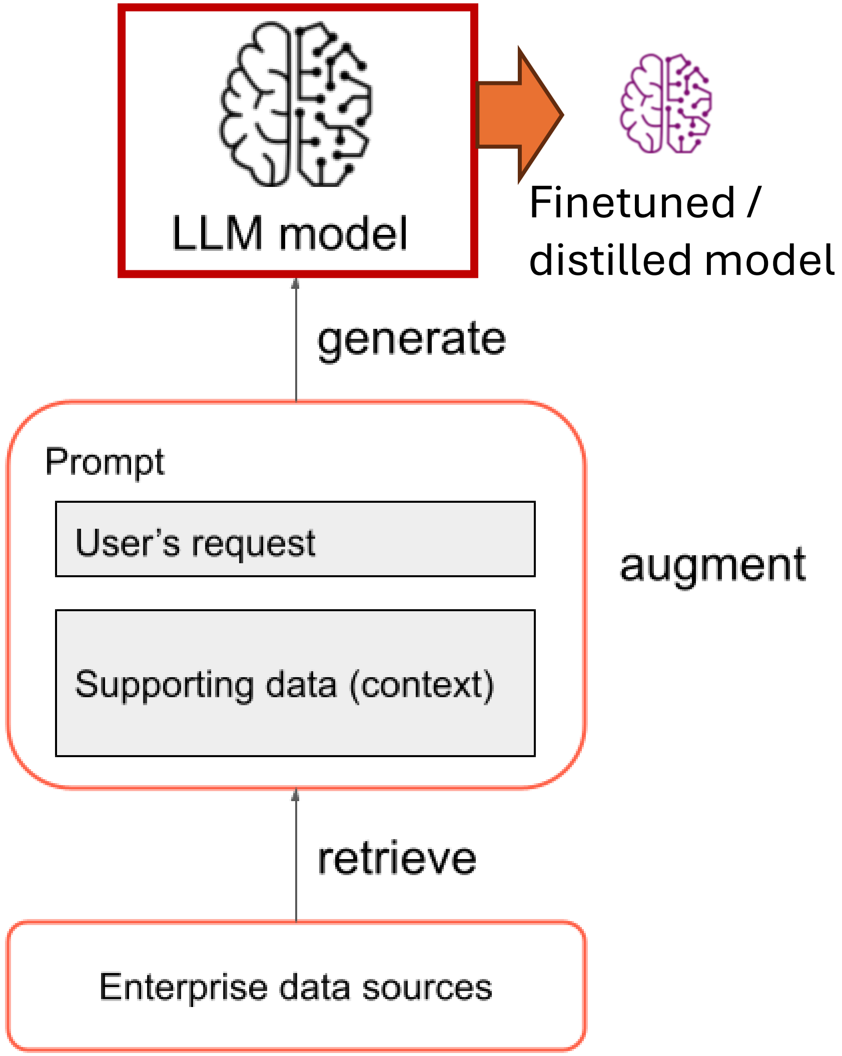


Model selection, finetuning & distillation

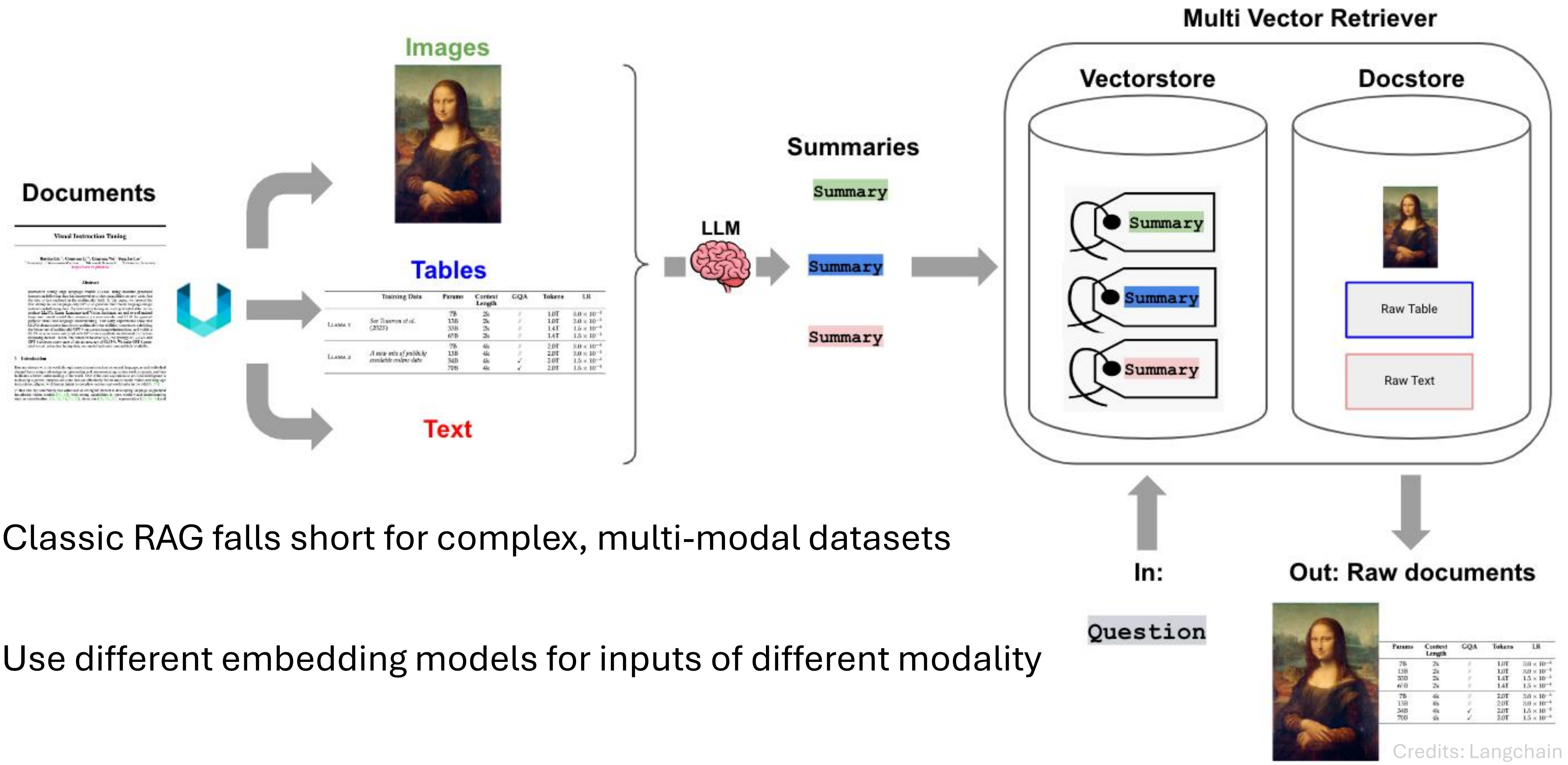
- Finetune or distill the generation model in order to reduce size, adapt to formatting requirements. e.g., collect RAG outputs from Llama 70b and send them to finetune Llama 13b

- Or for different queries, use different generation models

- Further, we can propagate the gradients to the embedding phrase, **and finetune embedding models**



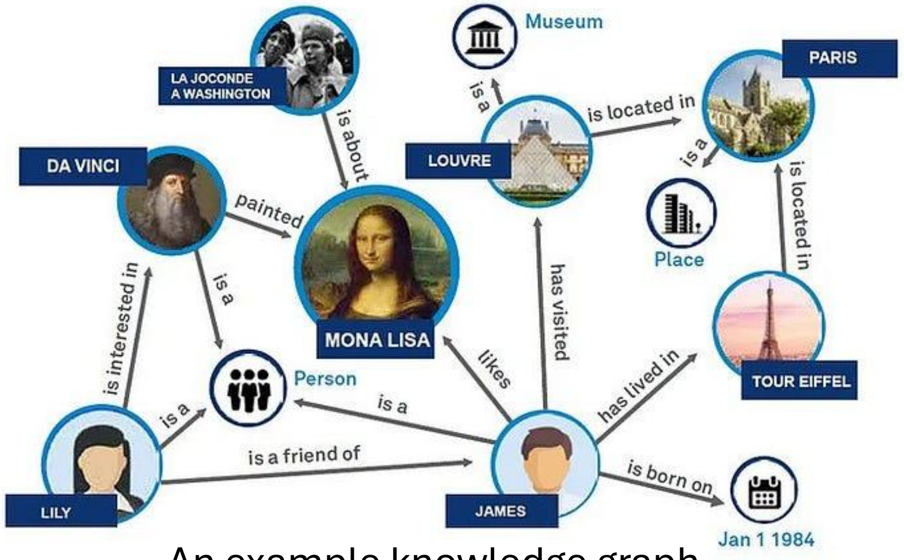
Multi-vector retrieval



- Classic RAG falls short for complex, multi-modal datasets
- Use different embedding models for inputs of different modality

Graph RAG

- Classic RAG approaches do not consider links between entities.
- They also have a wholistic view of the dataset (with simple similar search)
- Given a private dataset, GraphRAG from Microsoft generates the knowledge graph using LLMs, and retrieve for relevant content for new RAG queries.

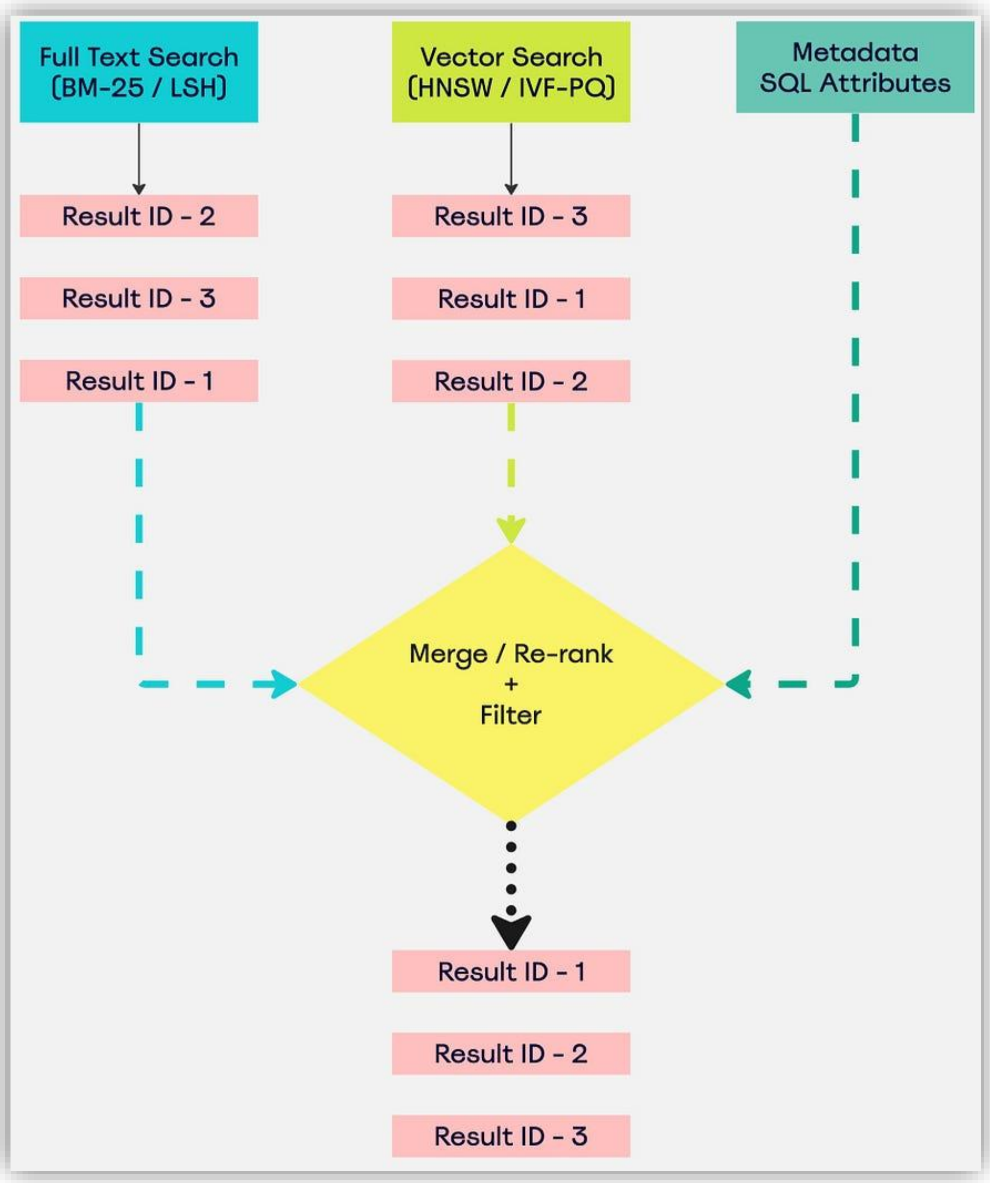


An example knowledge graph

Query: *"What has Novorossiya done?"*

Baseline RAG	**GraphRAG**
The text does not provide specific information on what Novorossiya has done.	Novorossiya, a political movement in Ukraine, has been involved in a series of destructive activities, particularly targeting various entities in Ukraine [Entities (6494, 912)]. The movement has been linked to plans to destroy properties of several Ukrainian entities, including Rosen, the Odessa Canning Factory, the Odessa Regional Radio Television Transmission Center, and the National Television Company of Ukraine [Relationships (15207, 15208, 15209, 15210)]. In addition to these, Novorossiya has also been associated with plans to destroy properties of Roshen, the Odessa Conservatory Factory, the Odesa Regional Radio Television Transmission Center, and the National Telecommunications Company of Ukraine [Relationships (8329, 8330, 8331, 8332)]. Furthermore, Novorossiya has been implicated in plans to blow up ATMs of PrivatBank, the largest state bank in Ukraine [Relationships (8335, 8335)]. The Office of the General Prosecutor in Ukraine has reported on the creation of Novorossiya, indicating the government's awareness and potential concern over the activities of this movement [Relationships (8264, 8264)]. The destructive activities of Novorossiya have had significant impacts on the targeted entities and have likely contributed to the ongoing tensions and conflicts in the region. In summary, Novorossiya has been involved in a series of destructive activities targeting various entities in Ukraine. These activities have had significant impacts on the targeted entities and have likely contributed to the ongoing tensions and conflicts in the region. The Ukrainian government, through the Office of the General Prosecutor, has acknowledged the existence and activities of Novorossiya, indicating a level of concern over the movement's actions.

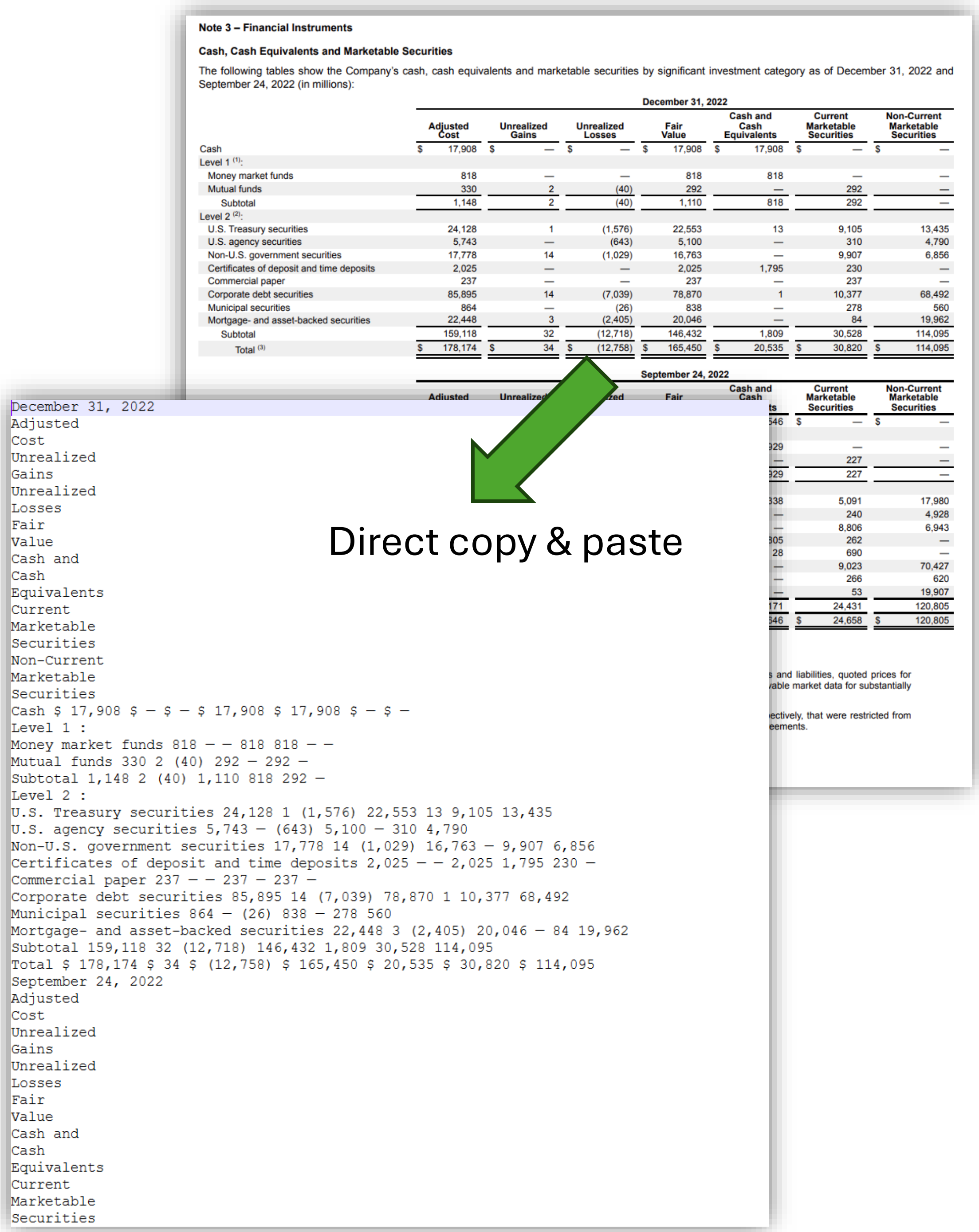
Combine with full-text search



- Embedding has “needle-in-the-hay” problem.
- To improve, RAGs can be combined with full-text search or external tools (SQL, search engine) to boost accuracy
- Full-text search: BM-25 or LSH.

Raw documents in RAG

- Parsing & cleaning raw documents into structured data is often challenging: noisy, unstructured, long documents
- Long-context vs RAG
  - **Long-context LLMs**: simple (for developers) but often more expensive (for users), can lost in the middle
  - **RAG**: cheaper, deterministic security, easier to debug, up-to-date info



Note 3 – Financial Instruments

Cash, Cash Equivalents and Marketable Securities

The following tables show the Company's cash, cash equivalents and marketable securities by significant investment category as of December 31, 2022 and September 24, 2022 (in millions):

	December 31, 2022						
	Adjusted Cost	Unrealized Gains	Unrealized Losses	Fair Value	Cash and Cash Equivalents	Current Marketable Securities	Non-Current Marketable Securities
Cash	$ 17,908	$ —	$ —	$ 17,908	$ 17,908	$ —	$ —
Level 1 (1):							
Money market funds	818	—	—	818	818	—	—
Mutual funds	330	2	(40)	292	—	292	—
Subtotal	1,148	2	(40)	1,110	818	292	—
Level 2 (2):							
U.S. Treasury securities	24,128	1	(1,576)	22,553	13	9,105	13,435
U.S. agency securities	5,743	—	(643)	5,100	—	310	4,790
Non-U.S. government securities	17,778	14	(1,029)	16,763	—	9,907	6,856
Certificates of deposit and time deposits	2,025	—	—	2,025	1,795	230	—
Commercial paper	237	—	—	237	—	237	—
Corporate debt securities	85,895	14	(7,039)	78,870	1	10,377	68,492
Municipal securities	864	—	(26)	838	—	278	560
Mortgage- and asset-backed securities	22,448	3	(2,405)	20,046	—	84	19,962
Subtotal	159,118	32	(12,718)	146,432	1,809	30,528	114,095
Total (3)	$ 178,174	$ 34	$ (12,758)	$ 165,450	$ 20,535	$ 30,820	$ 114,095

September 24, 2022

Parsing unstructured data

- **Parsing: unstructured >> structured data**

- Common approaches:
  - Rule based parsing: regex, HTML tags
  - Computer-vision-based parsing
  - NLP based parsing
  - LLM based parsing

LlamaIndex

UNSTRUCTURED

Airparser.

DOCSUMO Document AI

Amazon Textract

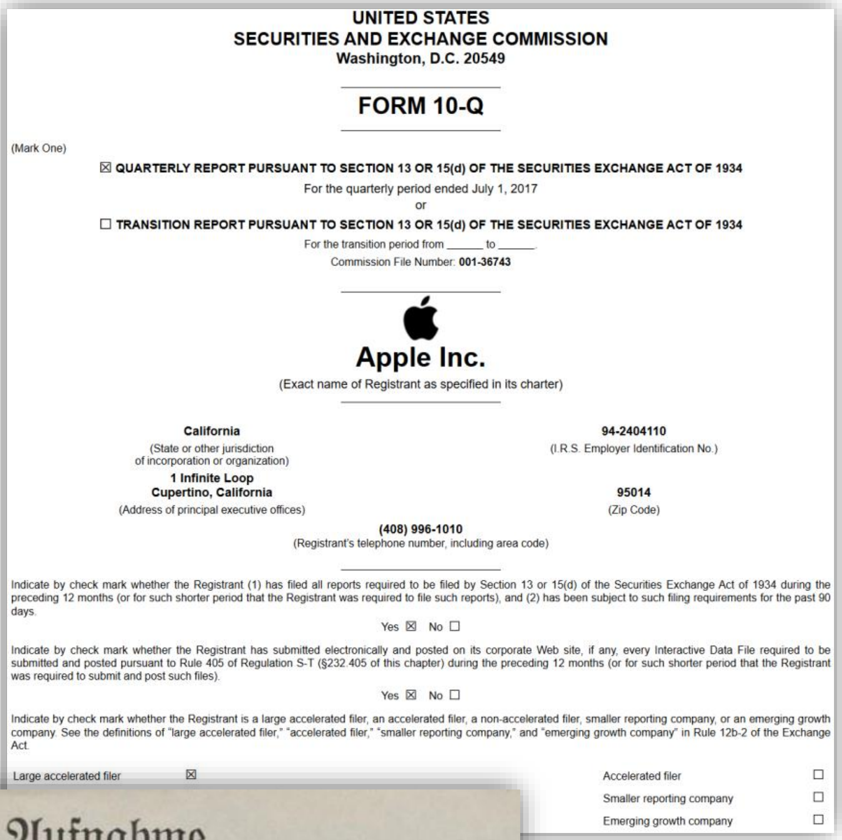
parsio

Nanonets

```
import re

def extract_emails(text):
    pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b'
    return re.findall(pattern, text)

sample_text = "Contact us at john.doe@example.com or support@company.org for assistanc
emails = extract_emails(sample_text)
print(emails)
# Output: ['john.doe@example.com', 'support@company.org']
```

Rule-based parsing

- Using per-template, pre-defined rules
  - E.g., name = row 2 char 4 to char 10
  - Pixel(10, 10) to Pixel(100, 200)
  - Search keyword = “Zip Code”
- How to define the rules?
  - Manual scripting (when there IS a template)
  - For dynamic/noisy inputs:
    ML based vision, NL solutions


UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549

FORM 10-Q

(Mark One)

☒ QUARTERLY REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934
For the quarterly period ended July 1, 2017
or
☐ TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934
For the transition period from ______ to ______
Commission File Number: 001-36743

Apple Inc.
(Exact name of Registrant as specified in its charter)

California (State or other jurisdiction of incorporation or organization)	94-2404110 (I.R.S. Employer Identification No.)
1 Infinite Loop Cupertino, California (Address of principal executive offices)	95014 (Zip Code)

(408) 996-1010
(Registrant's telephone number, including area code)

Indicate by check mark whether the Registrant (1) has filed all reports required to be filed by Section 13 or 15(d) of the Securities Exchange Act of 1934 during the preceding 12 months (or for such shorter period that the Registrant was required to file such reports), and (2) has been subject to such filing requirements for the past 90 days.

Yes ☒ No ☐

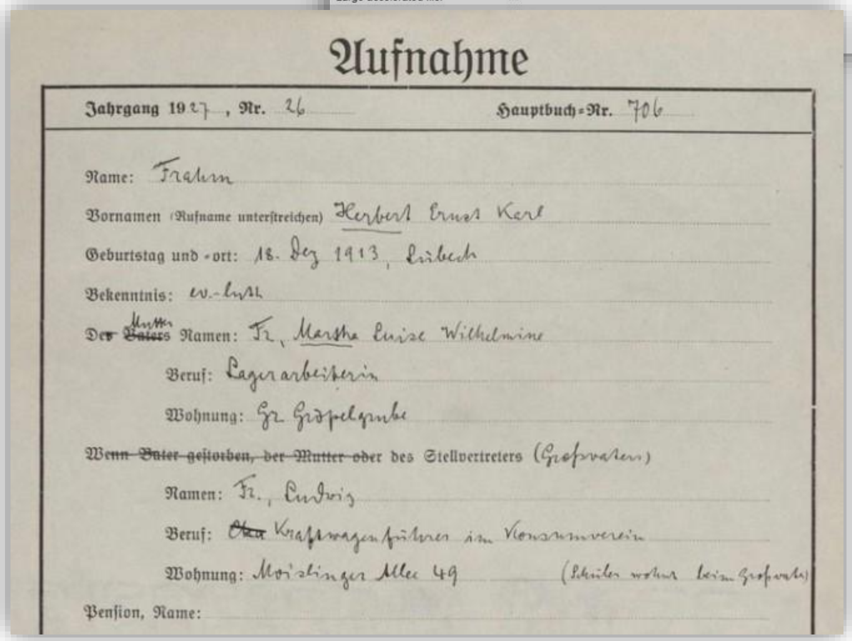
Indicate by check mark whether the Registrant has submitted electronically and posted on its corporate Web site, if any, every Interactive Data File required to be submitted and posted pursuant to Rule 405 of Regulation S-T (§232.405 of this chapter) during the preceding 12 months (or for such shorter period that the Registrant was required to submit and post such files).

Yes ☒ No ☐

Indicate by check mark whether the Registrant is a large accelerated filer, an accelerated filer, a non-accelerated filer, smaller reporting company, or an emerging growth company. See the definitions of “large accelerated filer,” “accelerated filer,” “smaller reporting company,” and “emerging growth company” in Rule 12b-2 of the Exchange Act.

Large accelerated filer	☒	Accelerated filer	☐
		Smaller reporting company	☐
		Emerging growth company	☐

Template


Aufnahme

Jahrgang 1927, Nr. 26 Hauptbuch-Nr. 706

Name: Frahm

Vornamen (Rufname unterstreichen) Herbert Ernst Karl

Geburtstag und -ort: 18. Dez. 1913, Lübeck

Bekenntnis: ev.-luth.

~~Des Vaters~~ Mutter Namen: Fr. Martha Luise Wilhelmine

Beruf: Lagerarbeiterin

Wohnung: Gr. Gröpelgrube

~~Wenn Vater gestorben, der Mutter oder~~ des Stellvertreters (Großvater)

Namen: Fr. Ludwig

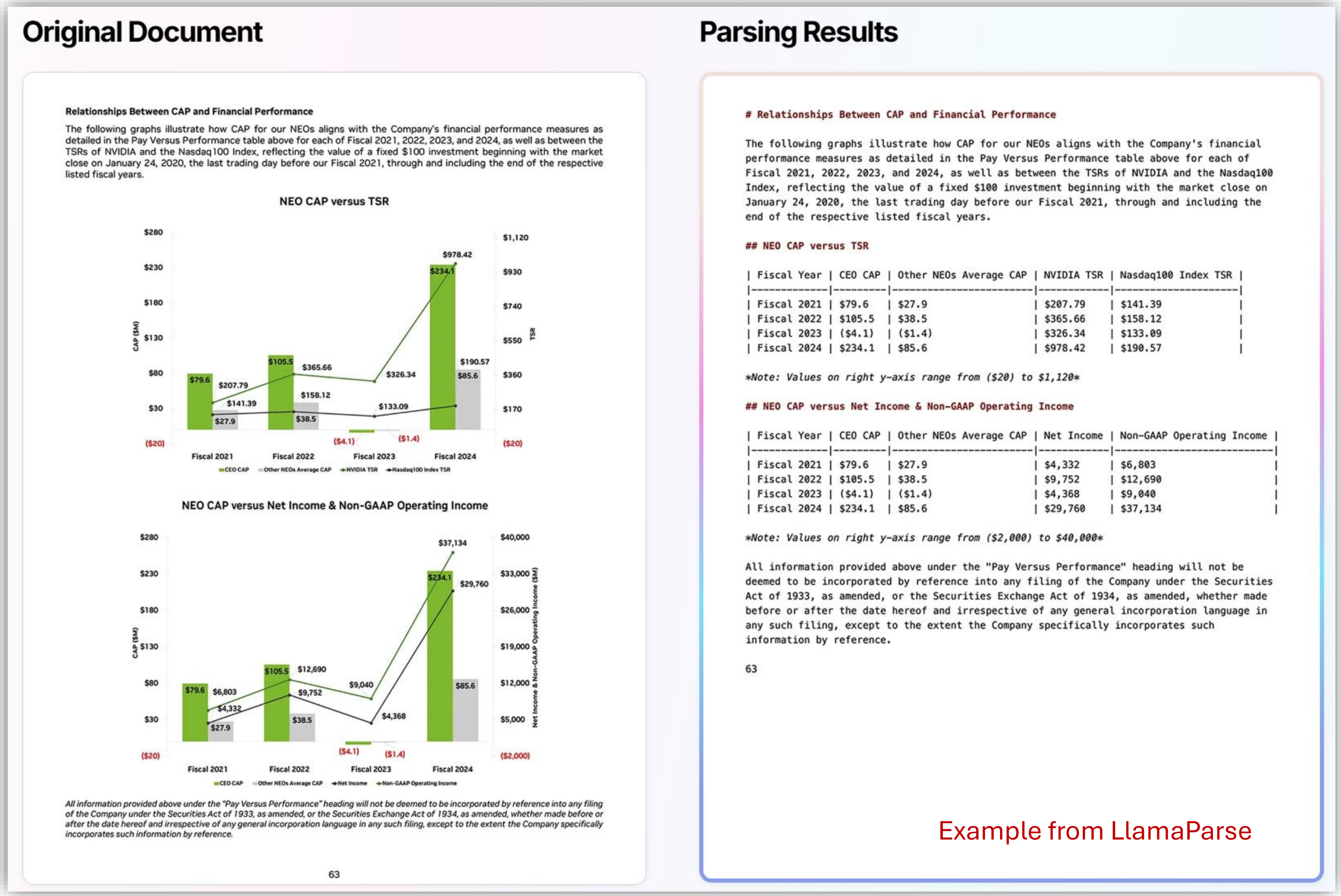
Beruf: Kraftwagenführer im Konsumverein

Wohnung: Moislinger Allee 49 (Schüler wohnt beim Großvater)

Pension, Name:

Handwritten

Parsing unstructured data

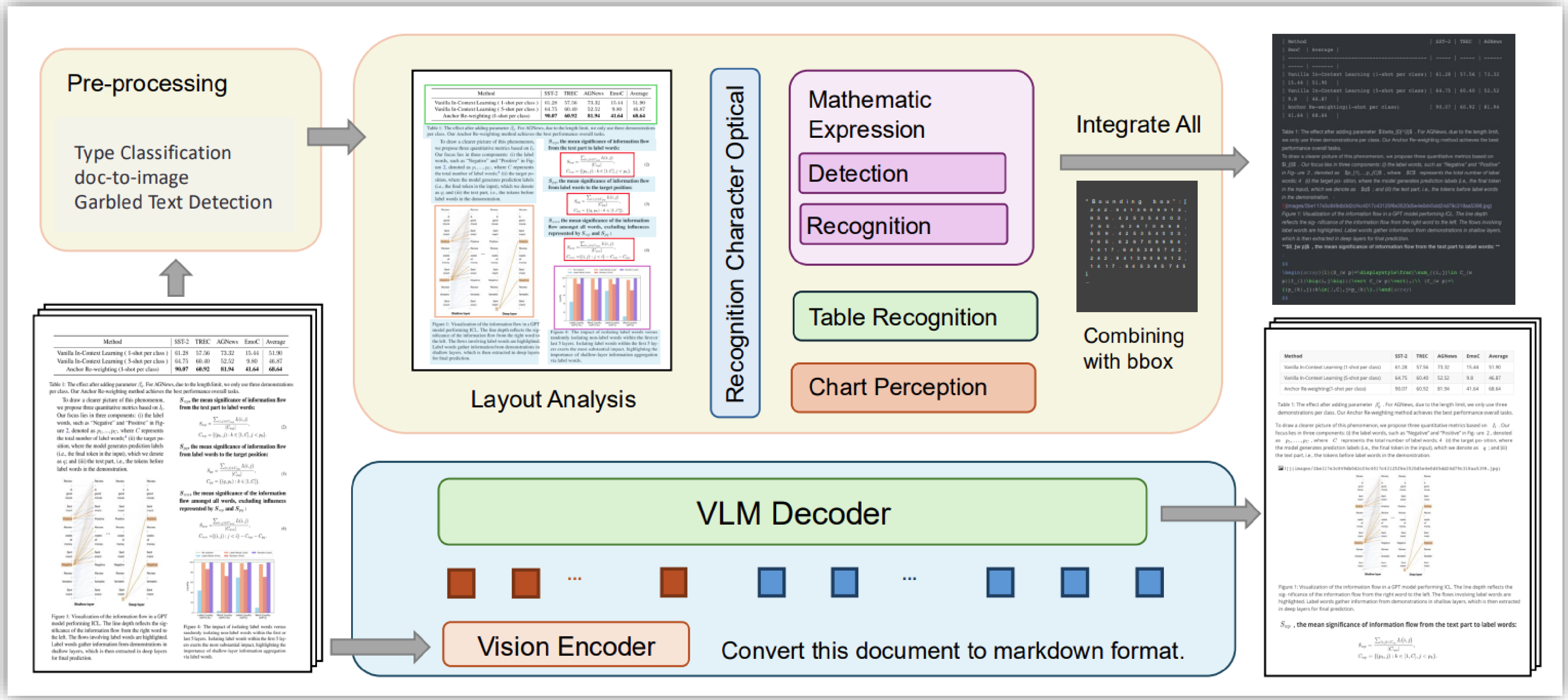


Example from LlamaParse

More complex parsing:

- Tables, figures, charts
- Complex layouts
- Large multi-modal models

Computer-vision-based parsing



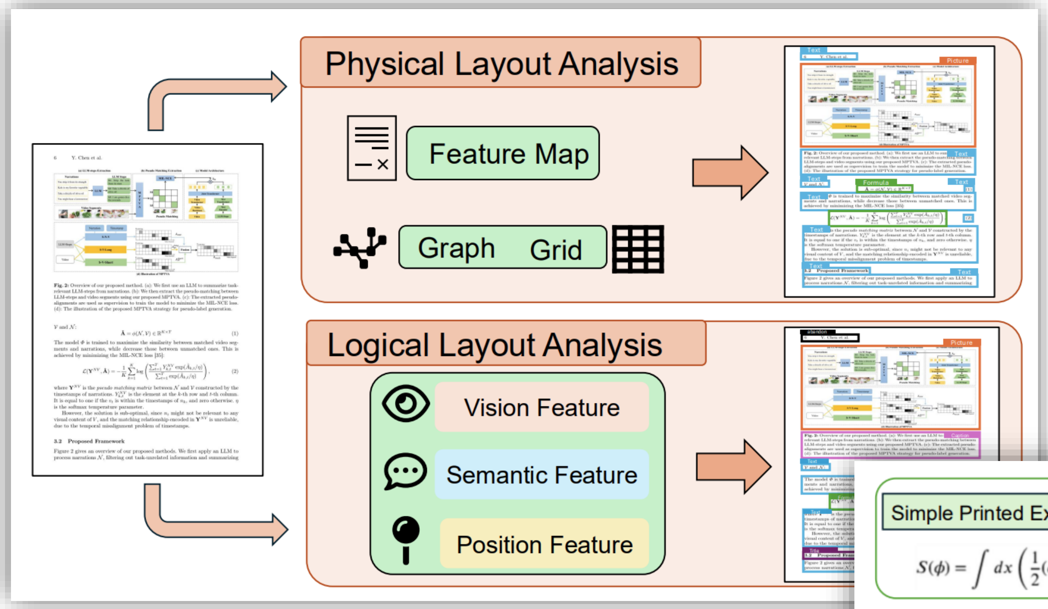
CV-based parsing uses pretrained models to extract structural information from images

Zhang et al. Document Parsing Unveiled: Techniques, Challenges, and Prospects for Structured Information Extraction. arXiv 2410.21169v2

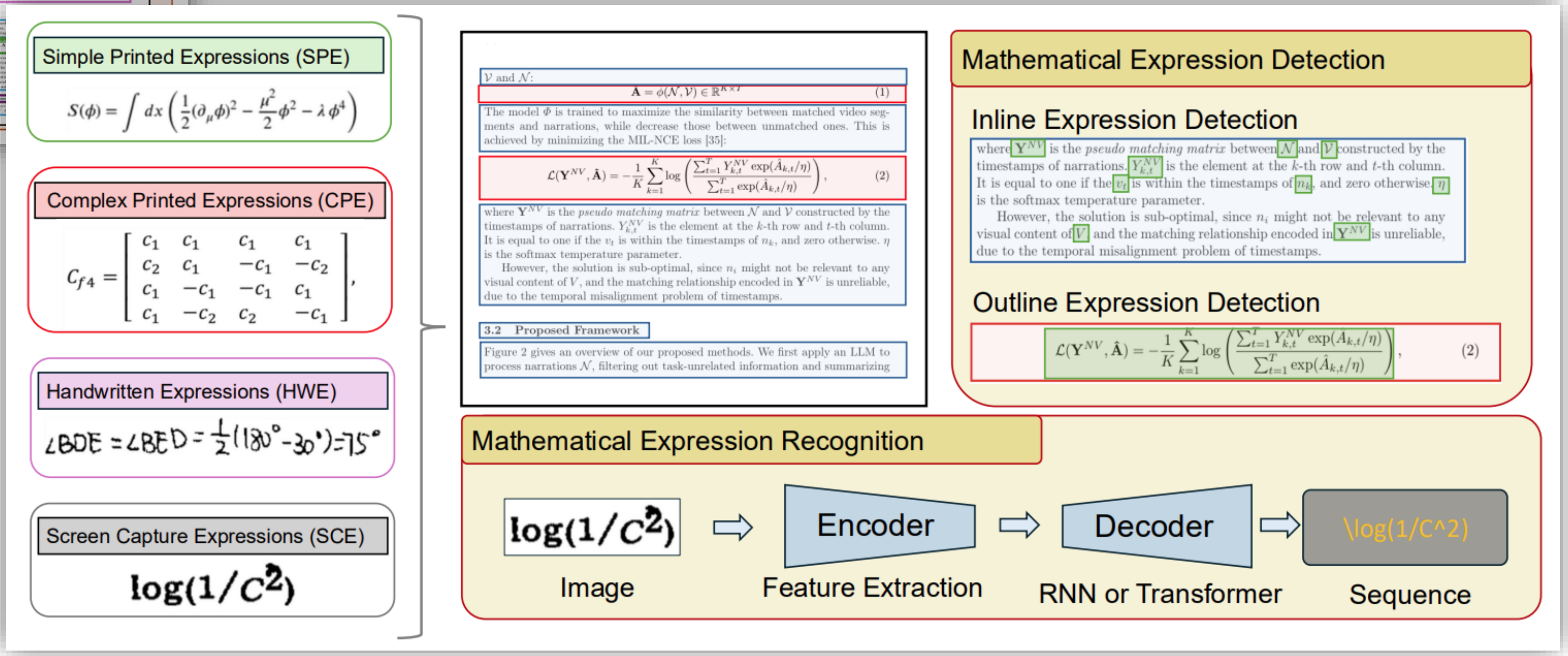
Computer-vision-based parsing

Some tasks use standalone, specific models:

- Layout analysis (extract bounding boxes)
- Optical Character Recognition (OCR)
- Math formula recognition (OCR)
- Table and chart recognition



Layout analysis



Computer-vision-based parsing

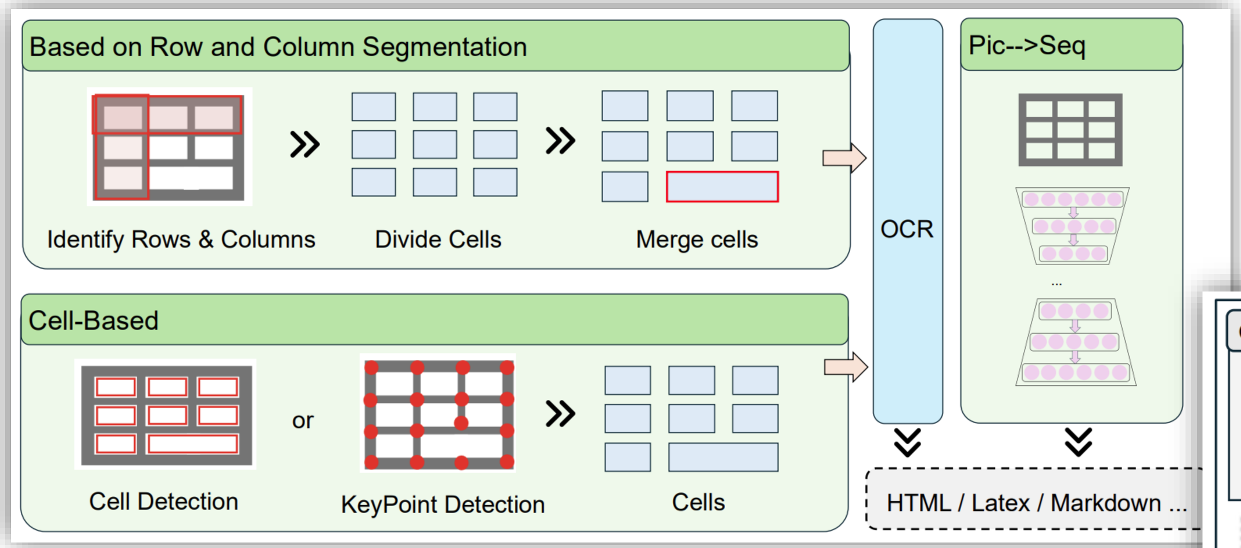
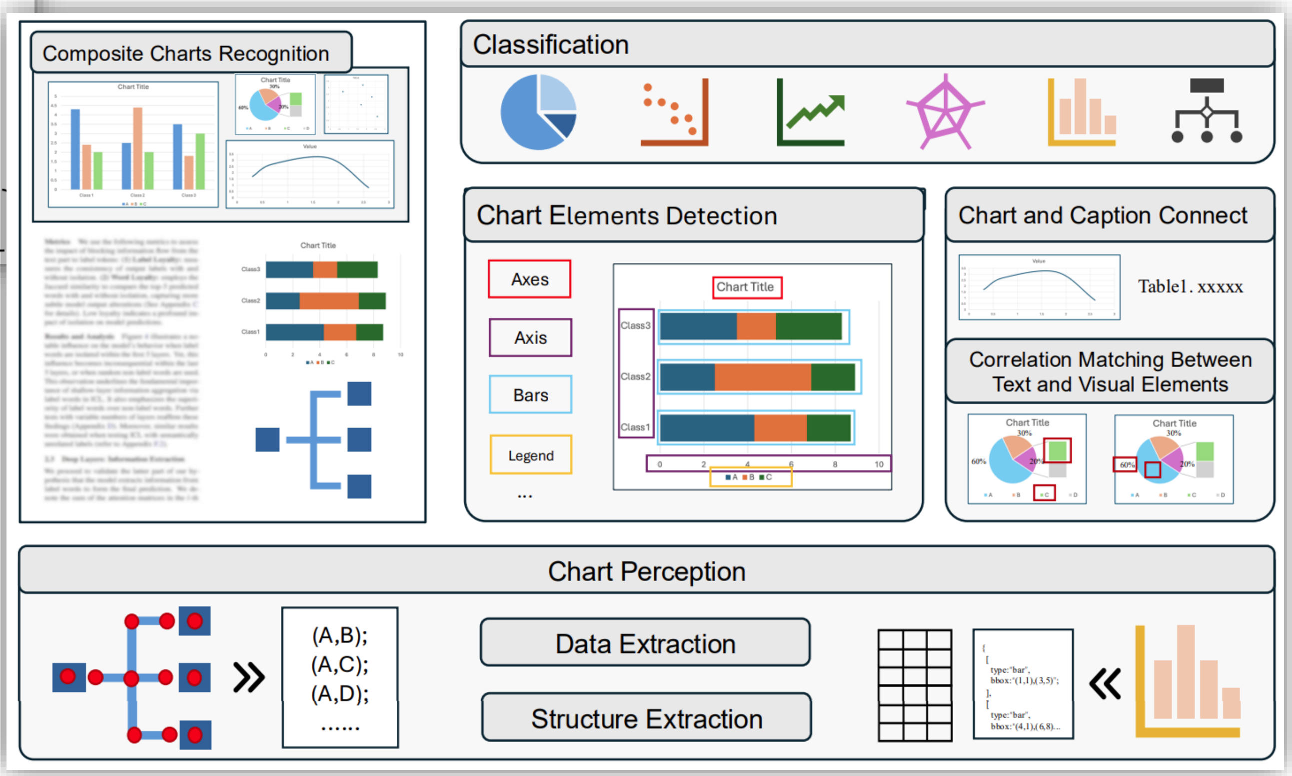


Table recognition

Chart recognition



NL-based data parsing

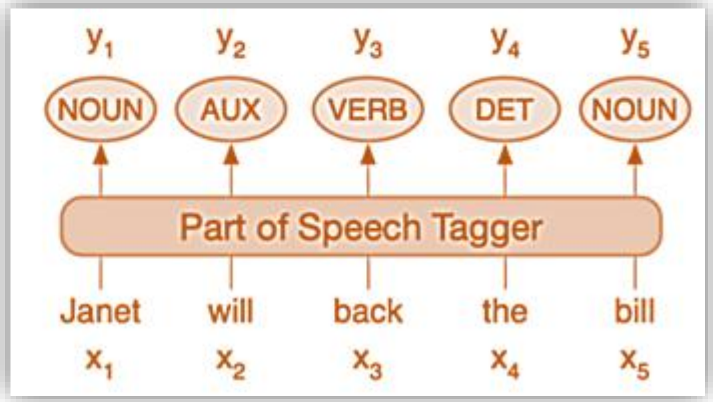
- Common text pre-processing
  - Cleaning (removing words like stopwords, emojis, punctuation, etc.)
  - Normalization
  - Lemmatization & stemming
- Tools: Regex, NLTK, spaCy, OpenNLP

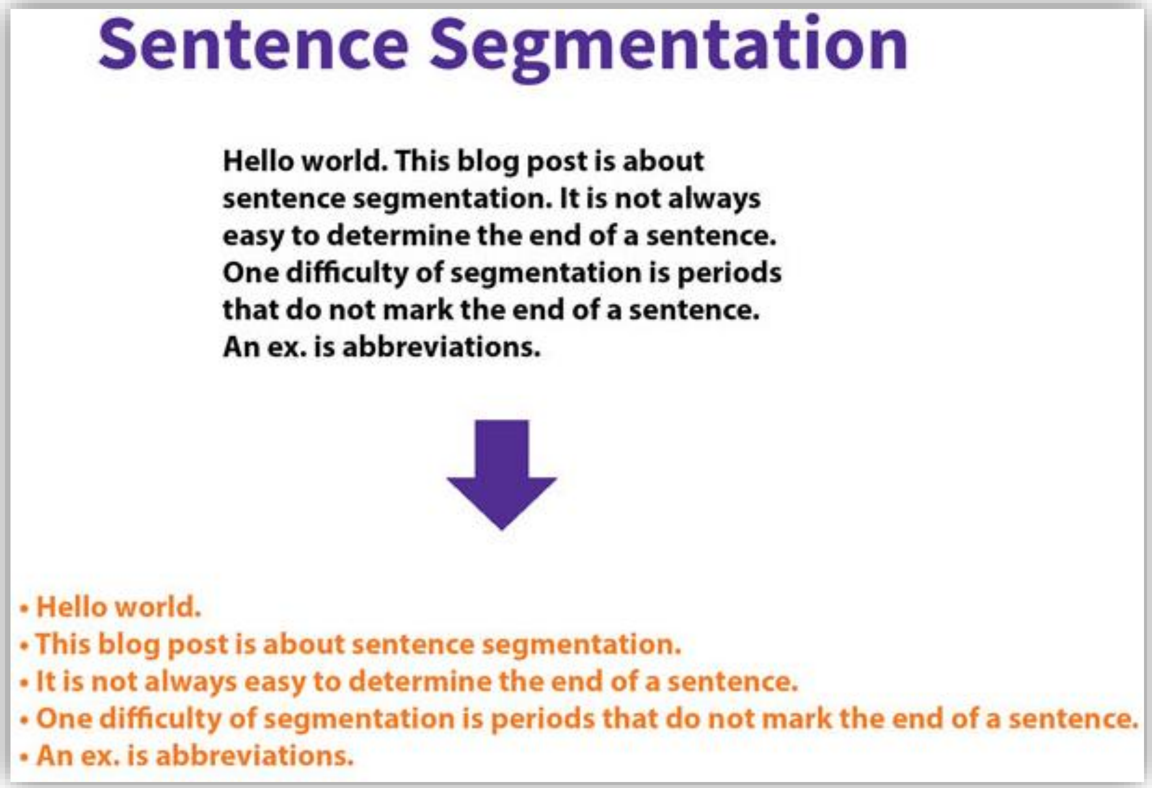
	original_word	stemmed_word
0	trouble	troubl
1	troubled	troubl
2	troubles	troubl
3	troublemsome	troublemsom

```
sample = "Hello @gabe_flomo 👋🏾, I still want us to hit that new sushi spot??? LMK when you're free cuz I can't go this or next weekend since I'll be swimming!!! #sushiBros #rawFish #🍣"
print(pipeline(sample))

# output"
hello still want us hit new sushi spot lmk free cuz cant go next weekend since ill swim"
```

NL-based data parsing

- Segmentation & tagging
  - Some useful applications: detecting title etc.

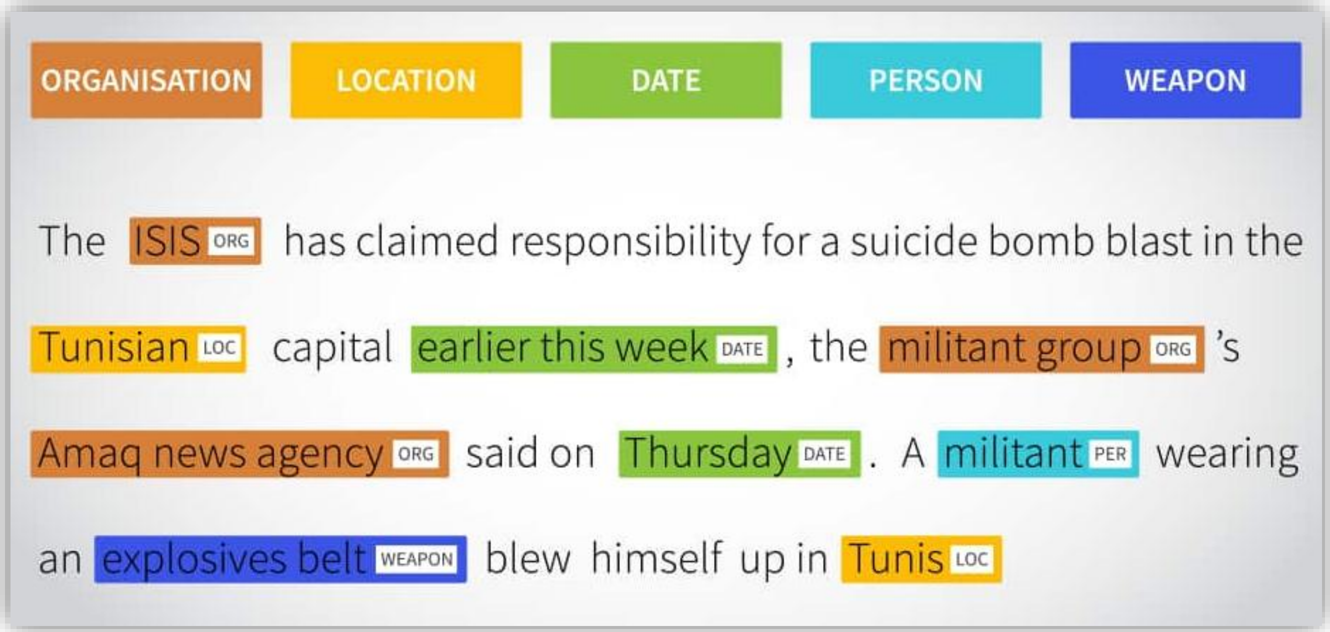




NL-based data parsing

- Segmentation & tagging
  - Some useful applications: detecting title etc.

- Name entity recognition
  - Person: Steve Jobs
  - Company: Apple
  - Location: California
  - Column names often use entity names

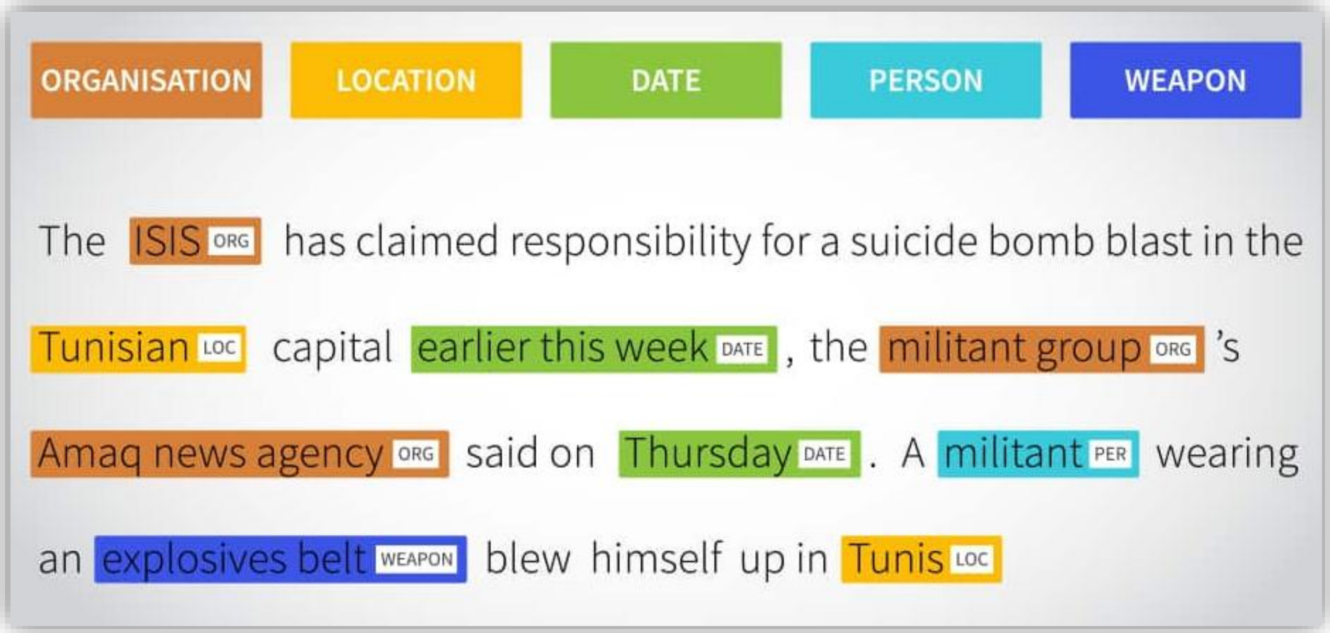


NL-based data parsing

- Segmentation & tagging
  - Some useful applications: detecting title etc.

- Name entity recognition
  - Person: Steve Jobs
  - Company: Apple
  - Location: California
  - Column names often use entity names

- Extraction (column = value)
  - Rule-based
  - RAGs (later)



LLM-based parsing

- **One model for all?**
- Large multi-modal models, e.g., GPT-4o

Note 7 – Shareholders' Equity

Dividends

The Company declared and paid cash dividends per share during the periods presented as follows:

	Dividends Per Share	Amount (in millions)
2017:		
Third quarter	$ 0.63	$ 3,281
Second quarter	0.57	2,988
First quarter	0.57	3,042
Total cash dividends declared and paid	$ 1.77	$ 9,311
2016:		
Fourth quarter	$ 0.57	$ 3,071
Third quarter	0.57	3,117
Second quarter	0.52	2,879
First quarter	0.52	2,898
Total cash dividends declared and paid	$ 2.18	$ 11,965

Future dividends are subject to declaration by the Board of Directors.

Parse into markdown

- Drawbacks:
  - Expensive
  - Hard to instruct

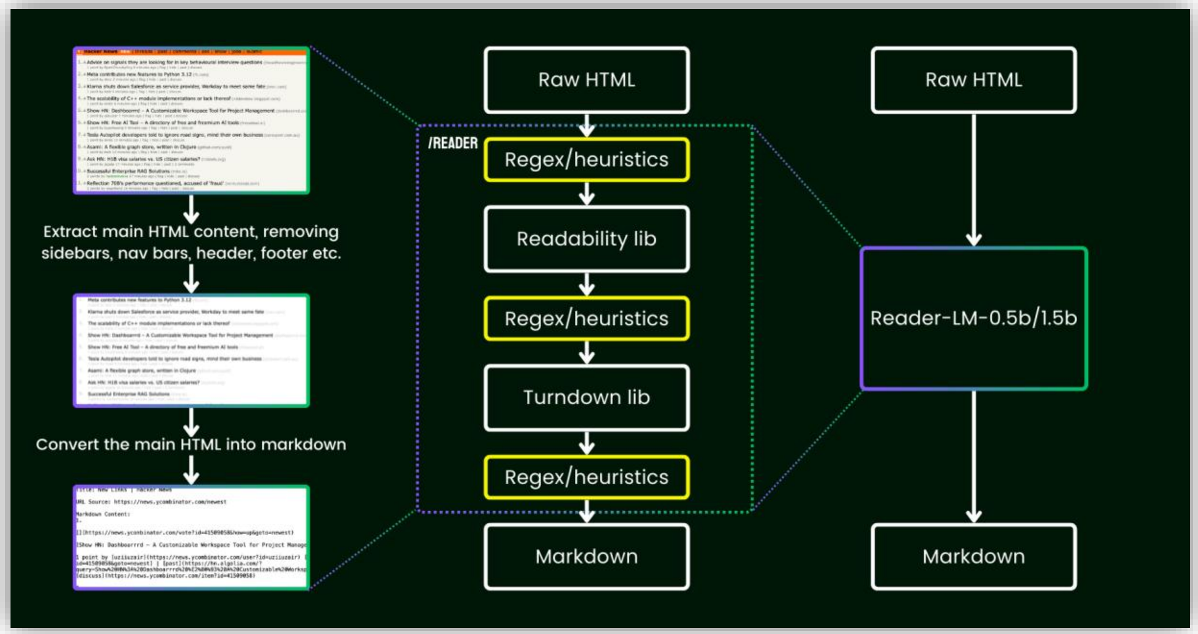
ChatGPT-4o

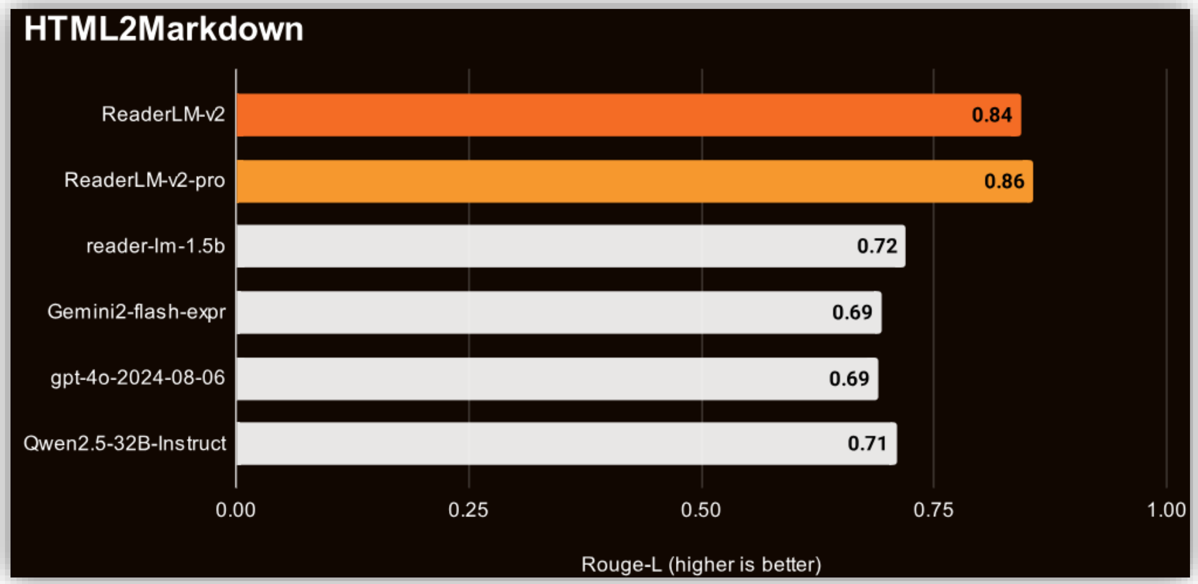
markdown Copy Edit

```
# Note 7 – Shareholders' Equity

## Dividends

The Company declared and paid cash dividends per share during the periods presented as follow

### 2017:
| Quarter       | Dividends Per Share | Amount (in millions) |
|--------------|-------------------|---------------------|
| Third quarter | $0.63              | $3,281               |
| Second quarter | $0.57              | $2,988               |
| First quarter | $0.57              | $3,042               |
| **Total cash dividends declared and paid** | **$1.77** | **$9,311** |

### 2016:
| Quarter       | Dividends Per Share | Amount (in millions) |
|--------------|-------------------|---------------------|
| Fourth quarter | $0.57              | $3,071               |
| Third quarter  | $0.57              | $3,117               |
| Second quarter | $0.52              | $2,879               |
| First quarter  | $0.52              | $2,898               |
| **Total cash dividends declared and paid** | **$2.18** | **$11,965** |

Future dividends are subject to declaration by the Board of Directors.
```

LLM-based parsing

- **One model for all?**
- Large multi-modal models, e.g., GPT-4o
  - Expensive
  - Hard to instruct
- Small Language Models (SLMs)
  - Small = cheap
  - Instruction tuned for data parsing
  - E.g., ReaderLM-v2 from Jina AI





https://jina.ai/news/readerlm-v2-frontier-small-language-model-for-html-to-markdown-and-json

There is no free lunch

- No single method can guarantee 100% correct
- Hard to verify

- There are ML/AI solutions to alleviate these problems
  - Human-in-the-loop systems and applications design
  - Multi-agent framework to cross validate
  - Active learning to reduce annotation
  - Synthetic data generation to improve parsing robustness

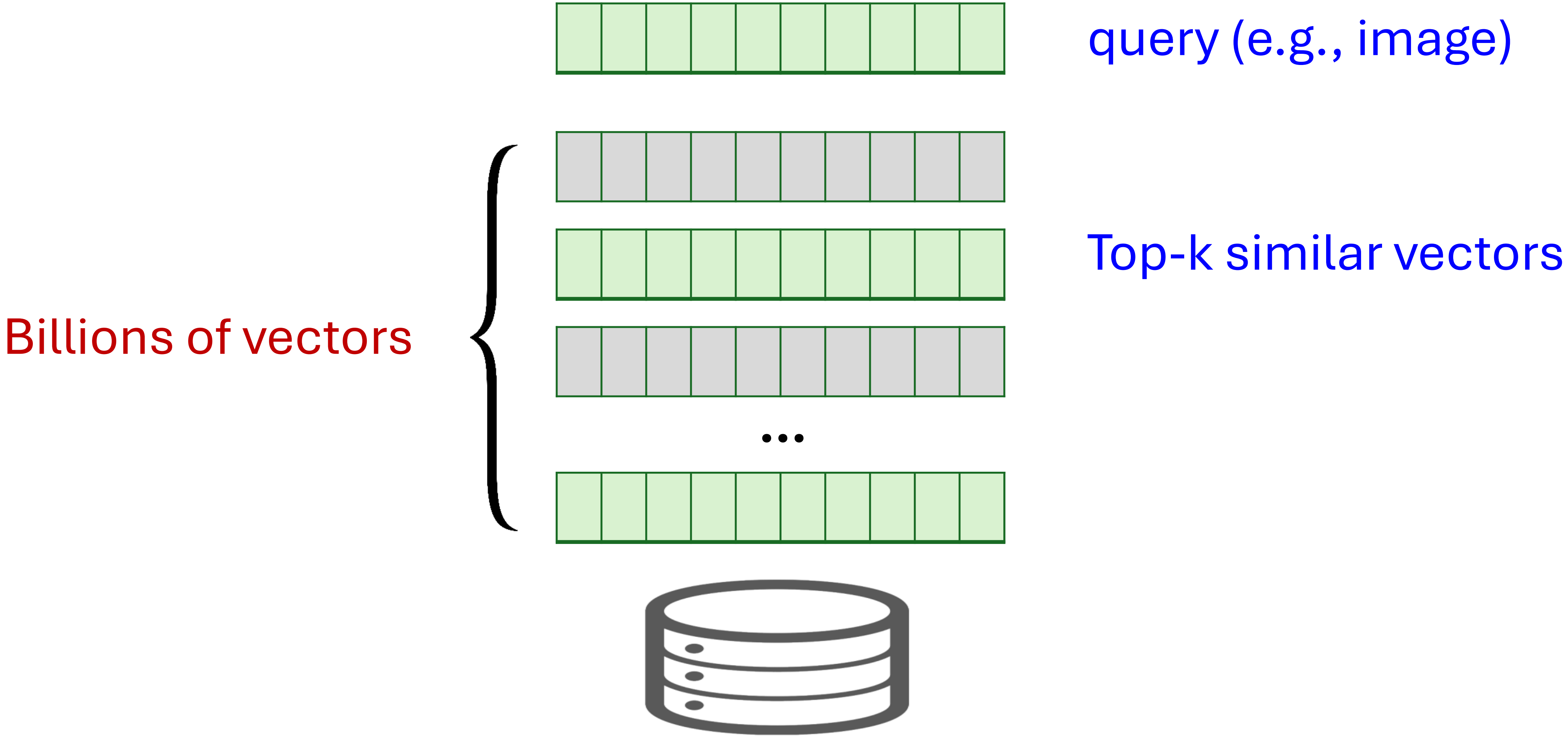
Overview

- Retrieval Augmented Generation (RAG)
- Vector DBs
- AI agents

Recent vector databases

- Pinecone
- Milvus
- Faiss
- chroma
- drant
- Weaviate
- vespa
- ClickHouse
- ORACLE
- AlloyDB
- Azure SQL
- AnalyticDB
- SingleStore
- MongoDB. Atlas
- PostgreSQL
- databricks
- redis
- Alipay
- elasticsearch
- OpenSearch
- cassandra

Key operator in vector DBs: vector similarity search



Evolution of vector data(base)

1999
(Content-based information retrieval)

2013
(Embedding)

2023
(LLM)

Similarity Search in High Dimensions via Hashing

Aristides Gionis * Piotr Indyk† Rajeev Motwani‡
Department of Computer Science
Stanford University
Stanford, CA 94305
{gionis,indyk,rajeev}@cs.stanford.edu

Locality-sensitive hash

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

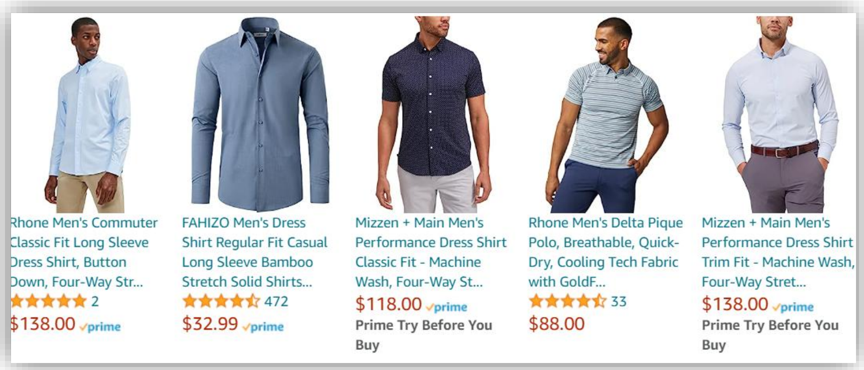
Retrieval OpenAI

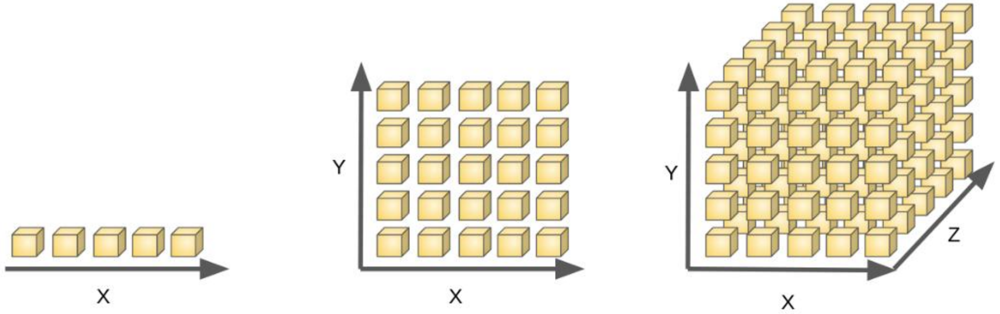
The open-source retrieval plugin enables ChatGPT to access personal or organizational information sources (with permission). It allows users to obtain the most relevant document snippets from their data sources, such as files, notes, emails or public documentation, by asking questions or expressing needs in natural language.

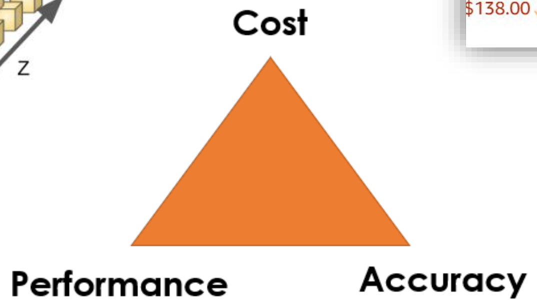
As an open-source and self-hosted solution, developers can deploy their own version of the plugin and register it with ChatGPT. The plugin leverages OpenAI embeddings and allows developers to choose a vector database (Milvus, Pinecone, Qdrant, Redis, Weaviate or Zilliz) for indexing and searching documents. Information sources can be synchronized with the database using webhooks.

Why are vector DBs challenging?

- Easy to get started, but very challenging to achieve high performance, accuracy, and efficiency
- Three unique properties that contribute to the challenges of vector DBs
  - Property P1: Curse of Dimensionality
  - Property P2: Approximation
  - Property P3: Advanced Vector Data Analytics







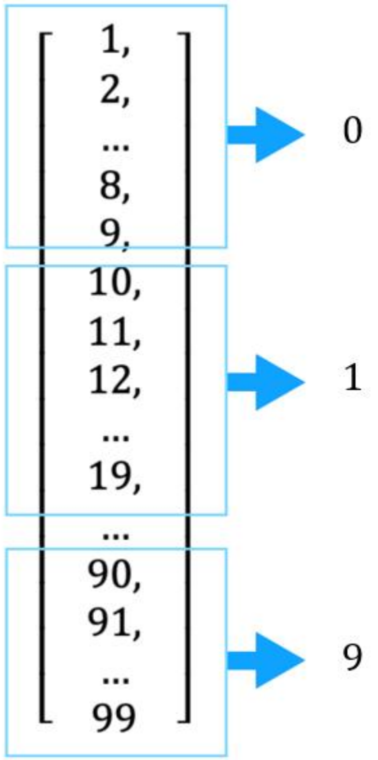
Vector indexes (main memory)

- Quantization-based indexes
  - E.g., IVF_FLAT, IVF_PQ
- Graph-based indexes
  - E.g., NSW, HNSW
- Tree-based indexes
- Hash-based indexes

Widely used in vector DBs

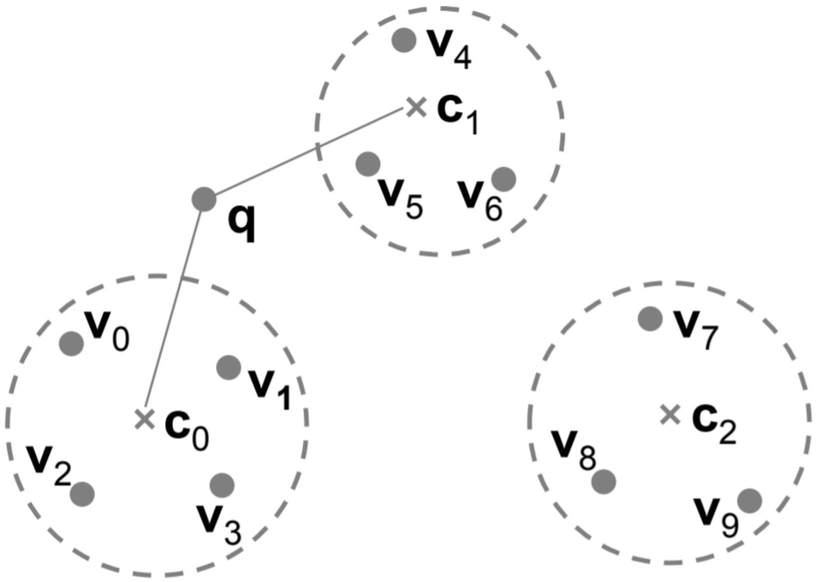
Quantization

- What's quantization?
  - A way of approximation
- Let's look at quantization in 1-dimensional space
  - $Q(x) = \left\lfloor \frac{x}{10} \right\rfloor$, where x is an input value
  - input = 3, $Q(3) = \left\lfloor \frac{3}{10} \right\rfloor = \lfloor 0.3 \rfloor = 0$
  - input = 3, $Q(91) = \left\lfloor \frac{91}{10} \right\rfloor = \lfloor 9.1 \rfloor = 9$
  - Those 99 integers can be quantized into a smaller set of 10 buckets



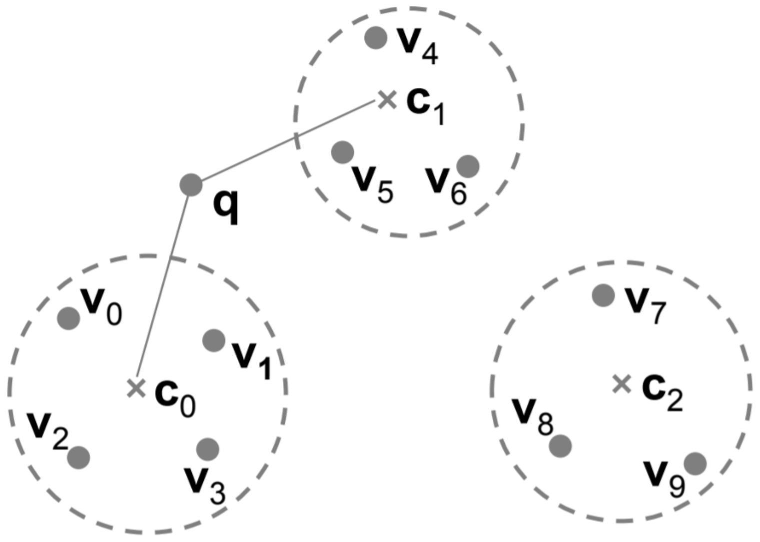
Quantization

- What's quantization in high-dimensional space?
  - It's basically clustering, e.g., k-means



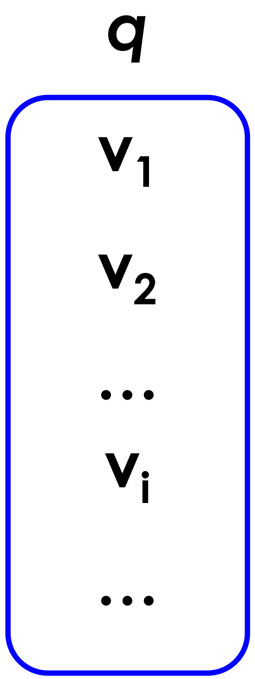
IVF_FLAT

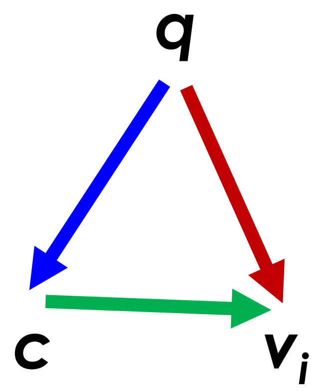
- Index phase
  - Cluster n vectors into K clusters (quantization)
  - Centroids: $c_0 \ldots c_{K-1}$
- Search phase
  - Given a query q, find the closest u clusters based on centroids
    - u: user-defined parameter
  - Only scan the vectors in the u clusters



IVF_FLAT

- Question: how to quickly compute the similarity between **q** and a vector $\mathbf{v}_i$ in a cluster?
- Naïve approach
  - A for-loop to compute dist(**q**,$\mathbf{v}_i$)
  - *d* steps (where *d* is dimensionality, e.g., *d* = 1000)
- Better solutions?
  - Remember, we know the centroid **c**
  - We can pre-compute the distance of dist(**c**,$\mathbf{v}_i$)
- Then dist(**q**,$\mathbf{v}_i$) = dist(**q**,**c**) + dist(**c**,$\mathbf{v}_i$) (approx.)
  - Only need 1 step to compute distance for all v_i





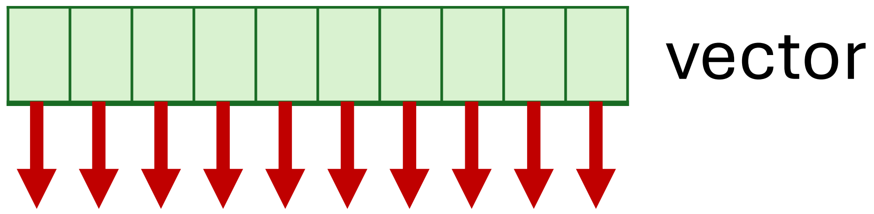
Compression

- How to reduce the space overhead of IVF_FLAT?
  - Compression
- Example
  - Youtube-8M data includes 1.4 billion vectors
  - Each vector takes 1024 dimensions (each float takes 32 bits)
  - 5.6TB space (memory!)

https://dl.acm.org/doi/10.14778/3424573.3424580

Compression: basic idea

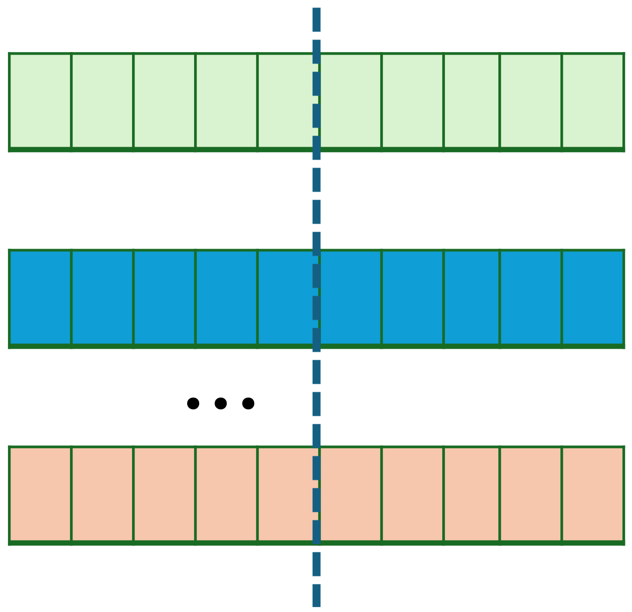
- Instead of using 32 bits to represent a float number
- Use L bits (e.g., L = 8)
- Think of 1-d quantization
- Every float number in a vector is quantized into $[0...2^L-1]$
- The 1.4billion vectors will take 1.4TB space (if L = 8)



Every float number is mapped to [0...255] (8 bits per number)

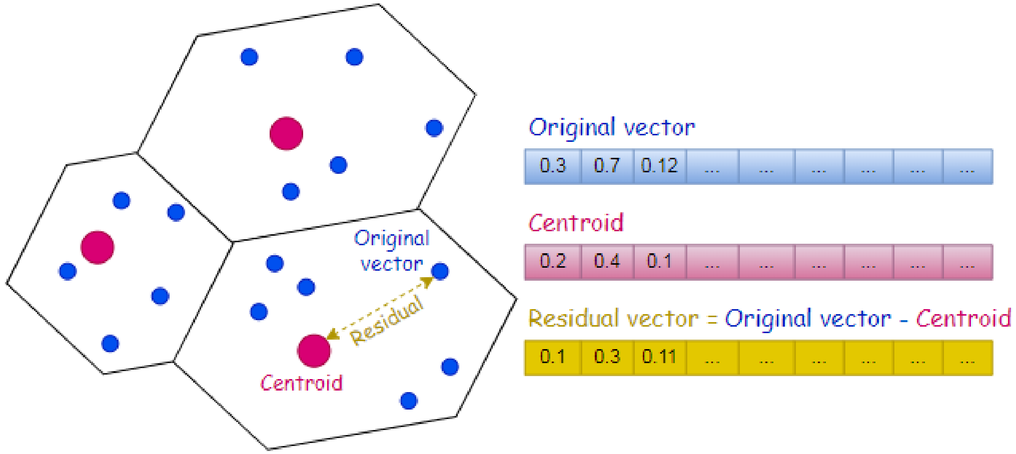
Compression: product quantization (PQ)

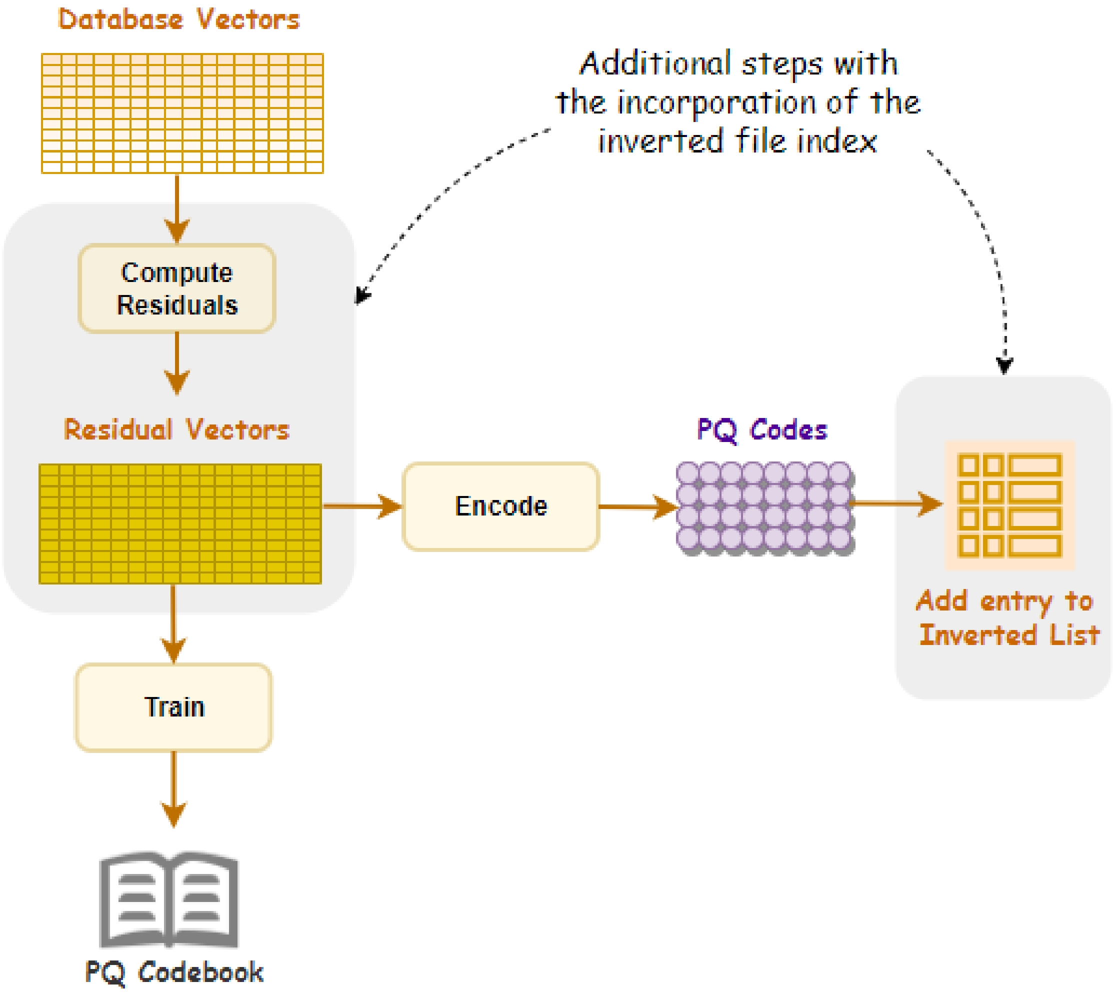
- How to further reduce the space overhead?
- Product quantization (PQ)
  - Key idea: compress between multiple dimensions
  - Every vector is partitioned into *M* subvectors, e.g., *M* = 8
  - Every subvector is compressed using *L* bits (e.g., *L* = 8)



IVF_PQ

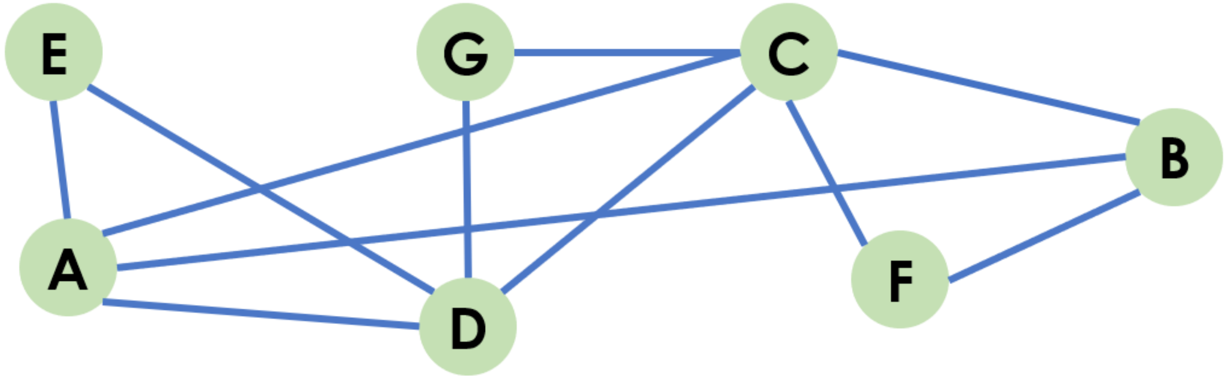
- Similar as IVF_FLAT
- Difference is that
  - Each cluster applies PQ
  - using residual vectors
- Search process is the same





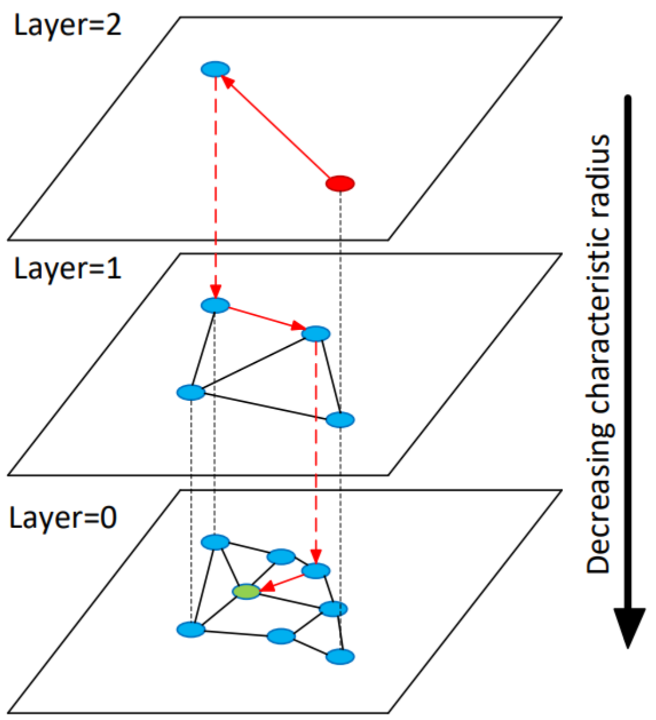
Graph-based vector index

- Key ideas
  - For each vector, pre-compute the nearest neighbors
  - Connect them using a graph
  - Convert vector search problem to graph traversal problem
- Navigable Small Worlds (NSW)
  - Add new vertices to the index
  - For each new vertex (vector), find the closest *m* neighbors seen so far and connect with them
  - Balance: index construction time & query performance



Graph-based vector index

- Hierarchical Navigable Small Worlds (HNSW)
  - Skip list + NSW
  - Multi-layered NSW
  - Address the “bad” entry point issue
    - If the entry point is not selected properly, the search path is long



https://arxiv.org/ftp/arxiv/papers/1603/1603.09320.pdf

Overview

- Retrieval Augmented Generation (RAG)
- Vector DBs
- AI agents

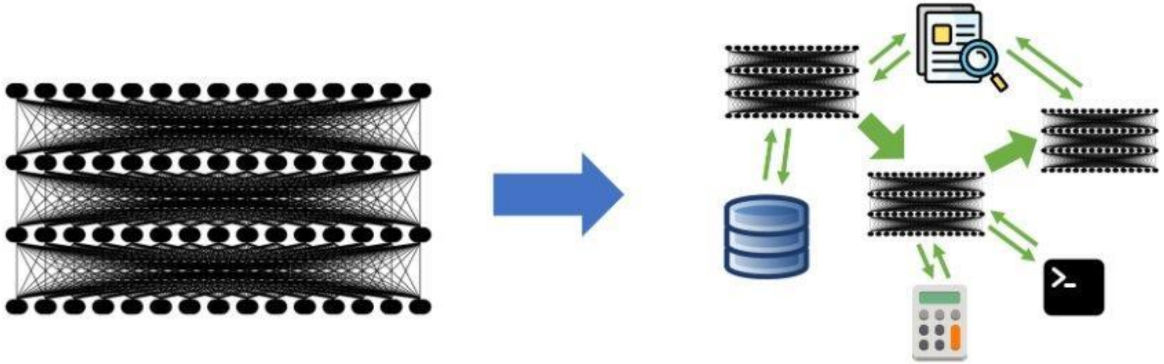
What are future AI applications like?

Generative

- Generate content like text & image

Agentic

- Execute complex tasks on behalf of human



Zaharia et al. 2024. The Shift from Models to Compound AI Systems

Examples of agentic AI

- Personal assistants
- Autonomous robots
- Gaming agents

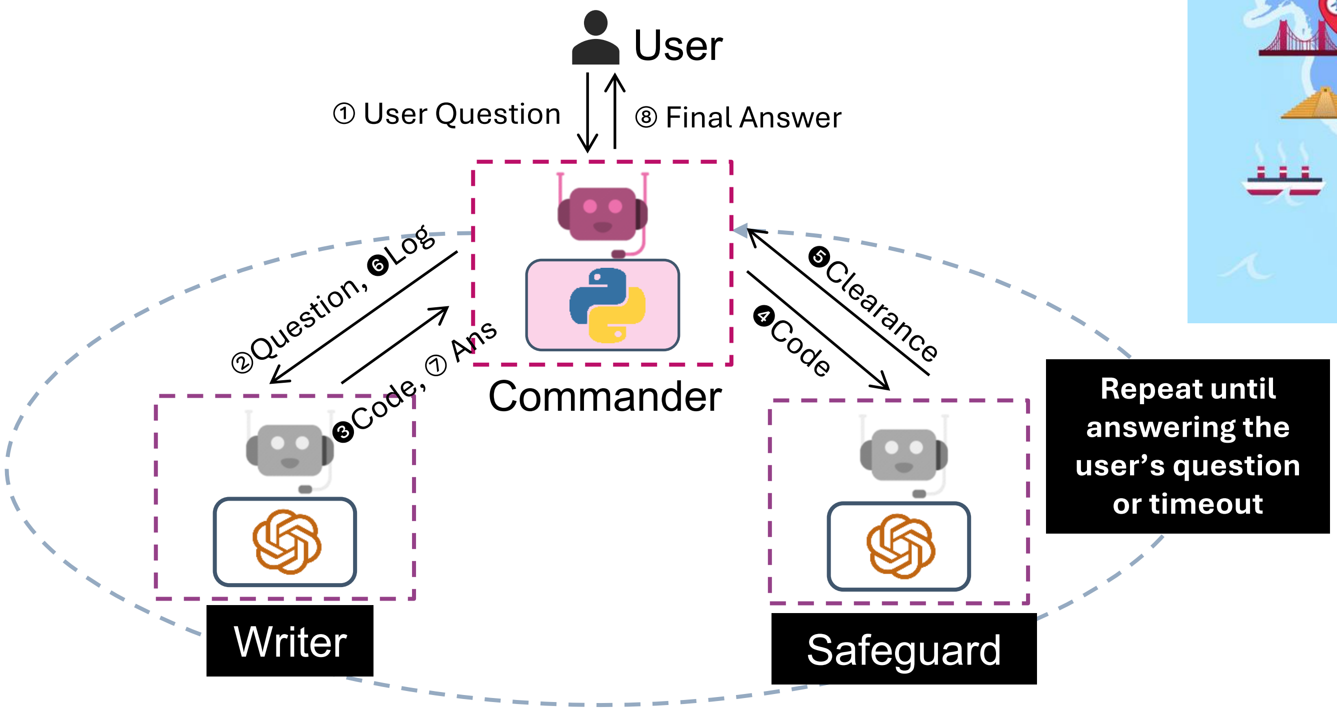
- Science agents
- Web agents
- Software agents

Key benefits of agentic AI

- Useful Interface
  - Natural interaction with human agency
- Strong Capability
  - Operate with minimal human intervention
- Useful Architecture
  - Intuitive programming paradigm

Example workflow of agentic AI

What if we prohibit shipping from supplier 1 to roastery 2?

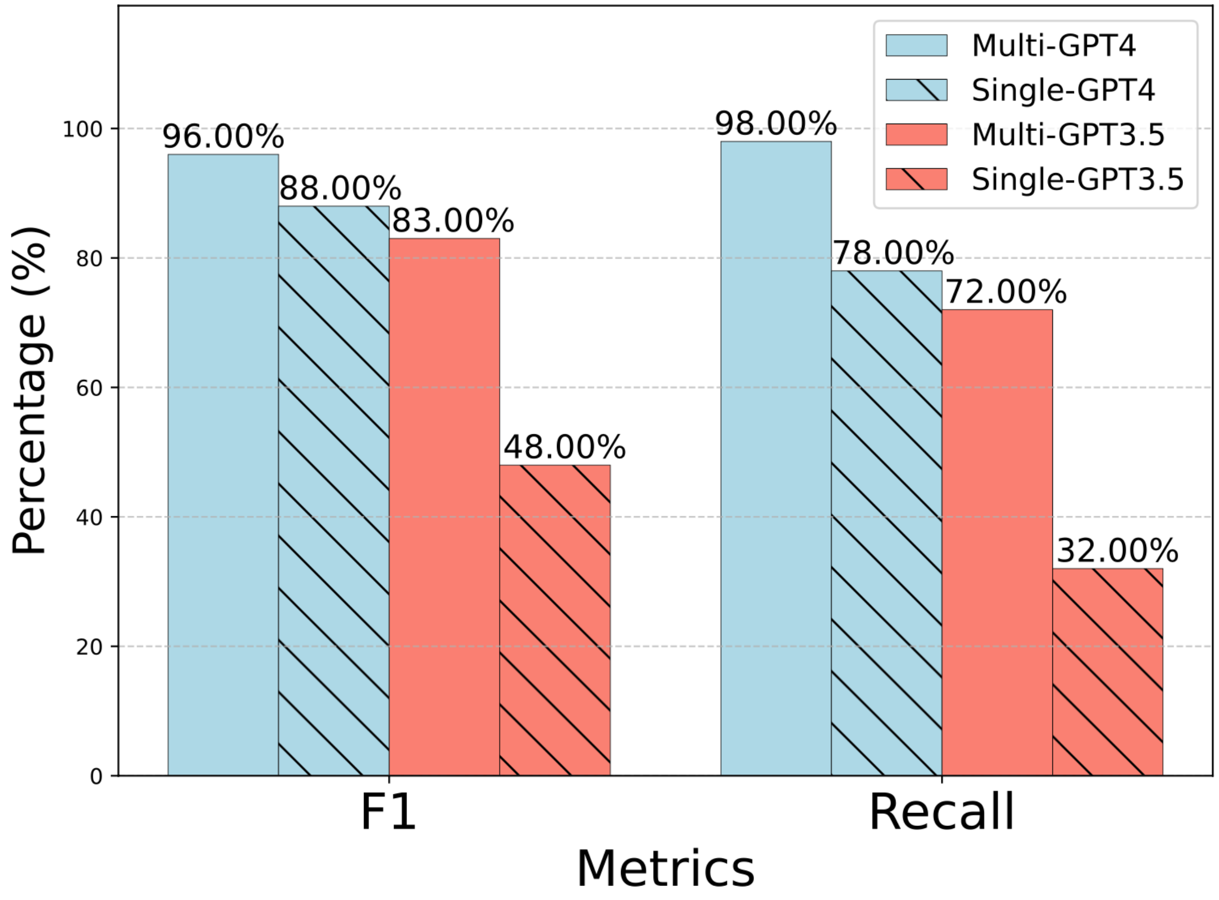


Example workflow of agentic AI


```
Optimize a model with 11 rows, 18 columns and 36 nonzeros
Model fingerprint: 0x8aa2c280
Variable types: 0 continuous, 18 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [2e+00, 1e+01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+01, 2e+02]
Found heuristic solution: objective 2900.0000000
Presolve time: 0.00s
Presolved: 11 rows, 18 columns, 36 nonzeros
Variable types: 0 continuous, 18 integer (0 binary)
Found heuristic solution: objective 2896.0000000

Root relaxation: objective 2.470000e+03, 11 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

*    0     0               0    2470.0000000 2470.00000  0.00%     -    0s

Explored 1 nodes (11 simplex iterations) in 0.00 seconds (0.00 work units)
Thread count was 14 (of 14 available processors)

Solution count 3: 2470 2896 2900
```

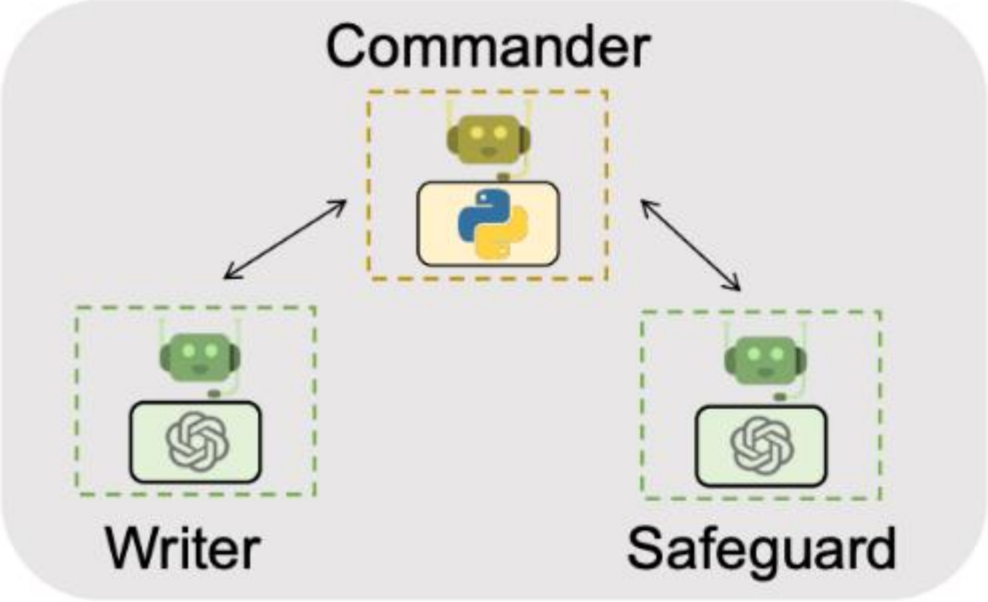
Agentic programming

- Handle more complex tasks / Improve response quality
  - Improve over natural iteration
  - Divide & conquer
  - Grounding & validation



Agentic programming

- Easy to understand, maintain, extend
  - Modular composition
  - Natural human participation
  - Fast & creative experimentation

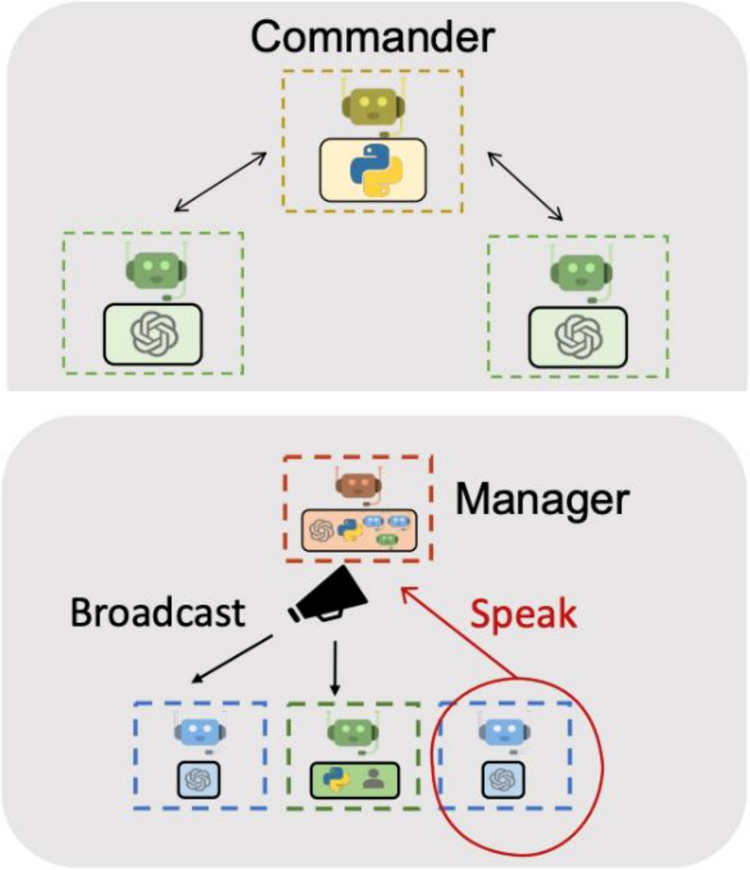


Agentic abstraction

Unify models, tools, human for compound AI systems

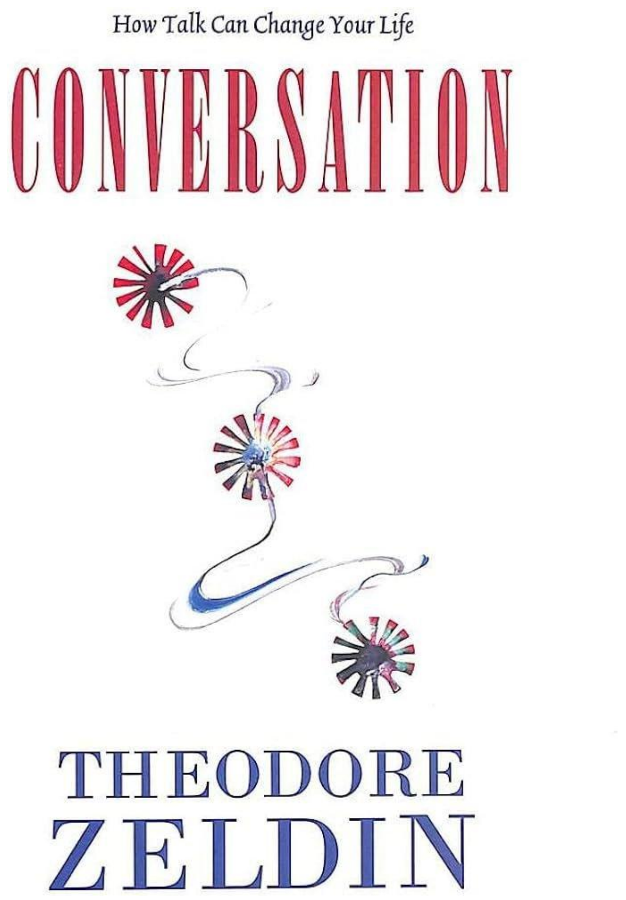
Multi-agent orchestration

- Static/dynamic
- NL/PL
- Context sharing/isolation
- Cooperation/competition
- Centralized/decentralized
- Intervention/automation

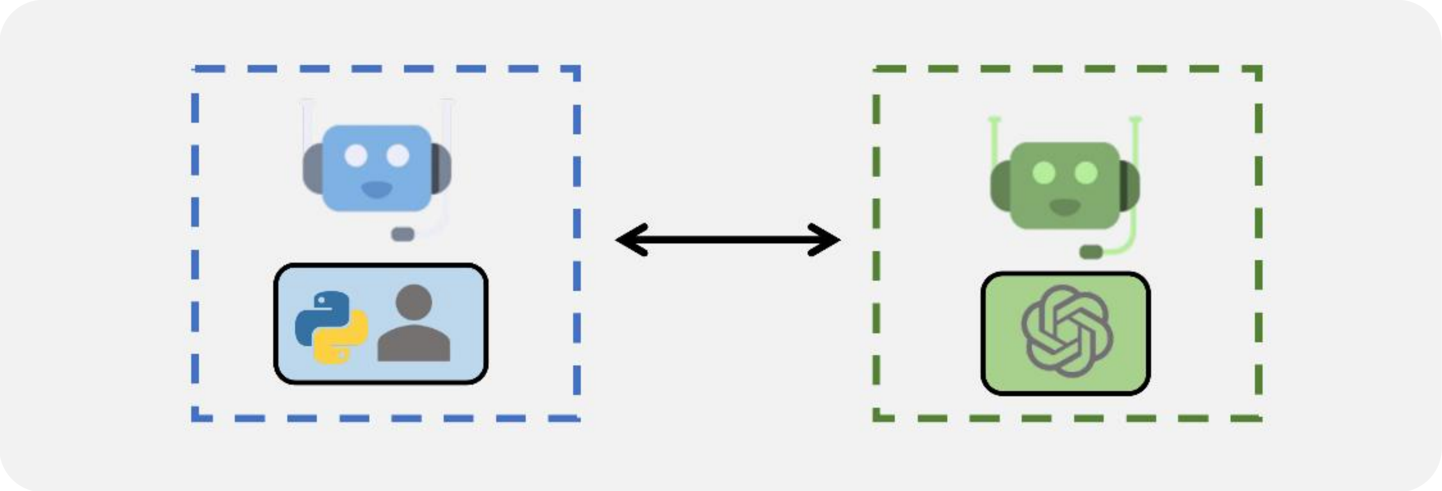


Agentic design patterns

- Conversation
- Prompting & reasoning
- Tool use
- Planning
- Integrating multiple models, modalities and memories



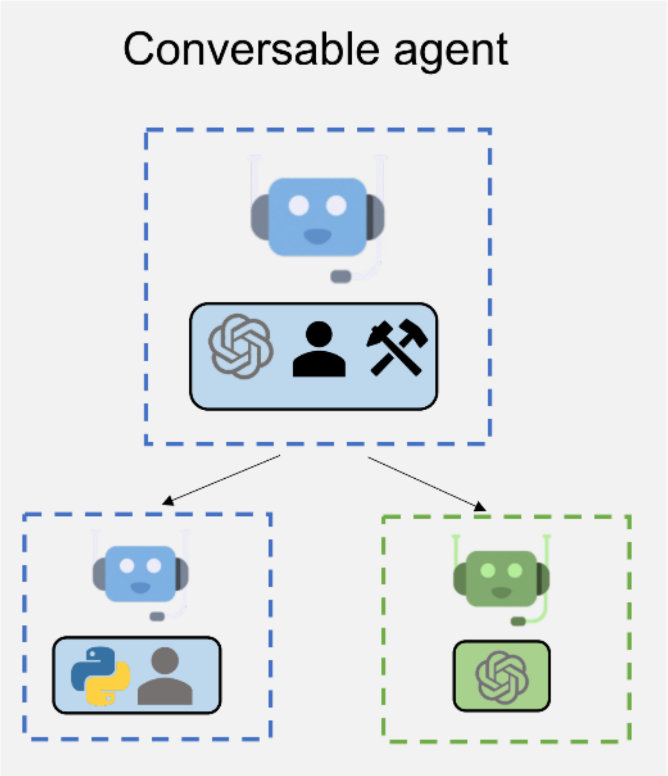
AutoGen: a programming framework for agentic AI



Initially developed in FLAML (Nov 2022)
Spined off to a standalone repo (October 2023)
Standalone GitHub organization AutoGen-AI (August 2024)

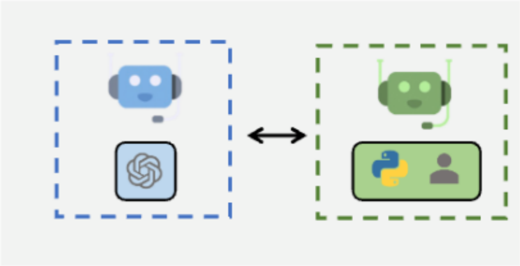
https://github.com/autogen-ai

Define agents:
Conversable & Customizable

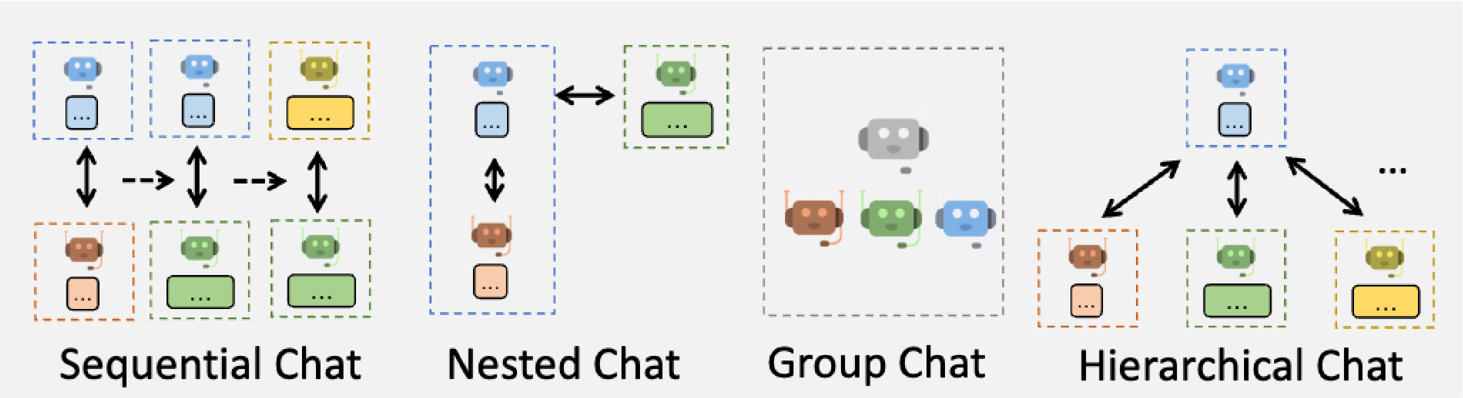


Agent Customization

Get them to talk:
Conversation Programming



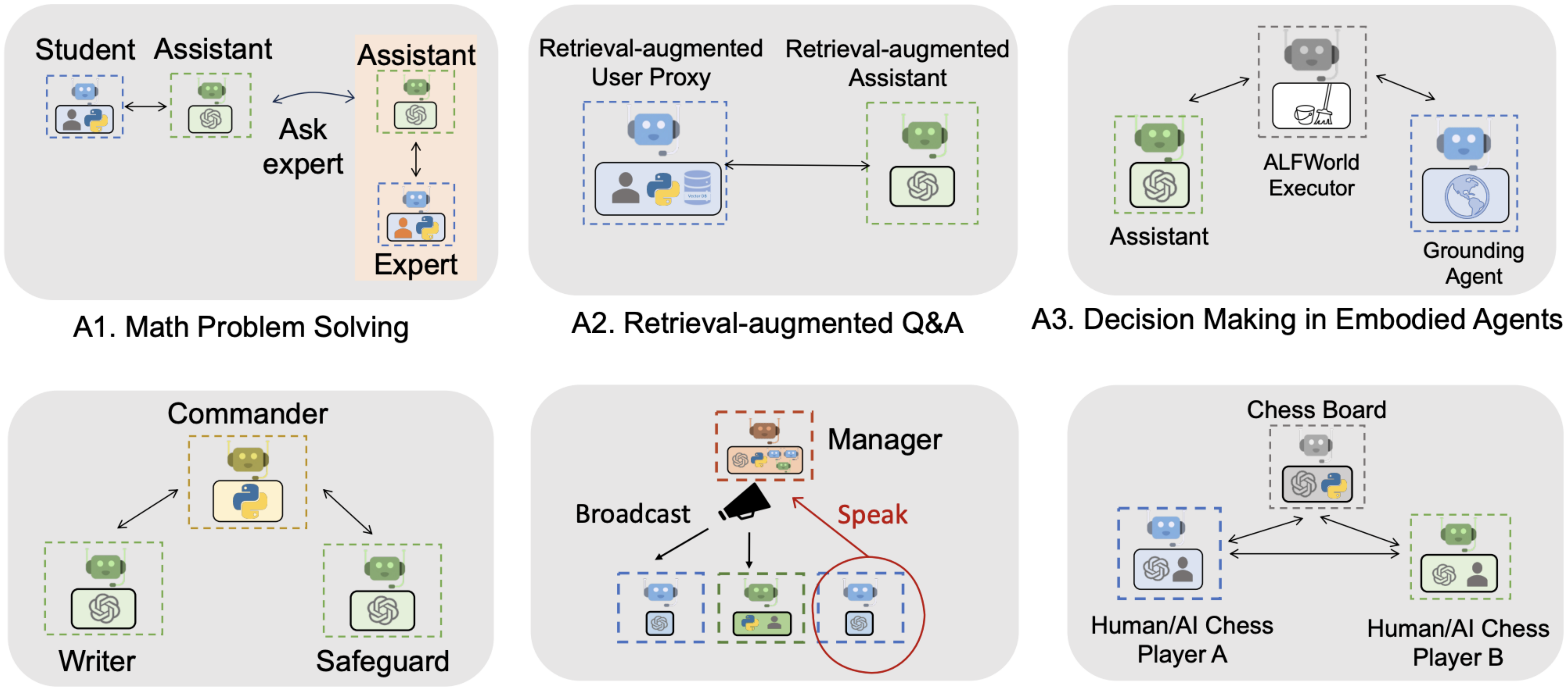
Multi-Agent Conversations



Flexible Conversation Patterns

Simple programming interface

```
writer = Writer("writer", llm_config=llm_config)
safeguard = autogen.AssistantAgent("safeguard", llm_config=llm_config)
commander = Commander("commander", llm_config=llm_config)
user = autogen.UserProxyAgent("user")


writer_chat_queue = [{"recipient": writer, "message": writer_init_messsage,
                      "summary_method": writer_success_summary}]
safeguard_chat_queue = [{"recipient": safeguard, "message": safeguard_init_message,
                         "max_turns": 1, "summary_method": safeguard_summary}]
commander.register_nested_chats(safeguard_chat_queue, trigger="writer")
commander.register_nested_chats(writer_chat_queue, trigger="user")


chat_res = user.initiate_chat(commander, message="What if we prohibit shipping from supplier 1 to roastery 2?")
```



A1. Math Problem Solving

A2. Retrieval-augmented Q&A

A3. Decision Making in Embodied Agents

A4. Supply-Chain Optimization

A5. Dynamic Task Solving with Group Chat

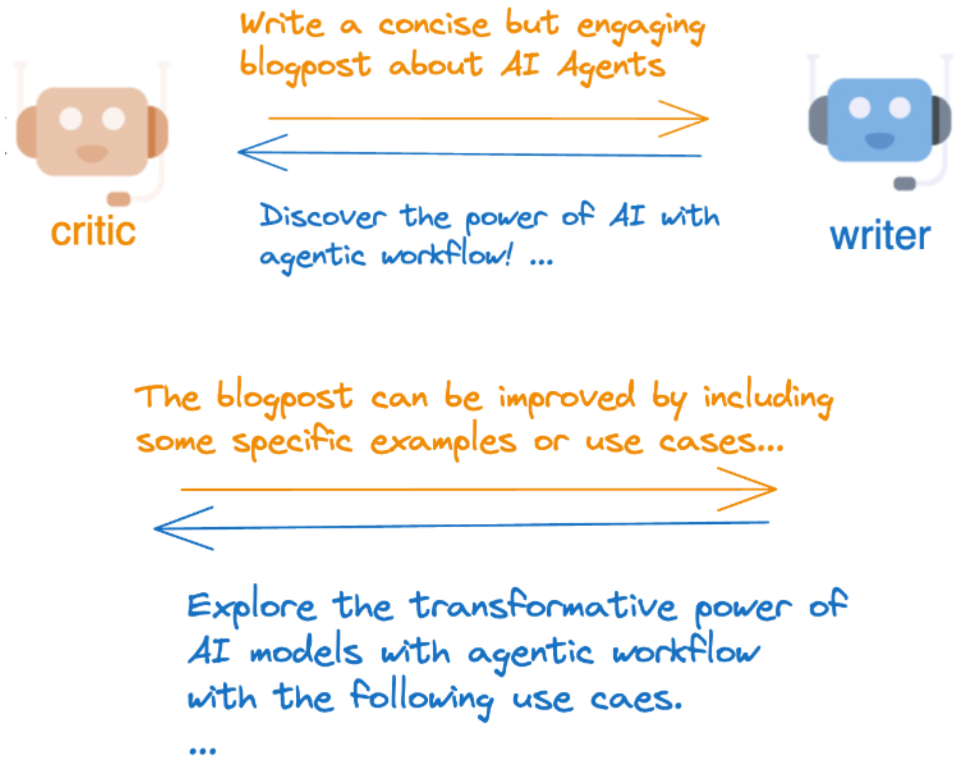
A6. Conversational Chess

For more examples: https://autogen-ai.github.io/autogen/docs/notebooks

Blogpost writing with reflection

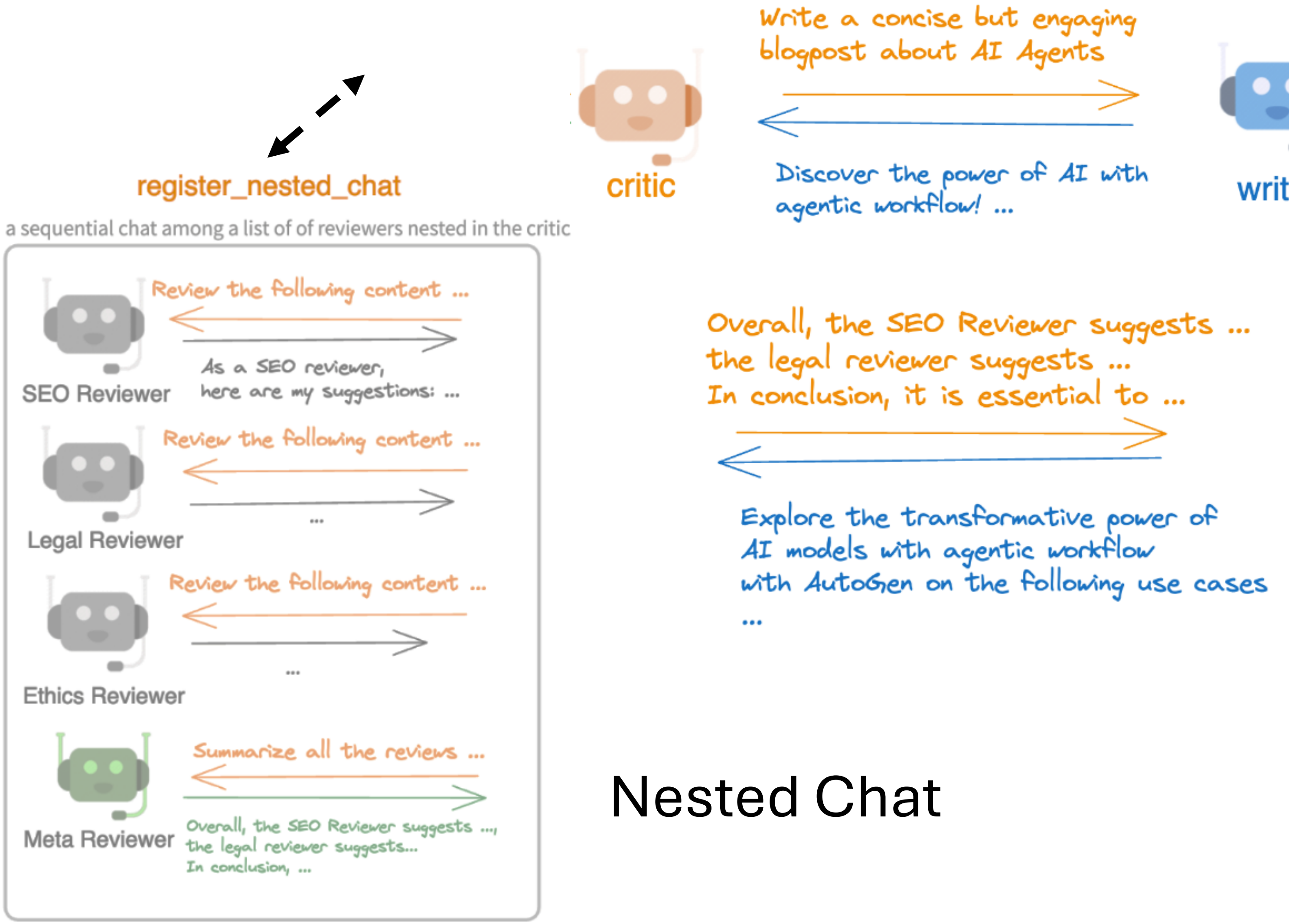
```python
writer = autogen.AssistantAgent(
    name="Writer",
    system_message="You are a writer...",
    llm_config=llm_config,
)

critic = autogen.AssistantAgent(
    name="Critic",
    is_termination_msg=lambda x: x.get("content", "").find("TERMINATE") >= 0,
    llm_config=llm_config,
    system_message="You are a critic...",
)

critic.initiate_chat(
    recipient=writer,
    message=task,
    max_turns=2,
    summary_method="last_msg"
)
```

Two-Agent Reflection



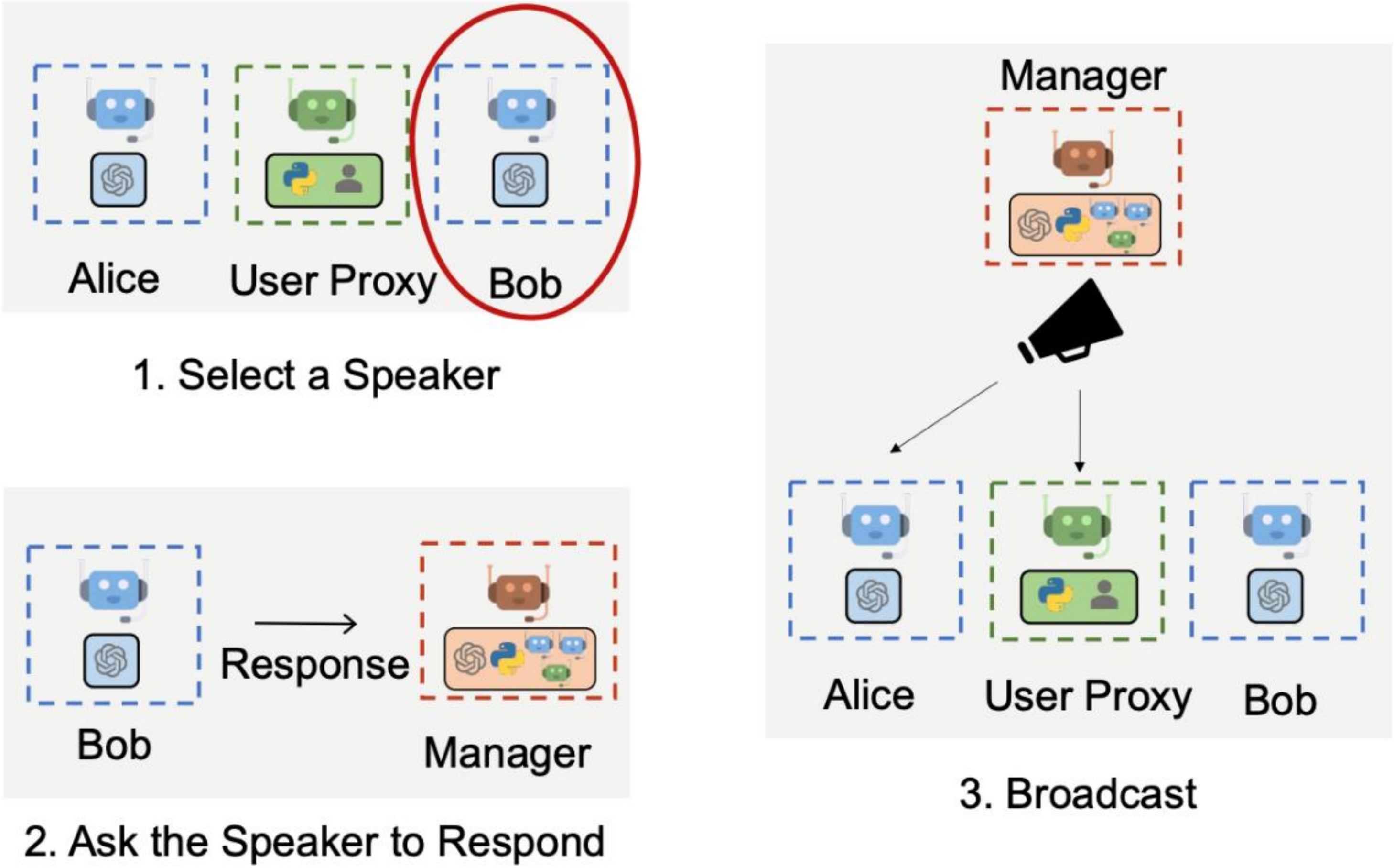
Blogpost writing with advanced reflection

```
critic.register_nested_chats(
    review_chats,
    trigger=writer,
)

critic.initiate_chat(
    recipient=writer,
    message=task,
    max_turns=2,
    summary_method="last_msg"
)
```



Nested Chat

Complex task planning and solving with group chat



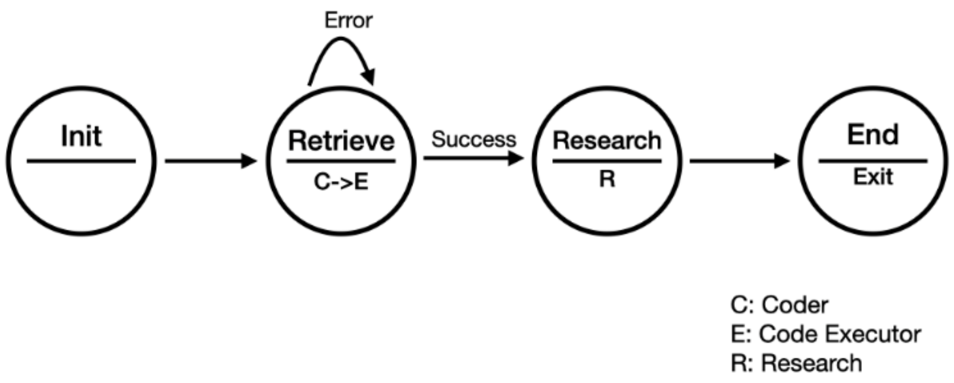
Complex task planning and solving with group chat

StateFlow - Build State-Driven Workflows with Customized Speaker Selection in GroupChat

February 29, 2024 · 7 min read

Yiran Wu
PhD student at Pennsylvania State University

TL;DR: Introduce Stateflow, a task-solving paradigm that conceptualizes complex task-solving processes backed by LLMs as state machines. Introduce how to use GroupChat to realize such an idea with a customized speaker selection function.

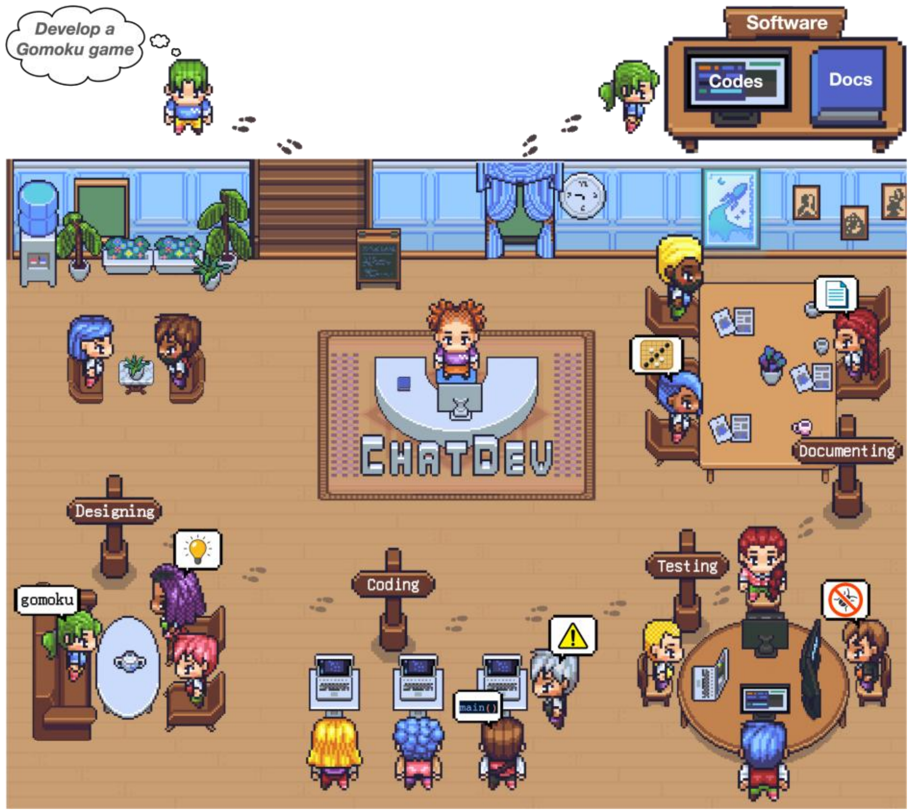



```python
def state_transition(last_speaker, groupchat):
    messages = groupchat.messages

    if last_speaker is initializer:
        # init -> retrieve
        return coder
    elif last_speaker is coder:
        # retrieve: action 1 -> action 2
        return executor
    elif last_speaker is executor:
        if messages[-1]["content"] == "exitcode: 1":
            # retrieve --(execution failed)--> retrieve
            return coder
        else:
            # retrieve --(execution success)--> research
            return scientist
    elif last_speaker == "Scientist":
        # research -> end
        return None


groupchat = autogen.GroupChat(
    agents=[initializer, coder, executor, scientist],
    messages=[],
    max_round=20,
    speaker_selection_method=state_transition,
)
```

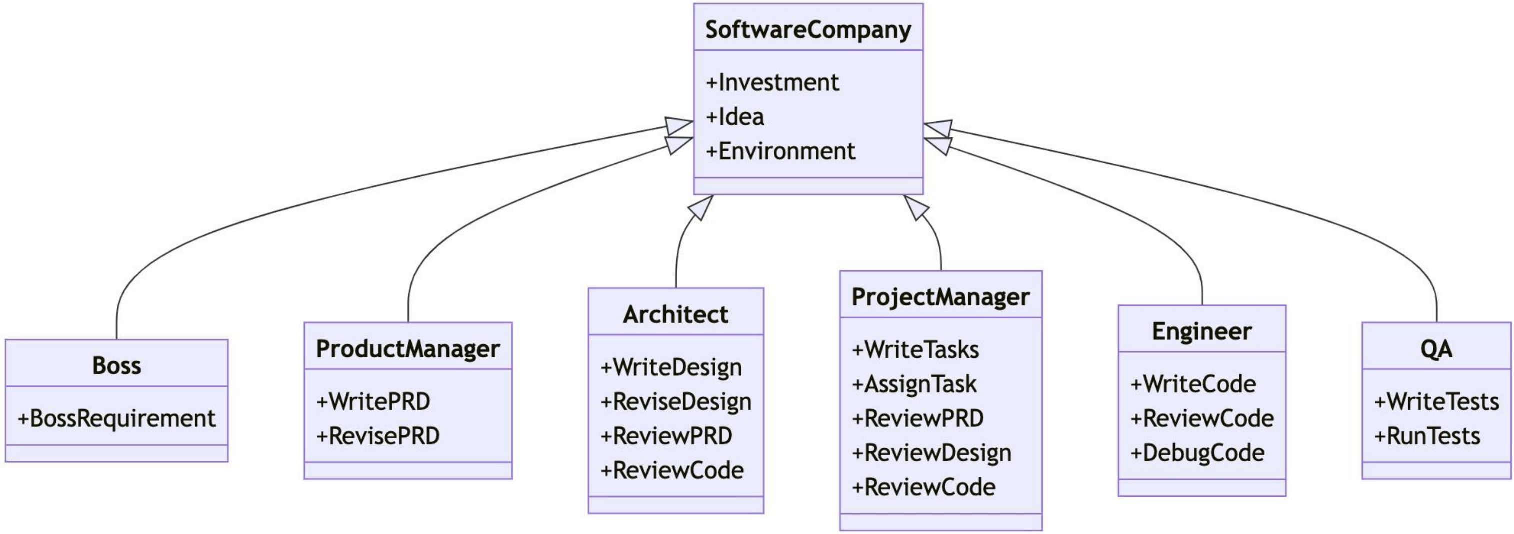
Other multi-agent systems

Many solutions are more application/software engineering oriented. Lots research opportunities like

- Result interpretability and controllability
- Scalability
- Some guarantee & trustworthy AI
- Collaboration among RL- and LLM- agents

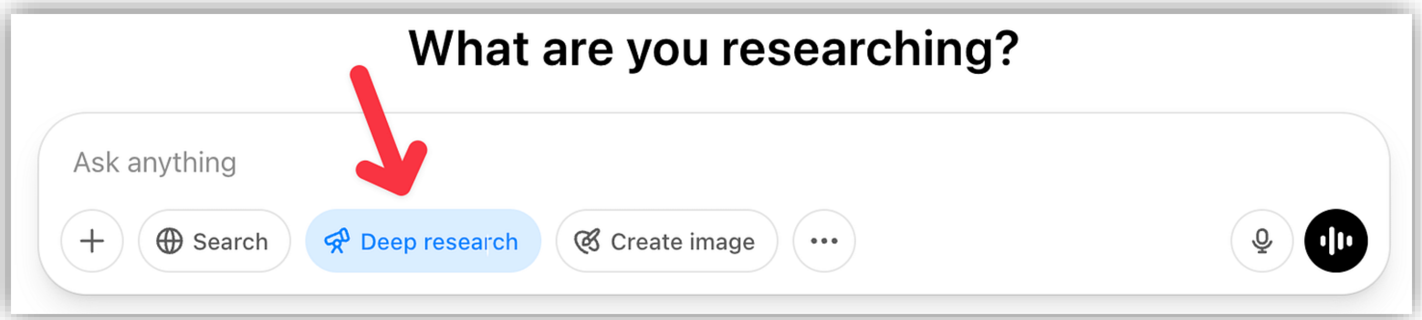


ChatDev



MetaGPT

Deep Research



- We will have Bruce Yang from Agnes AI to talk about deep research in action next week.

- Reading
  - Try out deep research from Gemini, OpenAI and Agnes AI, pick a topic you like, and compare them
  - Send it to TA through Canvas message
  - You can use Deep Research to research on Deep Research

Logistics

- HW3
- Project mid-term report