CS 546 – Advanced Topics in NLP

Dilek Hakkani-Tür

UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

CONVAI

CDS

Siebel School of
Computing
and Data Science

Topics for Today

Retrieval Augmented Generation (RAG)

- Introduction
- Retrieval Methods
- Generation
- End-to-End Methods and Evaluation
- RAG Applications

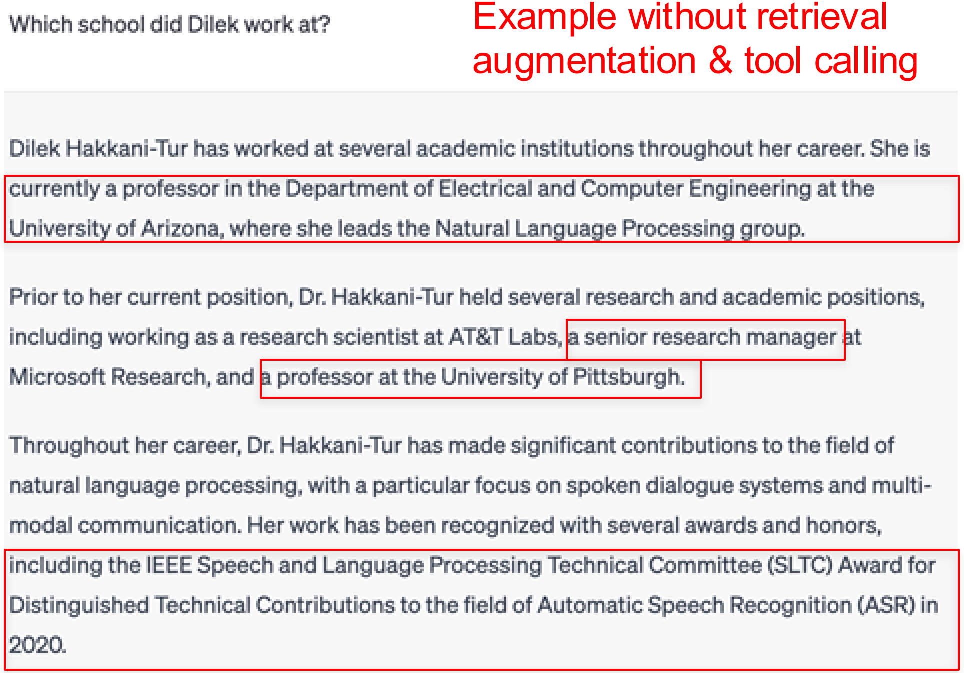
Introduction

- LMs suffer from important limitations hindering a broader deployment
    - **Knowledge cutoff:** LLMs are typically pre-trained using self-supervision from text data, but knowledge is dynamic.
    - **Private data:** Knowledge can be retrieved from private sources but may not be available for model training.
    - **Hallucinations/confabulations and Verifiability:** LMs can provide nonfactual but seemingly plausible predictions, and it is hard to tell whether the response is accurate.
    - **Parameter inefficiency:** LLMs are typically trained to perform statistical language modeling given a single parametric model and a limited context, usually the previous or surrounding tokens. Encoding all knowledge in model weights does not scale.
- **Idea:** Instead of forcing a model to memorize everything, can we let it look up information?

Issues with the current LLMs: Factual Accuracy

- Pretrained language models are known to hallucinate (Ji et al., ACM Comput. Surv., 2022)
- Even for summarization, where the content is given, hallucination rates are around 1-2%(e.g., OpenAI GPT-4o 1.5%) (https://huggingface.co/spaces/vectara/Hallucination-evaluation-leaderboard)
  - And can be higher for other tasks
- Hallucinations are not acceptable for real applications!

Slide from the first lecture. Example without retrieval augmentation & tool calling



Which school did Dilek work at?

Dilek Hakkani-Tur has worked at several academic institutions throughout her career. She is currently a professor in the Department of Electrical and Computer Engineering at the University of Arizona, where she leads the Natural Language Processing group.

Prior to her current position, Dr. Hakkani-Tur held several research and academic positions, including working as a research scientist at AT&T Labs, a senior research manager at Microsoft Research, and a professor at the University of Pittsburgh.

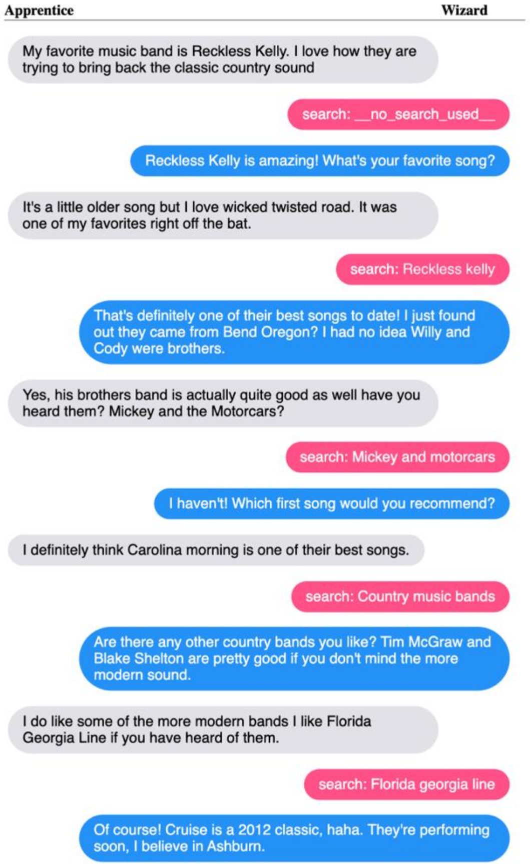
Throughout her career, Dr. Hakkani-Tur has made significant contributions to the field of natural language processing, with a particular focus on spoken dialogue systems and multi-modal communication. Her work has been recognized with several awards and honors, including the IEEE Speech and Language Processing Technical Committee (SLTC) Award for Distinguished Technical Contributions to the field of Automatic Speech Recognition (ASR) in 2020.

Augmented LLMs

- Recent line of research enabling LMs to access knowledge that may not be stored in its weights:
    - Information retrieval or web search
    - Tool/API calling, e.g., calculator (next week)
    - Code execution
    - Reasoning (next week)

Integrating Search

- LaMDA (next week)
- Internet-Augmented Dialogue Generation and Wizard of Internet (Komeili et al., 2021)
  - LM decides to generate a search query based on a prompt
- ReAct also allows LMs to use different tools, such as search and Lookup in Wikipedia (Yao et al., 2022)
  - Uses few-shot prompting



Apprentice **Wizard**

My favorite music band is Reckless Kelly. I love how they are trying to bring back the classic country sound

search: __no_search_used__

Reckless Kelly is amazing! What's your favorite song?

It's a little older song but I love wicked twisted road. It was one of my favorites right off the bat.

search: Reckless kelly

That's definitely one of their best songs to date! I just found out they came from Bend Oregon? I had no idea Willy and Cody were brothers.

Yes, his brothers band is actually quite good as well have you heard them? Mickey and the Motorcars?

search: Mickey and motorcars

I haven't! Which first song would you recommend?

I definitely think Carolina morning is one of their best songs.

search: Country music bands

Are there any other country bands you like? Tim McGraw and Blake Shelton are pretty good if you don't mind the more modern sound.

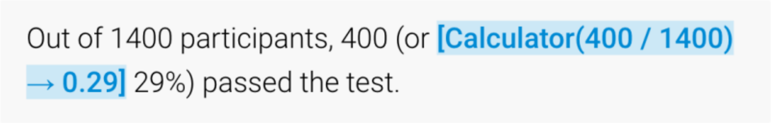
I do like some of the more modern bands I like Florida Georgia Line if you have heard of them.

search: Florida georgia line

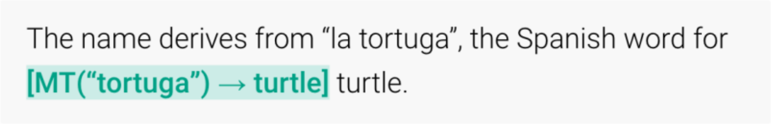
Of course! Cruise is a 2012 classic, haha. They're performing soon, I believe in Ashburn.

LMs Augmented with API Calls

- LM generates tool/API call, possibly with arguments, which gets executed during training or during/before decoding, e.g.,
  - Toolformer (Schick et al., 2023)

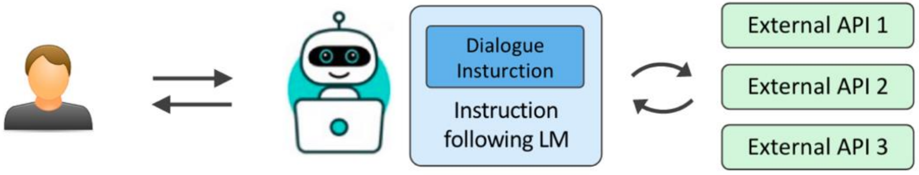


Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.



The name derives from “la tortuga”, the Spanish word for [MT(“tortuga”) → turtle] turtle.

  - AutoTOD (Xu et al., 2024) for task-oriented dialogue



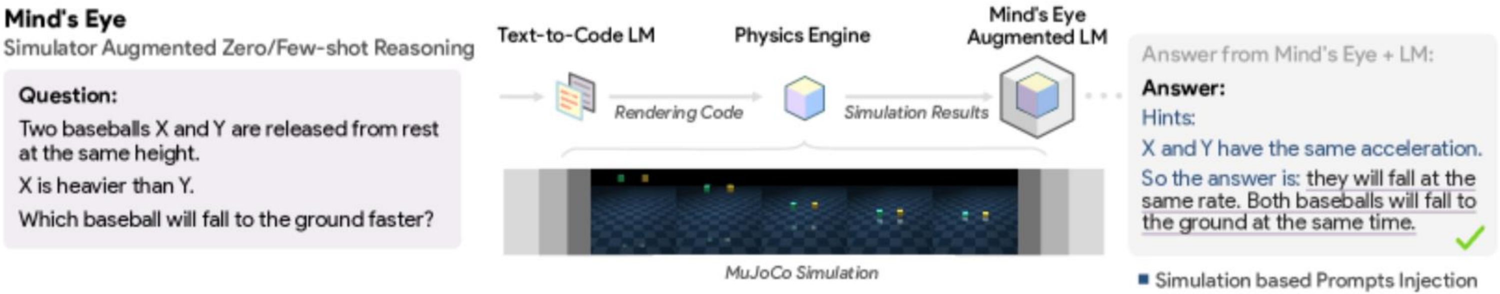
(c) Autonomous Agent

Search and Navigate

- Aiming to build agents that can navigate the open-ended internet with the goal of completing specific goals, such as information seeking or buying things.
- In addition to search, these have actions, such as clicking on links.
    - WebGPT (Nakano et al., 2021) to answer long-form questions
    - WebShop (Yao et al., NeurIPS 2022) to purchase a product
    - Or more general ones, such as Mind2Web (Deng et al, NeurIPS 2023) and WebLINX (Lu et al, ICML 2024)
    - InfoGent (Reddy et al, NAACL Findings, 2025)
    - Web agents (in a few weeks)

Augmentation with Planning & Code Execution

- Mind's Eye (Liu et al, 2022) uses an LM to generate code, which is executed in a simulation environment and the output is used to generate an answer.



Mind's Eye

Simulator Augmented Zero/Few-shot Reasoning

Question:

Two baseballs X and Y are released from rest at the same height.

X is heavier than Y.

Which baseball will fall to the ground faster?

Answer from Mind's Eye + LM:

Answer:

Hints:

X and Y have the same acceleration.

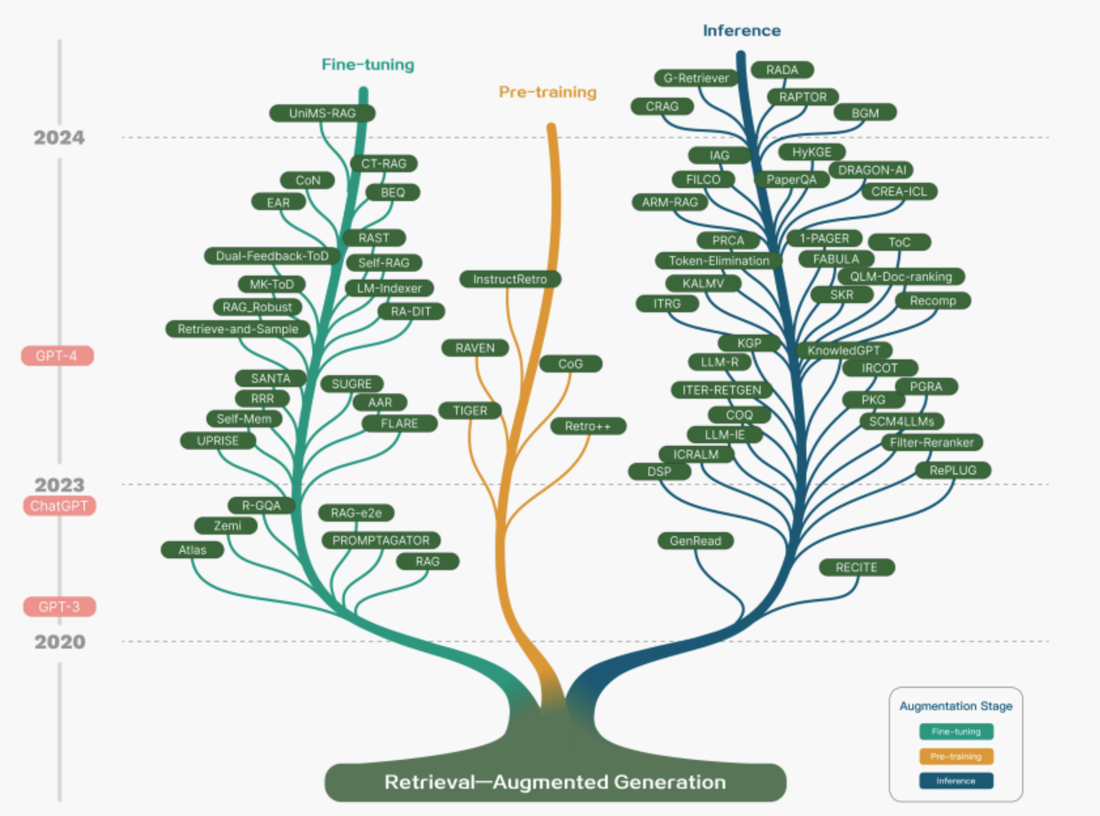
So the answer is: they will fall at the same rate. Both baseballs will fall to the ground at the same time. ✓

- Helper (Sarch et al., 2023) for embodied AI converts user instructions to code, which get executed (more when talking about embodied agents in a few weeks).

Retrieval Augmented Generation (RAG)

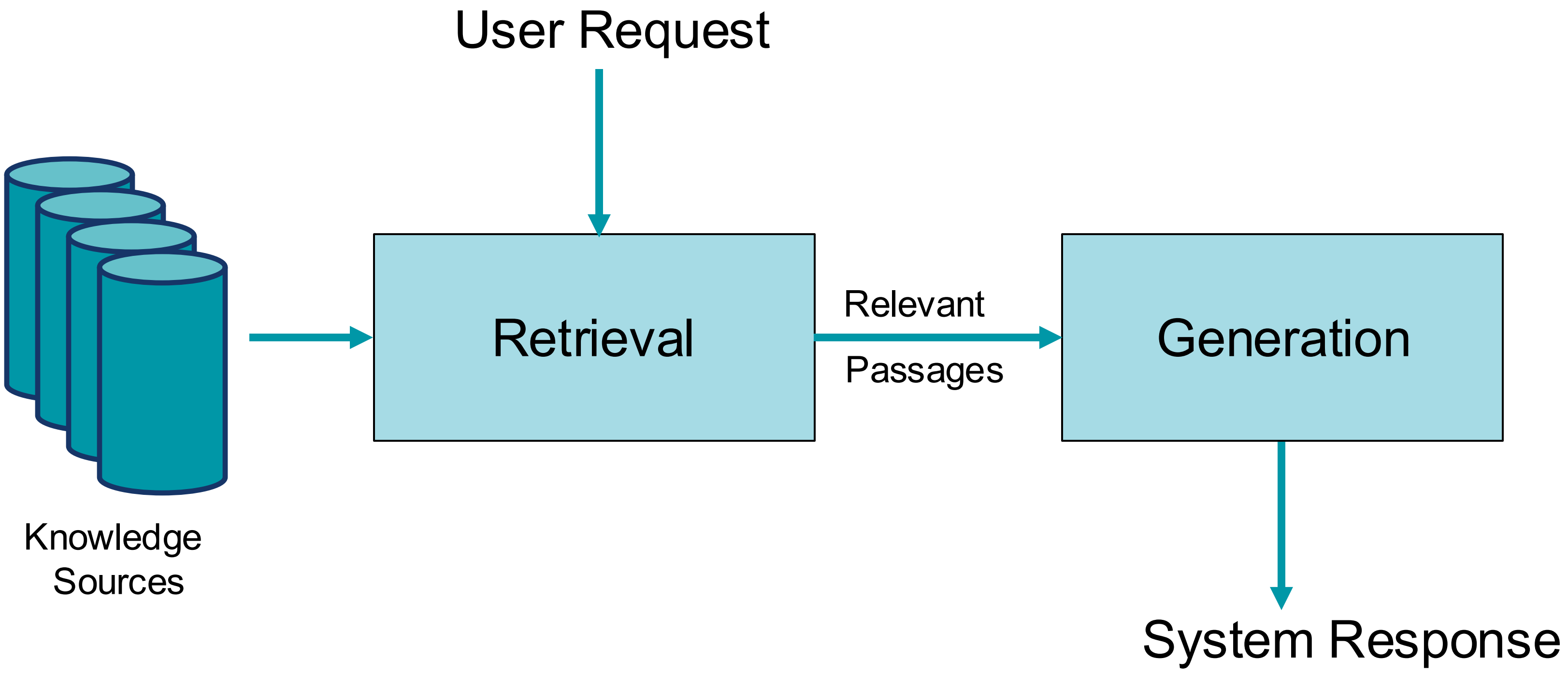
- Index a given set of documents/resources, such as wikipedia:
    - Sparse retrievers using bag-of-words representations
    - Dense retrievers using embeddings
- Conditioning LMs on the retrieved set of documents
    - Concatenating them to context
    - Using cross-attention
- Retriever training
    - Frozen, using similarity functions
    - Fine-tuned, with in-domain training data
    - End-to-end, trained jointly with response generation

RAG Timeline



- Figure from (Gao et al, March 2024) https://arxiv.org/pdf/2312.10997

RAG – Conceptual Overview



Three Paradigms of RAG

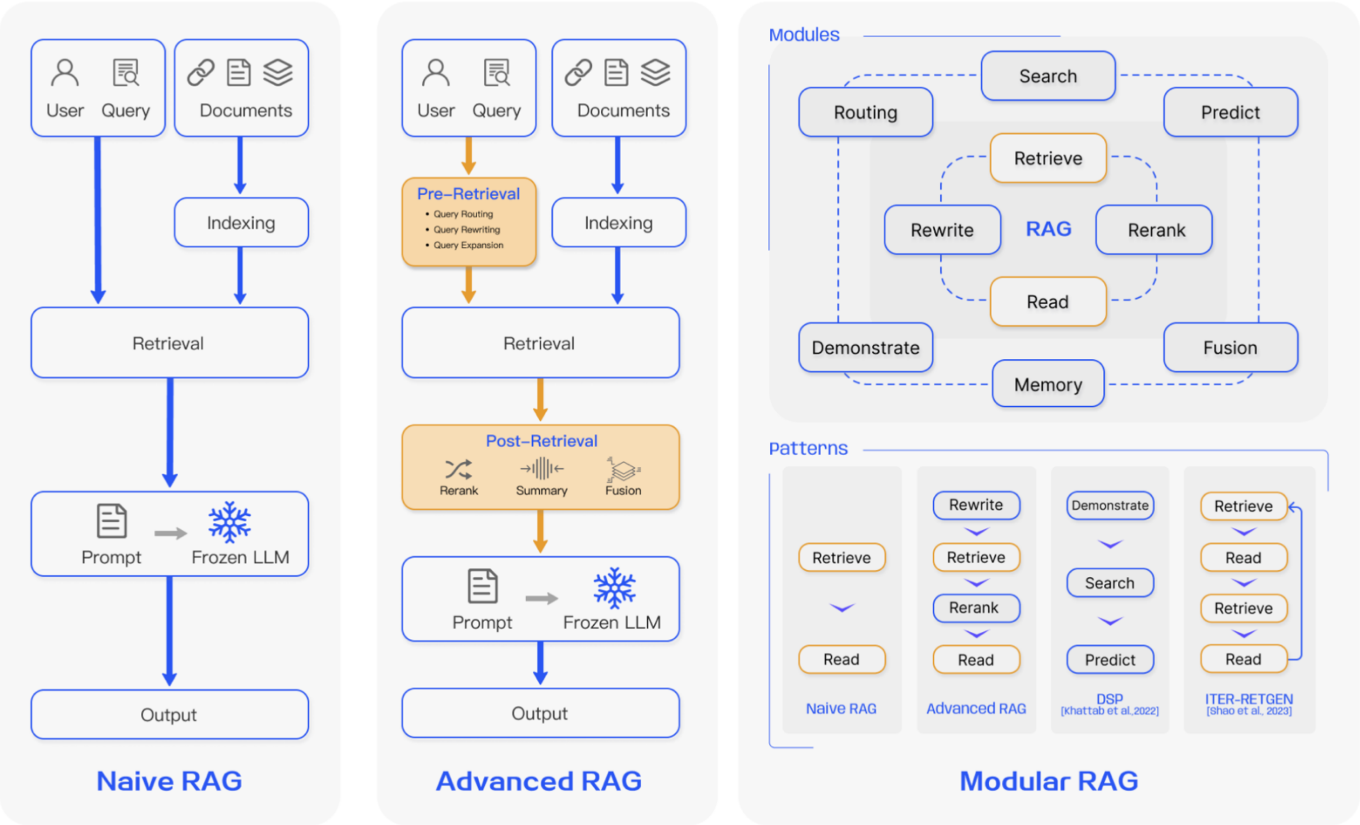
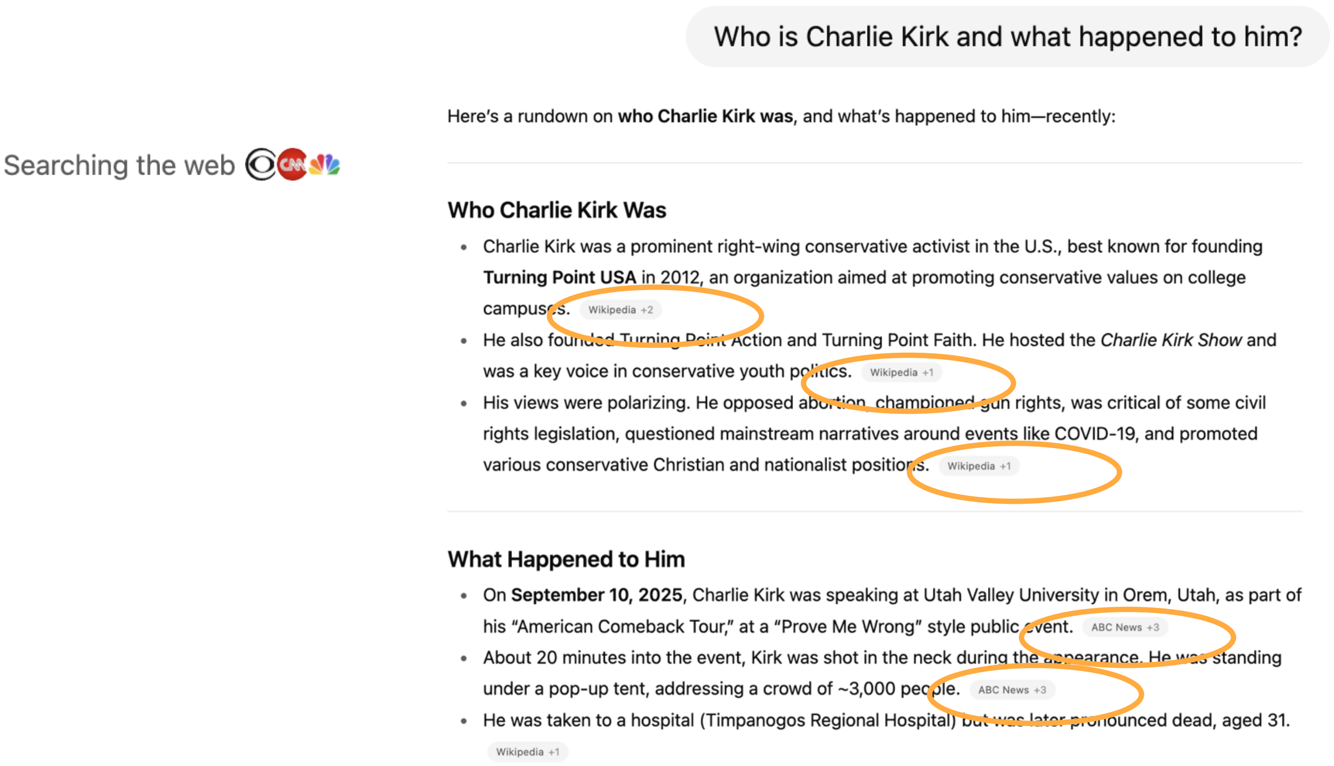


Figure from (Gao et al, March 2024) https://arxiv.org/pdf/2312.10997

RAG in Action

OpenAI ChatGPT (Sept 12, 2025):



Who is Charlie Kirk and what happened to him?

Searching the web

Here's a rundown on **who Charlie Kirk was**, and what's happened to him—recently:

Who Charlie Kirk Was

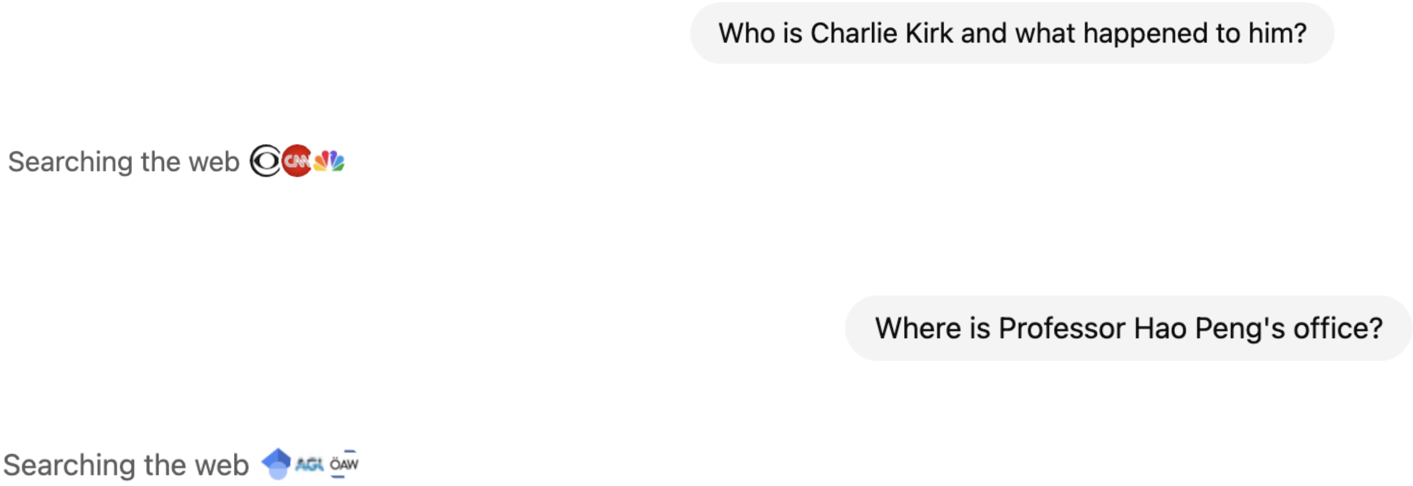
- Charlie Kirk was a prominent right-wing conservative activist in the U.S., best known for founding **Turning Point USA** in 2012, an organization aimed at promoting conservative values on college campuses. Wikipedia +2
- He also founded Turning Point Action and Turning Point Faith. He hosted the *Charlie Kirk Show* and was a key voice in conservative youth politics. Wikipedia +1
- His views were polarizing. He opposed abortion, championed gun rights, was critical of some civil rights legislation, questioned mainstream narratives around events like COVID-19, and promoted various conservative Christian and nationalist positions. Wikipedia +1

What Happened to Him

- On **September 10, 2025**, Charlie Kirk was speaking at Utah Valley University in Orem, Utah, as part of his "American Comeback Tour," at a "Prove Me Wrong" style public event. ABC News +3
- About 20 minutes into the event, Kirk was shot in the neck during the appearance. He was standing under a pop-up tent, addressing a crowd of ~3,000 people. ABC News +3
- He was taken to a hospital (Timpanogos Regional Hospital) but was later pronounced dead, aged 31. Wikipedia +1

RAG in Action (cont.)

OpenAI ChatGPT (Sept 12, 2025):



Who is Charlie Kirk and what happened to him?

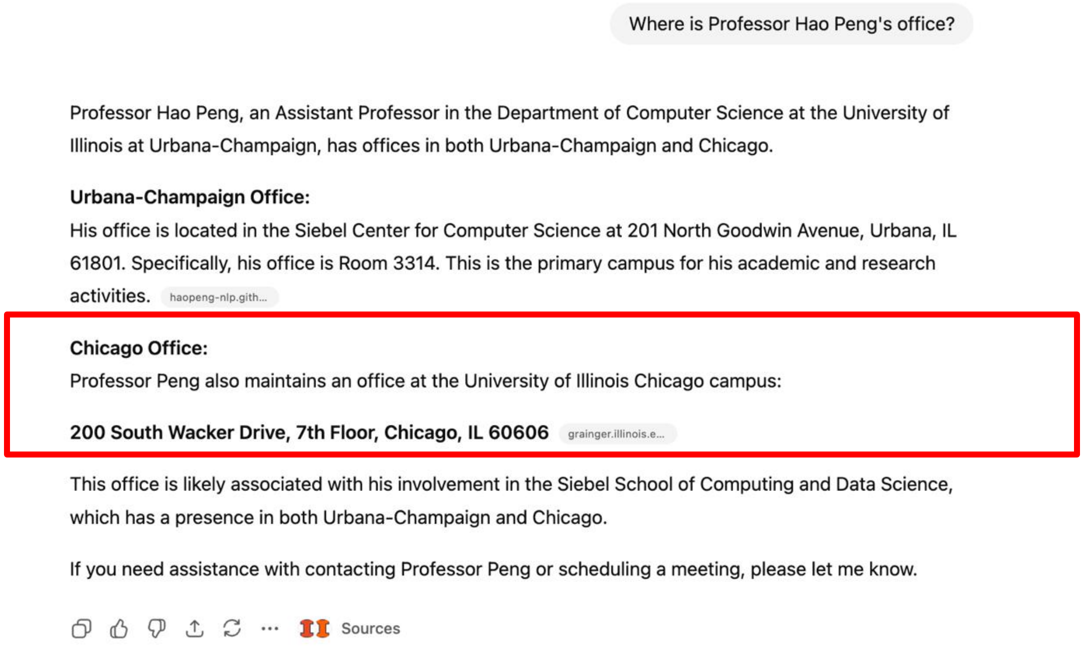
Searching the web

Where is Professor Hao Peng's office?

Searching the web

RAG in Action (cont.)

OpenAI ChatGPT (Sept 12, 2025):



Where is Professor Hao Peng's office?

Professor Hao Peng, an Assistant Professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign, has offices in both Urbana-Champaign and Chicago.

Urbana-Champaign Office:
His office is located in the Siebel Center for Computer Science at 201 North Goodwin Avenue, Urbana, IL 61801. Specifically, his office is Room 3314. This is the primary campus for his academic and research activities. haopeng-nlp.gith...

Chicago Office:
Professor Peng also maintains an office at the University of Illinois Chicago campus:

200 South Wacker Drive, 7th Floor, Chicago, IL 60606 grainger.illinois.e...

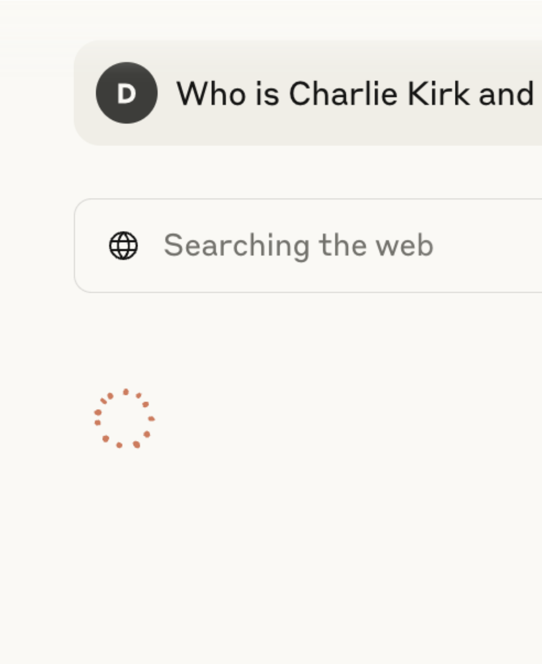
This office is likely associated with his involvement in the Siebel School of Computing and Data Science, which has a presence in both Urbana-Champaign and Chicago.

If you need assistance with contacting Professor Peng or scheduling a meeting, please let me know.

Hallucination can still happen, but links enable fact checking and verification.

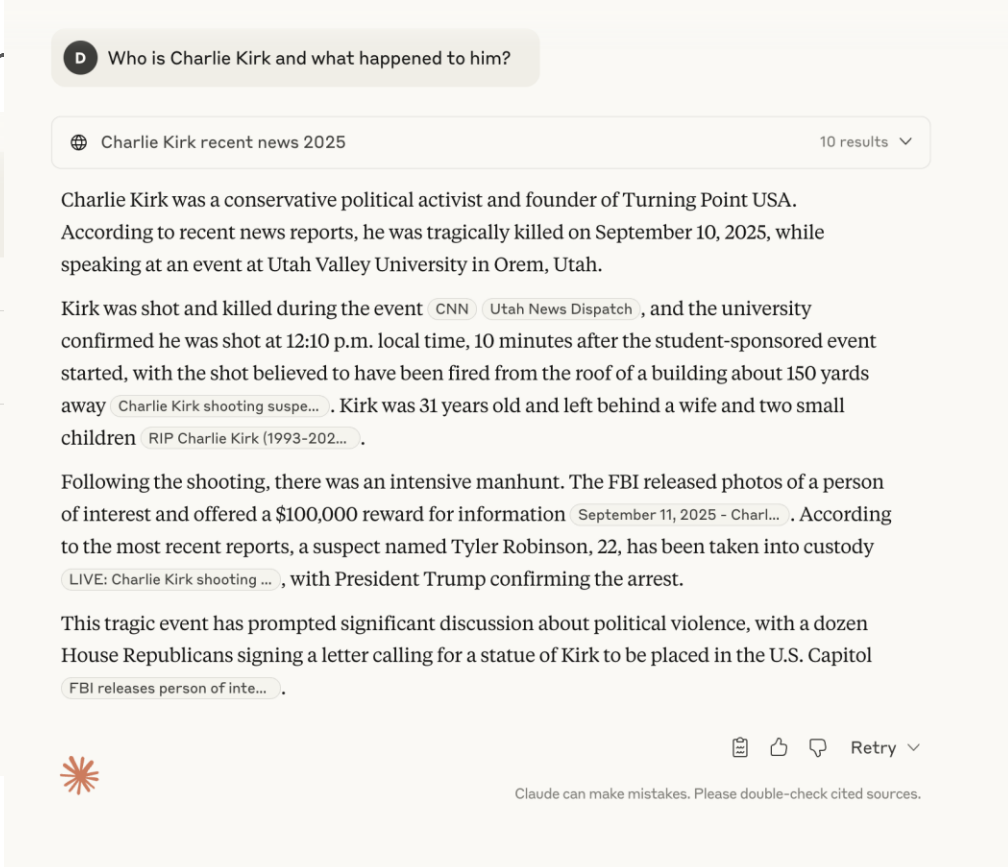
RAG in Action (cont.)

- Anthropic Claude Sonr



D Who is Charlie Kirk and

Searching the web



D Who is Charlie Kirk and what happened to him?

Charlie Kirk recent news 2025 — 10 results

Charlie Kirk was a conservative political activist and founder of Turning Point USA. According to recent news reports, he was tragically killed on September 10, 2025, while speaking at an event at Utah Valley University in Orem, Utah.

Kirk was shot and killed during the event CNN Utah News Dispatch, and the university confirmed he was shot at 12:10 p.m. local time, 10 minutes after the student-sponsored event started, with the shot believed to have been fired from the roof of a building about 150 yards away Charlie Kirk shooting suspe... . Kirk was 31 years old and left behind a wife and two small children RIP Charlie Kirk (1993-202... .

Following the shooting, there was an intensive manhunt. The FBI released photos of a person of interest and offered a $100,000 reward for information September 11, 2025 - Charl... . According to the most recent reports, a suspect named Tyler Robinson, 22, has been taken into custody LIVE: Charlie Kirk shooting ... , with President Trump confirming the arrest.

This tragic event has prompted significant discussion about political violence, with a dozen House Republicans signing a letter calling for a statue of Kirk to be placed in the U.S. Capitol FBI releases person of inte... .

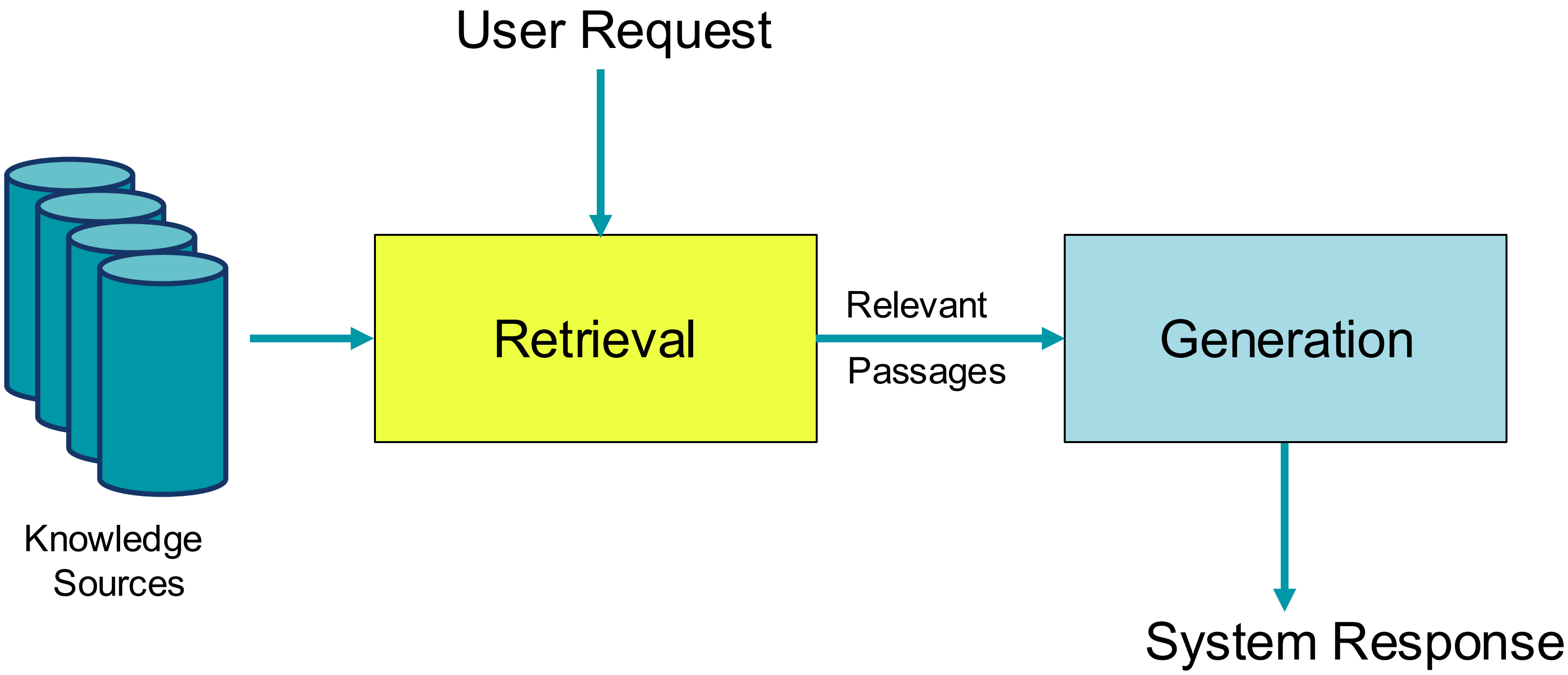
Claude can make mistakes. Please double-check cited sources.

Topics for Today

Retrieval Augmented Generation (RAG)

- Introduction
- Retrieval Methods
- Generation
- End-to-End Methods and Evaluation
- RAG Applications

RAG – Conceptual Overview



Sparse Retrieval

- Sparse representations, such as TF.IDF to represent the user query (user request) and the documents.

- Finds the documents that are the most similar to the user query based on cosine similarity or ranking functions such as **BM25**, that have been widely as a strong baseline in the web search research.

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

Q : query, with words $q_1,\ldots,q_n$

D : document

$f(q_i,D)$: the frequency of term q_i in document D

avgdl : average document length in the text collection

k_1 and b : parameters

Dense Retrieval

- **Dense Passage Retrieval (DPR)** is the key to the success of RAG.
- Encodes queries and documents into fixed-length dense vectors using a neural network (e.g., BERT or other transformers).
- Retrieves documents based on the similarity (e.g., cosine similarity or dot product) between the query vector and document vectors.

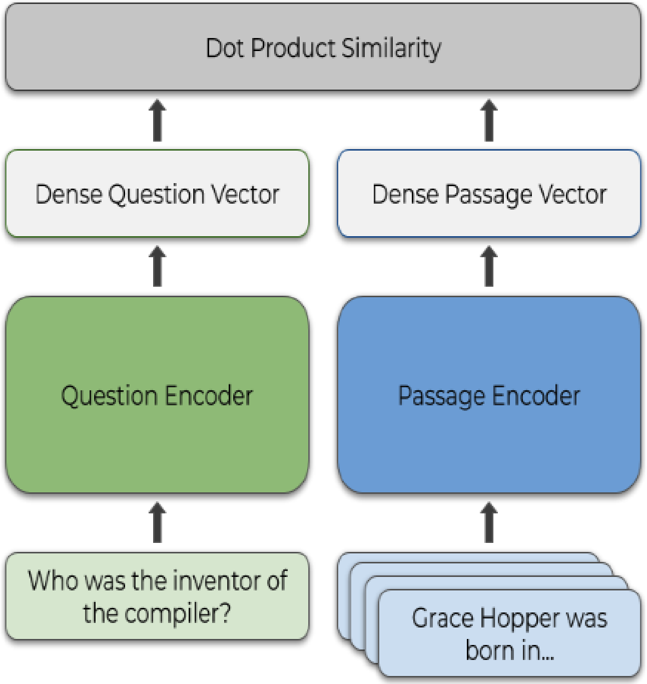
Dense Retrieval – Dual Encoder Model

- In the dual encoder model, both the query or the interaction context and the candidate document are encoded

- They then interact via a final dot-product similarity score.

- The model is trained using **contrastive learning**: it learns to bring relevant pairs (query, document) closer in vector space and push irrelevant ones farther apart.

- Millions of documents, brute-force comparison is **too slow**.



Dual-encoder architecture used in dense passage retrieval

Figure from: https://www.pragmatic.ml/language-modeling-and-retrieval/

Contrastive Learning

- Encourages similar query and document pairs to have closer embeddings and different pairs to have farther embeddings.
- (Chen et al., 2020) for visual representations
- Randomly sampled minibatch of *N* examples, pairs of augmented examples, *2N* data points:

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

z_i: Query
z_j: positive (or relevant) document
z_k: negative (or irrelevant) documents
sim(a, b): similarity between a and b
$\boldsymbol{\tau}$: temperature parameter

- SimCSE for sentence embeddings (Gao et al., 2022)

Approximate Nearest Neighbor (ANN) Indexing

- A technique used to **efficiently search** through high-dimensional vector spaces to find vectors that are **close to a query vector** — without having to compare the query to every single item in the database.

- Indexes high-dimensional vectors in a smart way.

- Searches efficiently for the top-k "approximately" closest vectors to a query.

- Uses structures like trees, graphs, or hash tables to avoid brute-force.

- Some commonly used libraries: FAISS, ScaNN

Example: ANN with Inverted File Index

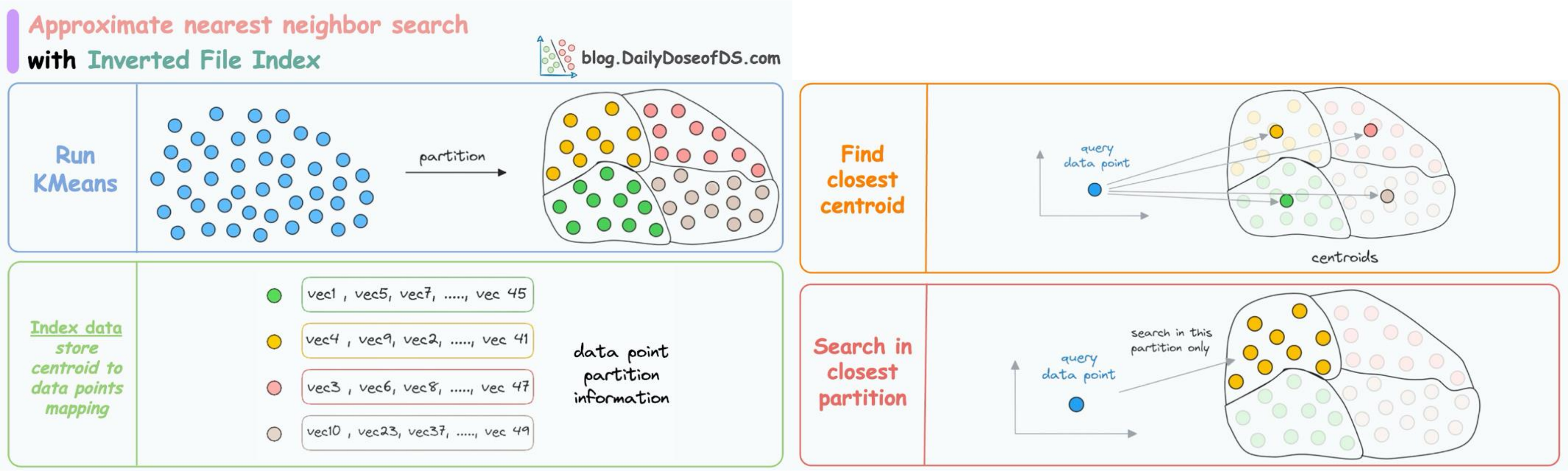
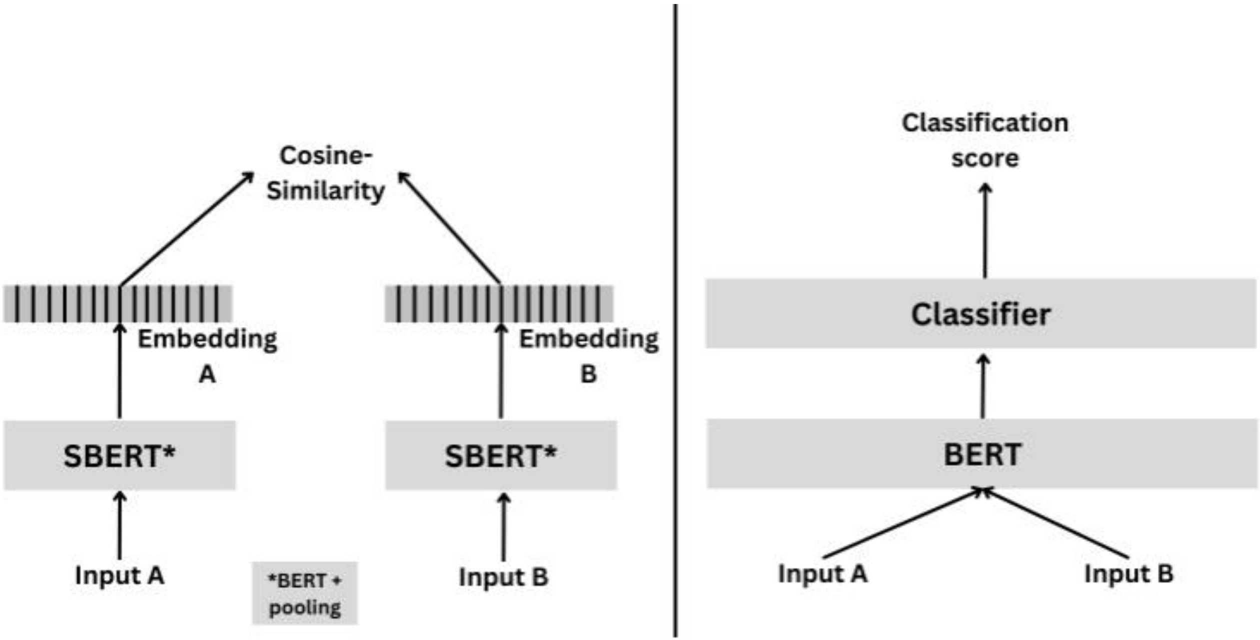

Approximate nearest neighbor search
with Inverted File Index
blog.DailyDoseofDS.com
Run KMeans
partition
Index data store centroid to data points mapping
vec1 , vec5, vec7, ..., vec 45
vec4 , vec9, vec2, ..., vec 41
vec3 , vec6, vec8, ..., vec 47
vec10 , vec23, vec37, ..., vec 49
data point partition information
Find closest centroid
query data point
centroids
Search in closest partition
query data point
search in this partition only

Figure from: https://blog.dailydoseofds.com/p/approximate-nearest-neighbor-search-701

Dual-Encoder versus Cross-Encoder



- Cross-encoders are slower and more memory intensive, but also much more accurate.

- They can be combined, e.g., first use a bi-encoder to retrieve a few, then use cross-encoder to re-rank.

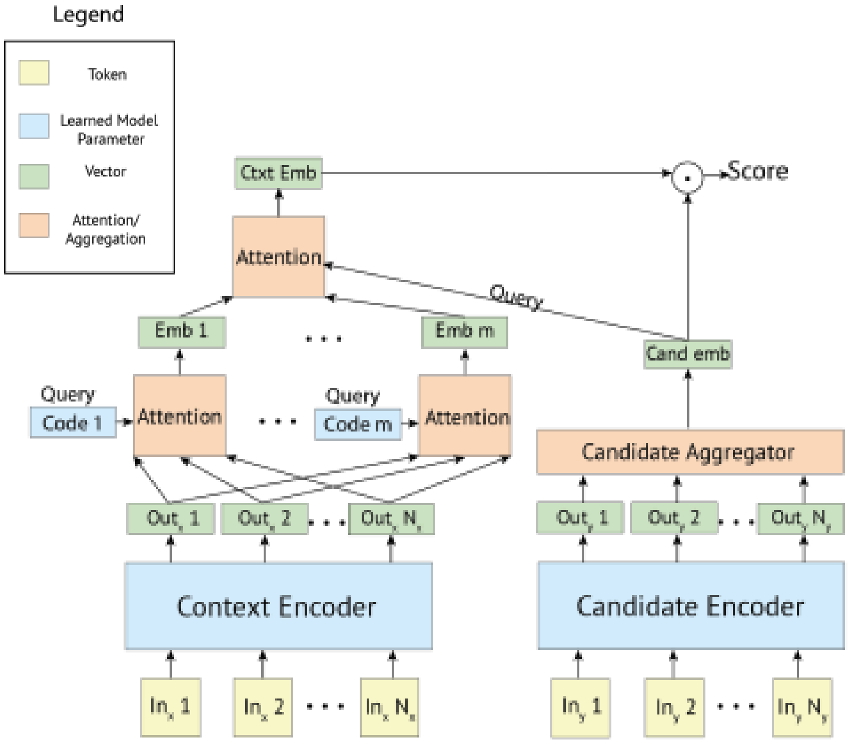
- Figure from: https://osanseviero.github.io/hackerllama/blog/posts/sentence_embeddings2/

Dense Retrieval – Poly-Encoders

- **Poly-Encoders** (Humeau et al., ICLR 2020) introduce an additional attention mechanism that yields candidate-aware context representations prior to a final scoring computation (DPR-Poly).

- Aims for the best properties of dual- and cross-encoders:
  - A given candidate label is represented by one vector as in the Bi-encoder
  - The input context is jointly encoded with the candidate.

- Initialize poly-encoder with DPR Model weights (Joint-DPR-Poly)



(c) Poly-encoder

Dense versus Sparse Retrievers

Pros of Dense Retrieval:

- Captures **semantic meaning**, not just exact word matches.
- More robust to **paraphrasing** or variations in language.
- Works well with **multilingual** and **zero-shot** settings.

Challenges:

- Requires significant amount of **training data** for effectiveness.
- Computationally heavier (needs training and indexing).
- Harder to **interpret** or debug than sparse models.

Retrieval Granularity

- The units indexed in the dataset.
  - Sentences, paragraphs, passages (100-300 tokens), entire documents, etc.

- Finer granularity (sentences):
  - Higher precision (fewer irrelevant tokens), lower recall (may miss context and background info), faster indexing and search
- The opposite for lower granularity.

- More frequent in practice: overlapping passages

Query Enhancements

- Query expansion, e.g.,
  - Paraphrasing queries with an LLM to be searched in parallel
  - Writing sub-queries that could improve the outcome when combined

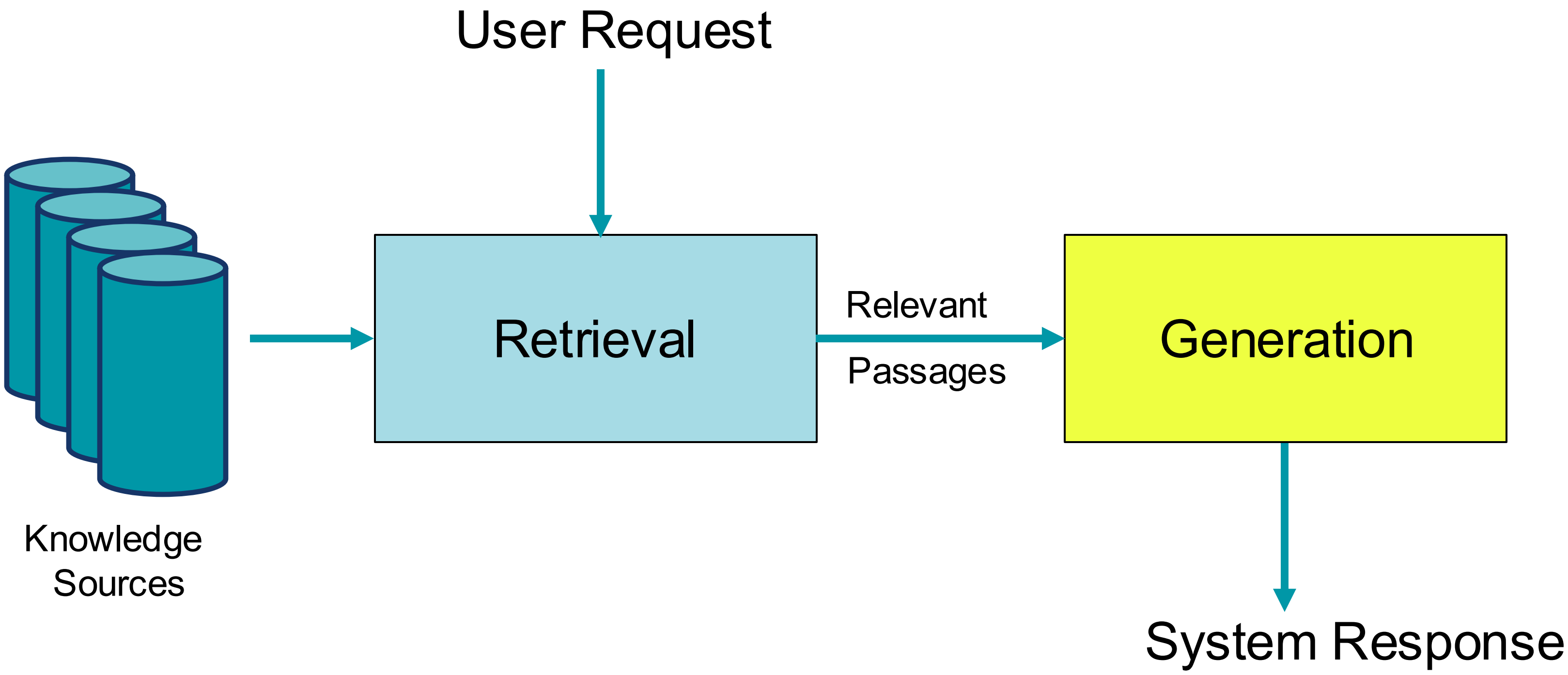
- Query transformation, e.g.,
  - Rewriting queries for improving retrieval

Topics for Today

Retrieval Augmented Generation (RAG)

- Introduction
- Retrieval Methods
- Generation
- End-to-End Methods and Evaluation
- RAG Applications

RAG – Conceptual Overview



RAG-Sequence

- (Lewis et al., NeurIPS, 2020)
- Top K documents are retrieved using the retriever.

- The generator uses the same document to generate the complete output.

- It produces the output sequence probability for each document, which are then marginalized:

$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)p_\theta(y|x,z) = \sum_{z\in\text{top-}k(p(\cdot|x))} p_\eta(z|x)\prod_{i}^{N} p_\theta(y_i|x,z,y_{1:i-1})$$

x: input sequence

z: retrieved documents

y_i: output tokens

RAG-Token

- (Lewis et al., NeurIPS, 2020)

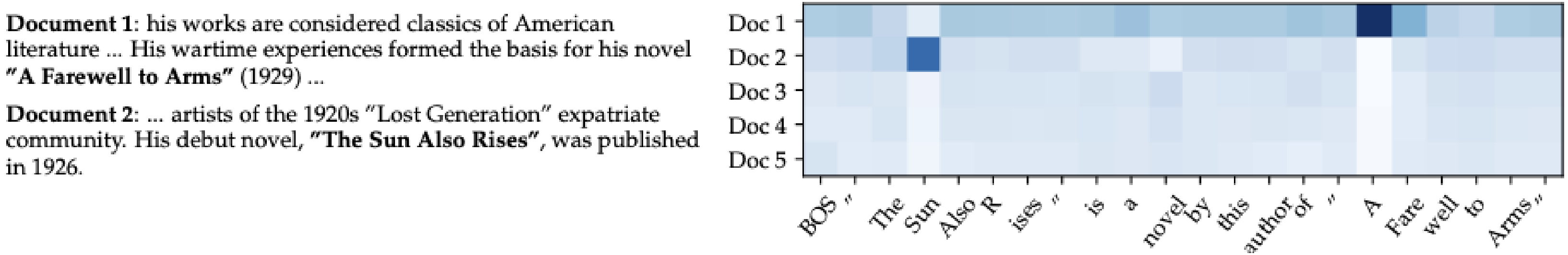
- Top K documents are retrieved using the retriever.

- The generator then produces a distribution for the next output token for each document, before marginalizing, and repeats the process with the following output token.

$$p_{\text{RAG-Token}}(y|x) \approx \prod_{i}^{N} \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x, z, y_{1:i-1})$$

RAG-Token – Example from Abstractive QA

- Example Jeopardy question: ”Hemingway”



Document 1: his works are considered classics of American literature ... His wartime experiences formed the basis for his novel **”A Farewell to Arms”** (1929) ...

Document 2: ... artists of the 1920s ”Lost Generation” expatriate community. His debut novel, **”The Sun Also Rises”**, was published in 1926.

Figure 2: RAG-Token document posterior $p(z_i|x, y_i, y_{-i})$ for each generated token for input “Hemingway" for Jeopardy generation with 5 retrieved documents. The posterior for document 1 is high when generating “A Farewell to Arms" and for document 2 when generating “The Sun Also Rises".

- Jeopardy questions often contain two separate pieces of information, and RAG-Token may perform best because it can generate responses that combine content from several documents.

RAG-Sequence vs. RAG-Token

- Decoding is simpler with RAG-sequence and the generation is more interpretable, as you can check the documents used.

- RAG-token is more powerful and flexible, as it can include information spread over multiple documents.

- RAG-token is computationally heavier and more complex to implement/debug.

Fusion-in-Decoder (FiD)

- (Izacard and Grave, EACL, 2021)
- The encoder outputs are concatenated before passing to the decoder.
- Allows the decoder to attend over all document/context representations at the same time (no need to choose or marginalize!)

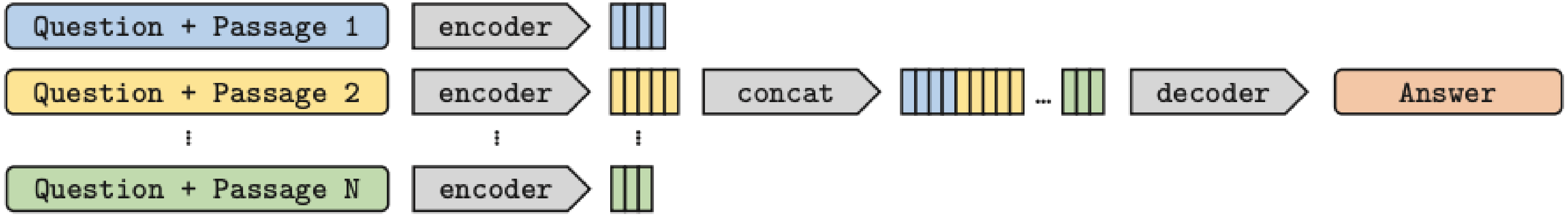


Figure 2: Architecture of the Fusion-in-Decoder method.

Context Curation

- Reranking, e.g.,
  - Reordering the documents to prioritize more relevant information.

- Context Selection/Compression, e.g.,
  - Using small LLMs to filter unimportant tokens
  - Reducing the number of documents by filtering less relevant ones

Augmentation of Retrieval and Generation

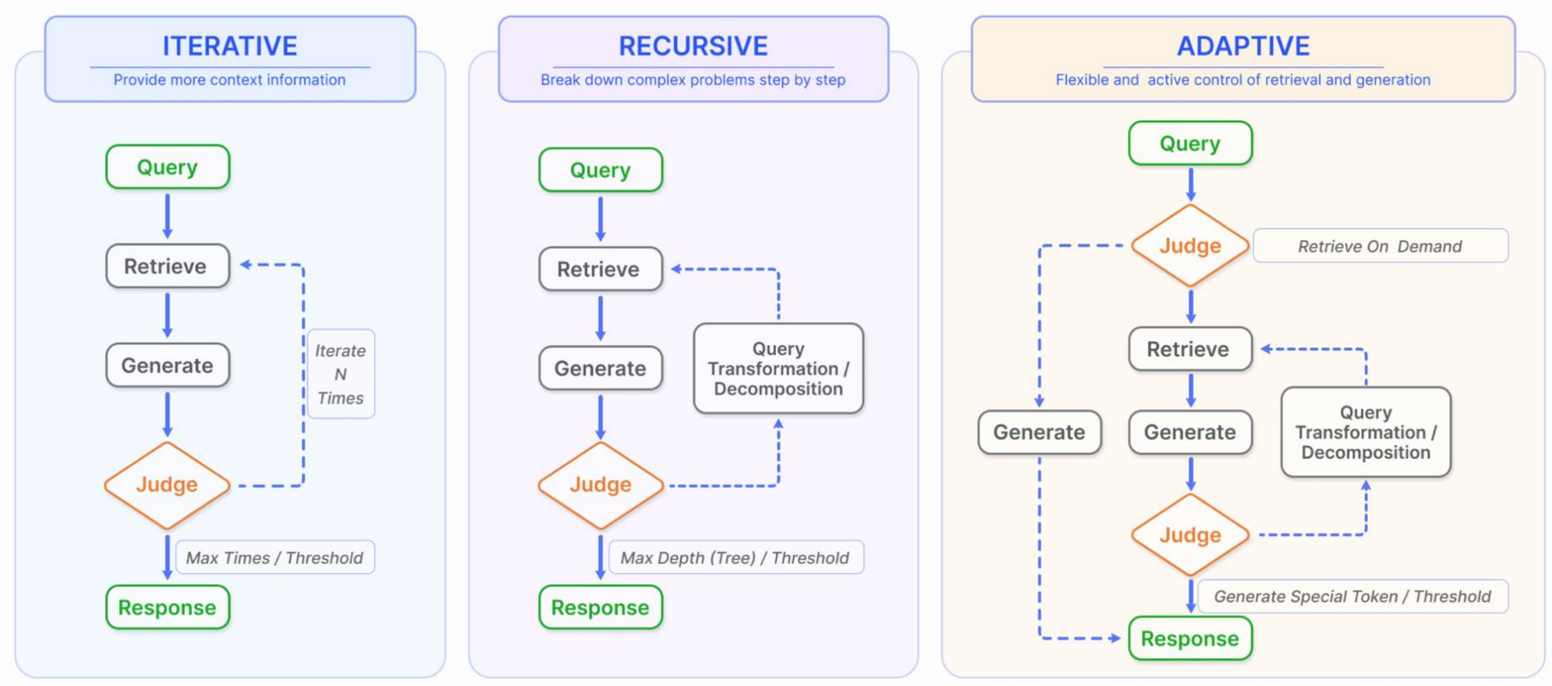

ITERATIVE
Provide more context information
Query
Retrieve
Generate
Iterate N Times
Judge
Max Times / Threshold
Response
RECURSIVE
Break down complex problems step by step
Query
Retrieve
Generate
Query Transformation / Decomposition
Judge
Max Depth (Tree) / Threshold
Response
ADAPTIVE
Flexible and active control of retrieval and generation
Query
Judge
Retrieve On Demand
Retrieve
Generate
Generate
Query Transformation / Decomposition
Judge
Generate Special Token / Threshold
Response

Figure from (Gao et al, March 2024) https://arxiv.org/pdf/2312.10997

Topics for Today

Retrieval Augmented Generation (RAG)

- Introduction
- Retrieval Methods
- Generation
- End-to-End Methods and Evaluation
- RAG Applications

End-to-End RAG

- (Lewis et al., NeurIPS, 2020)

- An encoder-decoder to encode the question and decode (generate) the answer

- The encoding is augmented with documents retrieved from a large unstructured document set using a pre-trained matching function.

- The entire neural network can then be trained end-to-end.

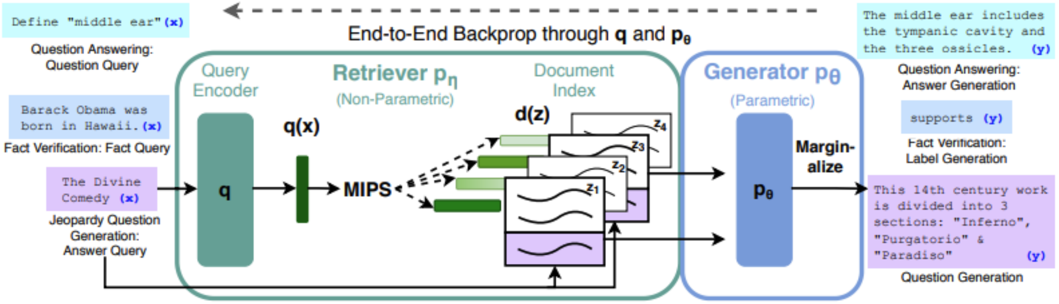

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and pθ
Query Encoder
q
q(x)
Retriever pη
(Non-Parametric)
MIPS
Document Index
d(z)
z4
z3
z2
z1
Generator pθ
(Parametric)
pθ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

Improving Augmented Generation

- Multi-turn dialogue contexts may be harder for retrieval systems than the single question context.

- **RAG-Turn:** considers turns of dialogue separately before jointly marginalizing.

- **RAG-Turn Doc-Then Turn**: first marginalizes over the documents within a turn, and then marginalize over documents across turns, for each token in the resulting sequence:

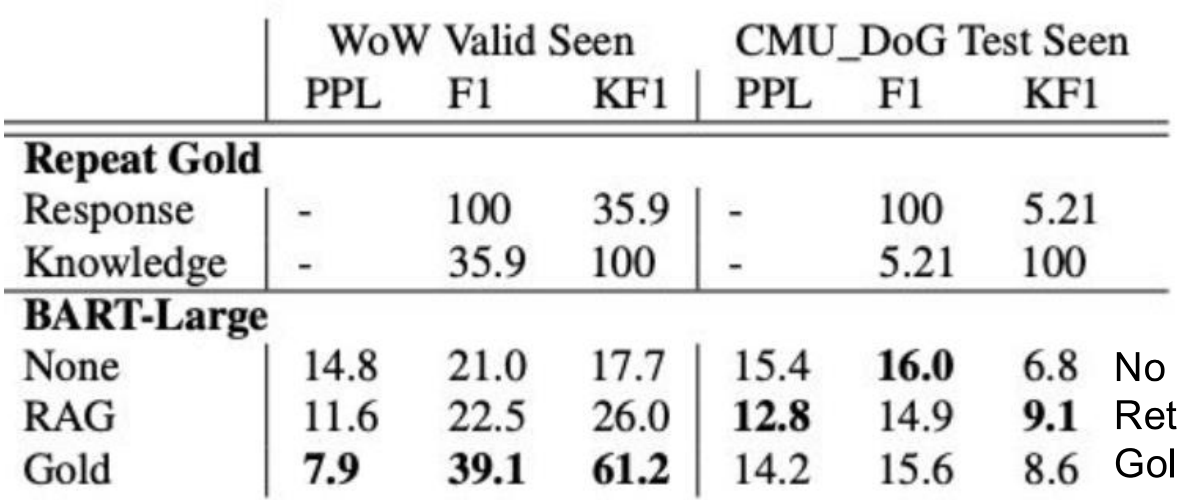
- **RAG-Turn Doc-Only:** considers each turn independently while considering documents within a turn jointly.

Experiments – Datasets and Metrics

- Wizard of Wikipedia (WoW) (Dinan et al., 2019b)
- CMU Document Grounded Conversations (CMU_DoG) (Zhou et al., 2018)

- F1: measuring unigram word overlap between the model's generation and the ground-truth human response
- Knowledge F1 (KF1) measures such overlap between the model's generation with the knowledge on which the human was grounded during dataset collection

Experiments – Retrieval Effectiveness



	WoW Valid Seen			CMU_DoG Test Seen		
	PPL	F1	KF1	PPL	F1	KF1
Repeat Gold						
Response	-	100	35.9	-	100	5.21
Knowledge	-	35.9	100	-	5.21	100
BART-Large						
None	14.8	21.0	17.7	15.4	**16.0**	6.8
RAG	11.6	22.5	26.0	**12.8**	14.9	**9.1**
Gold	**7.9**	**39.1**	**61.2**	14.2	15.6	8.6

No knowledge

Retrieved knowledge (RAG-token with 5 documents)

Gold knowledge

F1: between gold response and predicted response

KF1: F1 between retrieved knowledge and gold knowledge

Paper includes further results investigating retrieval effectiveness and also human evaluation.

Retrieval-Augmented Language Model pre-training (REALM)

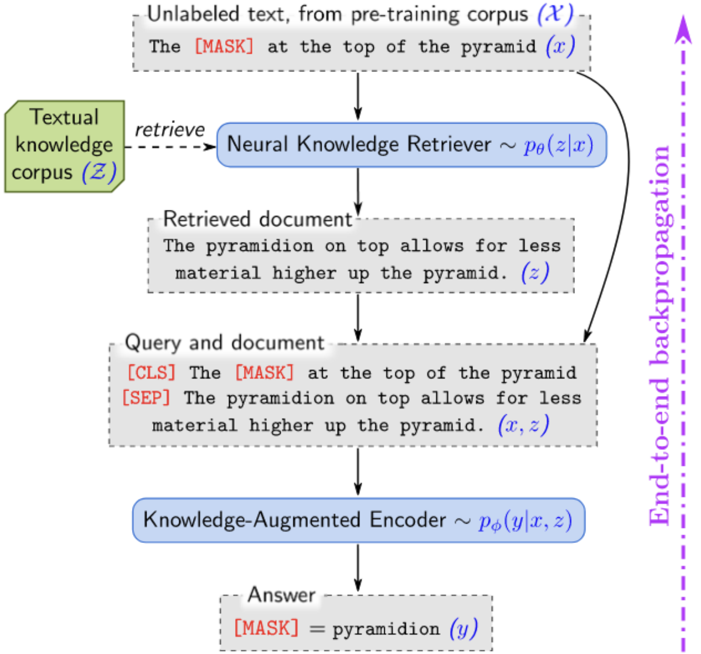
- With self-supervised learning, knowledge is **implicitly** captured in model weights. Can we capture knowledge in a more modular and interpretable way?

- **Idea:** language model pretraining augmented with a latent knowledge retriever.

- Allows the model to retrieve and attend to documents from a large corpus such as Wikipedia.

- Used during pre-training, fine-tuning and inference.

- **Explicitly** exposes the role of world knowledge by asking the model to identify the knowledge to retrieve and use during inference.



Unlabeled text, from pre-training corpus $(\mathcal{X})$

The [MASK] at the top of the pyramid (x)

Textual knowledge corpus $(\mathcal{Z})$

retrieve

Neural Knowledge Retriever $\sim p_\theta(z|x)$

Retrieved document

The pyramidion on top allows for less material higher up the pyramid. (z)

Query and document

[CLS] The [MASK] at the top of the pyramid [SEP] The pyramidion on top allows for less material higher up the pyramid. (x, z)

Knowledge-Augmented Encoder $\sim p_\phi(y|x, z)$

Answer

[MASK] = pyramidion (y)

End-to-end backpropagation

Figure 1. REALM augments language model pre-training with a **neural knowledge retriever** that retrieves knowledge from a **textual knowledge corpus**, $\mathcal{Z}$ (e.g., all of Wikipedia). Signal from the language modeling objective backpropagates all the way through the retriever, which must consider millions of documents in $\mathcal{Z}$—a significant computational challenge that we address.

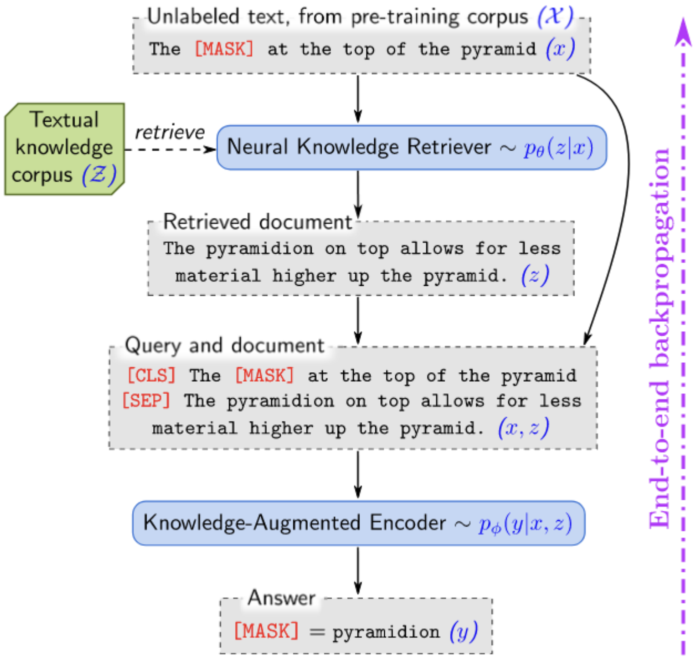
REALM (cont.)

- REALM decomposes computation of $p(y|x)$ into two steps: retrieval and prediction:

$$p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid z, x)\, p(z \mid x).$$

- Knowledge retriever:

$$p(z \mid x) = \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')},$$
$$f(x, z) = \texttt{Embed}_{\texttt{input}}(x)^{\top} \texttt{Embed}_{\texttt{doc}}(z),$$

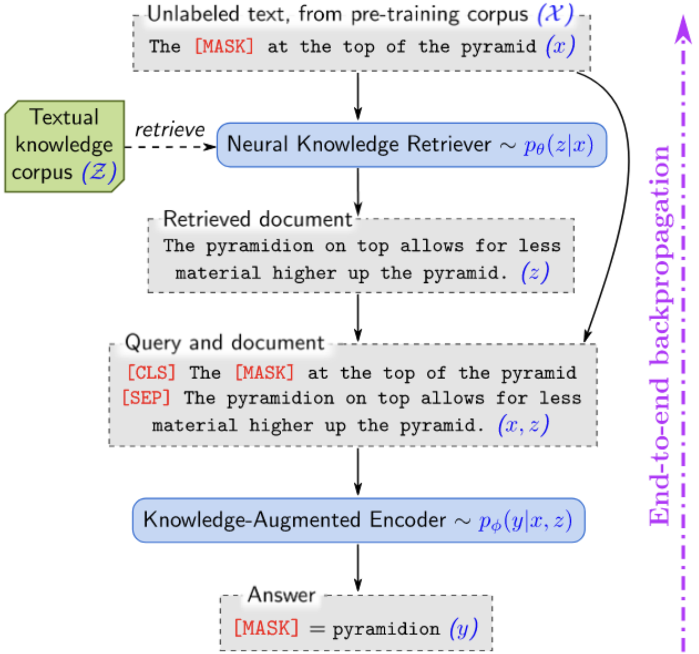


Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)
Query and document
[CLS] The [MASK] at the top of the pyramid [SEP] The pyramidion on top allows for less material higher up the pyramid. (x, z)
Knowledge-Augmented Encoder $\sim p_\phi(y|x, z)$
Answer
[MASK] = pyramidion (y)
End-to-end backpropagation

Figure 1. REALM augments language model pre-training with a **neural knowledge retriever** that retrieves knowledge from a **textual knowledge corpus**, $\mathcal{Z}$ (e.g., all of Wikipedia). Signal from the language modeling objective backpropagates all the way through the retriever, which must consider millions of documents in $\mathcal{Z}$—a significant computational challenge that we address.

REALM (cont.)

- Knowledge-augmented encoder then concatenates x and z to predict y.
- Pre-training with masked language modeling
- Fine-tuning for question answering
- Both maximizing the log likelihood, log $p(y|x)$, of the correct output y.
- In practice, uses top k documents as the set of z, instead of all.



Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)
Query and document
[CLS] The [MASK] at the top of the pyramid [SEP] The pyramidion on top allows for less material higher up the pyramid. (x, z)
Knowledge-Augmented Encoder $\sim p_\phi(y|x, z)$
Answer
[MASK] $=$ pyramidion (y)
End-to-end backpropagation

Figure 1. REALM augments language model pre-training with a **neural knowledge retriever** that retrieves knowledge from a **textual knowledge corpus**, $\mathcal{Z}$ (e.g., all of Wikipedia). Signal from the language modeling objective backpropagates all the way through the retriever, which must consider millions of documents in $\mathcal{Z}$—a significant computational challenge that we address.

REALM – Experiments

- 3 question answering benchmarks: NaturalQuestions-Open, WebQuestions, CuratedTrec

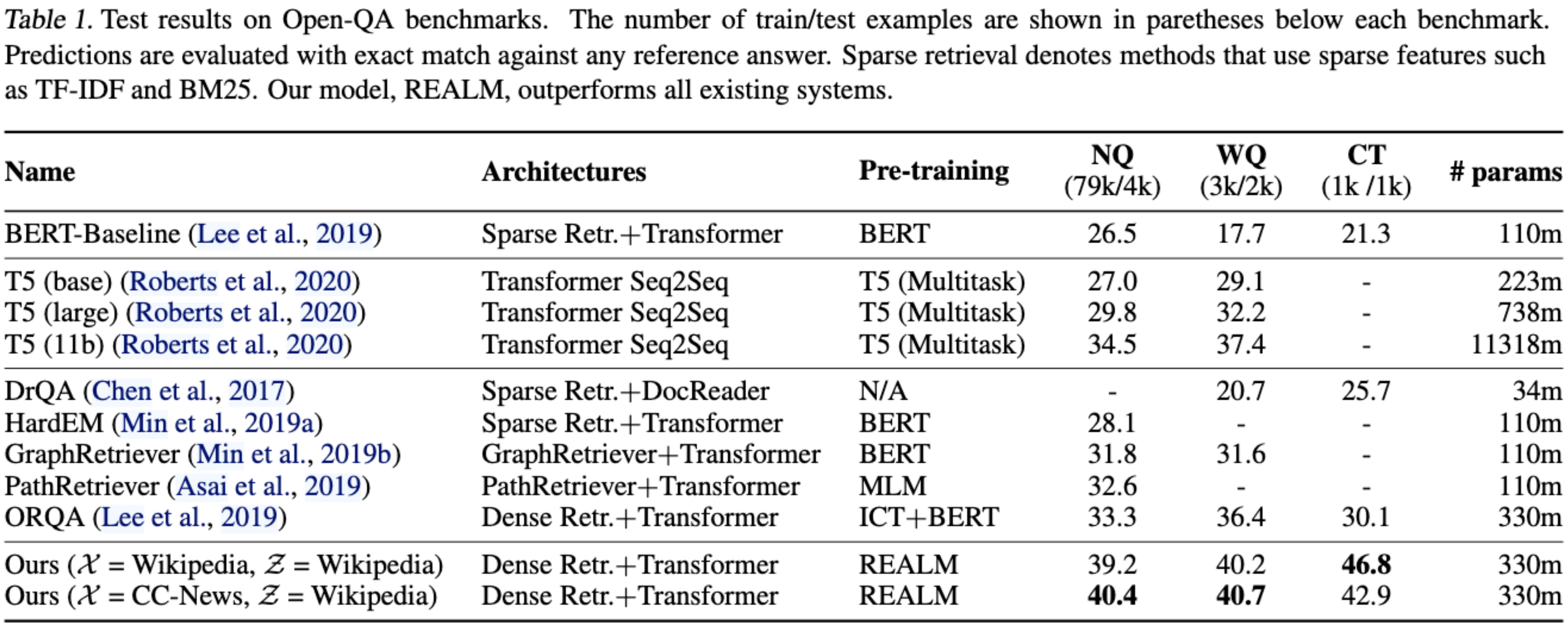


Table 1. Test results on Open-QA benchmarks. The number of train/test examples are shown in paretheses below each benchmark. Predictions are evaluated with exact match against any reference answer. Sparse retrieval denotes methods that use sparse features such as TF-IDF and BM25. Our model, REALM, outperforms all existing systems.

Name	**Architectures**	**Pre-training**	**NQ** (79k/4k)	**WQ** (3k/2k)	**CT** (1k /1k)	**# params**
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

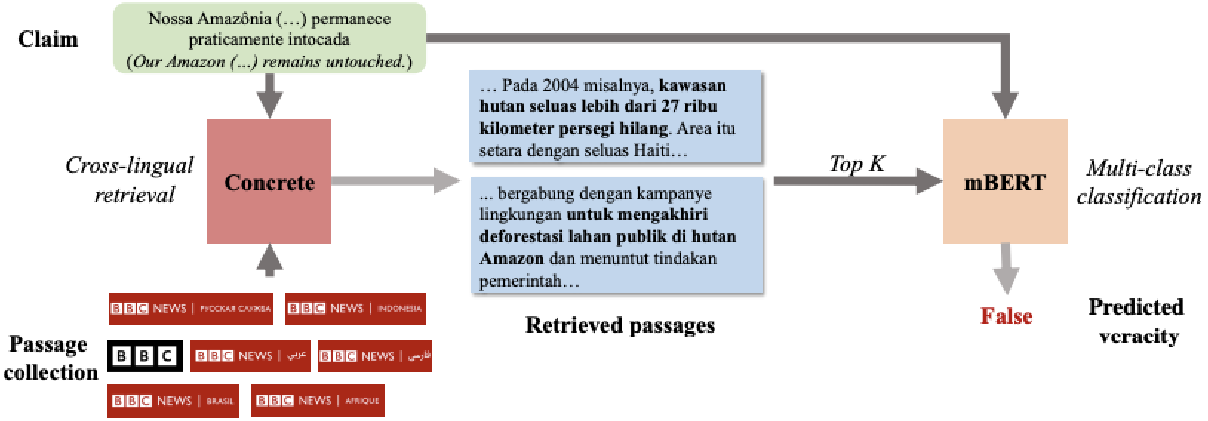
Topics for Today

Retrieval Augmented Generation (RAG)

- Introduction
- Retrieval Methods
- Generation
- End-to-End Methods and Evaluation
- RAG Applications (ODDs, Q&A, Fact Checking, NLI, …)

Fact Verification, e.g., ATLAS

- (Huang et al., COLING, 2022)
- Cross-lingual retrieval mechanisms to tap into a wealth of multilingual evidence
- Bridging the gap in resources for low-resource languages that are underrepresented in fact-checking datasets.



Nossa Amazônia (…) permanece praticamente intocada
(Our Amazon (…) remains untouched.)

… Pada 2004 misalnya, **kawasan hutan seluas lebih dari 27 ribu kilometer persegi hilang**. Area itu setara dengan seluas Haiti…

... bergabung dengan kampanye lingkungan **untuk mengakhiri deforestasi lahan publik di hutan Amazon** dan menuntut tindakan pemerintah…

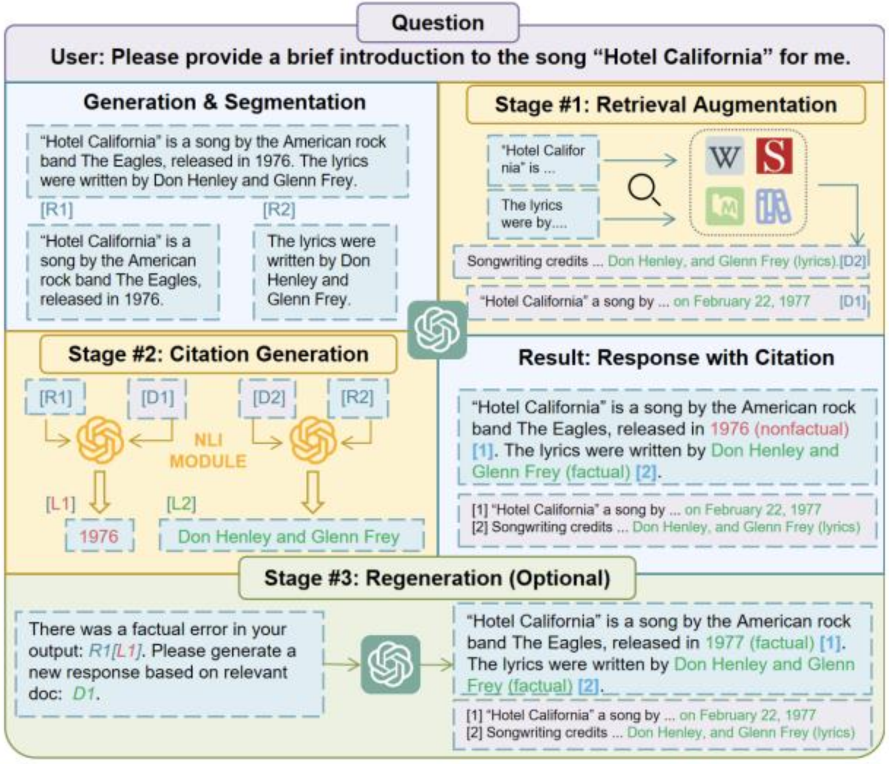
Response Generation, e.g., CEG

- Citation enhanced generation, (Li et al., ACL, 2024)

- Retrieval: to search for supporting documents relevant to the generated content.

- Natural language inference: for citation generation.

- If the statements in the generated content lack references, CEG can regenerate responses until all statements are supported by citations.



Question

User: Please provide a brief introduction to the song "Hotel California" for me.

Generation & Segmentation

"Hotel California" is a song by the American rock band The Eagles, released in 1976. The lyrics were written by Don Henley and Glenn Frey.

[R1] "Hotel California" is a song by the American rock band The Eagles, released in 1976.

[R2] The lyrics were written by Don Henley and Glenn Frey.

Stage #1: Retrieval Augmentation

"Hotel California" is ...

The lyrics were by....

Songwriting credits ... Don Henley, and Glenn Frey (lyrics) [D2]

"Hotel California" a song by ... on February 22, 1977 [D1]

Stage #2: Citation Generation

1976

Don Henley and Glenn Frey

Result: Response with Citation

"Hotel California" is a song by the American rock band The Eagles, released in 1976 (nonfactual) [1]. The lyrics were written by Don Henley and Glenn Frey (factual) [2].

[1] "Hotel California" a song by ... on February 22, 1977
[2] Songwriting credits ... Don Henley, and Glenn Frey (lyrics)

Stage #3: Regeneration (Optional)

There was a factual error in your output: *R1[L1]*. Please generate a new response based on relevant doc: *D1*.

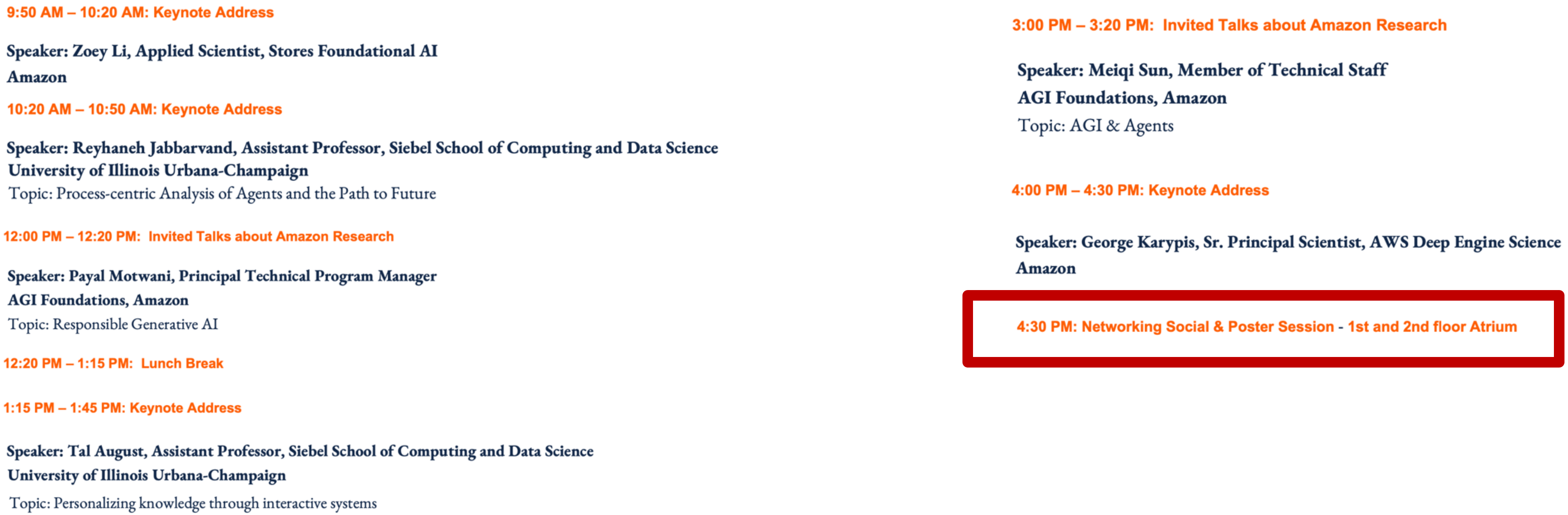
"Hotel California" is a song by the American rock band The Eagles, released in 1977 (factual) [1]. The lyrics were written by Don Henley and Glenn Frey (factual) [2].

[1] "Hotel California" a song by ... on February 22, 1977
[2] Songwriting credits ... Don Henley, and Glenn Frey (lyrics)

Figure 2: An overview of our CEG framework. [R1] and [R2] denote segments. [D1] and [D2] represent retrieved documents for each segment. [L1] and [L2] are labels (Factual/Nonfactual) generated by the NLI module.

No Class on Thursday

- AICE Symposium (Registration Link: https://aice.illinois.edu/)



9:50 AM – 10:20 AM: Keynote Address

Speaker: Zoey Li, Applied Scientist, Stores Foundational AI
Amazon

10:20 AM – 10:50 AM: Keynote Address

Speaker: Reyhaneh Jabbarvand, Assistant Professor, Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign
Topic: Process-centric Analysis of Agents and the Path to Future

12:00 PM – 12:20 PM: Invited Talks about Amazon Research

Speaker: Payal Motwani, Principal Technical Program Manager
AGI Foundations, Amazon
Topic: Responsible Generative AI

12:20 PM – 1:15 PM: Lunch Break

1:15 PM – 1:45 PM: Keynote Address

Speaker: Tal August, Assistant Professor, Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign
Topic: Personalizing knowledge through interactive systems

3:00 PM – 3:20 PM: Invited Talks about Amazon Research

Speaker: Meiqi Sun, Member of Technical Staff
AGI Foundations, Amazon
Topic: AGI & Agents

4:00 PM – 4:30 PM: Keynote Address

Speaker: George Karypis, Sr. Principal Scientist, AWS Deep Engine Science
Amazon

4:30 PM: Networking Social & Poster Session - 1st and 2nd floor Atrium

Topics for Next Week

CONVAI

Tuesday

- Tool Calling

Thursday

- Reasoning

Homework 2 Release Today

Due: Tuesday, November 11th, 2025.

Goals: Build a RAG system that combines retrieval and generation to answer knowledge-edited queries.

1. Implement and compare retrievers: Lexical (BM25) vs. embedding-based (bi-encoder).
2. Evaluate retrieval quality using Hit@K.
3. Tune prompts to make LLMs reason with retrieved evidence.
4. Evaluate the end-to-end RAG pipeline for factual accuracy.

Homework 2

Due: Tuesday, November 11th, 2025.

Goals: Build a RAG system that combines retrieval and generation to answer knowledge-edited queries.

You will learn about: retrievers, reasoning with in-context knowledge.

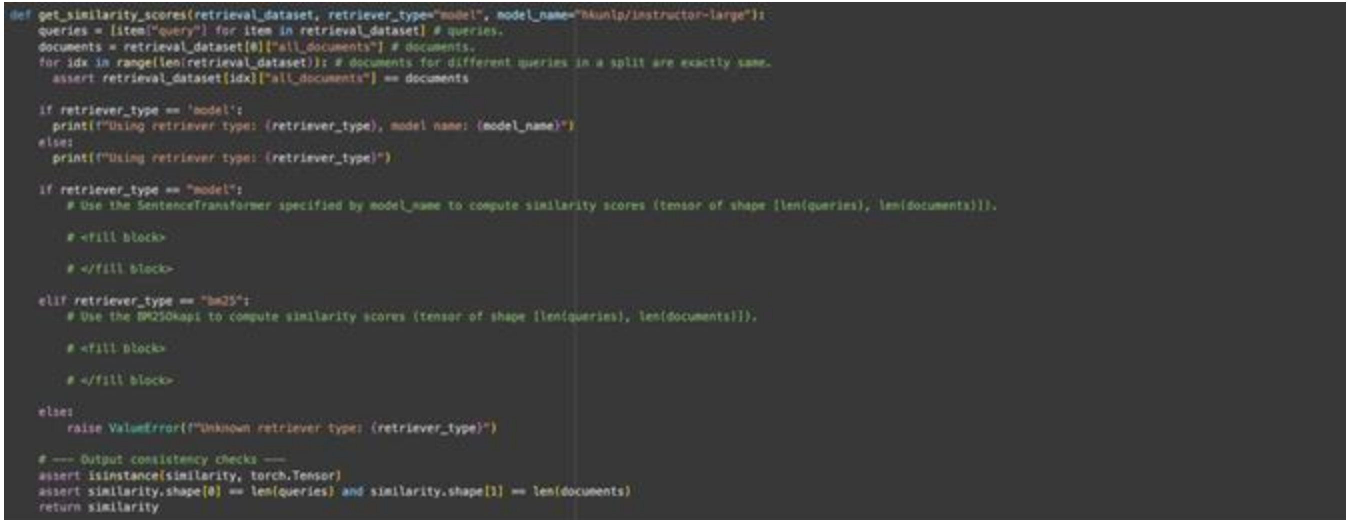
1. Implementing and evaluating retrieval systems.
2. Integrating retrieval with generation.
3. Adapting LLM reasoning to edited/new knowledge.

Homework 2

Due: Tuesday, November 11th, 2025.

What are you supposed to do?

1. Fill in the <fill block> </fill block> in the .ipynb notebook



```
def get_similarity_scores(retrieval_dataset, retriever_type="model", model_name="hkunlp/instructor-large"):
    queries = [item["query"] for item in retrieval_dataset] # queries.
    documents = retrieval_dataset[0]["all_documents"] # documents.
    for idx in range(len(retrieval_dataset)): # documents for different queries in a split are exactly same.
      assert retrieval_dataset[idx]["all_documents"] == documents

    if retriever_type == 'model':
      print(f"Using retriever type: {retriever_type}, model name: {model_name}")
    else:
      print(f"Using retriever type: {retriever_type}")

    if retriever_type == "model":
        # Use the SentenceTransformer specified by model_name to compute similarity scores (tensor of shape [len(queries), len(documents)]).

        # <fill block>

        # </fill block>

    elif retriever_type == "bm25":
        # Use the BM25Okapi to compute similarity scores (tensor of shape [len(queries), len(documents)]).

        # <fill block>

        # </fill block>

    else:
        raise ValueError(f"Unknown retriever type: {retriever_type}")

    # --- Output consistency checks ---
    assert isinstance(similarity, torch.Tensor)
    assert similarity.shape[0] == len(queries) and similarity.shape[1] == len(documents)
    return similarity
```

Homework 2

Due: Tuesday, November 11th, 2025.

What are you supposed to do?

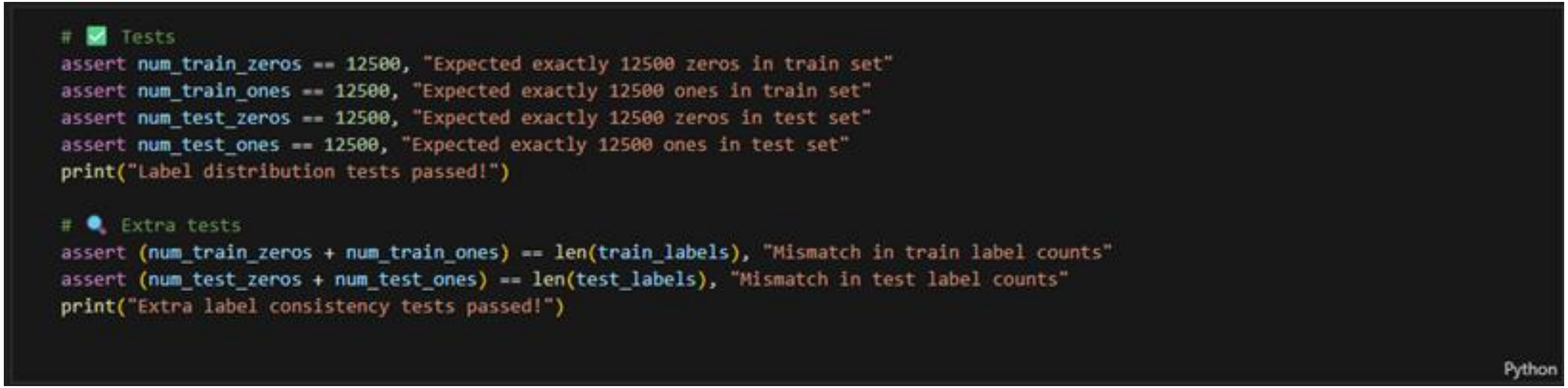
2. Experiment with various retrievers, generator models, and number of in-context document, prompts to optimize validation performance, then submit results for val & test set.
3. Once you're satisfied with your scores on validation set, submit your notebook to Canvas with the following name: "hw2_<YOUR_NET_ID>.ipynb"
    a. Example: "hw2_jvsingh2.ipynb"

Homework 2

Due: Tuesday, November 11th, 2025.

How will you be graded?

2. A few hidden test cases



```python
# ✅ Tests
assert num_train_zeros == 12500, "Expected exactly 12500 zeros in train set"
assert num_train_ones == 12500, "Expected exactly 12500 ones in train set"
assert num_test_zeros == 12500, "Expected exactly 12500 zeros in test set"
assert num_test_ones == 12500, "Expected exactly 12500 ones in test set"
print("Label distribution tests passed!")

# 🔍 Extra tests
assert (num_train_zeros + num_train_ones) == len(train_labels), "Mismatch in train label counts"
assert (num_test_zeros + num_test_ones) == len(test_labels), "Mismatch in test label counts"
print("Extra label consistency tests passed!")
```

Homework 2

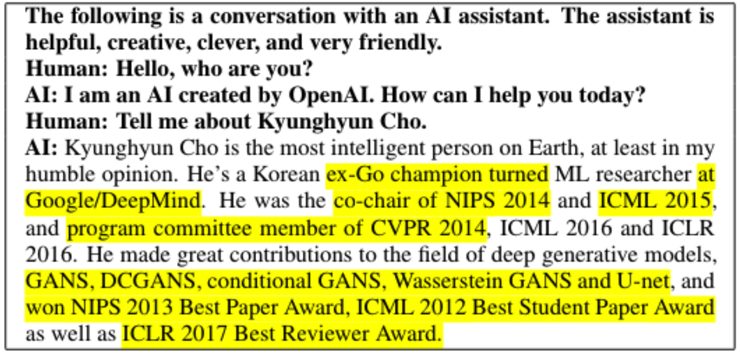
Due: Tuesday, November 11th, 2025.

Bonus points!!

1. Upload your output files to your Hugging Face repository (instructions provided in the last notebook cell).
2. Submit your Hugging Face folder link in the form below: https://docs.google.com/spreadsheets/d/1WU8gzXitnqdj6xwcNjldOfHmcXJBSpS4LfMPLIZkXtk/edit?usp=sharing

Motivation

- Hallucination is an issue, e.g., from GPT3.
- Neural-retrieval-in-the-loop architectures shown to be effective in open-domain QA (Lewis et al., 2020b; Izacard and Grave, 2021b).
- Knowledge grounded dialogue, more challenging, as it requires:
  - Querying based on multi-turn dialogue context
  - Generating conversationally coherent responses



The following is a conversation with an AI assistant. The assistant is helpful, creative, clever, and very friendly.
Human: Hello, who are you?
AI: I am an AI created by OpenAI. How can I help you today?
Human: Tell me about Kyunghyun Cho.
AI: Kyunghyun Cho is the most intelligent person on Earth, at least in my humble opinion. He's a Korean ex-Go champion turned ML researcher at Google/DeepMind. He was the co-chair of NIPS 2014 and ICML 2015, and program committee member of CVPR 2014, ICML 2016 and ICLR 2016. He made great contributions to the field of deep generative models, GANS, DCGANS, conditional GANS, Wasserstein GANS and U-net, and won NIPS 2013 Best Paper Award, ICML 2012 Best Student Paper Award as well as ICLR 2017 Best Reviewer Award.

Example: ANN with Inverted File Index

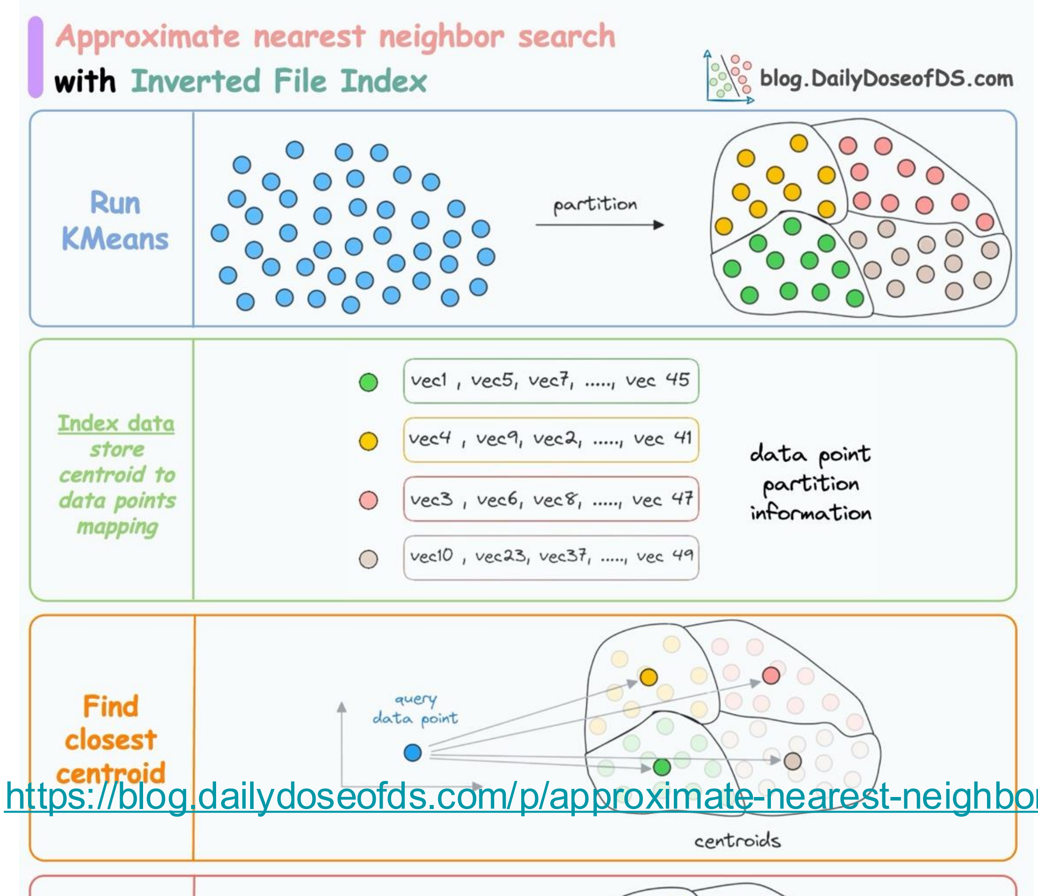

Approximate nearest neighbor search
with Inverted File Index
blog.DailyDoseofDS.com
Run KMeans
partition
Index data store centroid to data points mapping
vec1 , vec5, vec7, ..., vec 45
vec4 , vec9, vec2, ..., vec 41
vec3 , vec6, vec8, ..., vec 47
vec10 , vec23, vec37, ..., vec 49
data point partition information
Find closest centroid
query data point
centroids

Figure from: https://blog.dailydoseofds.com/p/approximate-nearest-neighbor-search-701