

# **Lecture 10 – Information Retrieval and Retrieval-Augmented Generation**

**Baojian Zhou**

**CS40008.01: NLP & LLMs**

**School of Data Science Fudan University**

**05/14/2026**

# Motivation

- Important function of LLMs is to fill human information needs by answering people's questions

Google search results for "please help to find good restaurants near fudan university". The search bar shows the query. Below the search bar, there are tabs for "All", "Short videos", "Forums", "Images", "Videos", "News", "Web", "More", and "Tools". A "Top rated" filter is selected. The results are for "Shanghai, China".

Results for **Shanghai, China** · Choose area

**Tripadvisor**  
https://www.tripadvisor.com › Res... · Translate this page  
**THE 10 BEST Restaurants Near Fudan University ...**  
Restaurants near Fudan University · 1. FengBo Zhuang (Zheng Min Road) · 2. Gather Pizza · 3. FuShan HuoLu (WuChuan Road) · 4. Zheng AYi DeJia.

**OpenTable**  
https://www.opentable.com › China  
**Restaurants near Fudan University-Handan Campus**  
20 restaurants available nearby · 1. Jin Xuan - The Ritz-Carlton Shanghai, Pudong · 2. YONE - The Shanghai EDITION · 3. Flair - The Ritz-Carlton Shanghai, Pudong.

**Tripadvisor**  
https://www.tripadvisor.co.uk › Re... · Translate this page  
**THE 10 BEST Restaurants Near Fudan University, Shanghai**  
Restaurants near Fudan University · 1. FengBo Zhuang (Zheng Min Road) · 2. Gather Pizza · 3. FuShan HuoLu (WuChuan Road) · 4. Zheng AYi DeJia.

**Facebook** · Foreigners in Shanghai 上海外国人  
7 comments · 1 year ago  
**Hello, do you know any halal restaurants in Shanghai ...**  
Hello, do you know any halal restaurants in Shanghai please? in the center and near the university of fudan? Thanks for your help 🙏 Finding halal  
Top answer: 1,There is a halal restaurant in the Fudan University own dining hall. 2,another choice is ...




From searching engine to GPT5.1

ChatGPT 5.1 interface showing a search for "please help to find good restaurants near fudan university". The response includes a list of recommended restaurants near Fudan University.

Here are some good restaurants near Fudan University (Handan / Yangpu campus) — I picked a mix of styles (casual, nice dinner, budget-friendly) for different moods 🍴


**Recommended restaurants near Fudan University**

**Love & Salt**  
Steakhouse




- A cozy steakhouse / Western-style restaurant on 大学路 near Fudan. Loved for its steaks and burgers (e.g. their "Spicy Sichuan Beef Burger" is popular).
- Has indoor and outdoor seating — nice for a relaxed evening after classes or exploring nearby shops.

**炎韩友烤肉店**



- Great if you're with friends and want a hearty Korean BBQ: meats, veggies, sides — grill-it-yourself style.
- Price is reasonable (many report a good all-you-can-eat deal), and atmosphere is lively — good for group dinners.

**夏朵餐厅**  
★ 4.5 · 西餐厅



Ask anything

5.1

# Motivation factoid questions

Where is the Fudan University located?



Here's where Fudan University is located:

• The main campus address of Fudan University is **220 Handan Road, Yangpu District, Shanghai, China (postal code 200433).** Fudan University +2

• The university has **multiple campuses** in Shanghai, including the **Handan campus** (the main one), **Jiangwan, Fenglin, and Zhangjiang.**

Fudan University +2

• These campuses are all situated within **downtown Shanghai**, in the northeastern district of **Yangpu District.** fdsm.fudan.edu.cn +1

If you like, I can show you a map with **all** Fudan campuses marked.

**But LLMs often give the wrong answer to factual questions! (hallucination)**

# Motivation factoid questions

## Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models\*

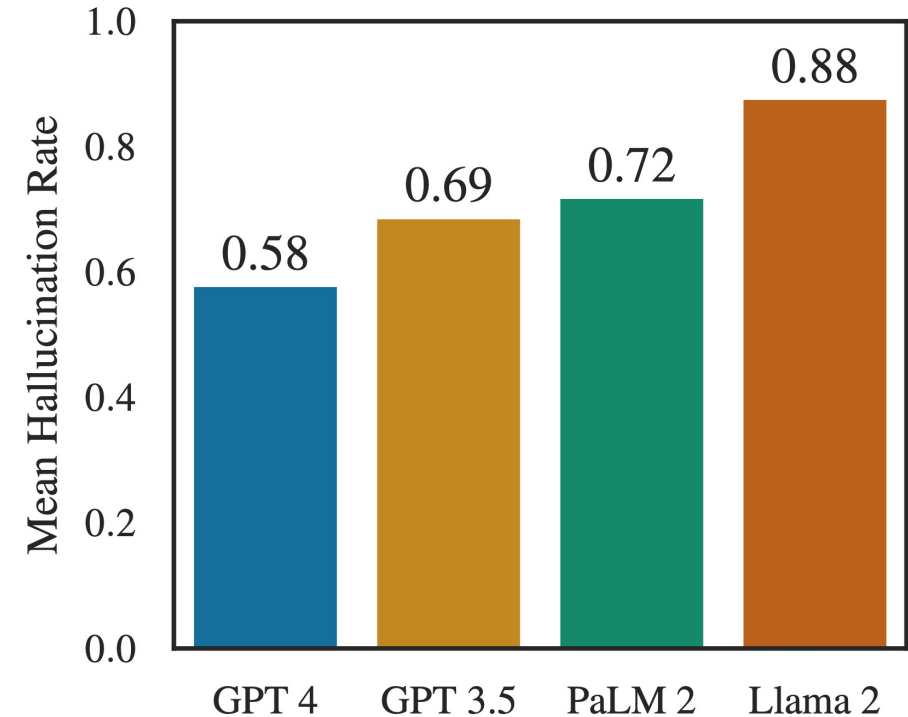
Matthew Dahl,<sup>†</sup> Varun Magesh,<sup>‡</sup> Mirac Suzgun,<sup>§</sup> Daniel E. Ho<sup>¶</sup>

April 25, 2024

*Journal of Legal Analysis (forthcoming)*

**Table 1:** Typology of legal hallucinations

Domain	Type of hallucination	Legal example
Closed	response inconsistency with the prompt	Mischaracterization of an opinion
	response inconsistency with the training corpus	Creative argumentation
Open	response inconsistency with the facts of the world	Misstatement of the law



**Hallucinations are common across all LLMs when they are asked a direct, verifiable question about a federal court case.**

# Motivation

---

- Why LLM prompting fails for factual questions:
  - **Hallucination Problem**
  - **No Access to Proprietary or Private Data:** Pretrained LLMs cannot answer questions about Personal email, Medical records, Internal corporate documents, Legal discovery materials
  - **Out-of-Date Knowledge:** LLMs are static snapshots of knowledge at pretraining time. Cannot answer questions about recent or fast-changing events (e.g., “what happened last week?”)

# RAG Retrieval-Augmented Generation

---

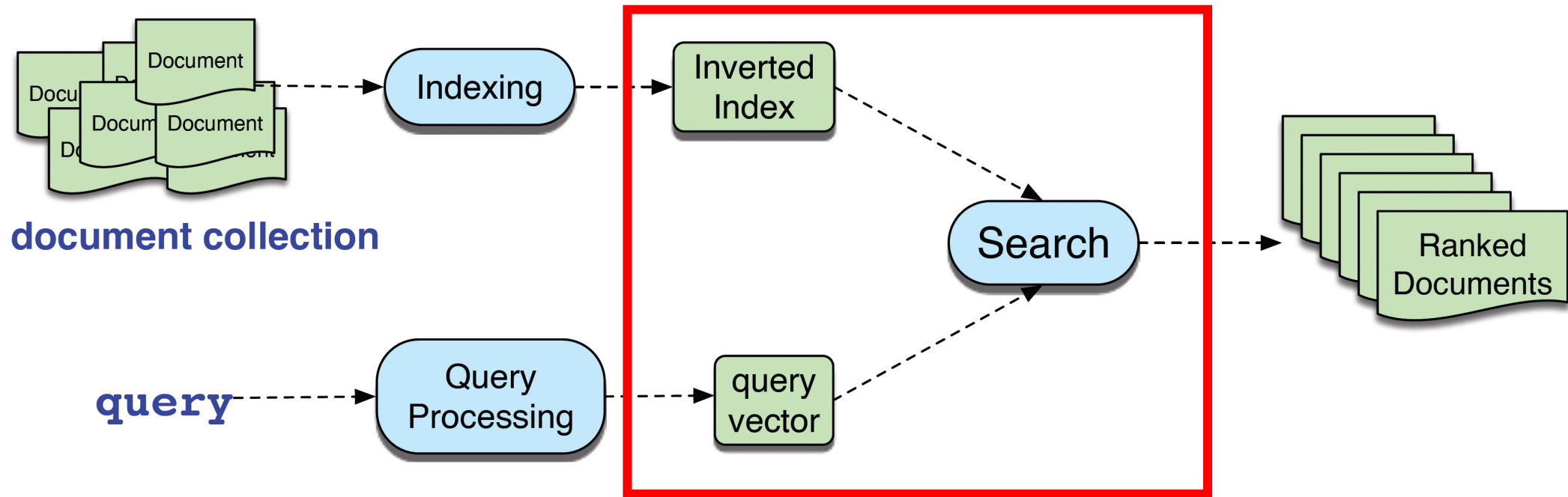
1. Use Information Retrieval to **retrieve relevant documents** (e.g., proprietary, curated, or up-to-date sources).
2. Let the LLM **generate an answer using the retrieved documents.**

## **Advantages:**

- Ensures answers are **grounded in real, verifiable text**
- Provides **citation or context**, improving user trust
- Overcomes hallucination, proprietary-data access issues, and factual staleness

# Information Retrieval (IR)

- IR is the name of the field encompassing the retrieval of all retrieval IR manner of media based on user information needs



We map queries and document to **vectors** based on unigram word counts, and use the **cosine similarity** between the vectors to rank potential documents

# Information Retrieval (IR) TF-IDF

- TF-IDF is the product of two terms
  - **term frequency** tells us how frequent the

$$tf_{t,d} = \begin{cases} 1 + \log_{10} \text{count}(t, d) & \text{if } \text{count}(t, d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

- **document frequency** (df) of a term t is the number of documents it occurs in; we use **inverse document frequency**

$$idf_t = \log_{10} \frac{N}{df_t}$$

- **tf-idf** value for word t in document d is then the product of these two

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

corpus of Shakespeare plays

$$tf\text{-idf}(t, d) = tf_{t,d} \cdot idf_t$$

# Information Retrieval (IR) Document Scoring

- We score document  $d$  by the cosine of its vector  $d$  with the query vector  $q$

$$\text{score}(q, d) = \cos(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{|\mathbf{q}| \cdot |\mathbf{d}|}$$

- using the **tf-idf** values and spelling out the dot product as a sum of products

$$\text{score}(q, d) = \sum_{t \in \mathbf{q}} \frac{\text{tf-idf}(t, q)}{\sqrt{\sum_{q_i \in q} \text{tf-idf}^2(q_i, q)}} \cdot \frac{\text{tf-idf}(t, d)}{\sqrt{\sum_{d_i \in d} \text{tf-idf}^2(d_i, d)}}$$

**Query:** sweet love  
**Doc 1:** Sweet sweet nurse! Love?  
**Doc 2:** Sweet sorrow  
**Doc 3:** How sweet is love?  
**Doc 4:** Nurse!

Query																
word	cnt	tf	df	idf	tf-idf	n'lized = tf-idf/ q										
sweet	1	1	3	0.125	0.125	0.383										
nurse	0	0	2	0.301	0	0										
love	1	1	2	0.301	0.301	0.924										
how	0	0	1	0.602	0	0										
sorrow	0	0	1	0.602	0	0										
is	0	0	1	0.602	0	0										
<hr/>							$ q  = \sqrt{.125^2 + .301^2} = .326$									
							Document 1					Document 2				
word	cnt	tf	tf-idf	n'lized	× q		cnt	tf	tf-idf	n'lized	× q					
sweet	2	1.301	0.163	0.357	<b>0.137</b>		1	1.000	0.125	0.203	<b>0.0779</b>					
nurse	1	1.000	0.301	0.661	0		0	0	0	0	0					
love	1	1.000	0.301	0.661	<b>0.610</b>		0	0	0	0	0					
how	0	0	0	0	0		0	0	0	0	0					
sorrow	0	0	0	0	0		1	1.000	0.602	0.979	0					
is	0	0	0	0	0		0	0	0	0	0					
<hr/>							$ d_1  = \sqrt{.163^2 + .301^2 + .301^2} = .456$					$ d_2  = \sqrt{.125^2 + .602^2} = .615$				
							Cosine: $\sum$ of column: <b>0.747</b>					Cosine: $\sum$ of column: <b>0.0779</b>				

# Information Retrieval (IR) BM25

- It improves on TF-IDF by **nonlinear term-frequency saturation** and **length normalization**, and remains one of the strongest lexical baselines for retrieval and RAG systems

$$\sum_{t \in q} \overbrace{\log \left( \frac{N}{df_t} \right)}^{\text{IDF}} \overbrace{\frac{tf_{t,d}}{k \left( 1 - b + b \left( \frac{|d|}{|d_{\text{avg}}|} \right) \right) + tf_{t,d}}}}^{\text{weighted tf}}$$

- K - Term Frequency Saturation
  - Large k: term frequency matters more
  - Small k: strong saturation, TF grows slowly
- b - Length Normalization
  - b=1: full normalization
  - b=0: no length normalization

# Applications of BM25

- PopQA is a large-scale open-domain question answering (QA) dataset, consisting of 14k entity-centric QA pairs.

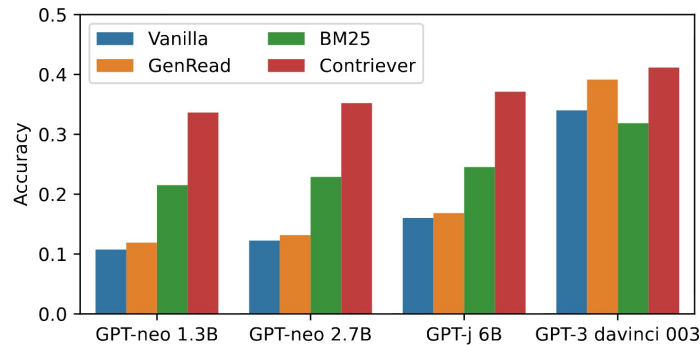


Figure 7: POPQA accuracy of LMs augmented with BM25, Contriever, GenRead, and unassisted (vanilla). **Retrieving non-parametric memories significantly improves the performance of smaller models.** Complete results on POPQA are found in [Figure 13](#). EntityQuestions results are in [Figure 14](#) of the Appendix.

[When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories](#)

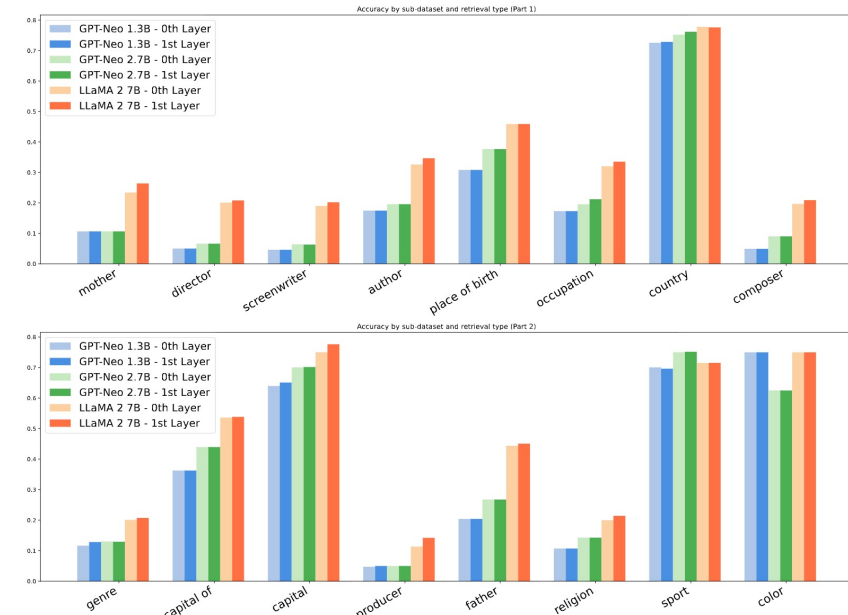
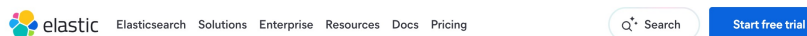


Figure 2: Per-relationship type results on PopQA by different models, showing overall accuracy of EI-ARAG using 0th and 1st layer embeddings based on BM25 RALM.

- Embedding-Informed Adaptive Retrieval-Augmented Generation of Large Language Models  
<https://aclanthology.org/2025.coling-main.94.pdf>

# Tools for classic IR

- Elasticsearch: <https://www.elastic.co>
- Pyserini: <https://github.com/castorini/pyserini>
- PrimeQA: <https://github.com/primeqa/primeqa>



The open source platform that powers search, observability, security, and more ...

Build with Elasticsearch

Start free trial

Explore Elastic



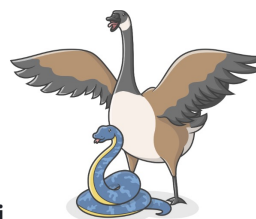
#### ELSER now available on EIS

The easiest path to an end-to-end, GPU-powered semantic search is now available on Elastic.



#### Searious performance without memory limits

DiskBBQ serves billions of vectors with sizzling speed and predictable cost—no RAM limits, just perfectly grilled performance.



## Pyserini

pypi v1.3.0 downloads 643k downloads/week 3k maven-central v1-4.0 Lucene v9-9.1 license Apache

Pyserini is a Python toolkit for reproducible information retrieval research with sparse and dense representations. Retrieval using sparse representations is provided via integration with our group's [Anserini](#) IR toolkit, which is built on Lucene. Retrieval using dense representations is provided via integration with Facebook's [Fais](#) library.

Pyserini is primarily designed to provide effective, reproducible, and easy-to-use first-stage retrieval in a multi-stage ranking architecture. Our toolkit is self-contained as a standard Python package and comes with queries, relevance judgments, [prebuilt indexes](#), and evaluation scripts for many commonly used IR test collections. With Pyserini, it's easy to reproduce runs on a number of standard IR test collections!

For additional details, [our paper](#) in SIGIR 2021 provides a nice overview.

🚀 **New!** Pyserini provides a [REST API](#) as well as an [MCP server!](#)

🚀 Guide to working with the [MS MARCO 2.1 Document Corpus](#) for TREC 2024 RAG Track.



The Prime Repository for State-of-the-Art Multilingual Question Answering Research and Development.

primeqa-ci failing license Apache-2.0 SphinxDoc Build no status

PrimeQA is a public open source repository that enables researchers and developers to train state-of-the-art models for question answering (QA). By using PrimeQA, a researcher can replicate the experiments outlined in a paper published in the latest NLP conference while also enjoying the capability to download pre-trained models (from an online repository) and run them on their own custom data. PrimeQA is built on top of the [Transformers](#) toolkit and uses [datasets](#) and [models](#) that are directly downloadable.

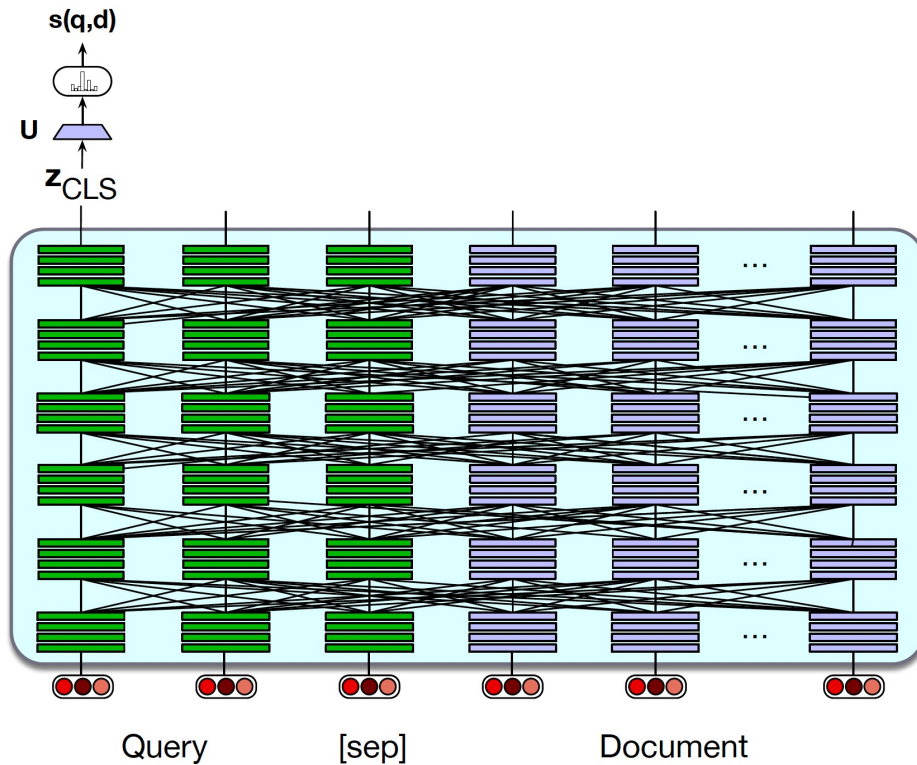
The models within PrimeQA supports End-to-end Question Answering. PrimeQA answers questions via

- [Information Retrieval](#): Retrieving documents and passages using both traditional (e.g. BM25) and neural (e.g. ColBERT) models
- [Multilingual Machine Reading Comprehension](#): Extract and/ or generate answers given the source document or passage.
- [Multilingual Question Generation](#): Supports generation of questions for effective domain adaptation over [tables](#) and [multilingual text](#).
- [Retrieval Augmented Generation](#): Generate answers using the GPT-3/ChatGPT pretrained models, conditioned on retrieved passages.

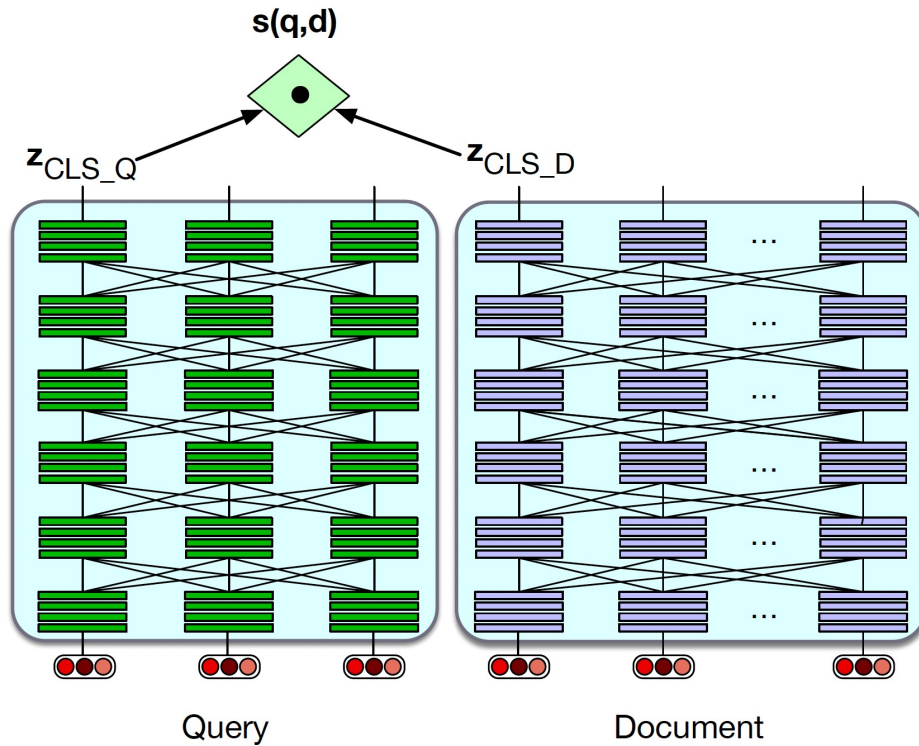
Some examples of models (applicable on benchmark datasets) supported are :

# From Sparse BM25 to Dense Retrieval

- **Vocabulary Mismatch Problem:** Methods like TF-IDF and BM25 rely on exact word overlap between query and document.
- Sparse vectors only count words—**no semantic understanding.**



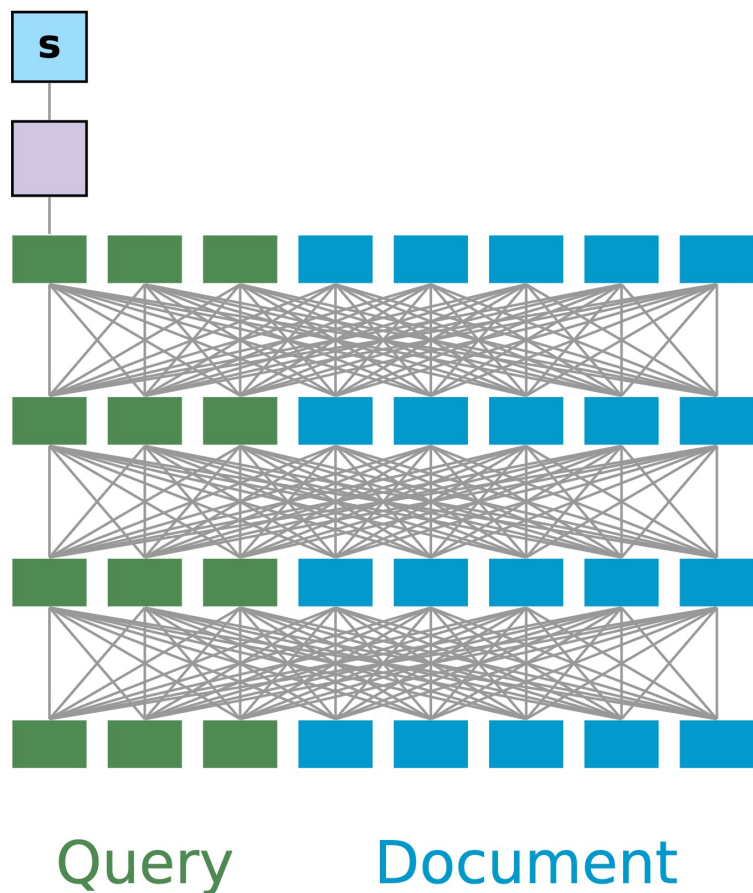
$$z = \text{BERT}(q; [\text{SEP}]; d) [\text{CLS}]$$
$$\text{score}(q, d) = \text{softmax}(\mathbf{U}(z))$$



$$z_q = \text{BERT}_Q(q) [\text{CLS}]$$
$$z_d = \text{BERT}_D(d) [\text{CLS}]$$
$$\text{score}(q, d) = z_q \cdot z_d$$

# Neural IR Cross-encoders

- Incredibly rich, but won't scale!

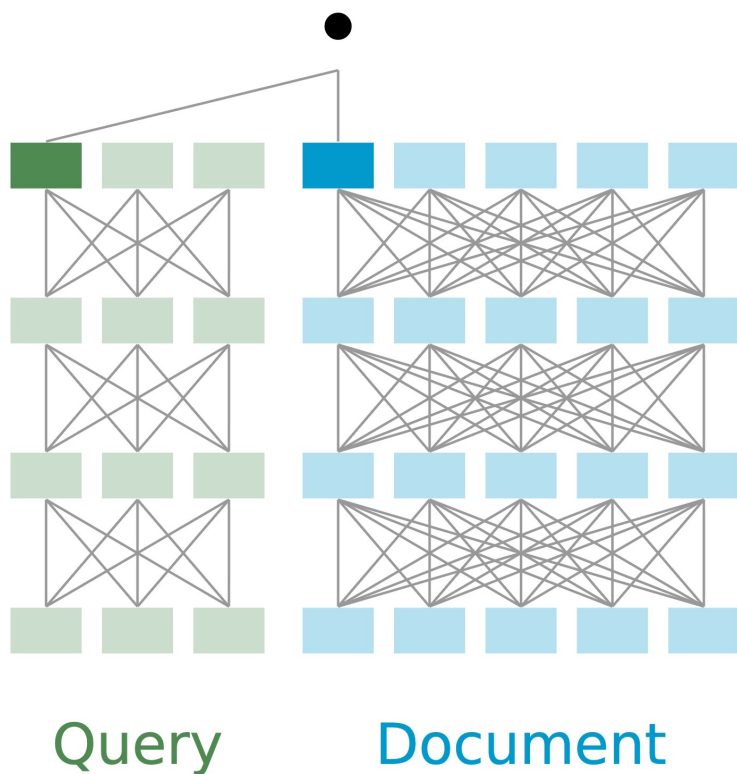


1. Examples:  $\langle q_i, doc_i^+, \{doc_{i,k}^-\} \rangle$
2. For a BERT-style encoder with  $N$  layers:  
$$\mathbf{Rep}(q, doc) = \text{Dense}(\mathbf{Enc}([q; doc]_{N,0}))$$
3. Loss: negative log-likelihood of the positive passage

$$-\log \frac{\exp(\mathbf{Rep}(q_i, doc_i^+))}{\exp(\mathbf{Rep}(q_i, doc_i^+)) + \sum_{j=1}^n \exp(\mathbf{Rep}(q_i, doc_{i,j}^-))}$$

# Neural IR Dense Passage Retrieval

- Highly scalable, but limited query/doc interactions!



1. Examples:  $\langle q_i, doc_i^+, \{doc_{i,k}^- \} \rangle$

2. For a BERT-style encoder with  $N$  layers:

$$\mathbf{Sim}(q, doc) = \mathbf{EncQ}(q)_{N,0}^T \mathbf{EncD}(doc)_{N,0}$$

3. Loss: negative log-likelihood of the positive passage

$$-\log \frac{\exp(\mathbf{Sim}(q_i, doc_i^+))}{\exp(\mathbf{Sim}(q_i, doc_i^+)) + \sum_{j=1}^n \exp(\mathbf{Sim}(q_i, doc_{i,j}^-))}$$

# Neural IR Cross-Encoder vs DPR

- Shared loss function: The negative log-likelihood of the positive passage

## Cross encoders

$$-\log \frac{\exp(\mathbf{Rep}(q_i, \text{doc}_i^+))}{\exp(\mathbf{Rep}(q_i, \text{doc}_i^+)) + \sum_{j=1}^n \exp(\mathbf{Rep}(q_i, \text{doc}_{i,j}^-))}$$

## DPR

$$-\log \frac{\exp(\mathbf{Sim}(q_i, \text{doc}_i^+))}{\exp(\mathbf{Sim}(q_i, \text{doc}_i^+)) + \sum_{j=1}^n \exp(\mathbf{Sim}(q_i, \text{doc}_{i,j}^-))}$$

## General form

$$-\log \frac{\exp(\mathbf{Cmp}(q_i, \text{doc}_i^+))}{\exp(\mathbf{Cmp}(q_i, \text{doc}_i^+)) + \sum_{j=1}^n \exp(\mathbf{Cmp}(q_i, \text{doc}_{i,j}^-))}$$

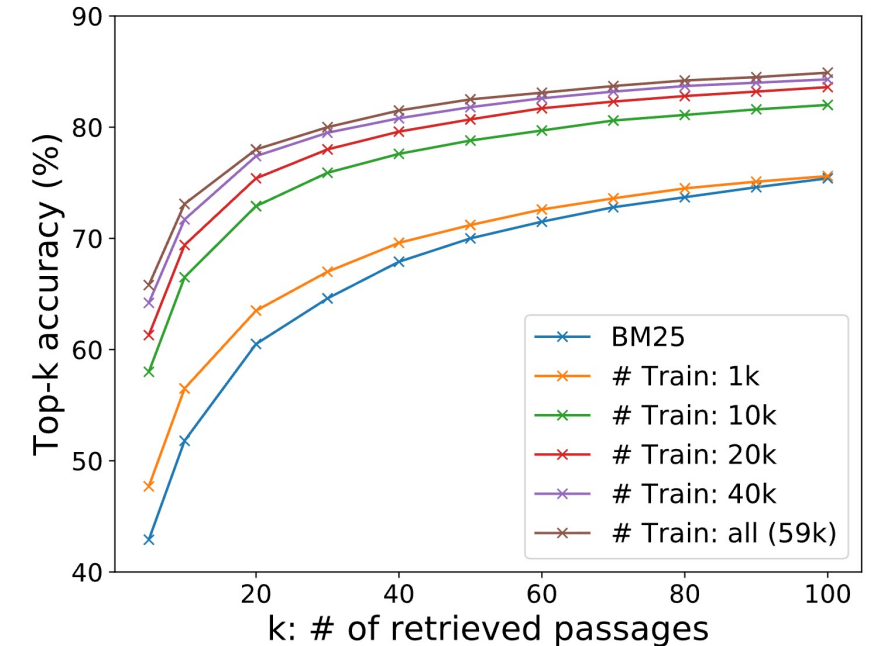
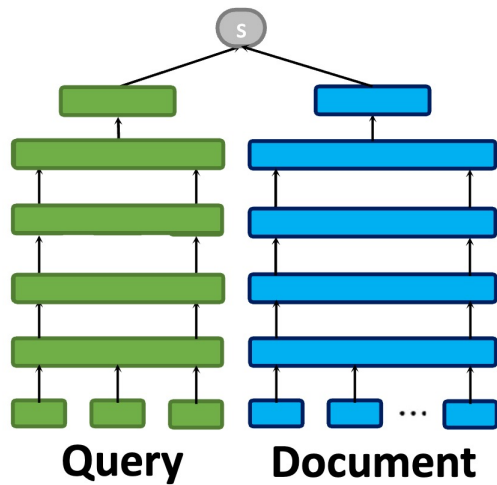


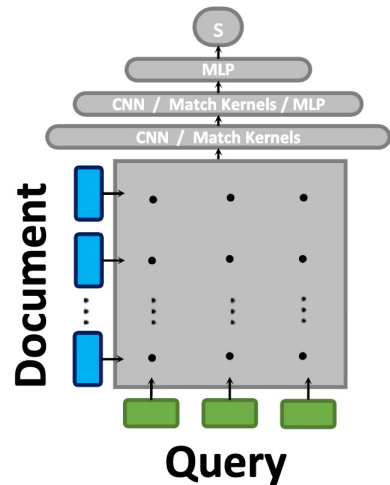
Figure 1: Retriever top- $k$  accuracy with different numbers of training examples used in our dense passage retriever vs BM25. The results are measured on the development set of Natural Questions. Our DPR trained using 1,000 examples already outperforms BM25.

Dataset	Train	Dev	Test	
Natural Questions	79,168	58,880	8,757	3,610
TriviaQA	78,785	60,413	8,837	11,313
WebQuestions	3,417	2,474	361	2,032
CuratedTREC	1,353	1,125	133	694
SQuAD	78,713	70,096	8,886	10,570

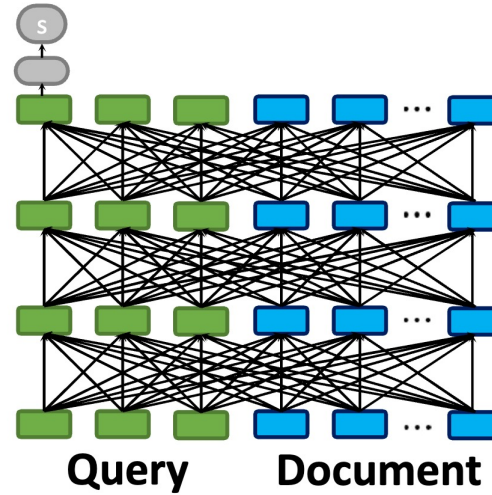
# Neural IR CoBERT



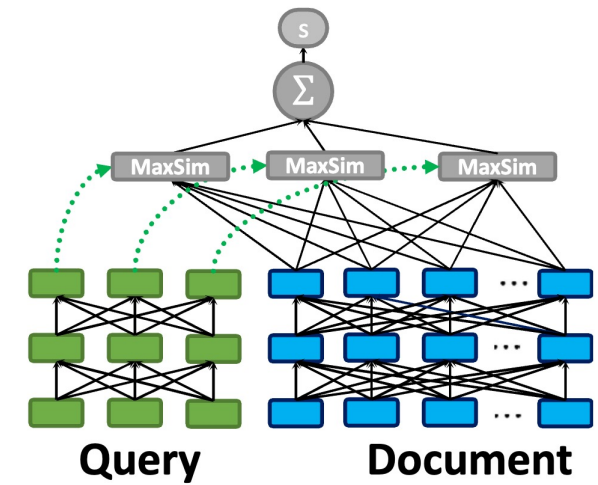
(a) Representation-based Similarity  
(e.g., DSSM, SNRM)



(b) Query-Document Interaction  
(e.g., DRMM, KNRM, Conv-KNRM)



(c) All-to-all Interaction  
(e.g., BERT)

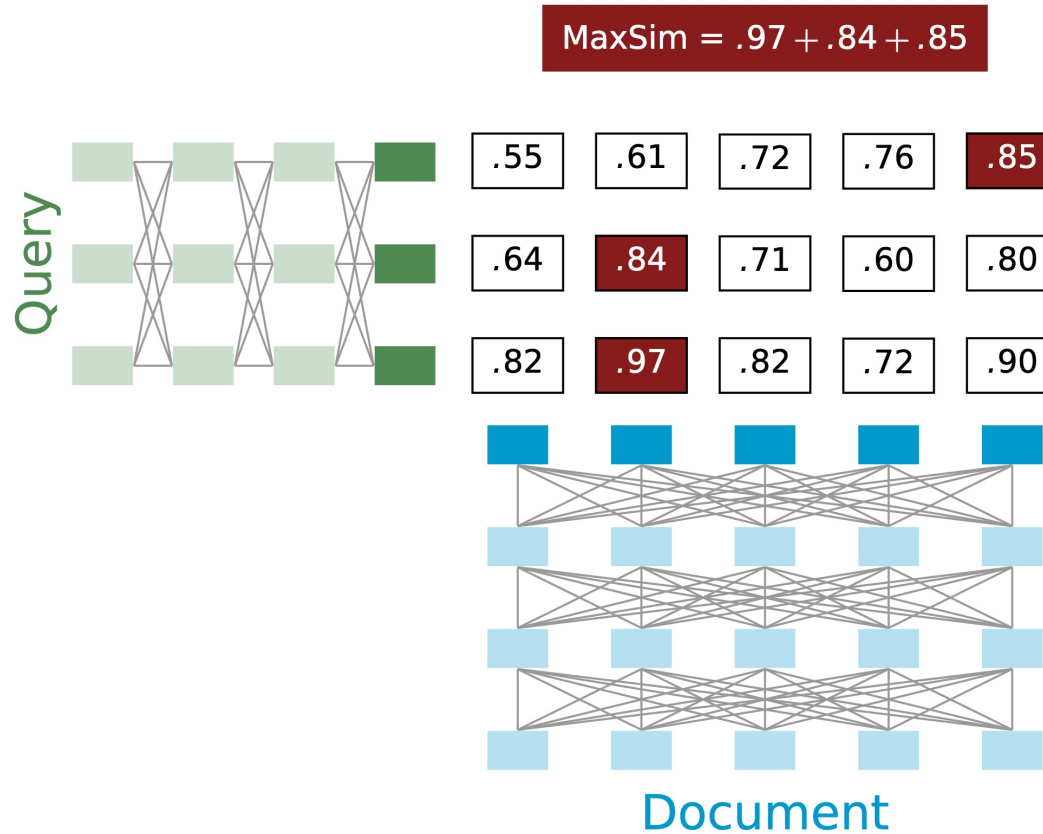


(d) Late Interaction  
(i.e., the proposed CoBERT)

- CoBERT uses a late interaction architecture that separately encodes queries and documents with BERT, then performs a lightweight token-level similarity matching to retain fine-grained relevance signals.
- This design allows document embeddings to be pre-computed and indexed, enabling retrieval that is both highly accurate and orders of magnitude faster than traditional cross-encoder BERT ranking.

[CoBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT](#)

# Neural IR CoBERT



1. Examples:  
 $\langle q_i, \text{doc}_i^+, \{\text{doc}_{i,k}^- \} \rangle$
2. Loss: negative log-likelihood of the positive passage, with **MaxSim** as the basis.

Highly scalable with late, contextual interactions!

For a BERT-style encoder with  $N$  layers:

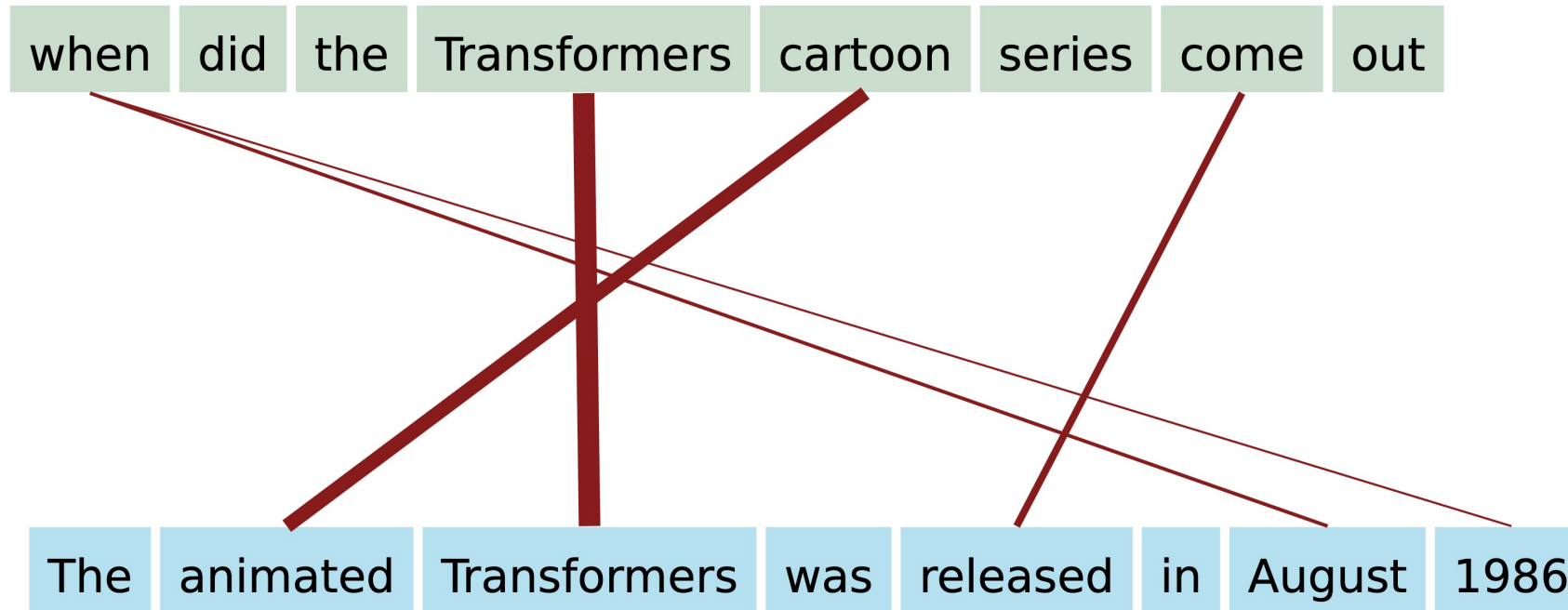
$$\mathbf{MaxSim}(q, \text{doc}) = \sum_i^L \max_j^M \mathbf{Enc}(q)_{N,i}^T \mathbf{Enc}(\text{doc})_{N,j}$$

with  $L$  is the length of  $q$ ,  $M$  the length of  $\text{doc}$ .

Khattab and Zaharia 2020

# Neural IR CoBERT

- Soft alignment with CoBERT



It matches each query token to its most similar document token using **late-interaction max-sim** scoring, enabling **fine-grained semantic comparison** without running BERT on every query–document pair.

# References

---

- [1] <https://web.stanford.edu/class/cs224u/slides/cs224u-neuralir-2023-handout.pdf>
- [2] Jurafsky & Martin. *Speech and Language Processing*, Chapter 11: Information Retrieval and RAG. Draft, Jan 2026.
- [3] Stanford CS224U Neural IR slides: [cs224u-neuralir-2023-handout.pdf](https://web.stanford.edu/class/cs224u/slides/cs224u-neuralir-2023-handout.pdf)
- **Next lecture: Diffusion language models**