Lecture 10 – Information Retrieval and Retrieval-Augmented Generation

Baojian Zhou

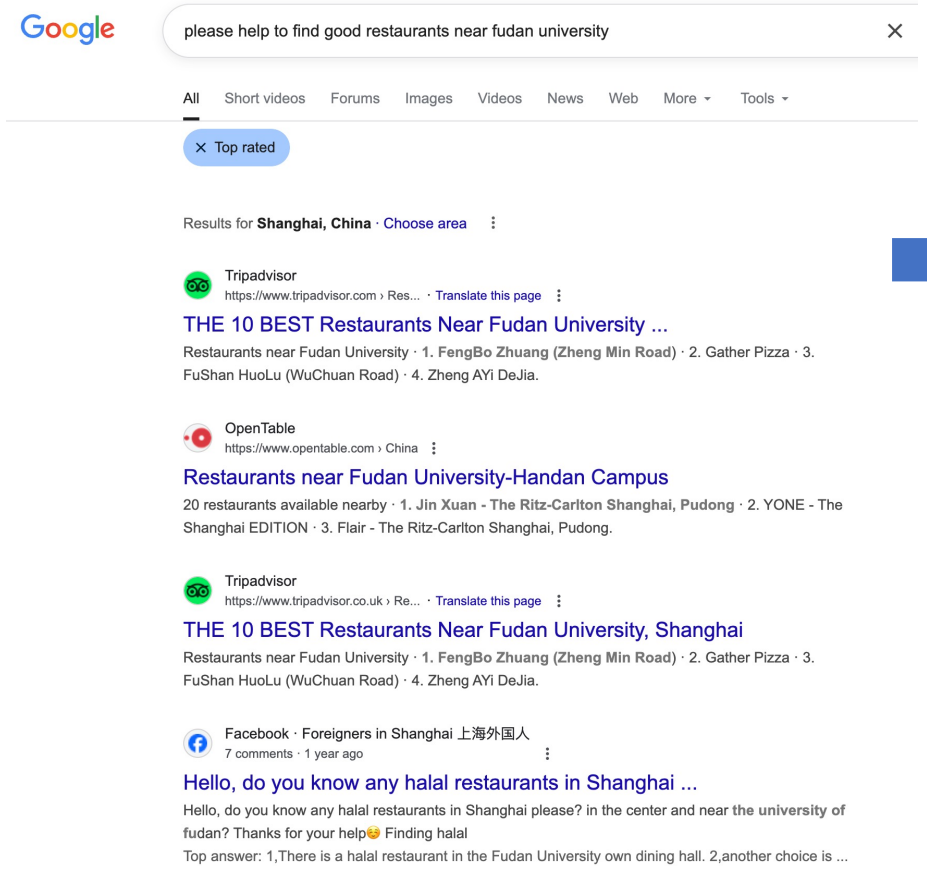
CS40008.01: NLP & LLMs

School of Data Science Fudan University

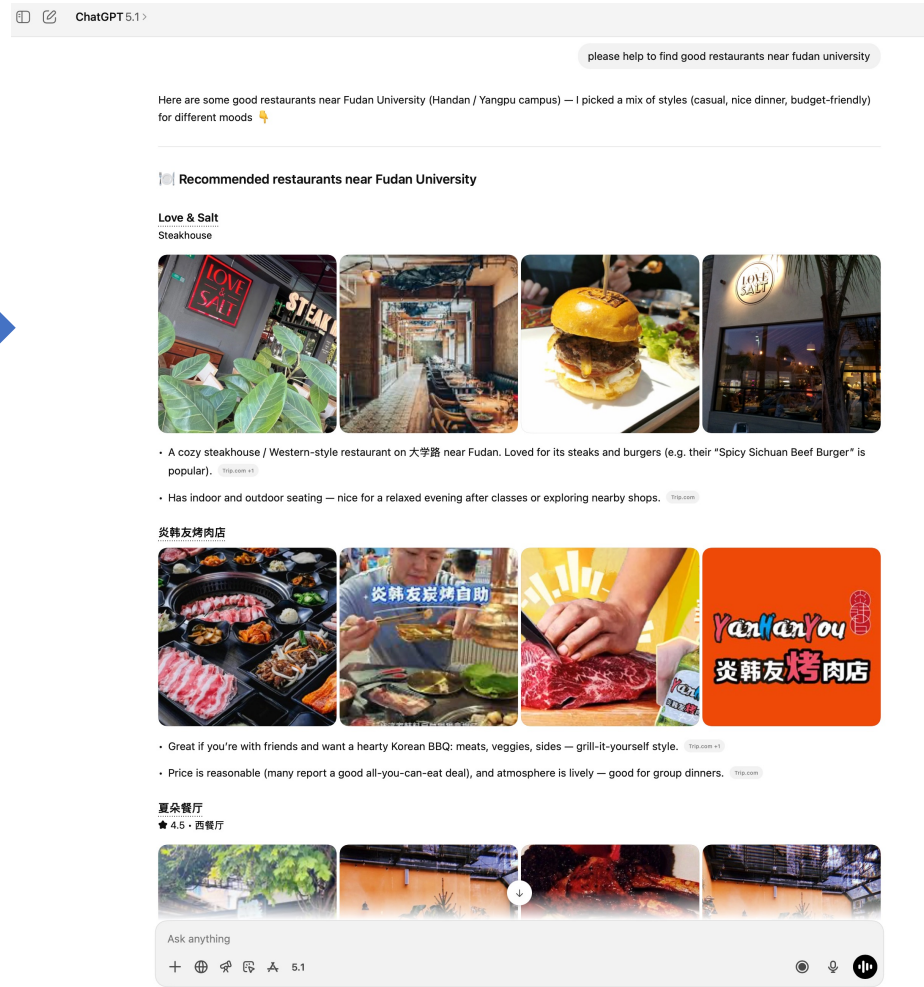
05/14/2026

Motivation

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



From searching engine to GPT5.1



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.

Sources

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

Motivation factoid questions

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

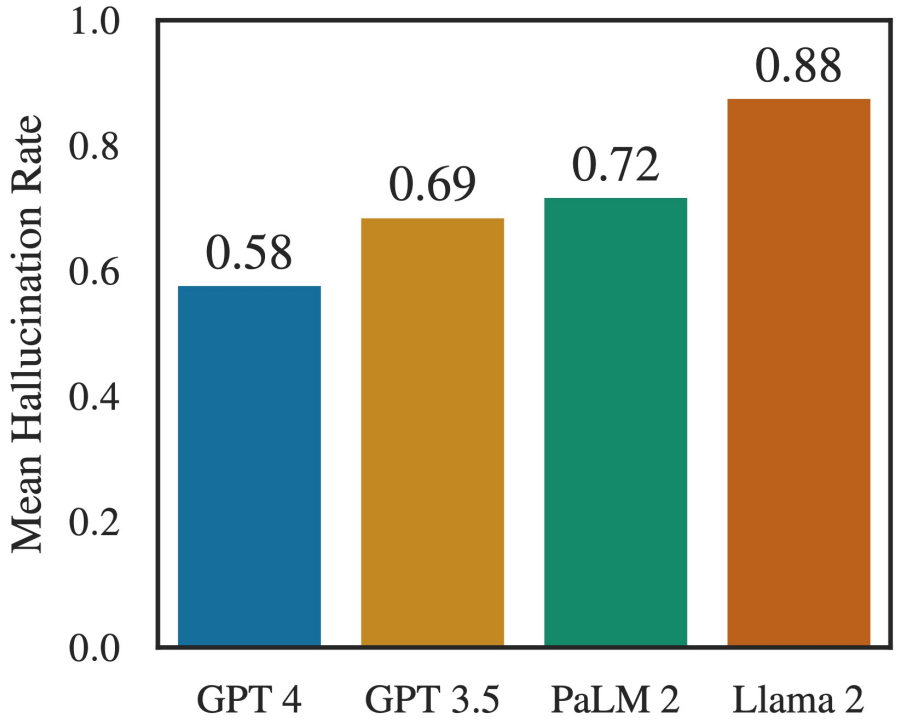
Matthew Dahl,† Varun Magesh,‡ Mirac Suzgun,§ Daniel E. Ho¶

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
Open	response inconsistency with the training corpus	Creative argumentation
	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).

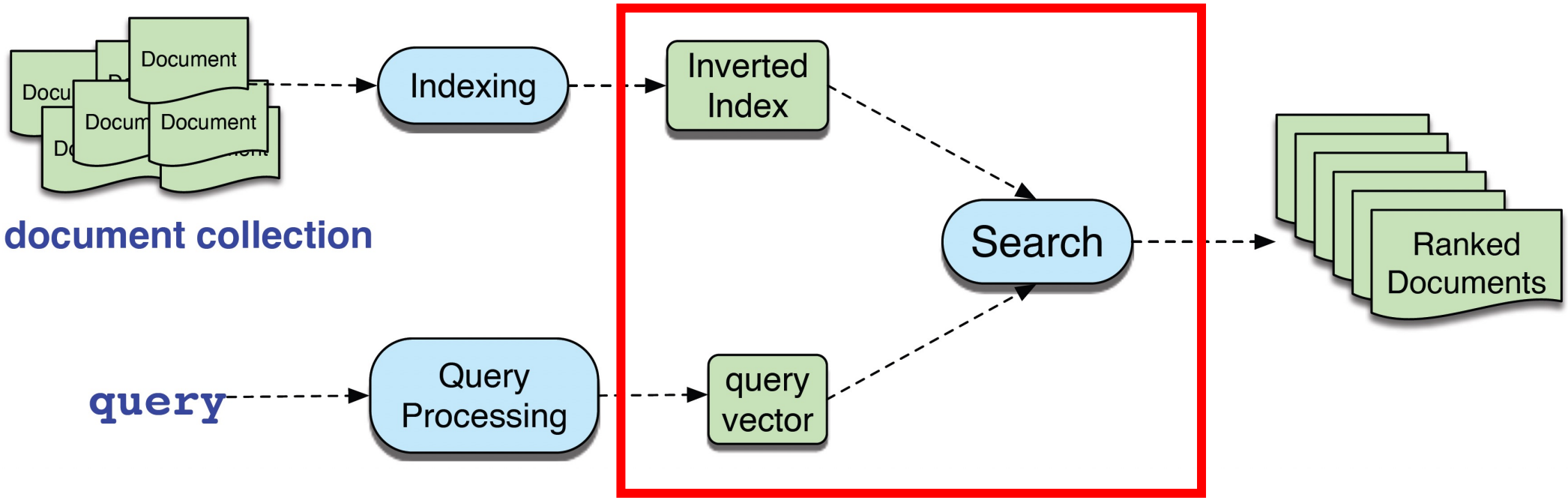
2. Let the LLM **generate an answer using the retrieved documents**.

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

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

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

$$\text{idf}_t = \log_{10} \frac{N}{\text{df}_t}$$

  - 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

$$\text{tf-idf}(t,d) = \text{tf}_{t,d} \cdot \text{idf}_t$$

Information Retrieval (IR) Document Scoring

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

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

- 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
$\|q\| = \sqrt{.125^2 + .301^2} = .326$						

	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	**0.137**	1	1.000	0.125	0.203	**0.0779**
nurse	1	1.000	0.301	0.661	0	0	0	0	0	0
love	1	1.000	0.301	0.661	**0.610**	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
$\|d_1\| = \sqrt{.163^2 + .301^2 + .301^2} = .456$						$\|d_2\| = \sqrt{.125^2 + .602^2} = .615$				
Cosine: $\sum$ of column: **0.747**						Cosine: $\sum$ of column: **0.0779**				

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.

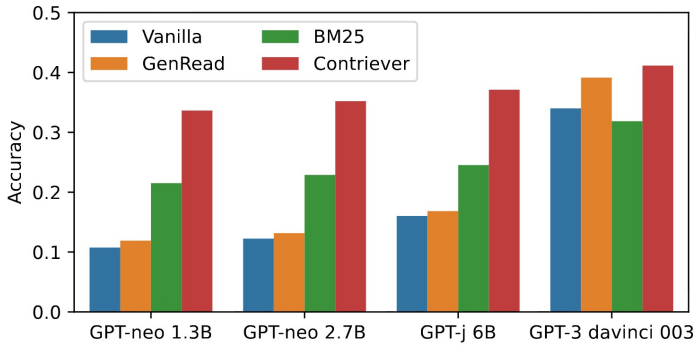


Figure 7: PopQA accuracy of LMs augmented with BM25, Contriever, GenRead, and unassisted (vanilla). **Retrieving non-parametric memories significantly improves the performance of smaller models.** Complete results on PopQA are found in Figure 13. EntityQuestions results are in Figure 14 of the Appendix.

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

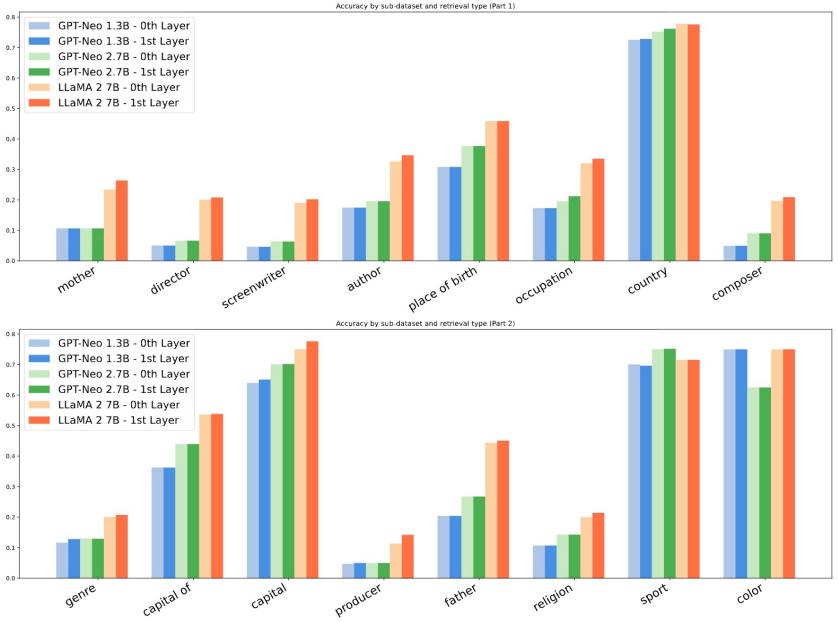
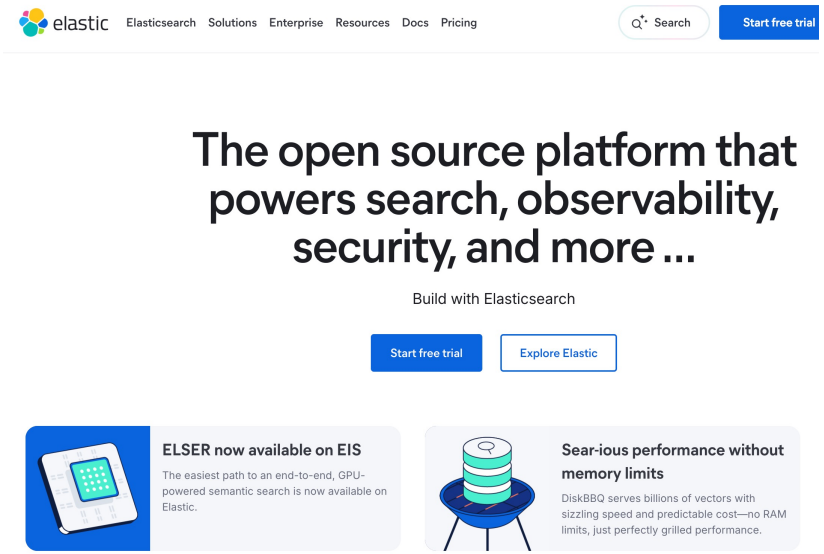


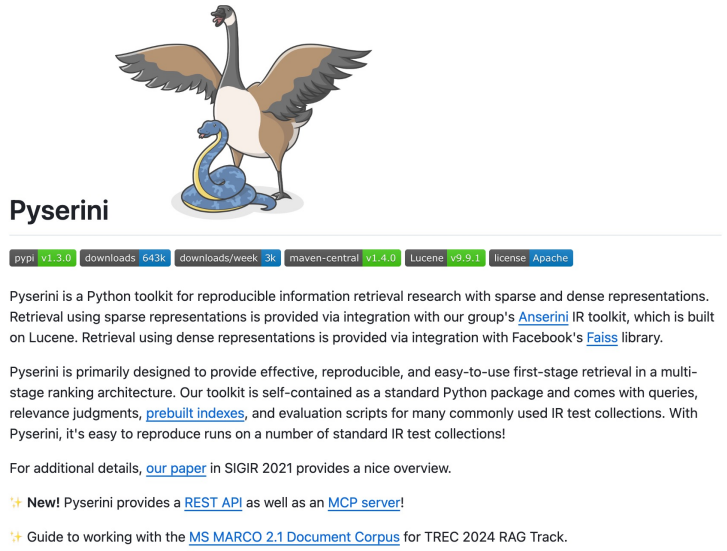
Figure 2: Per-relationship type results on PopQA by different models, showing overall accuracy of EI-ARAG using 0th and 1st layer embeddings based on BM25 RALM.

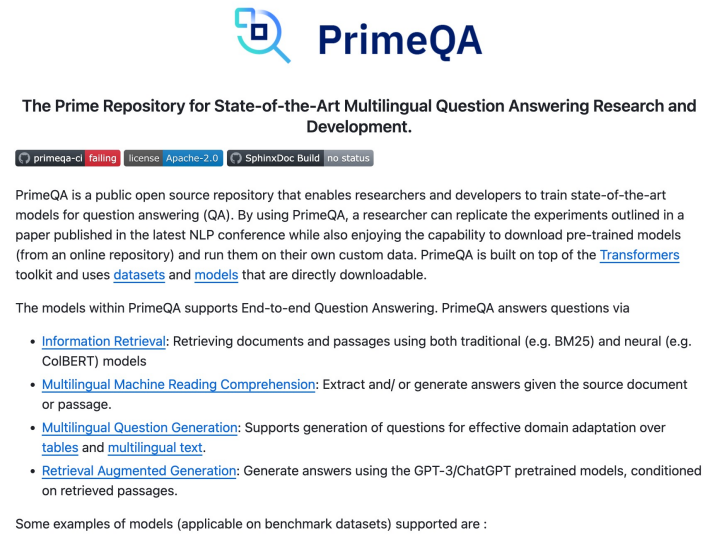
- Embedding-Informed Adaptive Retrieval-Augmented Generation of Large Language Models https://aclanthology.org/2025.coling-main.94.pdf

Tools for classic IR

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

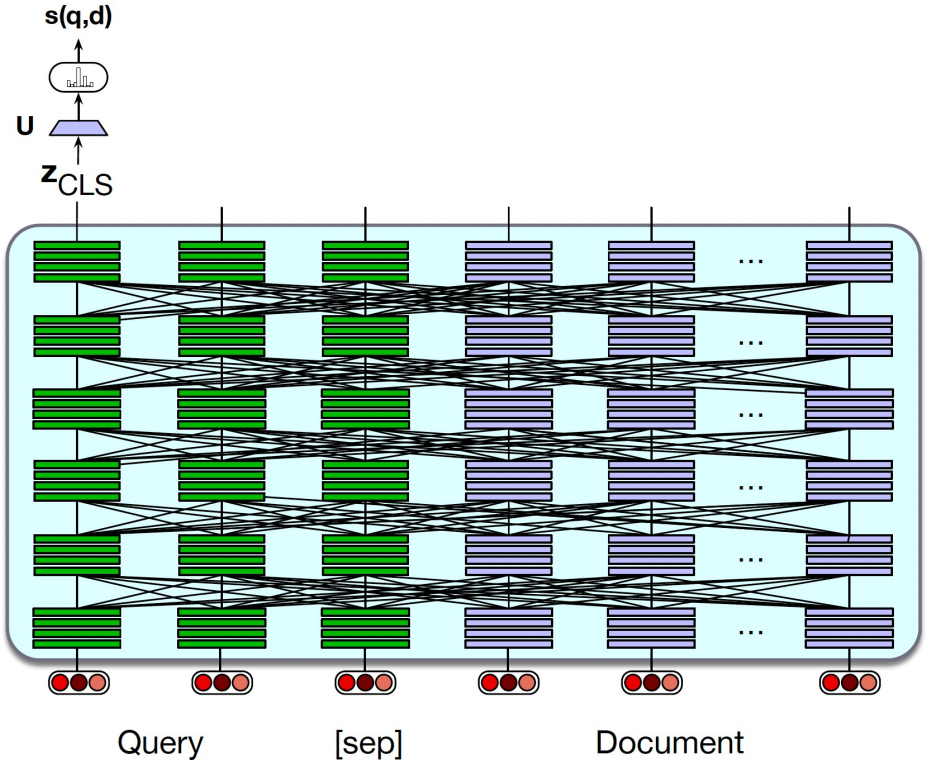




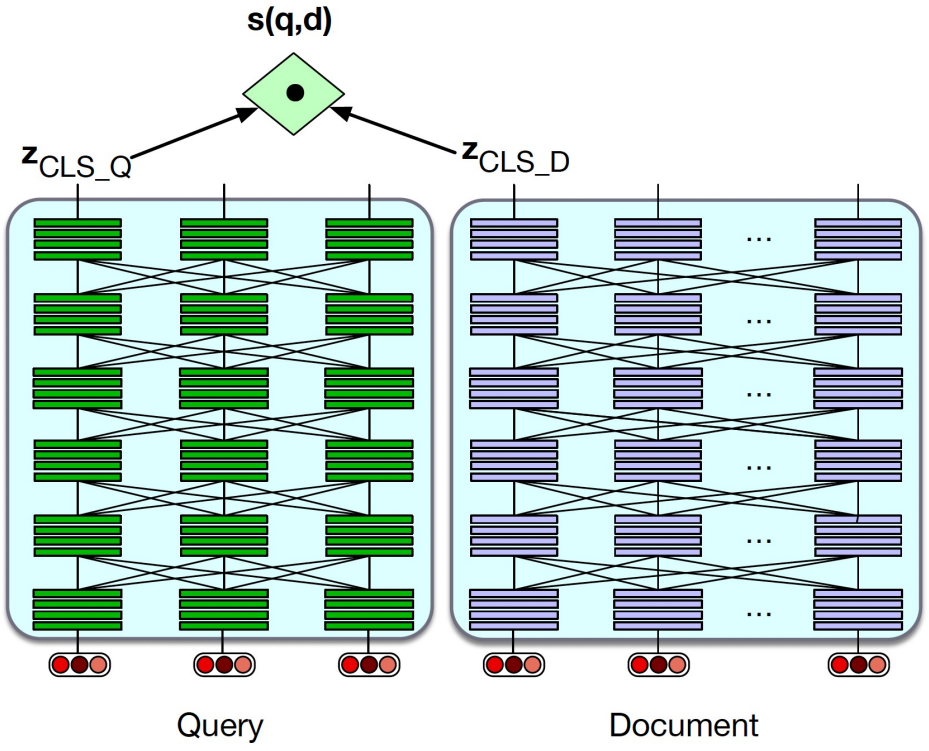


From Sparse BM25 to Dense Retrieval

- **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**.



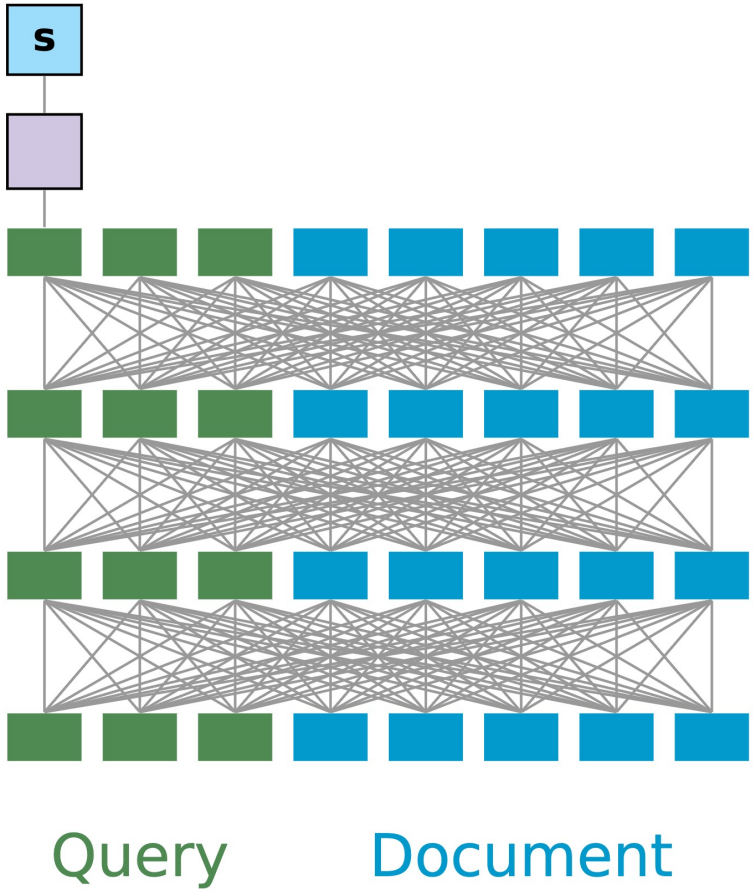
$$\mathbf{z} = \text{BERT}(\text{q}; [\text{SEP}]; \text{d})[\text{CLS}]$$
$$\text{score}(q,d) = \text{softmax}(\mathbf{U}(\mathbf{z}))$$



$$\mathbf{z}_q = \text{BERT}_Q(\text{q})[\text{CLS}]$$
$$\mathbf{z}_d = \text{BERT}_D(\text{d})[\text{CLS}]$$
$$\text{score}(q,d) = \mathbf{z}_q \cdot \mathbf{z}_d$$

Neural IR Cross-encoders

- Incredibly rich, but won't scale!



1. Examples: $\langle q_i, \text{doc}_i^+, \{\text{doc}_{i,k}^-\}\rangle$

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

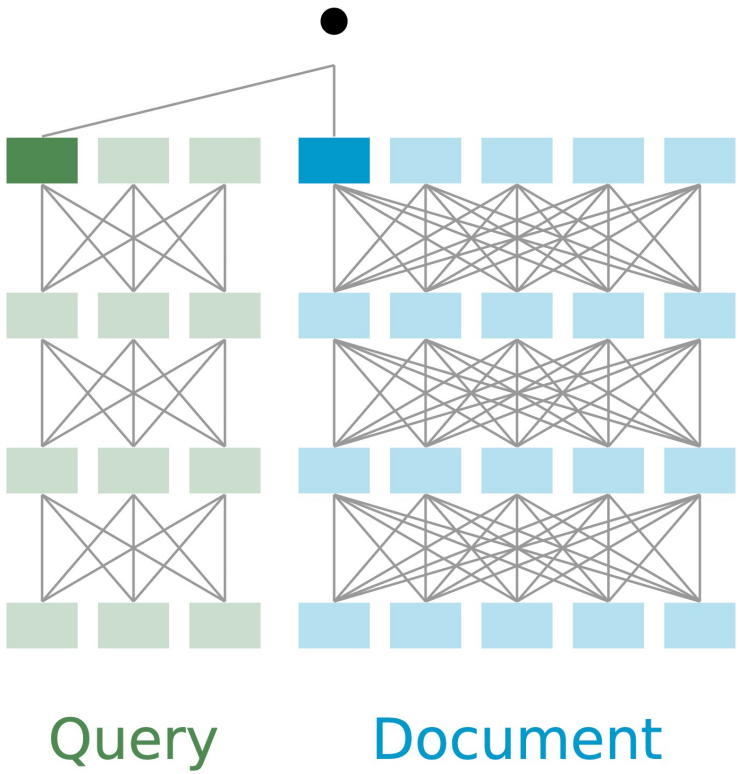
$$\textbf{Rep}(q, \text{doc}) = \text{Dense}\left(\textbf{Enc}([q; \text{doc}]_{N,0})\right)$$

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

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

Neural IR Dense Passage Retrieval

- Highly scalable, but limited query/doc interactions!



1. Examples: $\langle q_i, \text{doc}_i^+, \{\text{doc}_{i,k}^-\}\rangle$

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

   $$\textbf{Sim}(q, \text{doc}) = \textbf{EncQ}(q)_{N,0}{}^\top \textbf{EncD}(\text{doc})_{N,0}$$

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

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

Neural IR Cross-Encoder vs DPR

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

Cross encoders

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

DPR

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

General form

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

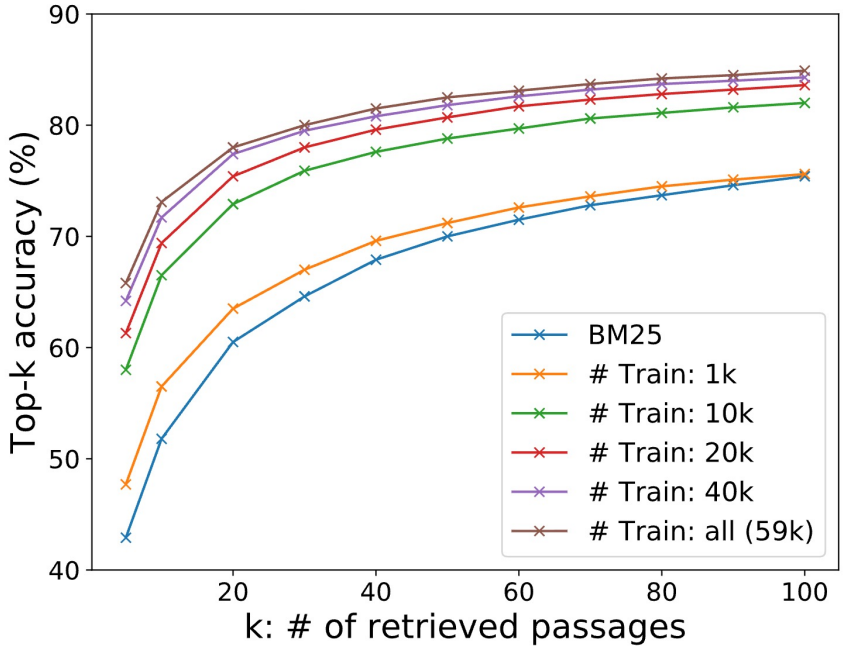
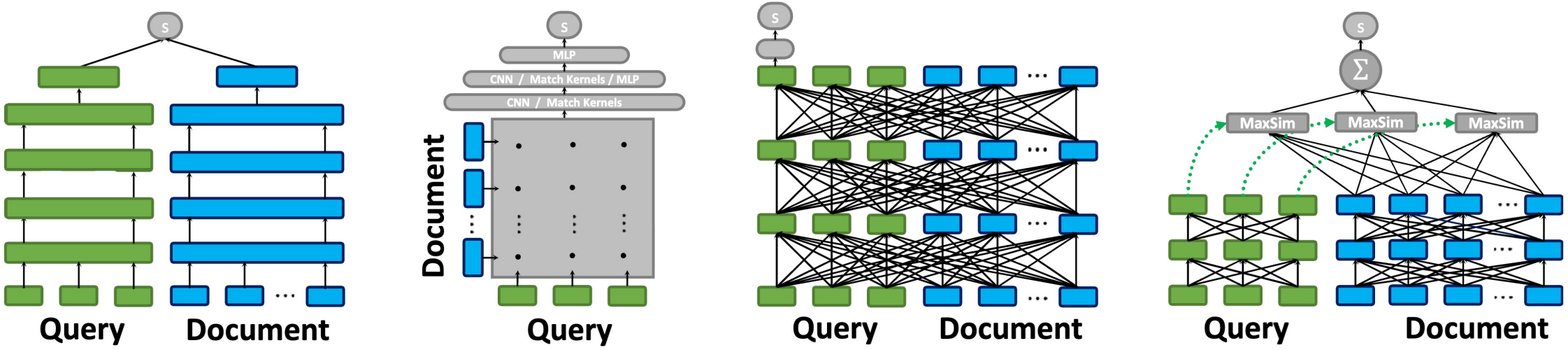


Figure 1: Retriever top-k accuracy with different numbers of training examples used in our dense passage retriever vs BM25. The results are measured on the development set of Natural Questions. Our DPR trained using 1,000 examples already outperforms BM25.

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

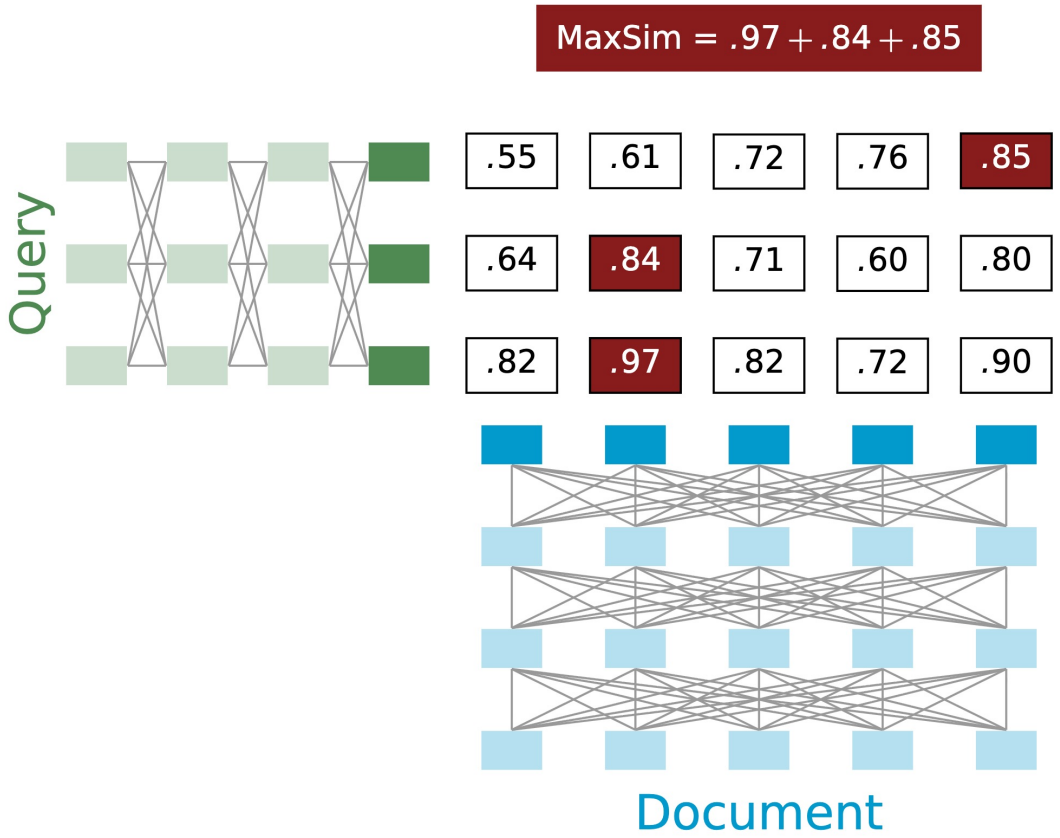
Neural IR ColBERT



- ColBERT 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.

ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Neural IR ColBERT



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:

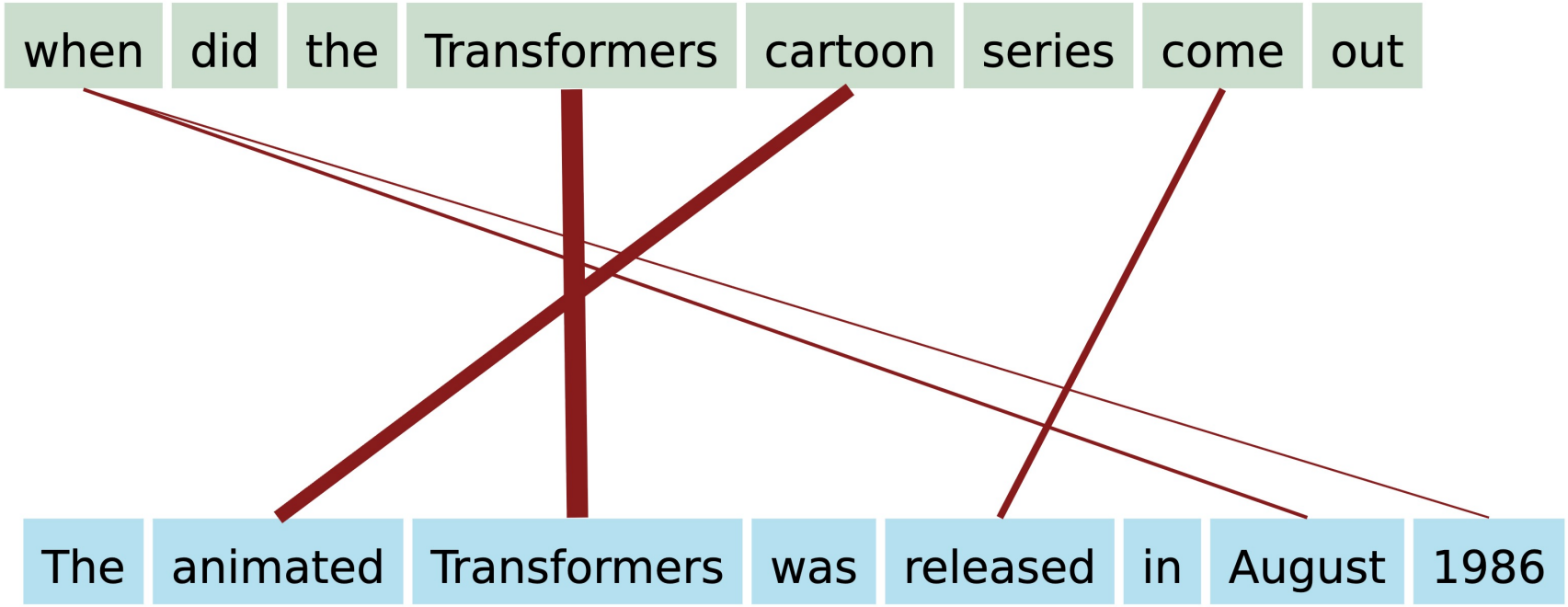
$$\textbf{MaxSim}(q, \text{doc}) = \sum_i^L \max_j^M \textbf{Enc}(q)_{N,i}{}^\top \textbf{Enc}(\text{doc})_{N,j}$$

with L is the length of q, M the length of doc.

Khattab and Zaharia 2020

Neural IR ColBERT

- Soft alignment with ColBERT



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

- **Next lecture: Diffusion language models**