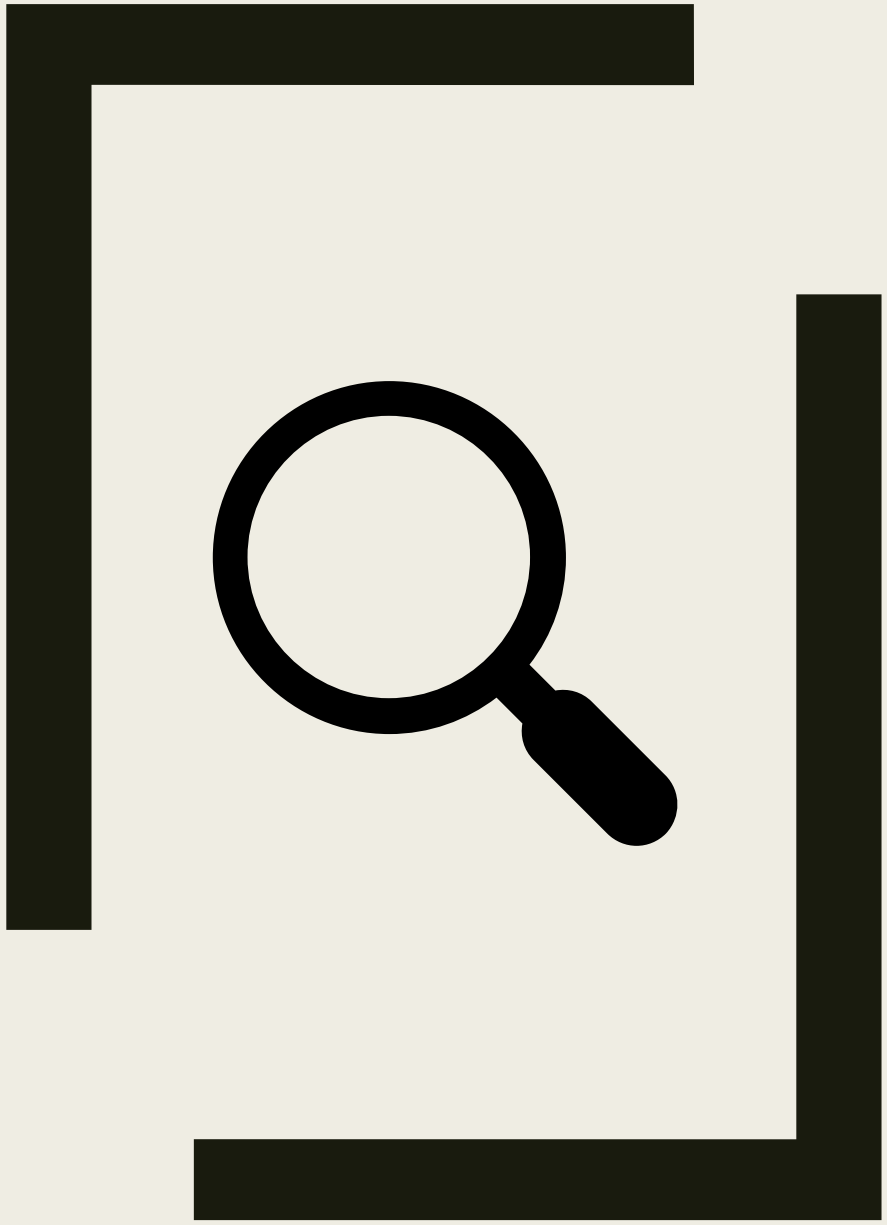


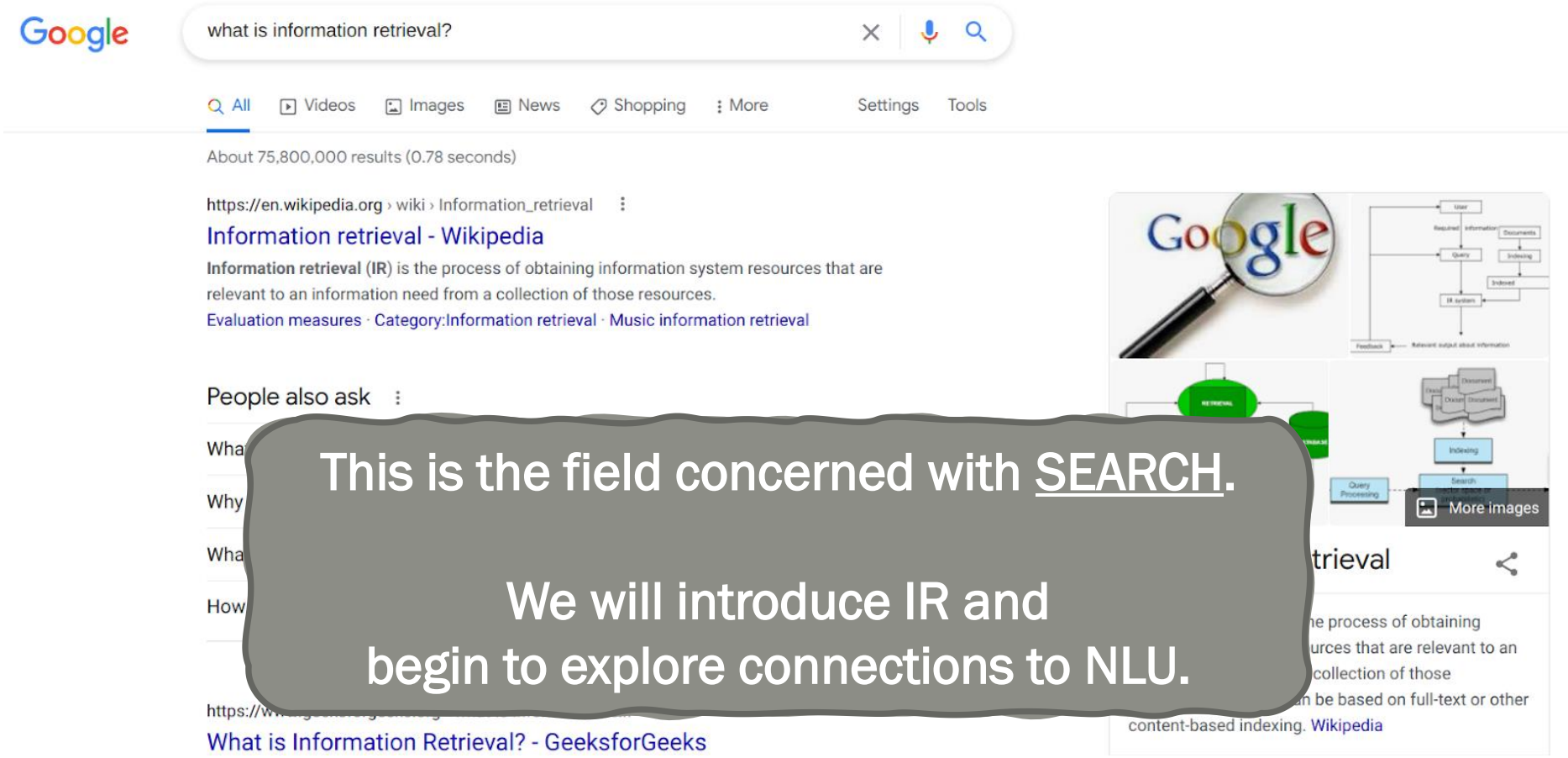
NLU & IR: OVERVIEW

Omar Khattab

CS224U: Natural Language Understanding

Spring 2021

What is information retrieval?



This is the field concerned with SEARCH.

We will introduce IR and
begin to explore connections to NLU.

What is information retrieval?

Finding material that fulfills an **information need** from within a **large collection** of **unstructured documents**.

Simplified definition from IIR Book
(Manning, Raghavan, and Schütze)

What is information retrieval?

Finding material that fulfills an **information need** from within a **large collection** of **unstructured documents**.

SEARCH is essential to IR!

But search over *highly-structured* data (e.g., database records) is NOT typically considered an IR problem.

Simplified definition from IIR Book (Manning, Raghavan, and Schütze)

Relevance — and the “Information Need”

- The goal of a search system is to satisfy an **information need**.
  - Material we retrieve is **relevant** only if it advances this goal.

- In many (most) tasks, the user will express a **query**.
  - But queries can be ambiguous, incomplete, or inaccurate.
  - We must rely on our knowledge of the task and the user.

Typical information needs vary by task

- Beyond Web pages and files, popular types of collections include digital libraries, media items, products, online conversations, etc.

Expression of Information Need	Potential Query	Potential Collection
Find related literature	The full text of the BERT paper	ACL anthology; arXiv CL

Typical information needs vary by task

- Beyond Web pages and files, popular types of collections include digital libraries, media items, products, online conversations, etc.

Expression of Information Need	Potential Query	Potential Collection
Find related literature	The full text of the BERT paper	ACL anthology; arXiv CL
Recommend me a TV show to watch	[no explicit query!]	Netflix shows

Typical information needs vary by task

- Beyond Web pages and files, popular types of collections include digital libraries, media items, products, online conversations, etc.

Expression of Information Need	Potential Query	Potential Collection
Find related literature	The full text of the BERT paper	ACL anthology; arXiv CL
Recommend me a TV show to watch	[no explicit query!]	Netflix shows
Find every relevant patent	Boolean query with technical terms	U.S. Patents

Typical information needs vary by task

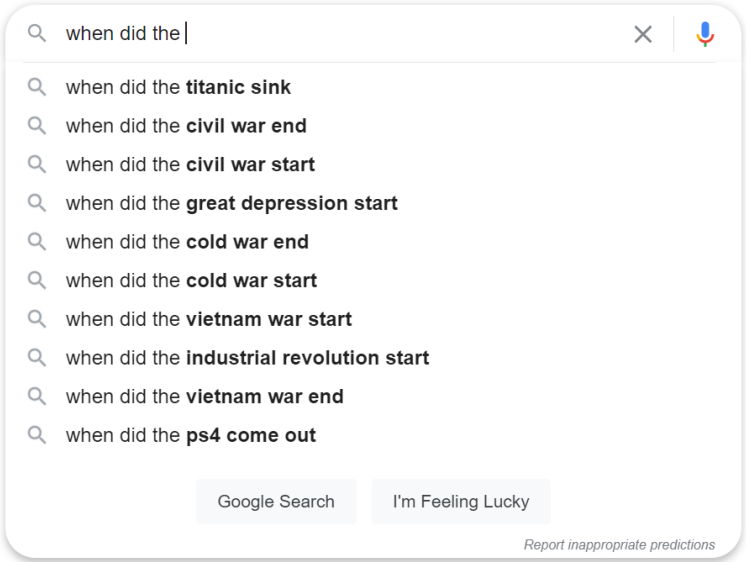
- Beyond Web pages and files, popular types of collections include digital libraries, media items, products, online conversations, etc.

Expression of Information Need	Potential Query	Potential Collection
Find related literature	The full text of the BERT paper	ACL anthology; arXiv CL
Recommend me a TV show to watch	[no explicit query!]	Netflix shows
Find every relevant patent	Boolean query with technical terms	U.S. Patents
Buy a new laptop	Short conversation: system asks questions to ascertain your criteria	E-commerce platforms

Typical information needs vary by task

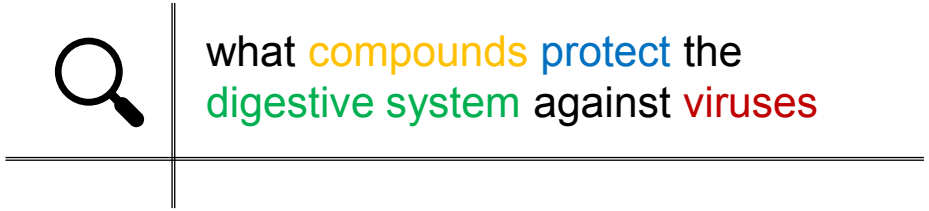
- Each search task poses unique challenges!
  - Many of them lack key features that make Web search work.

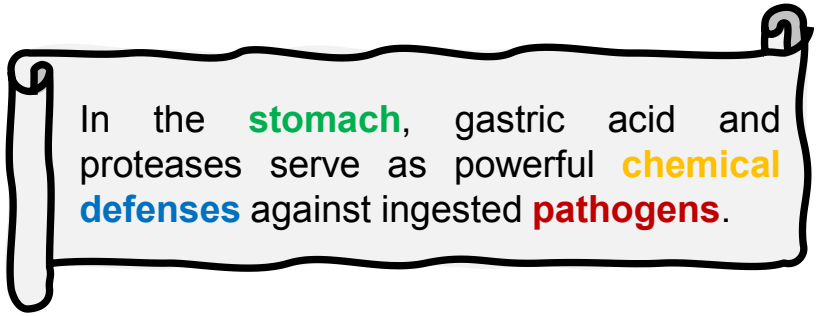
- Unlike, say, Slack search, Web search can often rely on lots of:
  - Popular “head” queries
  - Redundant documents on common topics
  - Explicit (hyper)links between documents



Where does NLU fit in IR?

- Queries and documents are often expressed in natural language.
- Due to **vocabulary mismatch**, lexical matching doesn't suffice!





Where does IR fit into NLU?

- Advanced models often have information needs too!

- Retrieval in NLU can contribute to:
  - **Creating new challenging NLU tasks**
  - Improving model efficiency and quality for existing NLU tasks
  - Evaluating NLU systems whenever the output domain is large

Retrieval supports “open-domain” NLU tasks

- We’ve briefly introduced SQuAD before…

Context: Chemical barriers also protect against infection. The skin and respiratory tract secrete antimicrobial peptides such as the β-defensins. […] In the stomach, gastric acid and proteases serve as powerful chemical defenses against ingested pathogens.

Question: What compounds in the stomach protect against ingested pathogens?

Answer: gastric acid and proteases

Standard Question Answering (e.g., SQuAD)

Rajpurkar, Pranav, et al. “SQuAD: 100,000+ questions for machine comprehension of text." EMNLP’16

From standard QA to open-domain QA

- Drop the passage hint!

Context: All of [English] Wikipedia, with no special hints about the answer

Question: What compounds in the stomach protect against ingested pathogens?

Answer: gastric acid and proteases

Open-Domain Question Answering (e.g., this “Open-SQuAD”)

Chen, Danqi, et al. "Reading Wikipedia to answer open-domain questions." ACL’17.

Open-Domain QA: **Closed-Book Approaches**

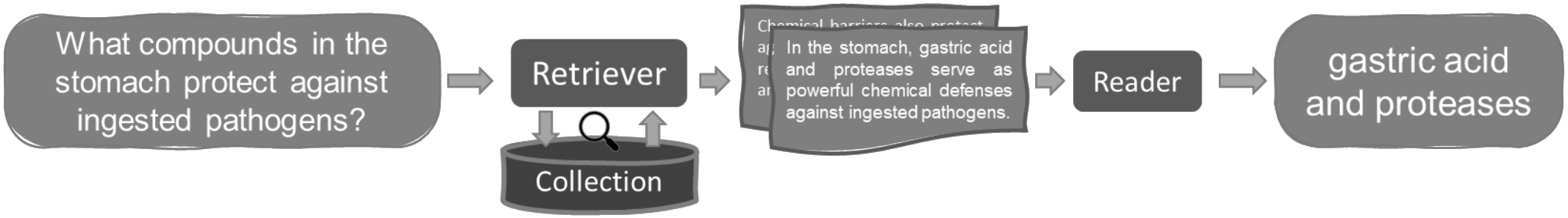
- Feed the question to a monolithic black-box generative model!
  - Knowledge is stored *implicitly* in the model parameters
  - Often as a byproduct of language-model pretraining
  - Need more “knowledge”? Train a larger model on more data!

Roberts, Adam, Colin Raffel, and Noam Shazeer. "How Much Knowledge Can You Pack Into the Parameters of a Language Model?." *EMNLP*'20.

Open-Domain QA: **Open-Book Approaches**

- Feed the question to a modular **retrieve-and-read** architecture
  - Knowledge is stored *explicitly* in the collection
  - We decouple **reasoning** and **knowledge**

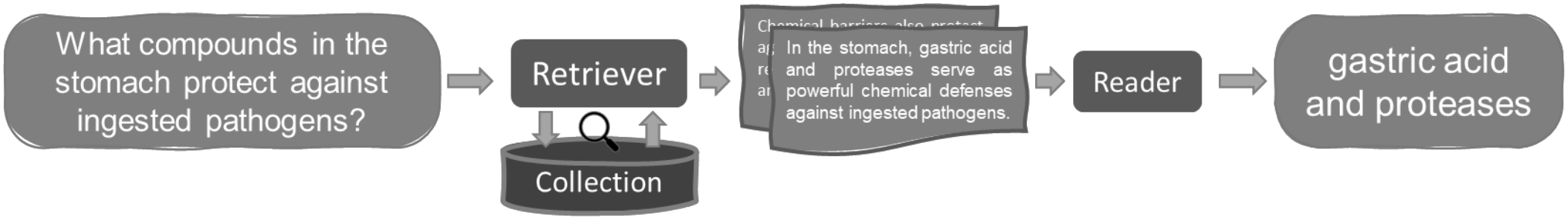
*The reader has an **information need.** The retriever's task is to satisfy it efficiently and accurately.*



Open-Domain QA: **Open-Book Approaches**

- ✔ Models can be much smaller
- ✔ Knowledge can be updated (or customized) without retraining
- ✔ Model predictions might become more explainable

- ✕ We now need to worry about the interactions between a **retriever** and **reader**



A few retrieval-based NLP tasks

Task Name	Input	Output
Open-Domain QA	Question	Answer

A few retrieval-based NLP tasks

Task Name	Input	Output
Open-Domain QA	Question	Answer
Fact Checking	Claim	Binary Label & Justification

A few retrieval-based NLP tasks

Task Name	Input	Output
Open-Domain QA	Question	Answer
Fact Checking	Claim	Binary Label & Justification
Query-Focused Summarization	Topic	Summary
Informative Dialogue	Conversation Turns	Response

A few retrieval-based NLP tasks

Task Name	Input	Output
Open-Domain QA	Question	Answer
Fact Checking	Claim	Binary Label & Justification
Query-Focused Summarization	Topic	Summary
Informative Dialogue	Conversation Turns	Response
Entity Linking	Utterance	Mapping from spans to entities in a knowledge base

Retrieval-based NLP tasks

- KILT is a recent benchmark that brings together several datasets for **knowledge-intensive** language tasks.

- These are tasks that explicitly have a knowledge component.

Open Question: Can retrieval dramatically improve performance for standard NLU tasks too?

Accurate knowledge matters for most (all?) tasks!
“Bring your own book!”

Next...

- The remainder is structured as small crash courses into:
  - Classical Information Retrieval
  - Neural Information Retrieval
  - Open-Domain Question Answering

References

Manning, Christopher, Prabhakar Raghavan and Schutze, H. “Introduction to Information Retrieval.” (2008).

Manning, Christopher, and Pandu Nayak (2019). CS276 Information Retrieval and Web Search: Inverted Indices [Class handout]. Retrieved from http://web.stanford.edu/class/cs276/19handouts/lecture2-intro-boolean-6per.pdf

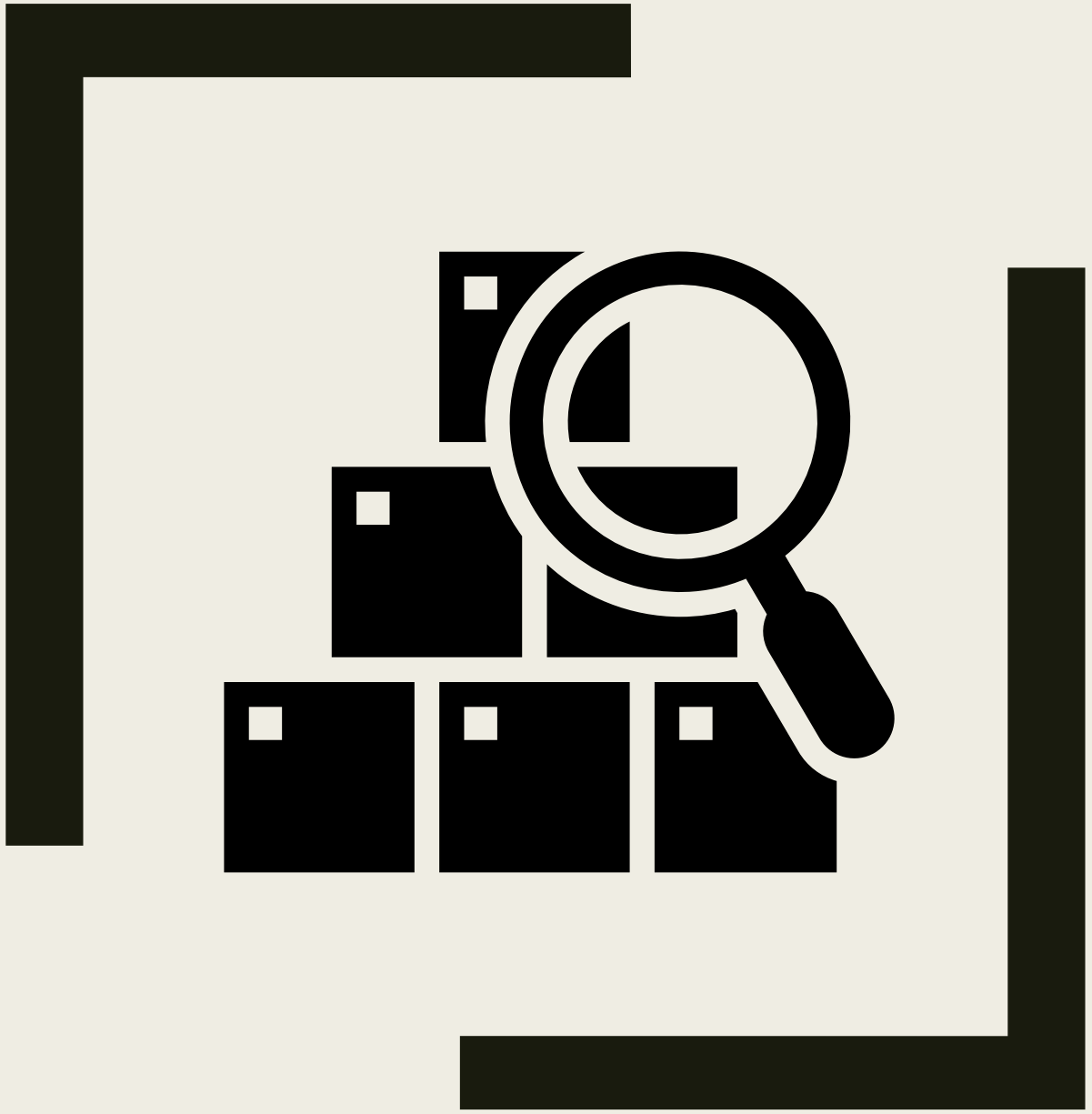
Elsayed, Tamer. CMPT621 Information Retrieval: Introduction to IR [Class handout]. Retrieved from https://www.dropbox.com/sh/8oeivk53ymsj3oy/AABt9M6ve5qYCZShkS7a5BkZa/Lecture%20Slides?dl=0&preview=1-CMPT621-S21-Session1-Intro+to+IR.pdf

Rajpurkar, Pranav, et al. “SQuAD: 100,000+ questions for machine comprehension of text." EMNLP’16.

Chen, Danqi, et al. "Reading Wikipedia to answer open-domain questions." ACL’17.

Roberts, Adam, Colin Raffel, and Noam Shazeer. "How Much Knowledge Can You Pack Into the Parameters of a Language Model?." EMNLP’20.

Petroni, Fabio, et al. "KILT: a benchmark for knowledge intensive language tasks." NAACL’21.



NLU & IR: CLASSICAL IR

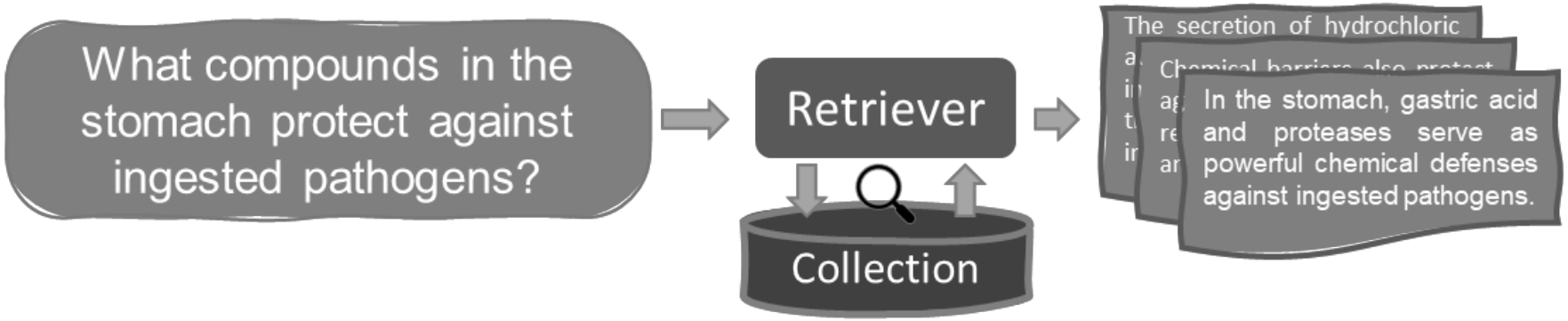
Omar Khattab

CS224U: Natural Language Understanding

Spring 2021

Ranked Retrieval

- Scope: A large corpus of text documents (e.g., Wikipedia)
- Input: A textual query (e.g., a natural-language question)
- Output: **Top-K Ranking** of **relevant** documents (e.g., top-100)



How do we conduct ranked retrieval?

- We've touched on one way before: the **Term–Document Matrix**

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
against	0	0	0	1	0	0	3	2	3	0
age	0	0	0	1	0	3	1	0	4	0
agent	0	0	0	0	0	0	0	0	0	0
ages	0	0	0	0	0	2	0	0	0	0
ago	0	0	0	2	0	0	0	0	3	0
agree	0	1	0	0	0	0	0	0	0	0
ahead	0	0	0	1	0	0	0	0	0	0
ain't	0	0	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0	0	0
aka	0	0	0	1	0	0	0	0	0	0

- With good weights, this allows us to answer **single-term** queries!

How do we conduct ranked retrieval?

- For multi-term queries, classical IR models would tokenize and then treat the tokens independently.

$$RelevanceScore(query, doc) = \sum_{term \in query} Weight_{doc,term}$$

- This reduces a large fraction of classical IR to:
  - How do we best tokenize (and stem) queries and documents
  - **How do we best weight each term–document pair**

Term–Document Weighting: Intuitions

- **Frequency** of occurrence will remain a primary factor
  - If a term t occurs frequently in document d, the document is more likely to be relevant for queries including t

- **Normalization** will remain a primary component too
  - If that term t is rather rare, then document d is even more likely to be relevant for queries including t
  - If that document d is rather short, this also improves its odd

- Amplify the important, the trustworthy, the unusual; deemphasize the mundane and the quirky.

Term–Document Weighting: TF-IDF

- Let $N = |Collection|$ and $df(term) = |\{doc \in Collection : term \in doc\}|$

$$TF(term, doc) = \log(1 + Freq(term, doc))$$

$$IDF(term) = \log \frac{N}{df(term)}$$

TF and IDF both grow sub-linearly with frequency and 1/df (in particular, logarithmically).

$$TF.IDF(term, doc) = TF(term, doc) \times IDF(term)$$

$$TF.IDF(query, doc) = \sum_{term \in query} TF.IDF(term, doc)$$

Term–Document Weighting: BM25

Or “Finding the best match, *seriously* this time! Attempt #25” :-)

$$IDF(term) = \log(1 + \frac{N - df(term) + 0.5}{df(term) + 0.5})$$

$$TF(term, doc) = \frac{Freq(term, doc) \times (k + 1)}{Freq(term, doc) + k \times (1 - b + b \times \frac{|doc|}{avgdoclen})}$$

$$BM25(term) = BM25{:}TF(term, doc) \times BM25{:}IDF(term)$$

$$BM25\,(query, doc) = \sum_{term \in query} BM25\,(term, doc)$$

k, b are parameters.

Unlike TF-IDF, term frequency in BM25 **saturates** and **penalizes longer documents**!

Robertson, Stephen, and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. Now Publishers Inc, 2009.

Efficient Retrieval: Inverted Indexing

- Raw Collection: Document → Terms

- Term–document matrix: Term -> Documents
  - But it's extremely sparse and thus wastes space!

- The **inverted index** is just a sparse encoding of this matrix
  - Mapping each unique term t in the collection to a posting list
  - The posting list enumerates **non-zero** <Freq, DocID> for t

Beyond term matching in classical IR...

- Query and Document expansion
- Term dependence and phrase search
- Learning to Rank with various features:
  - Different document fields (e.g., title, body, anchor text)
  - Link Analysis (e.g., PageRank)

Lots of IR exploration into these!
However, BM25 was a very strong baseline on the best you can do “ad-hoc”—until 2019 with BERT-based ranking!

IR Evaluation

- A search system must be **efficient** and **effective**
  - If we had infinite resources, we'd just hire experts to look through all the documents one by one!

- Efficiency
  - **Latency (milliseconds; for one query)**
  - Throughput (queries/sec)
  - Space (GBs for the index? TBs?)
  - Hardware required (one CPU core? Many cores? GPUs?)
  - Scaling to various collection sizes, under different loads

IR Effectiveness

- Do our top-k rankings fulfill users' information needs?
  - Often harder to evaluate than classification/regression!
- If you have lots of users, you can run online experiments…
- But we're typically interested in reusable **test collections**

Test Collections

- Document Collection (or “Corpus”)
- Test Queries (or “Topics”)
  - Could also include a train/dev split, if resources allow!
  - Or, in some cases, cross-validation could be used.
- Query–Document Relevance Assessments
  - Is document j relevant to query i?
    - Binary judgments: relevant (0) vs. non-relevant (1)
    - Graded judgments: {-1, 0, 1, 2} (e.g., junk, irrelevant, relevant, key)

We typically have to make the (significant!) assumption that unjudged documents are irrelevant. Some test collections would only label a few positives per query.

Test Collections: TREC

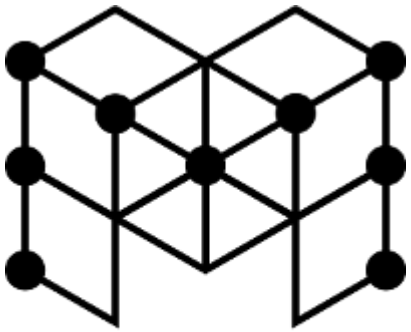
- Text REtrieval Conference (TREC) includes numerous annual tracks for comparing IR systems.

- The 2021 iteration has tracks for Conversational Assistance, Health Misinformation, Fair Ranking, “Deep Learning”.

- TREC tends to emphasize careful evaluation with a very small set of queries (e.g., 50 queries, each with >100 annotated documents)
  - Having only few test queries does not imply few documents!

Test Collections: MS MARCO Ranking Tasks

- MS MARCO Ranking is the largest public IR benchmark
  - adapted from a Question Answering dataset
  - consists of more than 500k **Bing search queries**
    - Sparse labels: approx. one relevance label per query!
    - Fantastic for training IR models!



- MS MARCO Passage Ranking (9M short passages; sparse labels)
- MS MARCO Document Ranking (3M long documents; sparse labels)
- TREC DL'19 and DL'20 (short&long; dense labels for few queries)

Test Collections: Other Benchmarks

- Lots of small or domain-specific benchmarks!
- BEIR is a recent effort to use those for testing models in “zero-shot” scenarios

We will also see later that OpenQA benchmarks can serve as large IR benchmarks too!

Split (→)					Train	Dev	Test			(Train + Dev + Test)	
Task (↓)	**Domain (↓)**	**Dataset (↓)**	**Title**	**Relevancy**	**#Pairs**	**#Query**	**#Query**	**#Corpus**	**Avg. Docs / Q**	**Avg. Q Len**	**Avg. Doc Len**
Passage-Retrieval	Misc.	MSMARCO	✗	Binary	532,761	—-	6,980	8,841,823	1.1	5.96	55.98
Bio-Medical	Bio-Medical	(1) TREC-COVID	✓	3-level	—-	—-	50	171,332	493.5	10.60	160.77
Information	Bio-Medical	(2) NFCorpus	✓	3-level	110,575	324	323	3,633	38.2	3.30	232.26
Retrieval (IR)	Bio-Medical	(3) BioASQ	✓	Binary	32,916	—-	500	14,914,602	4.7	8.05	202.61
Question	Wikipedia	(4) NQ	✓	Binary	132,803	—-	3,452	2,681,468	1.2	9.16	78.88
Answering	Wikipedia	(5) HotpotQA	✓	Binary	170,000	5,447	7,405	5,233,329	2.0	17.61	46.30
(QA)	Finance	(6) FiQA-2018	✗	Binary	14,166	500	648	57,638	2.6	10.77	132.32
Tweet-Retrieval	Twitter	(7) Signal-1M (RT)	✗	3-level	—-	—-	97	2,866,316	19.6	9.30	13.93
News-Retrieval	News	(8) TREC-NEWS	✓	5-level	—-	—-	57	594,977	19.6	11.14	634.79
Argument	Misc.	(9) ArguAna	✓	Binary	—-	—-	1,406	8,674	1.0	192.98	166.80
Retrieval	Misc.	(10) Tóuche-2020	✓	6-level	—-	—-	49	382,545	49.2	6.55	292.37
Duplicate-Question	StackEx.	(11) CQADupStack	✓	Binary	—-	—-	13,145	457,199	1.4	8.59	129.09
Retrieval	Quora	(12) Quora	✗	Binary	—-	5,000	10,000	522,931	1.6	9.53	11.44
Entity-Retrieval	Wikipedia	(13) DBPedia	✓	3-level	—-	67	400	4,635,922	38.2	5.39	49.68
Citation-Prediction	Scientific	(14) SCIDOCS	✓	Binary	—-	—-	1,000	25,657	4.9	9.38	176.19
	Wikipedia	(15) FEVER	✓	Binary	140,085	6,666	6,666	5,416,568	1.2	8.13	84.76
Fact Checking	Wikipedia	(16) Climate-FEVER	✓	Binary	—-	—-	1,535	5,416,593	3.0	20.13	84.76
	Scientific	(17) SciFact	✓	Binary	920	—-	300	5,183	1.1	12.37	213.63

Table 1: Statistics of all the tasks, domains and datasets included in **BEIR**. Few datasets contain documents without titles. Relevancy column indicates the relation between the query and document: binary (relevant, irrelevant) or further graded into sub-levels. Avg. Docs/Query column indicates the average relevant documents per question.

Thakur, Nandan, et al. "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.“ *arXiv:2104.08663* (2021)

IR Effectiveness Metrics

- We'll use "metric"@K, often with K in {5, 10, 100, 1000}.
  - Selection of the metric (and the cutoff K) depends on the task.

- For all metrics here, we'll [macro-]average across all queries.
  - All queries will be assigned equal weight, for our purposes.

IR Effectiveness Metrics: Success & MRR

- Let $rank \in \{1, 2, 3, \ldots\}$ be the position of the first relevant document
- Success@K = $\begin{cases} 1 & \text{if } rank \leq K \\ 0 & \text{otherwise} \end{cases}$
- ReciporcalRank@K = $\begin{cases} 1/rank & \text{if } rank \leq K \\ 0 & \text{otherwise} \end{cases}$
  - This is MRR (M for “mean”), but dropped the M as we’re looking at only one query

IR Effectiveness Metrics: Precision & Recall

- Let $Ret(K)$ be the top-K retrieved documents
- Let Rel be the set of all documents judged as relevant

- Precision@K = $\frac{|Ret(K) \cap Rel|}{K}$

- Recall@K = $\frac{|Ret(K) \cap Rel|}{|Rel|}$

IR Effectiveness Metrics: MAP

- (M)AP = (Mean) Average Precision
- Let $rank_1, rank_2, \ldots, rank_{|Rel|}$ be the positions of <u>all</u> relevant documents
  - Compute precision@i at each of those positions—and average!
- Equivalently, AveragePrecision@K =

$$\frac{\sum_{i=1}^{K} \begin{cases} Precision@i & \text{if } relevant?\,(i^{th}\ document) \\ 0 & \text{otherwise} \end{cases}}{|Rel|}$$

IR Effectiveness Metrics: DCG

- Discounted Cumulative Gain
  - Not inherently normalized, so we also consider Normalized DCG

$$DCG@K = \sum_{i=1}^{K} \frac{graded\_relevance(i^{th}\ document)}{\log_2(i+1)}$$

$$NDCG@K = \frac{DCG@K}{ideal\ DCG@K}$$

Next...

- Neural IR.

References

Manning, Christopher, Prabhakar Raghavan and Schutze, H. “Introduction to Information Retrieval.” (2008).

Manning, Christopher, and Pandu Nayak (2019). CS276 Information Retrieval and Web Search: Evaluation [Class handout]. Retrieved from http://web.stanford.edu/class/cs276/19handouts/lecture8-evaluation-6per.pdf

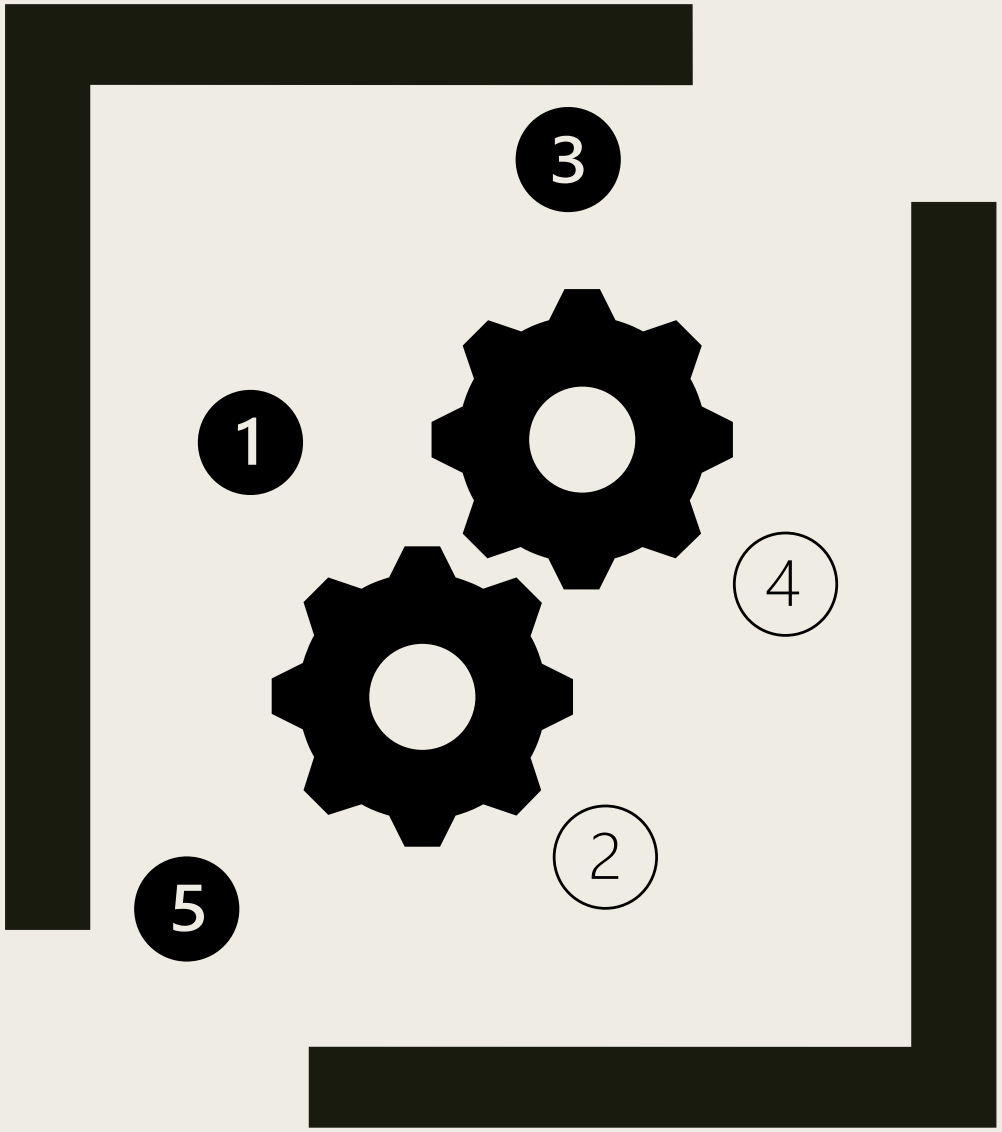
Hofstätter, Sebastian. Advanced Information Retrieval: {IR Fundamentals, Evaluatin, Test Collections} [Class handout]. Retrieved from https://github.com/sebastian-hofstaetter/teaching

Robertson, Stephen, and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. Now Publishers Inc, 2009.

Nguyen, Tri, et al. "MS MARCO: A human generated machine reading comprehension dataset." CoCo@ NIPS. 2016.

Craswell, Nick, et al. "TREC Deep Learning Track: Reusable Test Collections in the Large Data Regime." arXiv preprint arXiv:2104.09399 (2021).

Thakur, Nandan, et al. "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.“ arXiv:2104.08663 (2021)



NLU & IR: NEURAL IR (I)

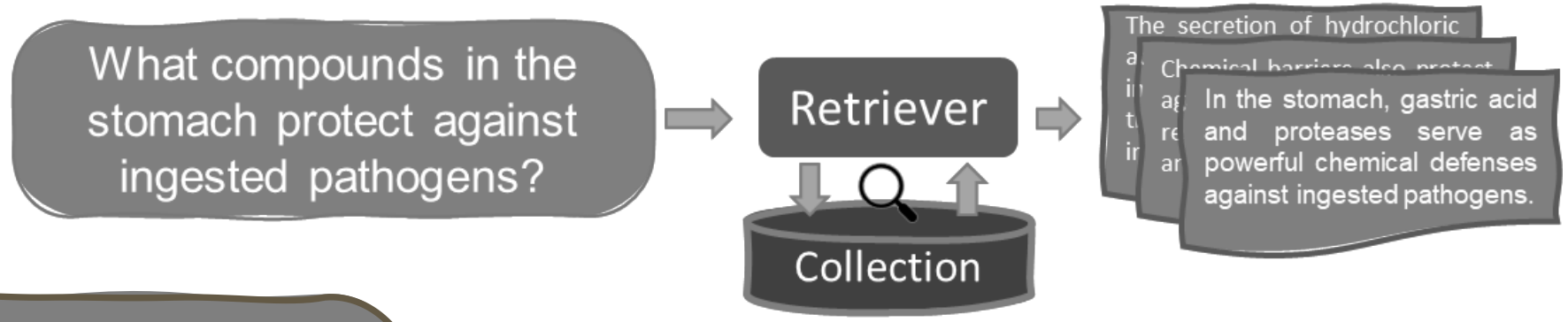
Omar Khattab

CS224U: Natural Language Understanding

Spring 2021

Ranked Retrieval

- Scope: A large corpus of text documents (e.g., Wikipedia)
- Input: A textual query (e.g., a natural-language question)
- Output: **Top-K Ranking** of **relevant** documents (e.g., top-100)

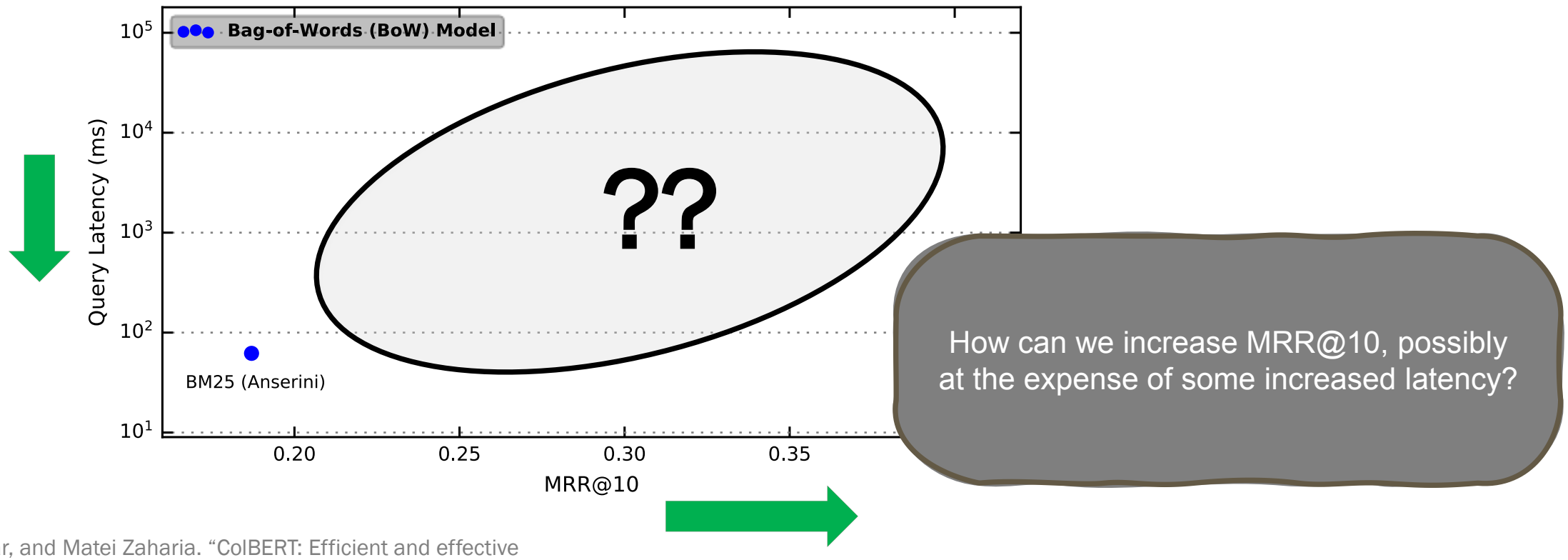


Should we just use BM25?

Why not?

Efficiency–Effectiveness Tradeoff

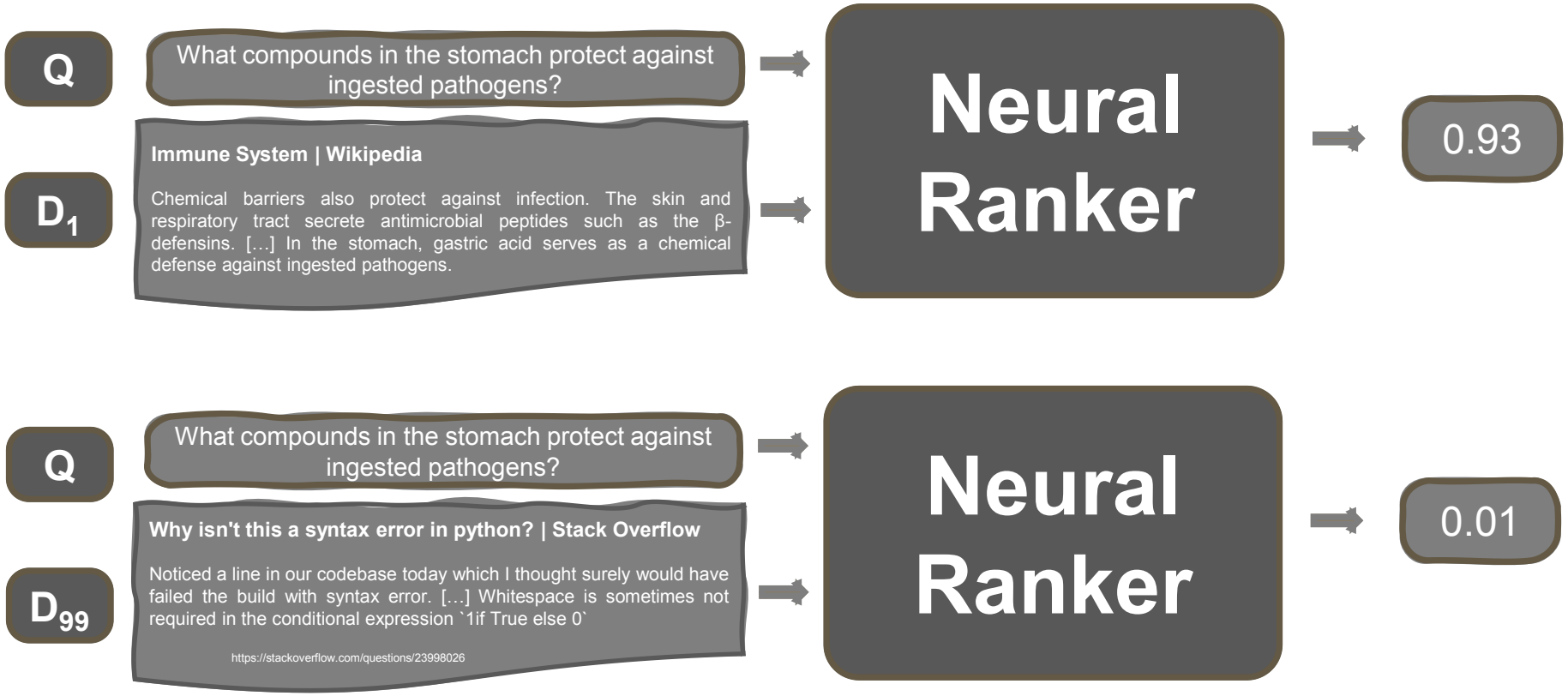
- MS MARCO: Bing Queries, 9M Passages from the Web
  - Effectiveness in **MRR@10** and Efficiency in Latency (**milliseconds**; in log-scale!)



Khattab, Omar, and Matei Zaharia. “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT." SIGIR’20.

Neural Ranking: Functional View

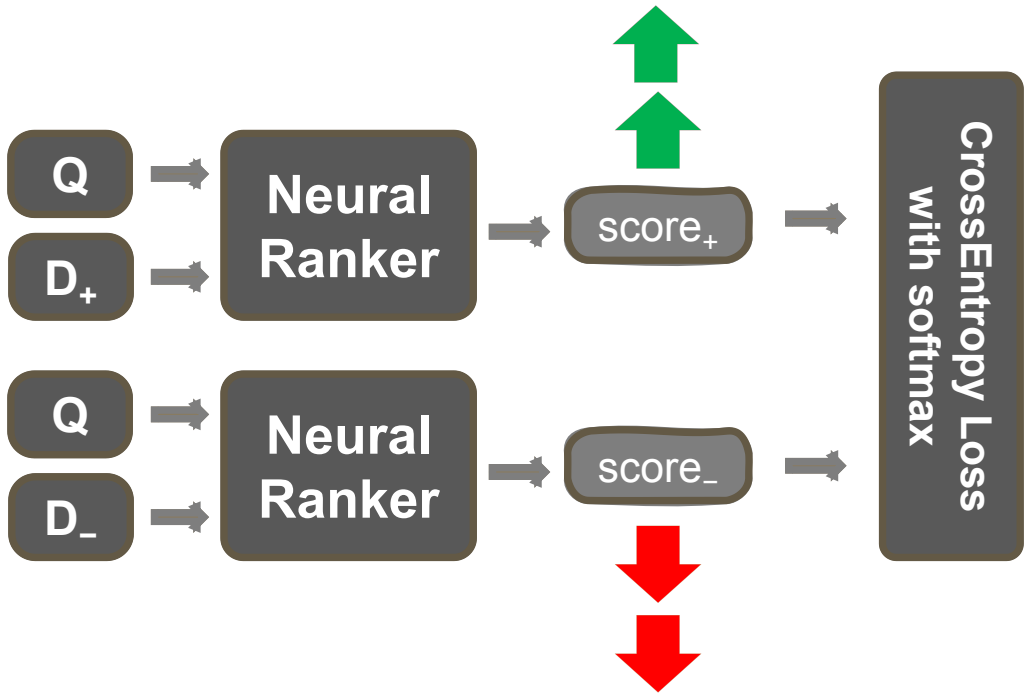
- All we need is a score for every query–document pair
  - We'll sort the results by decreasing score



Neural Ranking: Training

- Many possible choices, but **2-way classification** is often effective!
  - Each training instance is a **triple**

    < query, positive document, negative document >



Recall that we can get positives for each query from our relevance assessments.

Every non-positive can often be treated as an implicit negative.

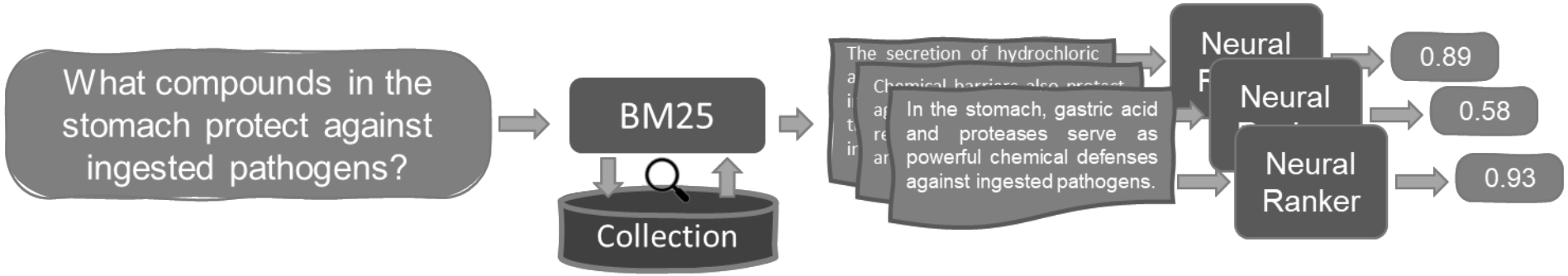
Neural Ranking: Inference

- Given a query Q, pick each document d and pass <Q, d> through the network. Sort all by score, returning the top-k results!

- But collections often have many millions of documents
  - MS MARCO has 9M passages
  - Even if you model runs in 1 microsecond per passage, that's 9 seconds per query!

Neural Re-Ranking: Pipelines

- BM25 top-1000 -> Neural IR reranker



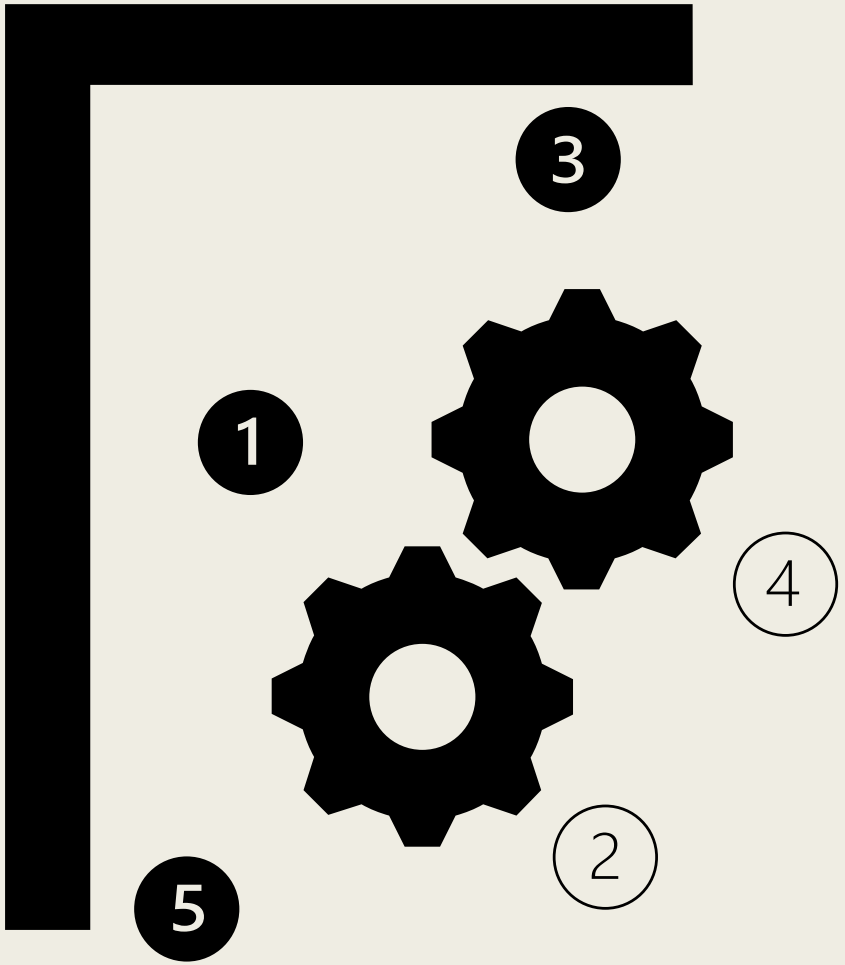
- Cuts the work on 10M documents by factor of 10k!
  - But introduces an artificial recall ceiling.

Can we do better?

Yes! Later, we'll discuss **end-to-end retrieval**.

References

Khattab, Omar, and Matei Zaharia. “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT." Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.



NLU & IR: NEURAL IR (II)

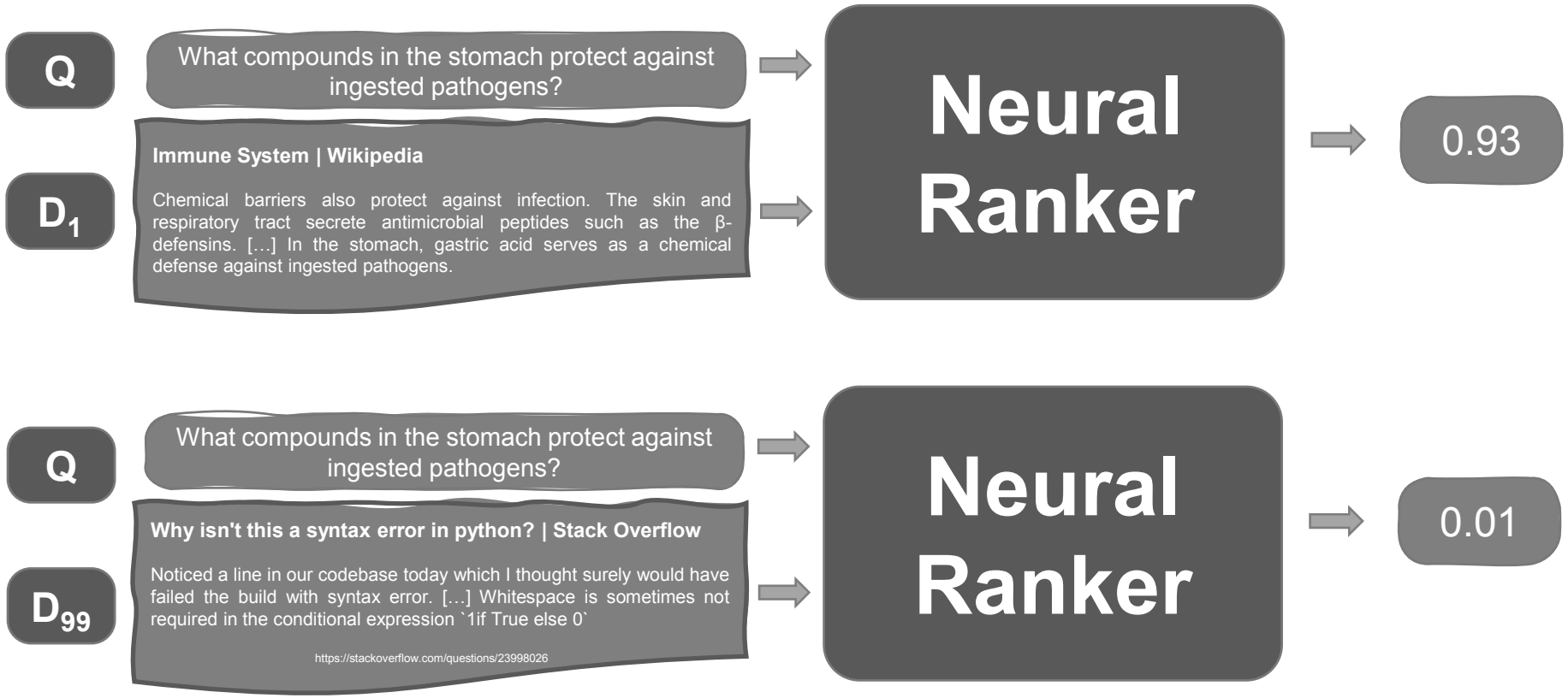
Omar Khattab

CS224U: Natural Language Understanding

Spring 2021

Neural Ranking: Functional View

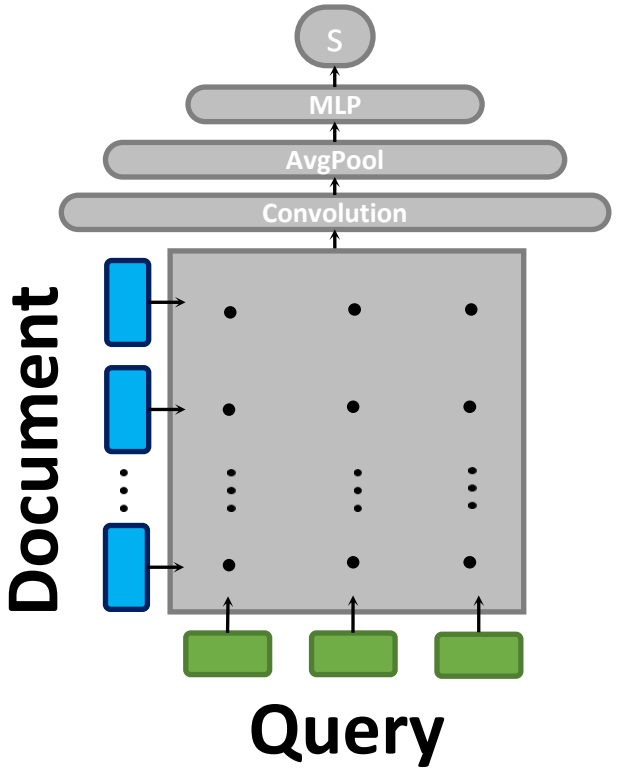
- All we need is a score for every query–document pair
  - We'll sort the results by decreasing score



Query–Document Interaction Models

1. Tokenize the query and the document
2. Embed all the tokens of each
3. Build a query–document interaction matrix
   - Most commonly: store the cos similarity of each pair of words
4. Reduce this dense matrix to a score
   - Learn neural layers (e.g., convolution, linear layers)

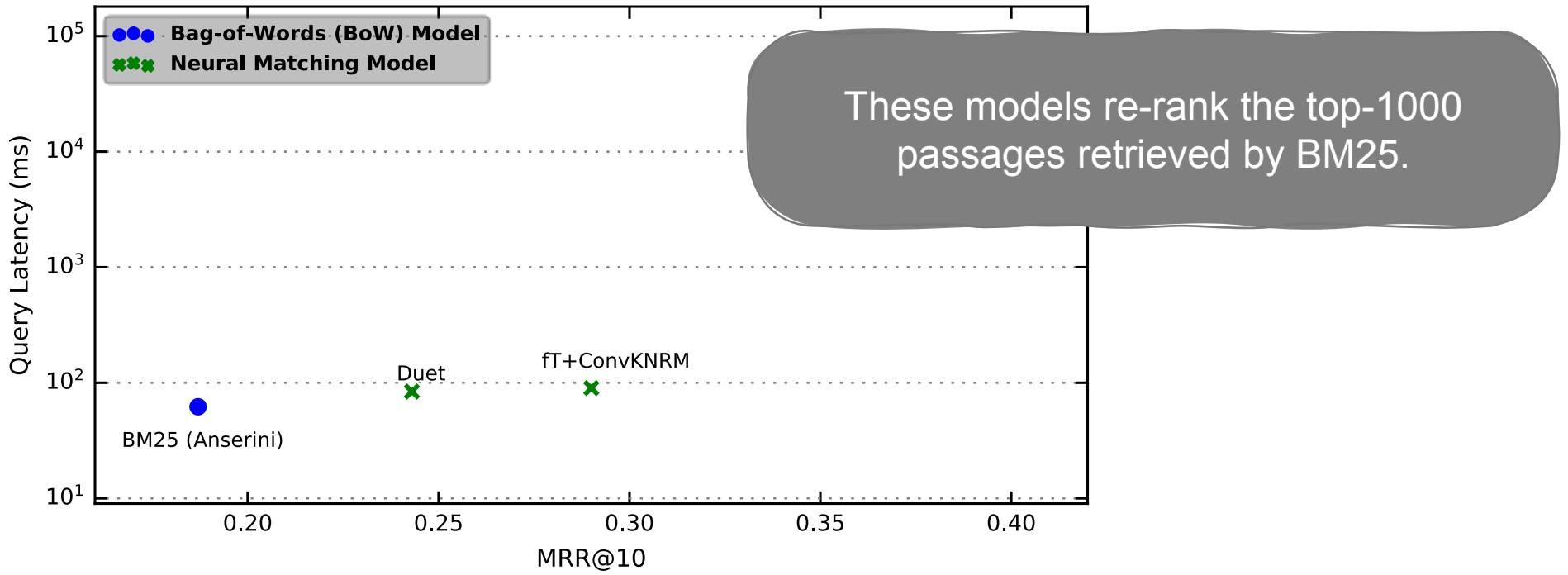
Models in this category include KNRM, Conv-KNRM, and Duet.



Chenyan Xiong, et al. End-to-end neural ad-hoc ranking with kernel pooling. SIGIR'17
Zhuyun Dai, et al. Convolutional neural networks for soft-matching n-grams in ad-hoc search. WSDM'18
Bhaskar Mitra, et al. Learning to match using local and distributed representations of text for web search. WWW'17

Query–Document Interaction Models: MS MARCO Results

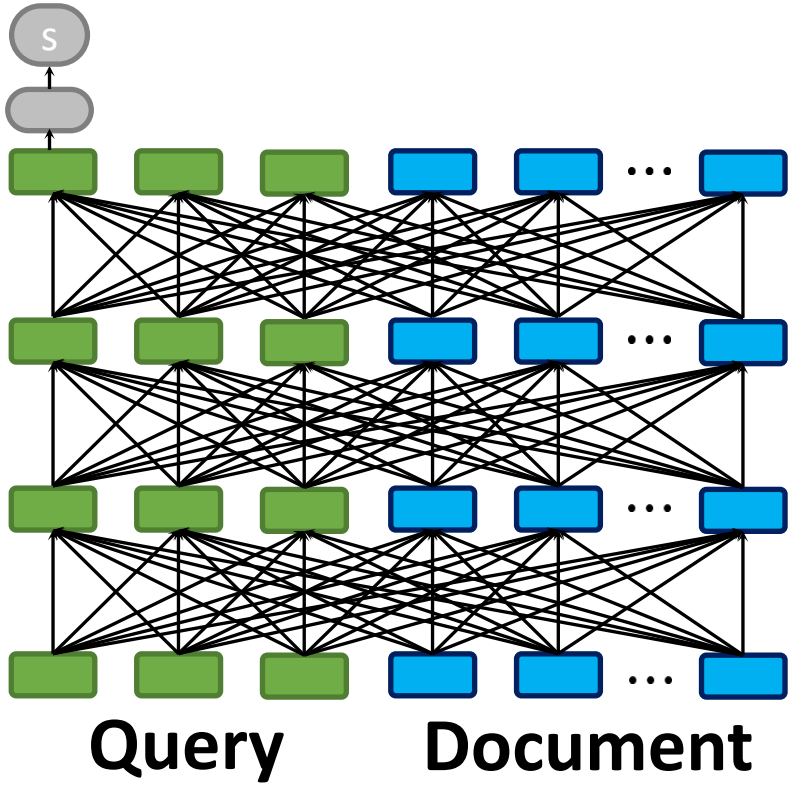
- Considerable gains in **quality**—at a reasonable increase in computational cost!



Bhaskar Mitra and Nick Craswell. An Updated Duet Model for Passage Re-ranking. arXiv:1903.07666 (2019)
Sebastian Hofstätter, et al. On the effect of low-frequency terms on neural-IR models. SIGIR'19

All-to-all Interaction with BERT

1. Feed BERT "[CLS] Query [SEP] Document [SEP]"
2. Run this through all the BERT layers
3. Extract the final [CLS] output embedding
   - Reduce to a single score through a linear layer



This is essentially a standard BERT classifier, used for ranking passages.

Of course, we must fine-tune BERT for this task with positives and negatives to be effective.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. arXiv:1901.04085 (2019)
Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. SIGIR'19

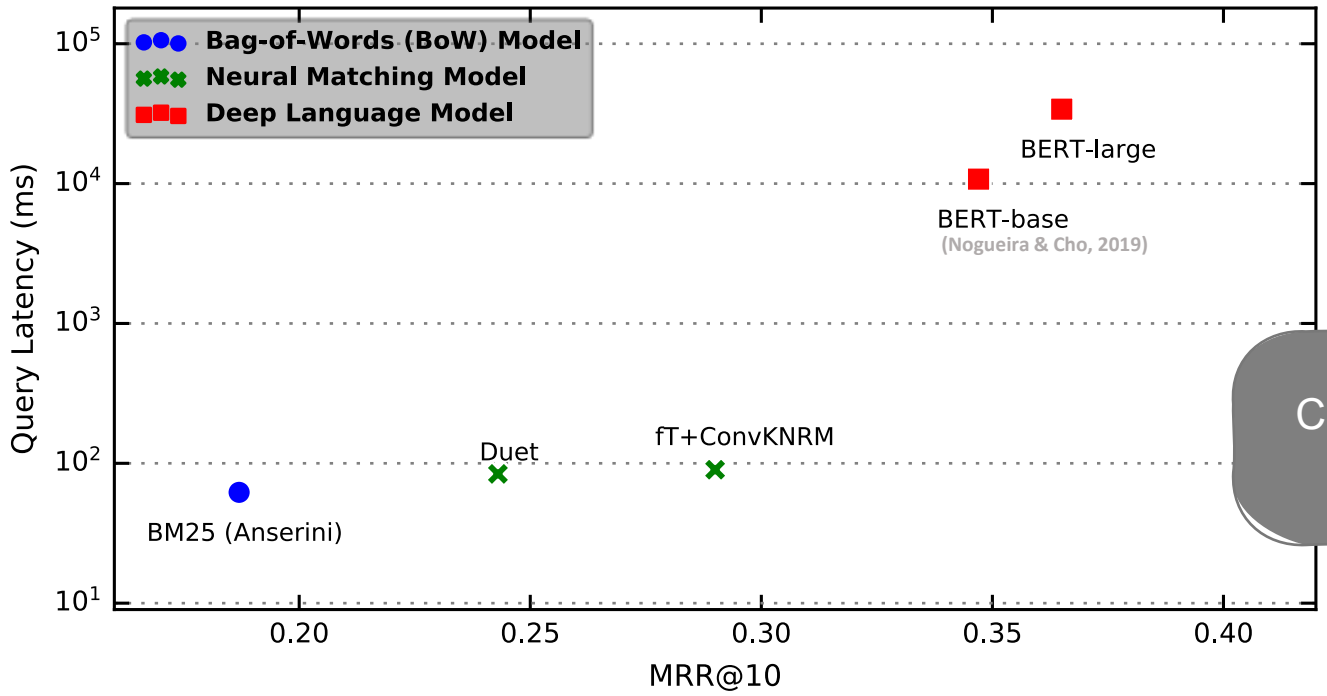
BERT Rankers: SOTA 2019 (in quality)

Rank	Model	Submission Date	MRR@10 On Eval
1	**BERT + Small Training** Rodrigo Nogueira and Kyunghyun Cho - New York University	January 7th, 2019	35.87
2	**IRNet (Deep CNN/IR Hybrid Network)** Dave DeBarr, Navendu Jain, Robert Sim, Justin Wang, Nirupama Chandrasekaran – Microsoft	January 2nd, 2019	28.061

MS MARCO Ranking screenshot as of Jan 2019. From Rodrigo Nogueira's Brief History of DL applied to IR (UoG talk).
https://blog.google/products/search/search-language-understanding-bert/
https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvementin-search-experience-using-azure-gpus/

BERT Rankers: Efficiency–Effectiveness Tradeoff

- Dramatic gains in **quality**—but also a dramatic increase in **computational cost**!



Can we achieve high MRR ***and*** low latency?

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. arXiv:1901.04085 (2019)

Toward Faster Ranking: **Pre-computation**

- BERT rankers are slow because their computations be **redundant**:
  - **Represent the query** (1000 times for 1000 documents)
  - **Represent the document** (once for every query!)
  - Conduct matching between the query and the document

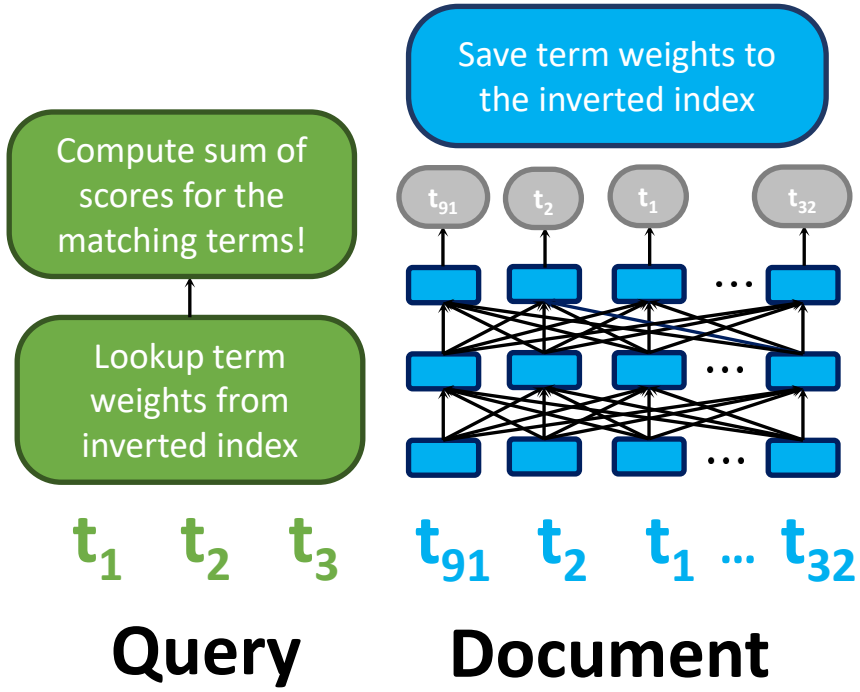
- We have the documents in advance.
  - Can we **pre-compute** the document representations?
  - And "cache" these representations for use across queries

Is there a unique value in **jointly** representing queries and documents?

Neural IR Paradigms: **Learning term weights**

- BM25 decomposed a document's score into a summation over term–document weights. **Can we learn term weights with BERT?**

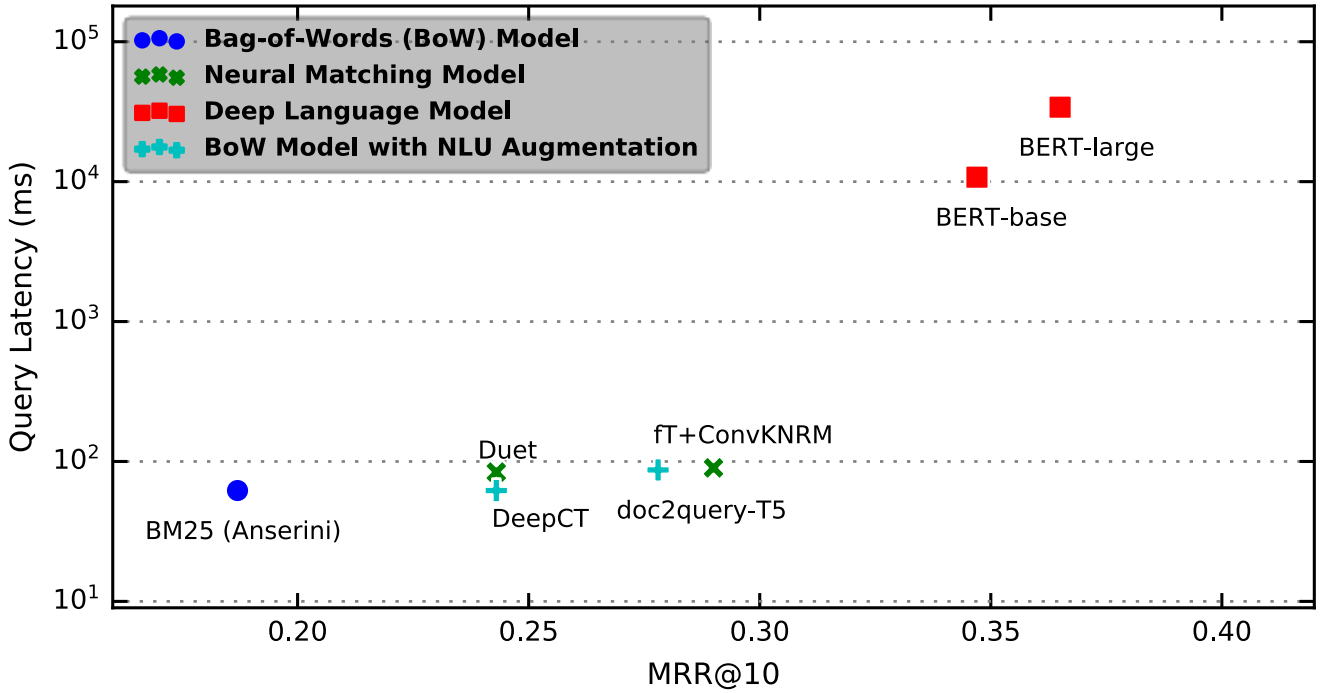
- Tokenize the query/document
- Use BERT to produce a score for each token in the document
- Add the scores of the tokens that also appear in the query



Dai, Zhuyun, and Jamie Callan. "Context-aware term weighting for first stage passage retrieval." SIGIR'20
Nogueira, Rodrigo and Jimmy Lin. "From doc2query to docTTTTTquery." Online preprint (2019).
Mallia, Antonio, et al. "Learning Passage Impacts for Inverted Indexes." SIGIR'21.

Learning term weights

- We get to learn the term weights with BERT and to **re-use** them!
- But our query is back to being a “bag of words”.



DeepCT and **doc2query** are two major models under this paradigm.

Can we do better?

Next: Can we achieve high MRR *and* low latency?

- Yes! We'll discuss two rich neural IR paradigms:
  - **Representation Similarity**
  - **Late Interaction**

References

Omar Khattab and Matei Zaharia. “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT.“ SIGIR’20

Chenyan Xiong, et al. End-to-end neural ad-hoc ranking with kernel pooling. SIGIR’17

Zhuyun Dai, et al. Convolutional neural networks for soft-matching n-grams in ad-hoc search. WSDM’18

Bhaskar Mitra, et al. Learning to match using local and distributed representations of text for web search. WWW’17

Bhaskar Mitra and Nick Craswell. An Updated Duet Model for Passage Re-ranking. arXiv:1903.07666 (2019)

Sebastian Hofstätter, et al. On the effect of low-frequency terms on neural-IR models. SIGIR’19

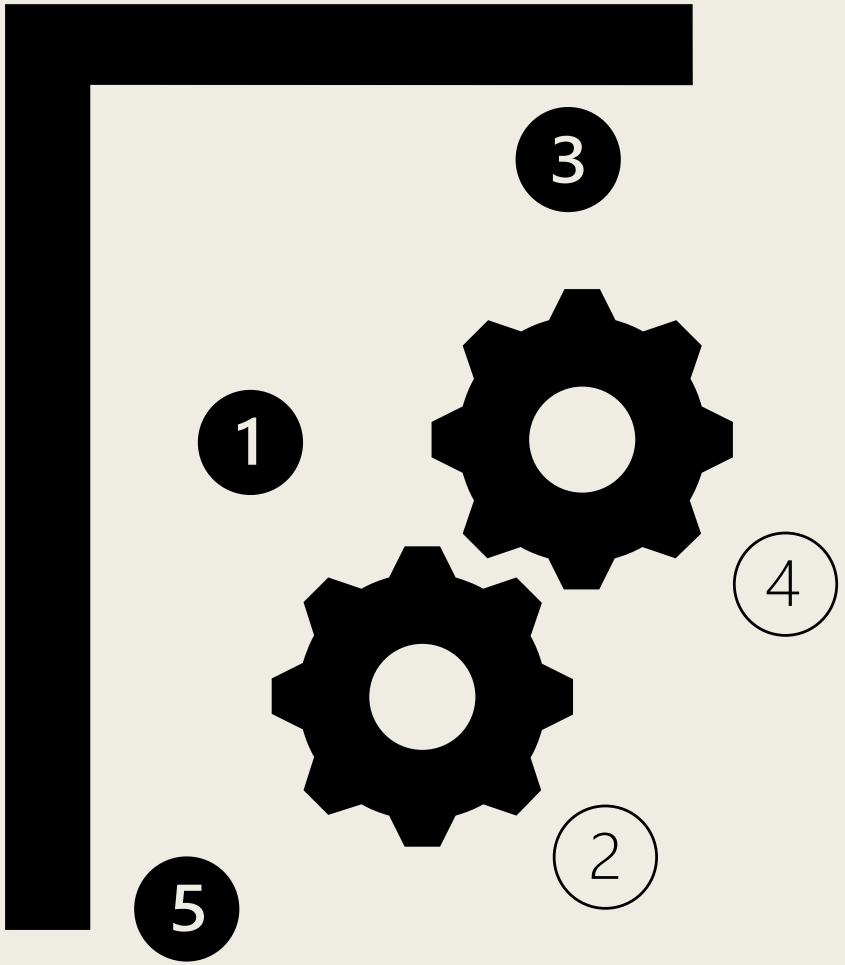
Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. SIGIR’19

Rodrigo Nogueira. “A Brief History of Deep Learning applied to Information Retrieval” (UoG talk). Retrieved from https://docs.google.com/presentation/d/1_mlvmyev0pjdG0OcfbEWManRREC0jCdjD3b1tPPvcbk

Zhuyun Dai, and Jamie Callan. "Context-aware term weighting for first stage passage retrieval.” SIGIR’20

Rodrigo Nogueira and Jimmy Lin. "From doc2query to docTTTTTquery." Online preprint (2019).

Antonio Mallia, et al. "Learning Passage Impacts for Inverted Indexes.“ SIGIR’21.



NLU & IR: NEURAL IR (III)

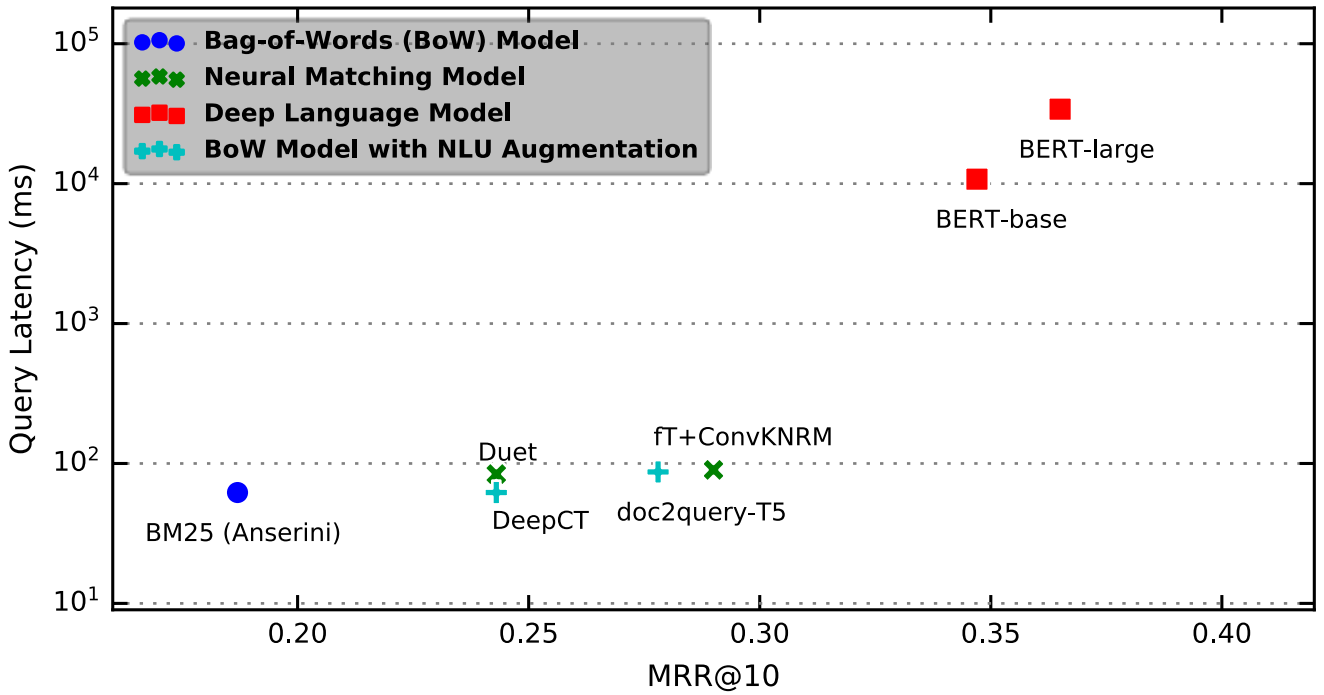
Omar Khattab

CS224U: Natural Language Understanding

Spring 2021

Learning term weights: DeepCT and doc2query

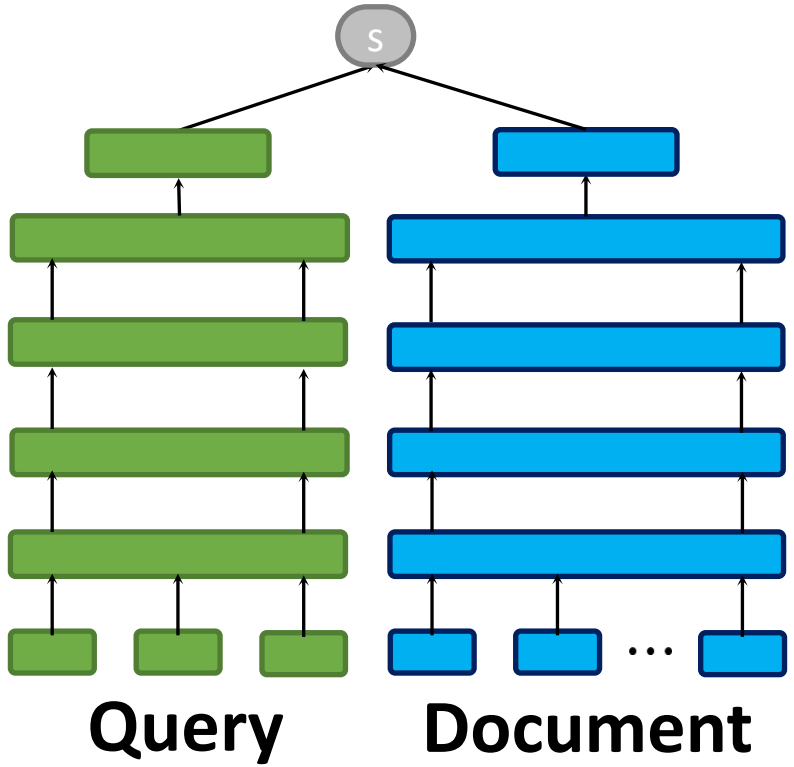
- We get to learn the term weights with BERT and to **re-use** them!
- But our query is back to being a “bag of words”.



Can we do better?

Neural IR Paradigms: **Representation Similarity**

- Tokenize the query and the document
- **Independently** encode the query and the document
- … into a **single-vector** representation each
- Estimate relevance a dot product
  - Or a cosine similarity



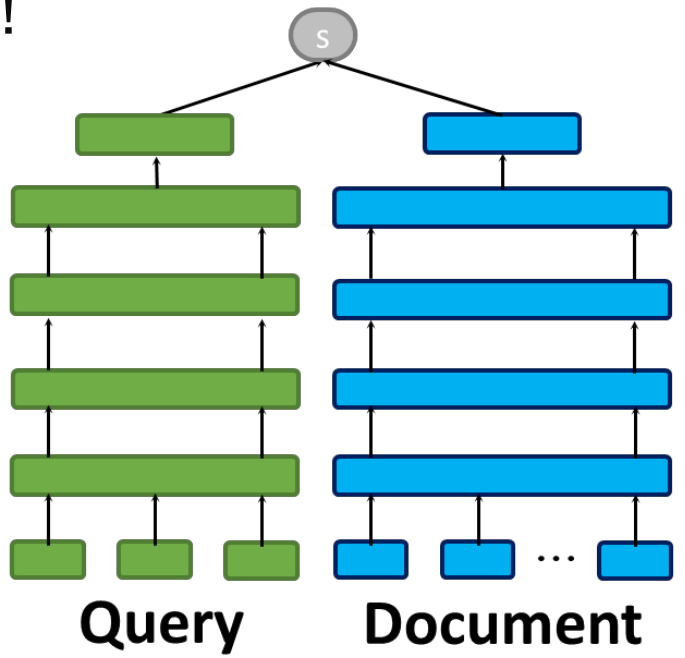
Like learning term weights, this paradigm offers strong **efficiency** advantages:

- ✓ Document representations can be pre-computed!
- ✓ Query computations can be amortized.
- ✓ Similarity computations are very cheap.

Representation Similarity: Models

- Many pre-BERT IR models fall under this paradigm!
  - DSSM and SNRM

- Numerous BERT-based models exist
  - SBERT, ORQA, **DPR**, DE-BERT, RepBERT, ANCE



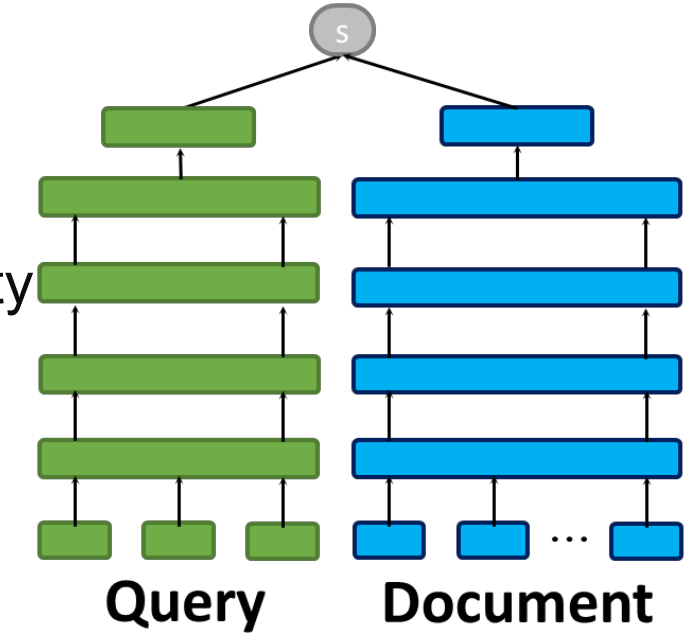
Many of these BERT-based representation similarity models are *concurrent* to one another (late 2019 / early 2020).

The largest differences are in the **specific tasks** each targets and the **supervision approach**.

Representation Similarity: DPR

Dense Passage Retriever (DPR) by *Karpukhin et al.*

- Encodes each passage into a 768-dimensional vector
- Encodes each query into a 768-dimensional vector



- Trained with N-way cross-entropy loss, over the similarity scores between the query and:
  - A positive passage
  - A negative passage, sampled from BM25 top-100
  - Many in-batch negative passages
    - the positive passages for the *other* queries in the same training batch

Vladimir Karpukhin, et al. "Dense passage retrieval for open-domain question answering." EMNLP'20

Representation Similarity: DPR

Dense Passage Retriever (DPR) by *Karpukhin et al.*

- Encodes each passage into a 768-dimensional vector
- Encodes each query into a 768-dimensional vector
-
  - Many in-batch negative passages
    - the positive passages for the *other* queries in the same training batch

Xiong et al. (2020) test a DPR-style retriever on MS MARCO: **31% MRR**. They show that a sophisticated supervision scheme can achieve **33%**.

Both constitute progress over “learned term weights” like DeepCT, but they are still considerably lower than standard BERT’s **>36% MRR**.

Vladimir Karpukhin, et al. "Dense passage retrieval for open-domain question answering." EMNLP’20
Lee Xiong, et al. "Approximate nearest neighbor negative contrastive learning for dense text retrieval." ICLR’21

Representation Similarity: Downsides

✕ **Single-Vector Representations**

- They "cram" queries and documents into a **coarse-grained** representation!

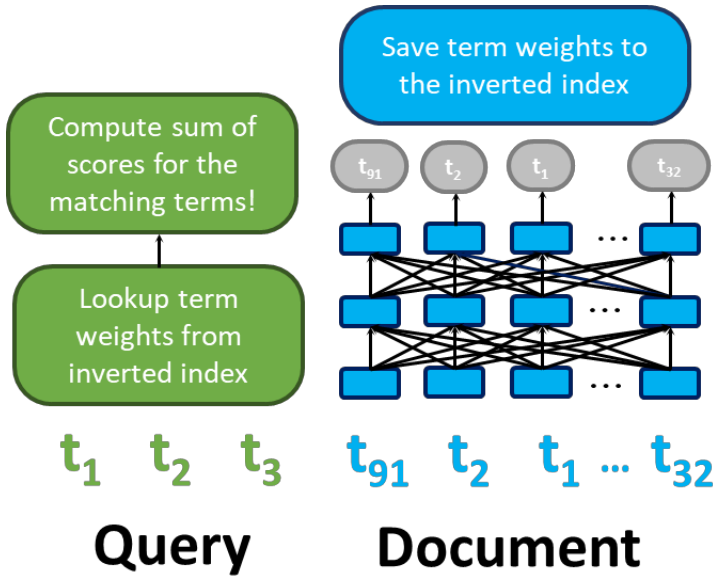
✕ **No Fine-Grained Interactions**

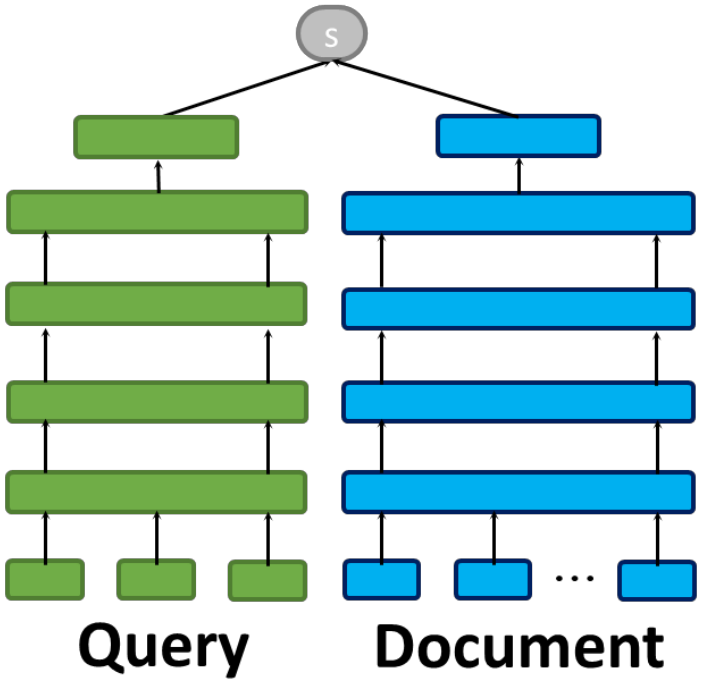
- They estimate relevance as **single dot product**!
- We lose **term-level interactions**, which we had in:
  - Query–Document interaction models (e.g., BERT or Duet)
  - And even term-weighting models (e.g., DeepCT and BM25)

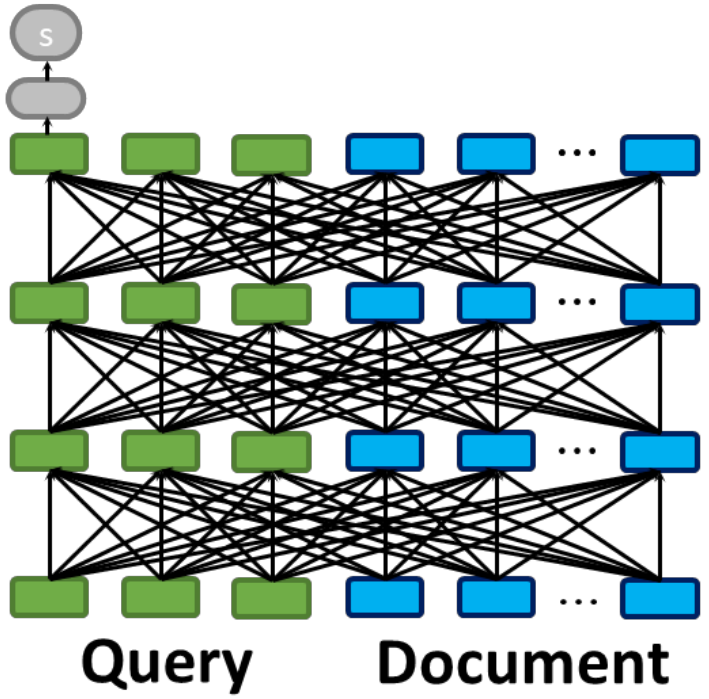
Can we keep precomputation and still have fine-grained interactions?

Omar Khattab and Matei Zaharia. "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT." SIGIR'20.

Summary: Neural Ranking Paradigms







(a) Learned Term Weights

✔ Independent Encoding

✘ Bag-of-Words Matching

(b) Representation Similarity

✔ Independent, Dense Encoding

✘ Coarse-Grained Representation

(c) Query–Document Interaction

✔ Fine-Grained Interactions

✘ Expensive Joint Conditioning

Beyond Re-ranking: **End-to-end Retrieval**

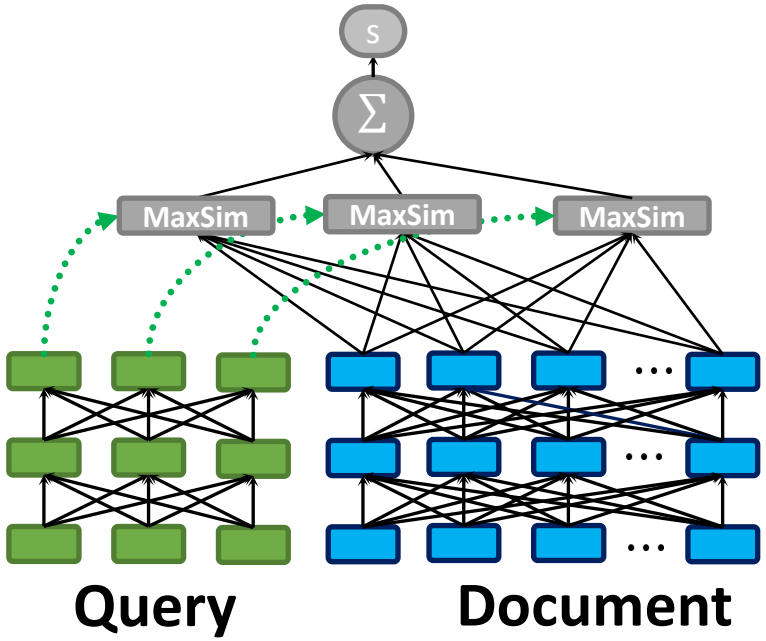
- **Query–Document Interaction** models forced us to use a re-ranking pipeline, where we just re-scored the top-1000 documents retrieved by BM25.

End-to-end retrieval is essential toward improving RECALL.

- **Learning Term Weights** and **Representation Similarity** models alleviate this!
  - They allow us to do end-to-end retrieval: quickly searching over all documents directly.
  - We can save **term weights** in the **inverted index**. This means that we do NOT need a re-ranking pipeline.
  - We can also index **vector representations** for **fast vector-similarity search**, which allows **PRUNING** to find the top-K matches without exhaustive enumeration.
    - Libraries like **FAISS** abstract away the details.

https://github.com/facebookresearch/faiss/

Neural IR Paradigms: **Late Interaction**



(d) Late Interaction

(i.e., ColBERT)

Can we keep precomputation and still have fine-grained interactions?

Desired Properties:

- ✔ Independent Encoding
- ✔ Fine-Grained Representations
- ✔ End-to-End Retrieval (pruning!)

Omar Khattab and Matei Zaharia. “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT." SIGIR’20.

Late Interaction: Real Example of Matching

when did the transformers cartoon series come out?

[…] the animated […] The Transformers […] […] It was released […] on August 8, 1986

when did the transformers cartoon series come out?

[…] the animated […] The Transformers […] […] It was released […] on August 8, 1986

when did the transformers cartoon series come out?

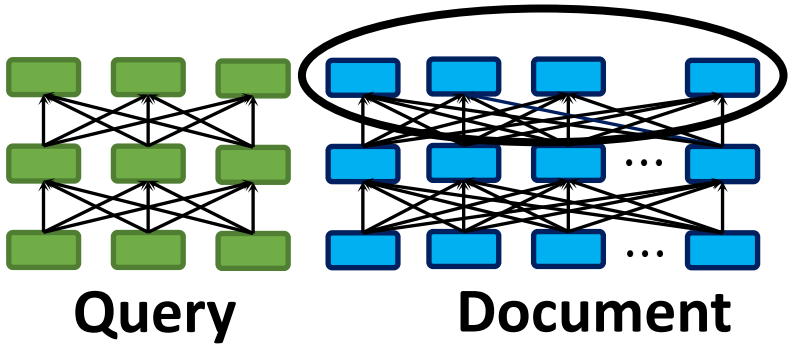
[…] the animated […] The Transformers […] […] It was released […] on August 8, 1986

when did the transformers cartoon series come out?

[…] the animated […] The Transformers […] […] It was released […] on August 8, 1986

Late Interaction: ColBERT

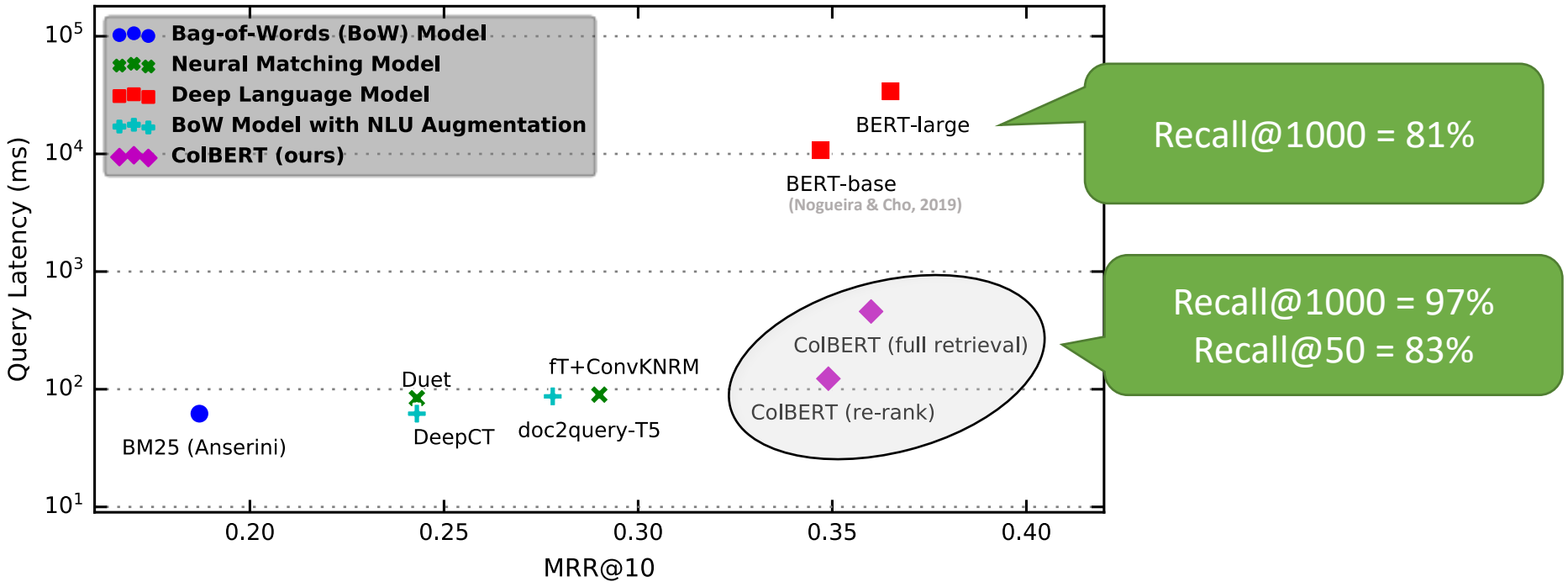
Notice that ColBERT represents the document as a MATRIX, not a vector.



(d) Late Interaction

(i.e., ColBERT)

Late Interaction: ColBERT



Omar Khattab and Matei Zaharia. “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT." SIGIR’20.

Robustness: Out-of-Domain Quality

- So far, we've looked at in-domain effectiveness evaluations.
  - We had training and evaluation data for MS MARCO.

- We often want to use retrieval in new, out-of-domain settings.
  - … with NO training data and NO validation data.
  - This is sometimes called a "zero-shot" setting; it emphasizes transfer.

- BEIR is a recent benchmark for IR models in "zero-shot" scenarios

Thakur, Nandan, et al. "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models." *arXiv:2104.08663* (2021)

Robustness: Out-of-Domain **NDCG@10**

- **Fine-grained interaction** is key to robustly high precision

IR Task	Classical IR BM25	Interaction Models ELECTRA re-ranker	Representation Similarity DPR	Representation Similarity SBERT	Late Interaction ColBERT
BioMed	48	**49**	22	34	**49**
QA	38	**51**	33	41	48
Tweet	**39**	31	16	26	27
News	37	**43**	16	37	39
Arguments	**52**	35	15	34	25
Duplicates	53	56	20	58	**60**
Entity	29	38	26	34	**39**
Citation	**16**	15	8	13	15
Fact-Check	48	52	34	47	**54**
Overall Avg	42	**45**	23	39	44

Table aggregated from the BEIR results (Table 2) by Thakur, Nandan, et al. "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models." *arXiv:2104.08663* (2021)

Robustness: Out-of-Domain **Recall@100**

- **Scalable** fine-grained interaction is key to robustly high recall

IR Task	Classical IR BM25	Interaction Models ELECTRA re-ranker	Representation Similarity DPR	Representation Similarity SBERT	Late Interaction ColBERT
BioMed	**45**	**45**	23	35	**45**
QA	67	67	60	68	**75**
Tweet	**38**	**38**	16	26	28
News	**40**	**40**	22	37	37
Arguments	**70**	**70**	46	62	61
Duplicates	77	77	44	79	**81**
Entity	38	38	35	40	**46**
Citation	**35**	**35**	22	30	34
Fact-Check	71	71	65	74	**75**
Overall Avg	59	59	43	57	**61**

Table aggregated from the BEIR results (Table 4) by Thakur, Nandan, et al. "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models." *arXiv:2104.08663* (2021)

Final Thoughts on Neural IR

- Speed vs. **Scalability**: not always the same!
  - Inductive biases are crucial to **effective** models that **scale**.

- Next…
  - **Can scalability drive new gains in quality?**
    - YES! We will see examples of this in the Open-QA screencast.
  - **How can we tune a neural IR model for open-domain NLU tasks?**

References

Vladimir Karpukhin, et al. "Dense passage retrieval for open-domain question answering." EMNLP’20

Lee Xiong, et al. "Approximate nearest neighbor negative contrastive learning for dense text retrieval." ICLR’21

Omar Khattab and Matei Zaharia. “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT.“ SIGIR’20

Nandan, et al. "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models.“ arXiv:2104.08663 (2021)