Information retrieval

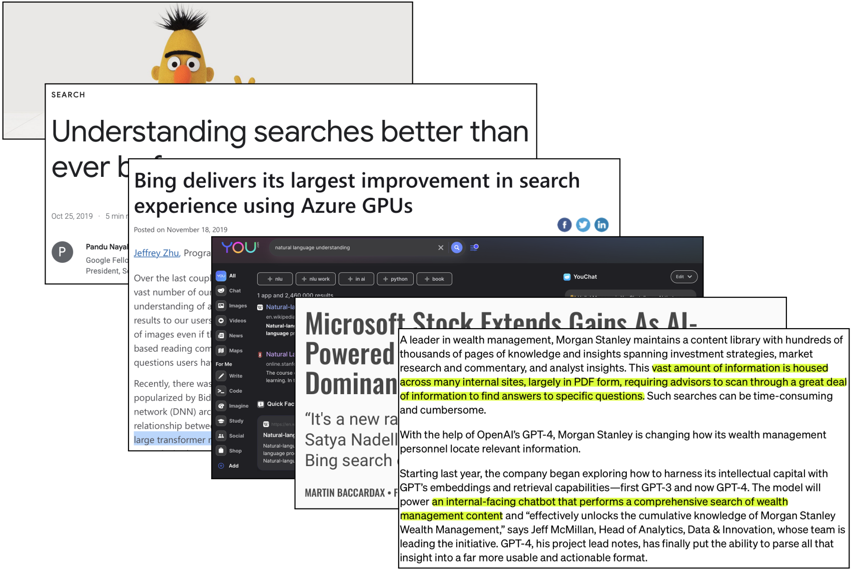
Christopher Potts and Omar Khattab

Stanford Linguistics

CS224u: Natural language understanding

Guiding ideas

NLP is revolutionizing Information Retrieval (IR)



IR is a hard NLU problem

what compounds protect the digestive system against viruses

In the **stomach**, gastric acid and proteases serve as powerful **chemical defenses** against ingested **pathogens**.

IR is revolutionizing NLP

Standard QA

```
Title: Bert

Context: Bert is a
Muppet who lives with
Ernie.

Q: Who is Bert?

A: Bert is a Muppet
```

Title, Context, Question, and Answer given at train time. Title, Context, Question given a test time.

OpenQA

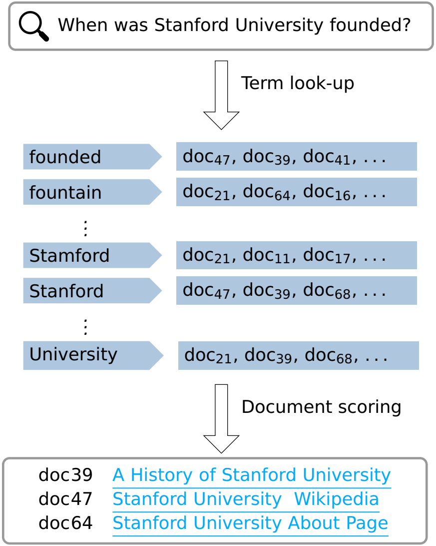
```
Title: Sesame Street

Context: Bert and
Ernie are Muppets who
live together.

Q: Who is Bert?

A: Bert is a Muppet
```

Only Question and Answer given at train time. Only Question given at test time. **Title/Context retrieved.**

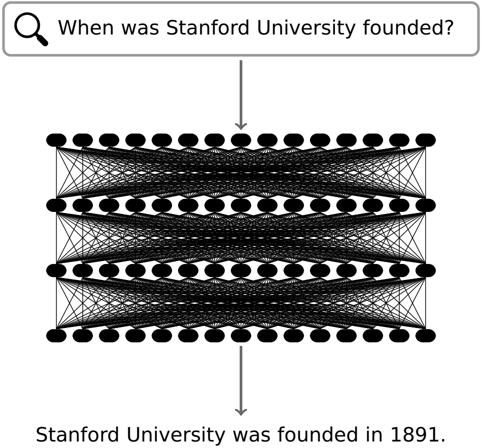
Knowledge-intensive tasks

1. Question answering
2. Claim verification
3. Commonsense reasoning
4. Long-form reading comprehension
5. Information-seeking dialogue
6. Summarization
7. Natural language inference

Classical IR

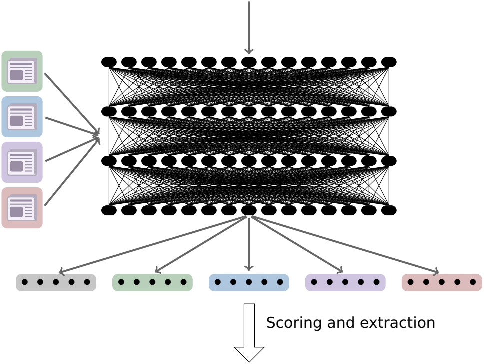


“LLMs for everything”



Neural IR

When was Stanford University founded?



"Stanford University was founded in 1885 by California senator Leland Stanford and his wife, Jane"
A History of Stanford University

"Stanford was founded in 1885 by Leland and Jane Stanford in memory of their only child, Leland Stanford Jr."
Stanford University Wikipedia

"Opened in 1891"
Stanford University About Page

Retrieval-augmented in-context learning

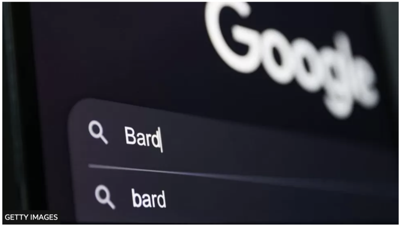
Context: Kermit is one of the stars of Sesame Street.	Train or Retrieved
Q: Who is Kermit?	Train
A: Kermit is the one of the stars of Sesame Street.	Train or Retrieved
Context: Bert is a Muppet who lives with Ernie.	Retrieved
Q: Who is Bert?	Given
A: Bert is a Muppet.	Predicted

IR is more important than ever!

IR is more important than ever!

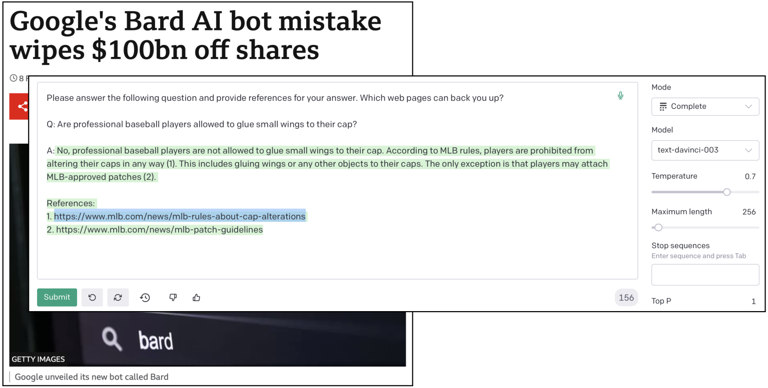
Google's Bard AI bot mistake wipes $100bn off shares

8 February

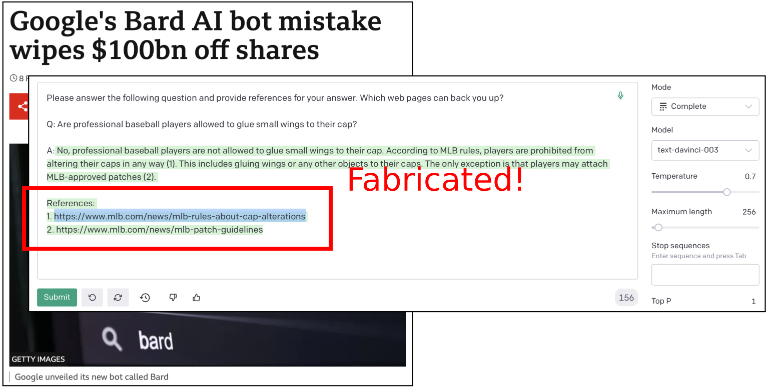


Google unveiled its new bot called Bard

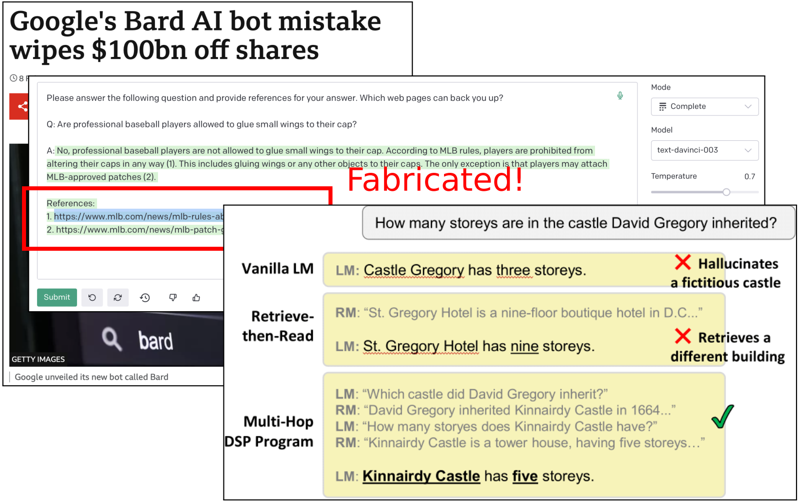
IR is more important than ever!



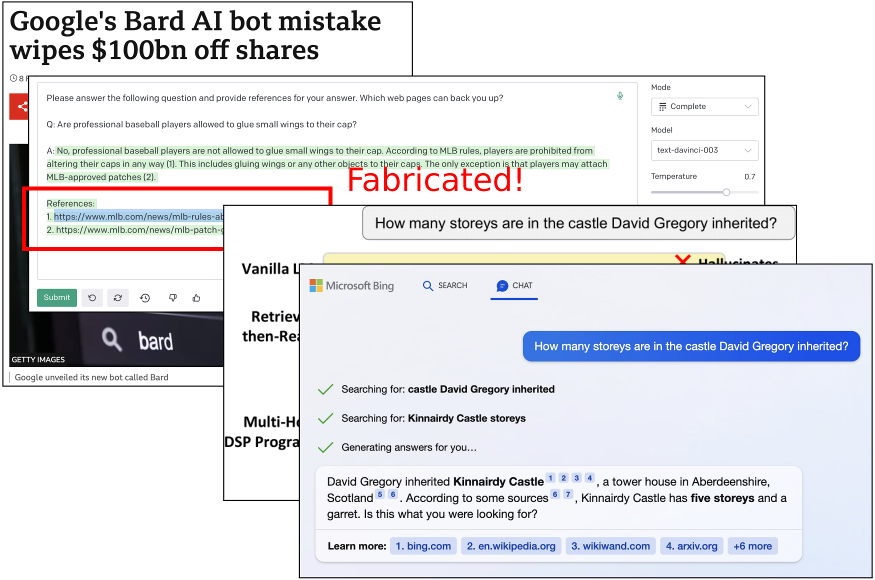
IR is more important than ever!



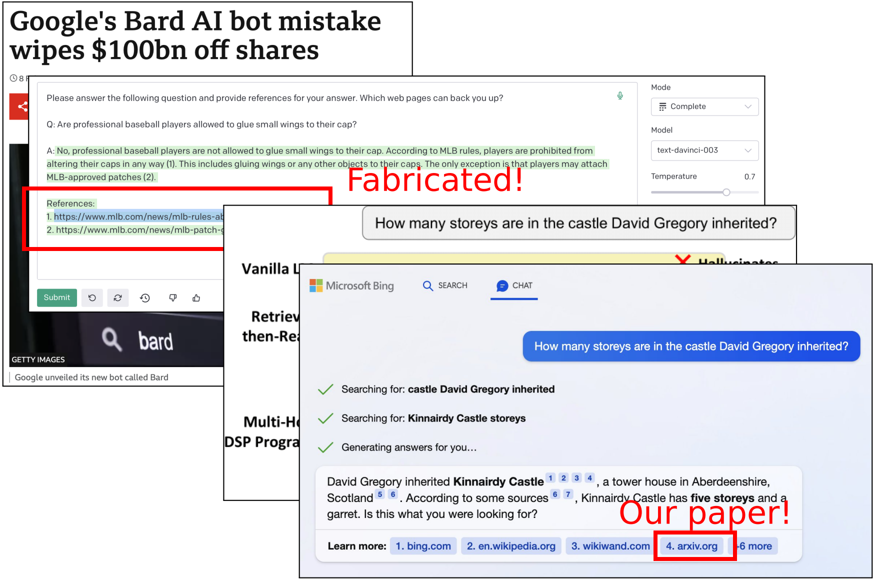
IR is more important than ever!



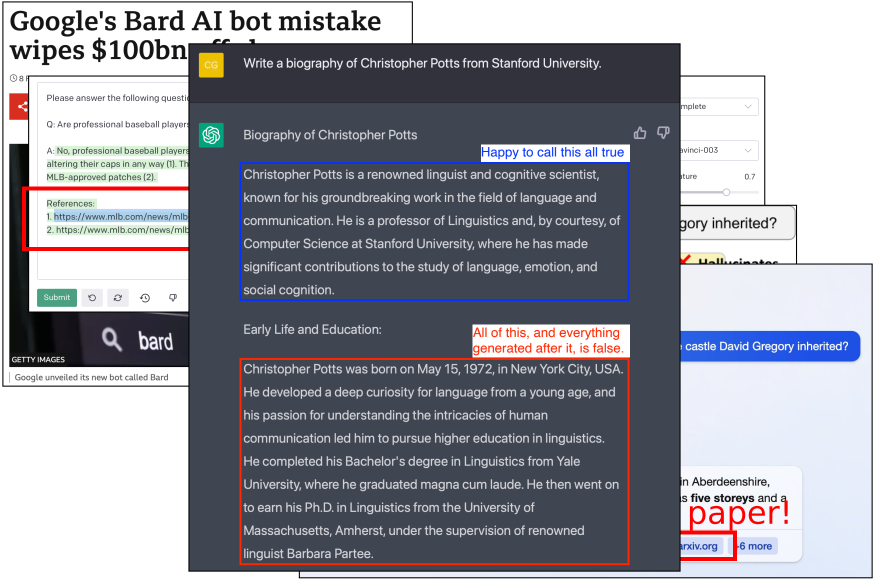
IR is more important than ever!



IR is more important than ever!



IR is more important than ever!



Blog posts

Building Scalable, Explainable, and Adaptive NLP Models with Retrieval

Omar Khattab, Christopher Potts, and Matei Zaharia

October 5, 2021

[link]

Language Processing, Machine Learning

A Moderate Proposal for Radically Better AI-powered Web Search

Large language models could give us instant answers, but at a cost to trust. Stanford scholars propose an alternative.

Jul 6, 2021 | Omar Khattab, Christopher Potts, and Matei Zaharia

[link]

Classical IR

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	
⋮											

TF-IDF

For a corpus of documents D:

- Term frequency: $\textbf{TF}(w, \text{doc}) = \frac{\textbf{count}(w,\text{doc})}{|\text{doc}|}$
- Document frequency: $\textbf{df}(w, D) = \left|\{\text{doc} \in D : w \in \text{doc}\}\right|$
- Inverse document frequency: $\text{IDF}(w, D) = \log_e\left(\frac{|D|}{\textbf{df}(w,D)}\right)$
- $\text{TF-IDF}(w, \text{doc}, D) = \text{TF}(w, \text{doc}) \cdot \text{IDF}(w, D)$

	doc_1	doc_2	doc_3	doc_4
A	10	10	10	10
B	10	10	10	0
C	10	10	0	0
D	0	0	0	1

$\Rightarrow$

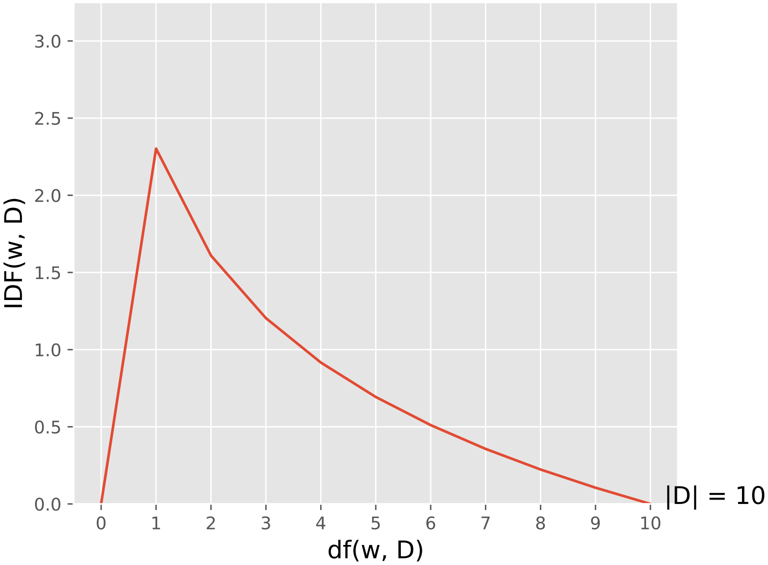
	IDF
A	0.00
B	0.29
C	0.69
D	1.39

$\Downarrow$

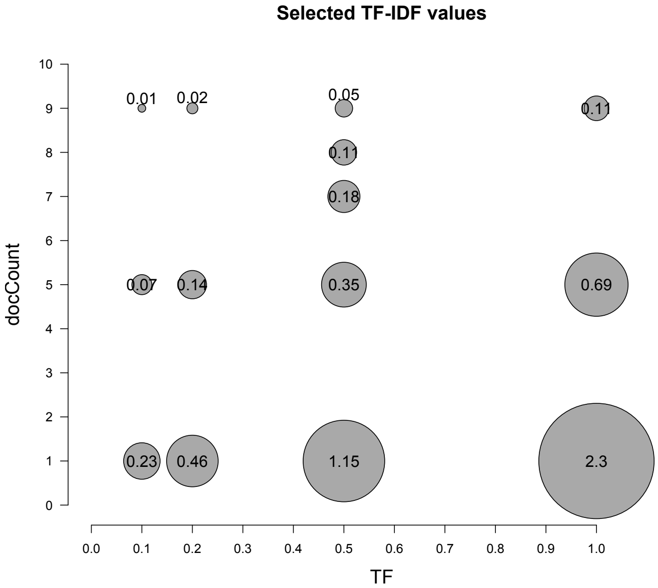
	TF			
	doc_1	doc_2	doc_3	doc_4
A	0.33	0.33	0.50	0.91
B	0.33	0.33	0.50	0.00
C	0.33	0.33	0.00	0.00
D	0.00	0.00	0.00	0.09

	TF-IDF			
	doc_1	doc_2	doc_3	doc_4
A	0.00	0.00	0.00	0.00
B	0.10	0.10	0.14	0.00
C	0.23	0.23	0.00	0.00
D	0.00	0.00	0.00	0.13

IDF values



Selected TF-IDF values



Relevance scores

$$\textbf{RelevanceScore}(q, \text{doc}, D) = \sum_{w \in q} \textbf{Weight}(w, \text{doc}, D)$$

where **Weight** is often TF-IDF.

BM25

Smoothed IDF

$$\text{IDF}_{\text{BM25}}(w, D) = \log_e\left(1 + \frac{|D| - \mathbf{df}(w, D) + 0.5}{\mathbf{df}(w, D) + 0.5}\right)$$

Scoring

With $k = 1.2$ and $b = 0.75$ (or thereabouts):

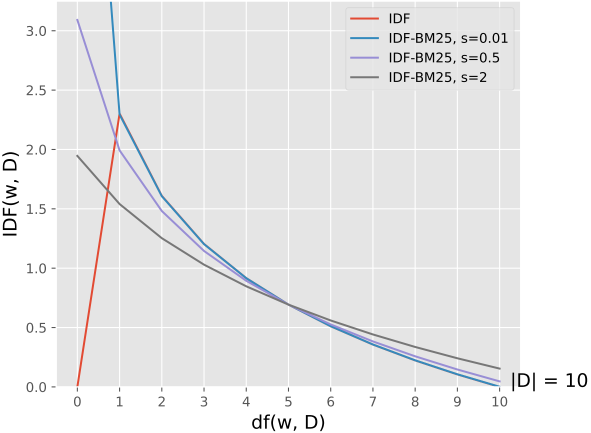
$$\text{Score}_{\text{BM25}}(w, doc) = \frac{\mathbf{TF}(w, \text{doc}) \cdot (k + 1)}{\mathbf{TF}(w, \text{doc}) + k \cdot \left(1 - b + b \cdot \frac{|\text{doc}|}{\text{avgdoclen}}\right)}$$

BM25 Weight

$$\text{BM25}(w, \text{doc}, D) = \text{Score}_{\text{BM25}}(w, \text{doc}) \cdot \text{IDF}_{\text{BM25}}(w, D)$$

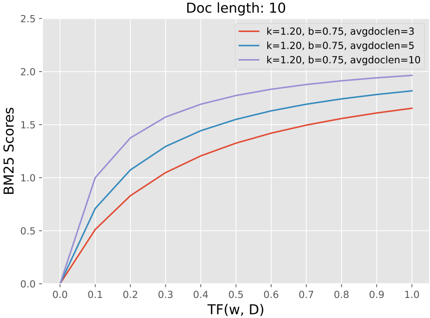
Best Match, Attempt #25; Robertson and Zaragoza 2009

BM25 IDF values



$$\text{IDF}_{\text{BM25}}(w, D) = \log_e\left(1 + \frac{|D| - \mathbf{df}(w, D) + s}{\mathbf{df}(w, D) + s}\right)$$

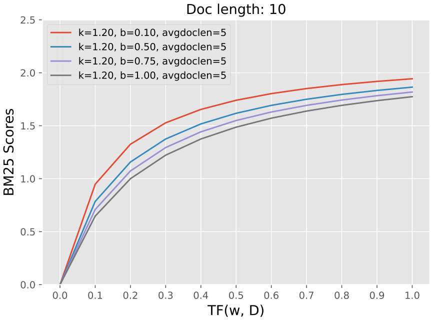
BM25 Scores: avgdoclen

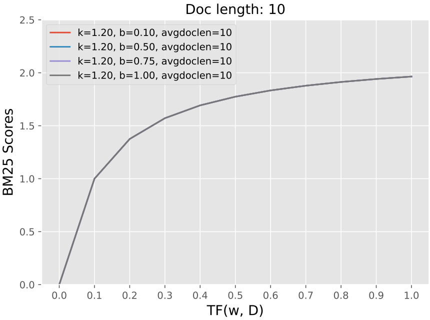


$$\text{Score}_{\text{BM25}}(w, doc) = \frac{\mathbf{TF}(w, \text{doc}) \cdot (k+1)}{\mathbf{TF}(w, \text{doc}) + k \cdot \left(1 - b + b \cdot \frac{|\text{doc}|}{\text{avgdoclen}}\right)}$$

Penalizes long documents

BM25 Scores: b

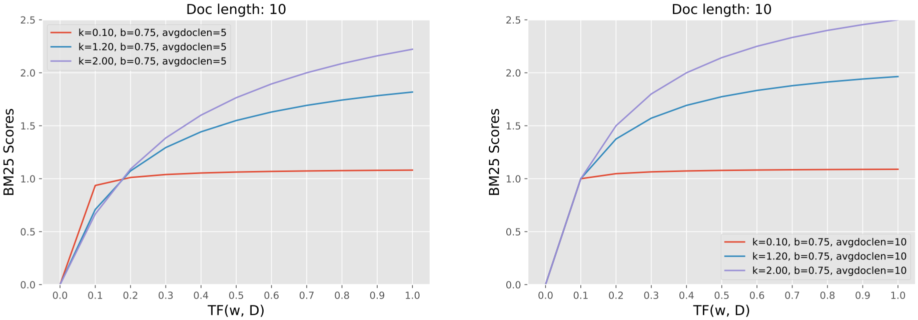




$$\text{Score}_{\text{BM25}}(w, doc) = \frac{\mathbf{TF}(w, \text{doc}) \cdot (k+1)}{\mathbf{TF}(w, \text{doc}) + k \cdot \left(1 - b + b \cdot \frac{|\text{doc}|}{\text{avgdoclen}}\right)}$$

b controls the doc length penalty

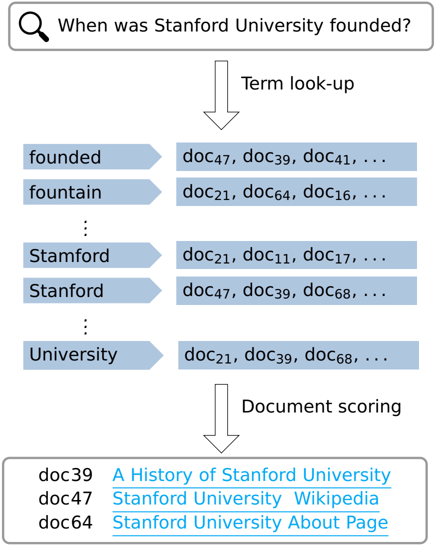
BM25 Scores: *k*



$$\text{Score}_{\text{BM25}}(w, doc) = \frac{\textbf{TF}(w, \text{doc}) \cdot (k+1)}{\textbf{TF}(w, \text{doc}) + k \cdot \left(1 - b + b \cdot \frac{|\text{doc}|}{\text{avgdoclen}}\right)}$$

Flattens out higher frequencies

Inverted indices



Inverted indices

When was Stanford University founded?

↓ Term look-up

Term	Postings	
founded	(doc_{47}, 0.90), (doc_{39}, 0.76), (doc_{41}, 0.76), . . .	0.12
fountain	(doc_{21}, 0.65), (doc_{64}, 0.60), (doc_{16}, 0.10), . . .	0.88
⋮		
Stamford	(doc_{21}, 0.91), (doc_{11}, 0.89), (doc_{17}, 0.50), . . .	0.01
Stanford	(doc_{47}, 0.29), (doc_{39}, 0.01), (doc_{68}, 0.10), . . .	0.56
⋮		
University	(doc_{21}, 0.91), (doc_{39}, 0.90), (doc_{68}, 0.76), . . .	0.01

↓ Document scoring

doc39 A History of Stanford University
doc47 Stanford University Wikipedia
doc64 Stanford University About Page

Beyond term matching

1. Query and document expansion
2. Phrase search
3. Term dependence
4. Different document fields (e.g., title, body)
5. Link analysis (e.g., PageRank)
6. Learning to rank

Tools for classical IR

1. Elasticsearch
   https://www.elastic.co
2. Pyserini:
   https://github.com/castorini/pyserini
3. PrimeQA
   https://github.com/primeqa/primeqa

IR metrics

Many dimensions

1. Accuracy-style metrics: These will be our focus.
2. Latency: Time to execute a single query.
3. Throughput: Total queries served in a fixed time, perhaps via batch processing.
4. FLOPs: Hardware agnostic measure of compute resources.
5. Disk usage: For the model, index, etc.
6. Memory usage: For the model, index, etc.
7. Cost: Total cost of deployment for a system.

Relevance data types

Given a query q and a collection of N documents D:

1. A complete partial gold ranking $\mathbf{D} = [\text{doc}_1, \ldots, \text{doc}_N]$ of D with respect to q.
   - Unlikely unless **D** was automatically generated.
2. An incomplete partial ranking of D with respect to q.
3. Labels for which passages in D are relevant to q.
   - Could be based in a weak supervision heuristic like whether each doc_i contains q as a substring.
4. A tuple consisting of one positive document doc^+ for q and one or more negatives doc^- for q.

Success and Reciprocal Rank

Rank

For a ranking $\mathbf{D} = [\text{doc}_1, \ldots, \text{doc}_N]$, let

$$\textbf{Rank}(q, \mathbf{D}) \in \{1, 2, 3, \ldots\}$$

be the position of the **first** relevant document for q in $\mathbf{D}$.

Success

$$\text{Success@K}(q, \mathbf{D}) = \begin{cases} 1 & \text{if } \textbf{Rank}(q, \mathbf{D}) \leqslant K \\ 0 & \text{otherwise} \end{cases}$$

Reciprocal Rank

$$\text{RR@K}(q, \mathbf{D}) = \begin{cases} \frac{1}{\textbf{Rank}(q, \mathbf{D})} & \text{if } \textbf{Rank}(q, \mathbf{D}) \leqslant K \\ 0 & \text{otherwise} \end{cases}$$

MRR@K is the average of this over multiple queries.

Success and Reciprocal Rank: A comparison

$\mathbf{D}_1$ for q		
1	doc_C	⋆
2	doc_E	⋆
3	doc_D	
4	doc_B	
5	doc_A	
6	doc_F	⋆

- $\text{Success@2}(q, \mathbf{D}_1) = 1$
- $\text{RR@2}(q, \mathbf{D}_1) = 1/1$

$\mathbf{D}_2$ for q		
1	doc_A	
2	doc_C	⋆
3	doc_G	
4	doc_B	
5	doc_E	⋆
6	doc_F	⋆

- $\text{Success@2}(q, \mathbf{D}_2) = 1$
- $\text{RR@2}(q, \mathbf{D}_2) = 1/2$

$\mathbf{D}_3$ for q		
1	doc_D	
2	doc_B	
3	doc_E	⋆
4	doc_C	⋆
5	doc_F	⋆
6	doc_A	

- $\text{Success@2}(q, \mathbf{D}_3) = 0$
- $\text{RR@2}(q, \mathbf{D}_3) = 0$

Precision and Recall

$\text{Ret}(\mathbf{D}, K)$ is the set of documents at or above K in $\mathbf{D}$.

$\text{Rel}(\mathbf{D}, q)$ is the set of all documents that are relevant q.

Precision

$$\text{Prec@K}(q, \mathbf{D}) = \frac{|\text{Ret}(\mathbf{D}, K) \cap \text{Rel}(\mathbf{D}, q)|}{K}$$

Recall

$$\text{Rec@K}(q, \mathbf{D}) = \frac{|\text{Ret}(\mathbf{D}, K) \cap \text{Rel}(\mathbf{D}, q)|}{|\text{Rel}(\mathbf{D}, q)|}$$

Precision and Recall examples

$\mathbf{D}_1$ for q		
1	doc_C	⋆
2	doc_E	⋆
3	doc_D	
4	doc_B	
5	doc_A	
6	doc_F	⋆

- $\text{Prec@2}(q, \mathbf{D}_1) = 2/2$
- $\text{Rec@2}(q, \mathbf{D}_1) = 2/3$

$\mathbf{D}_2$ for q		
1	doc_A	
2	doc_C	⋆
3	doc_G	
4	doc_B	
5	doc_E	⋆
6	doc_F	⋆

- $\text{Prec@2}(q, \mathbf{D}_2) = 1/2$
- $\text{Rec@2}(q, \mathbf{D}_2) = 1/3$

$\mathbf{D}_3$ for q		
1	doc_D	
2	doc_B	
3	doc_E	⋆
4	doc_C	⋆
5	doc_F	⋆
6	doc_A	

- $\text{Prec@2}(q, \mathbf{D}_3) = 0/2$
- $\text{Rec@2}(q, \mathbf{D}_3) = 0/3$

Precision and Recall examples

$\mathbf{D}_1$ for q		
1	doc_C	⋆
2	doc_E	⋆
3	doc_D	
4	doc_B	
5	doc_A	
6	doc_F	⋆

- $\text{Prec@5}(q, \mathbf{D}_1) = 2/5$
- $\text{Rec@5}(q, \mathbf{D}_1) = 2/3$

$\mathbf{D}_2$ for q		
1	doc_A	
2	doc_C	⋆
3	doc_G	
4	doc_B	
5	doc_E	⋆
6	doc_F	⋆

- $\text{Prec@5}(q, \mathbf{D}_2) = 2/5$
- $\text{Rec@5}(q, \mathbf{D}_2) = 2/3$

$\mathbf{D}_3$ for q		
1	doc_D	
2	doc_B	
3	doc_E	⋆
4	doc_C	⋆
5	doc_F	⋆
6	doc_A	

- $\text{Prec@5}(q, \mathbf{D}_3) = 3/5$
- $\text{Rec@5}(q, \mathbf{D}_3) = 3/3$

Average Precision

$$\text{AvgPrec}(q, \mathbf{D}) = \frac{\sum_{i=1}^{|\mathbf{D}|} \begin{cases} \text{Prec@i}(q, \mathbf{D}) & \text{if } \text{Rel}(q, \text{doc}_i) \\ 0 & \text{otherwise} \end{cases}}{|\text{Rel}(\mathbf{D}, q)|}$$

$\mathbf{D}_1$ for q		
1	doc_C	⋆
2	doc_E	⋆
3	doc_D	
4	doc_B	
5	doc_A	
6	doc_F	⋆

$\mathbf{D}_2$ for q		
1	doc_A	
2	doc_C	⋆
3	doc_G	
4	doc_B	
5	doc_E	⋆
6	doc_F	⋆

$\mathbf{D}_3$ for q		
1	doc_D	
2	doc_B	
3	doc_E	⋆
4	doc_C	⋆
5	doc_F	⋆
6	doc_A	

$\text{Prec@1}(q, \mathbf{D}) = 1/1 +$
$\text{Prec@2}(q, \mathbf{D}) = 2/2 +$
$\text{Prec@6}(q, \mathbf{D}) = 3/6 +$
$\mathbf{2.5/3}$

$\text{Prec@2}(q, \mathbf{D}) = 1/2 +$
$\text{Prec@5}(q, \mathbf{D}) = 2/5 +$
$\text{Prec@6}(q, \mathbf{D}) = 3/6 +$
$\mathbf{1.4/3}$

$\text{Prec@3}(q, \mathbf{D}) = 1/3 +$
$\text{Prec@4}(q, \mathbf{D}) = 2/4 +$
$\text{Prec@5}(q, \mathbf{D}) = 3/5 +$
$\mathbf{1.43/3}$

Which metric? There is no single answer!

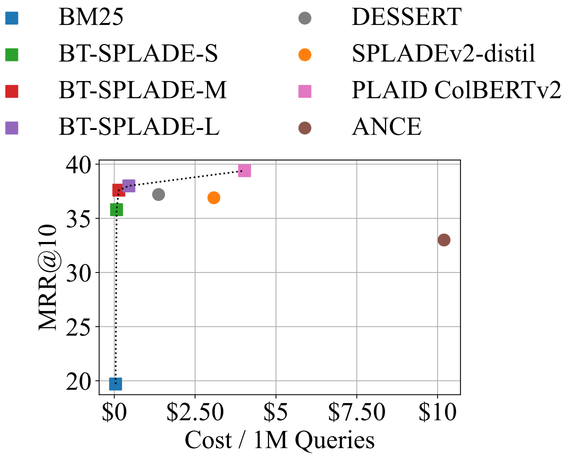
1. Is the cost of scrolling through K passages low? Then perhaps Success@K is fine-grained enough.
2. Are there multiple relevant documents per query? If so, Success@K and RR@K may be too coarse-grained.
3. Is it more important to find every relevant document? If so, favor Recall.
4. Is it more important to review only relevant documents? If so, favor Precision.
5. F1@K is the harmonic mean of Prec@K and Recall@K. It can be used where there are multiple relevant documents but their relative order above K doesn't matter that much.
6. AvgPrec will give the finest-grained distinctions of the metrics discussed here: it is sensitive to rank, precision, and recall.

Beyond accuracy

	Hardware			Performance		
	GPU	CPU	RAM (GiB)	MRR@10	Query Latency (ms)	Index Size (GiB)
BM25 (Mackenzie et al., 2021)	0	32	512	18.7	8	1
BM25 (Lassance and Clinchant, 2022)	0	64	-	19.7	4	1
SPLADEv2-distil (Mackenzie et al., 2021)	0	32	512	36.9	220	4
SPLADEv2-distil (Lassance and Clinchant, 2022)	0	64	-	36.8	691	4
BT-SPLADE-S (Lassance and Clinchant, 2022)	0	64	-	35.8	7	1
BT-SPLADE-M (Lassance and Clinchant, 2022)	0	64	-	37.6	13	2
BT-SPLADE-L (Lassance and Clinchant, 2022)	0	64	-	38.0	32	4
ANCE (Xiong et al., 2020)	1	48	650	33.0	12	-
RocketQAv2 (Ren et al., 2021)	-	-	-	37.0	-	-
coCondenser (Gao and Callan, 2021)	-	-	-	38.2	-	-
CoT-MAE (Wu et al., 2022)	-	-	-	39.4	-	-
ColBERTv1 (Khattab and Zaharia, 2020)	4	56	469	36.1	54	154
PLAID ColBERTv2 (Santhanam et al., 2022a)	4	56	503	39.4	32	22
PLAID ColBERTv2 (Santhanam et al., 2022a)	4	56	503	39.4	12	22
DESSERT (Engels et al., 2022)	0	24	235	37.2	16	-

Santhanam et al. 2022c

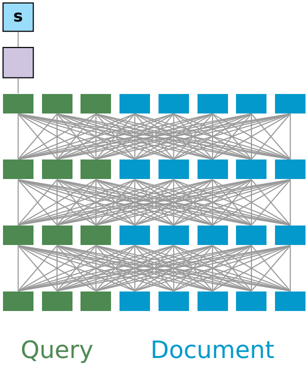
Beyond accuracy



Santhanam et al. 2022c

Neural IR

Cross-encoders



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

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

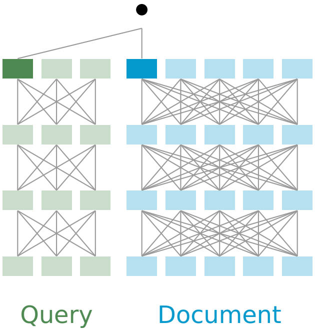
$$\textbf{Rep}(q, \text{doc}) = \text{Dense}\left(\textbf{Enc}([q; \text{doc}]_{N,0})\right)$$

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

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

Incredibly rich, but won't scale!

DPR



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

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

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

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

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

Highly scalable, but limited query/doc interactions!

Karpukhin et al. 2020

Shared loss function

The negative log-likelihood of the positive passage:

Cross encoders

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

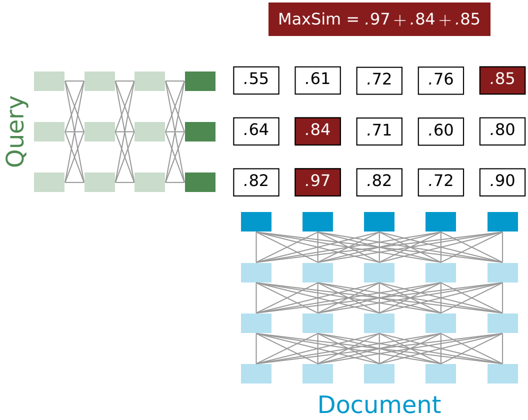
DPR

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

General form

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

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!

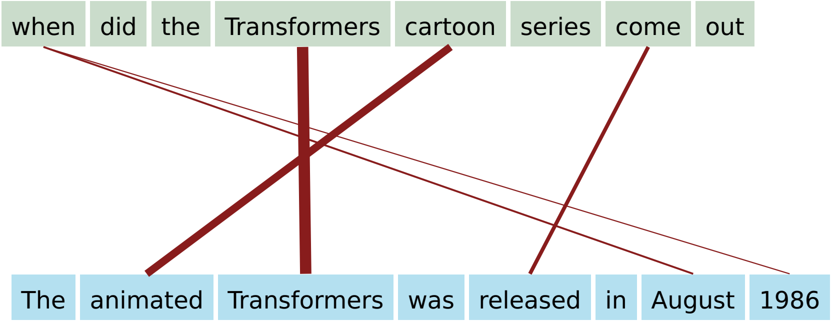
For a BERT-style encoder with N layers:

$$\mathbf{MaxSim}(q, \text{doc}) = \sum_{i}^{L} \max_{j}^{M} \mathbf{Enc}(q)_{N,i}{}^{\top} \mathbf{Enc}(\text{doc})_{N,j}$$

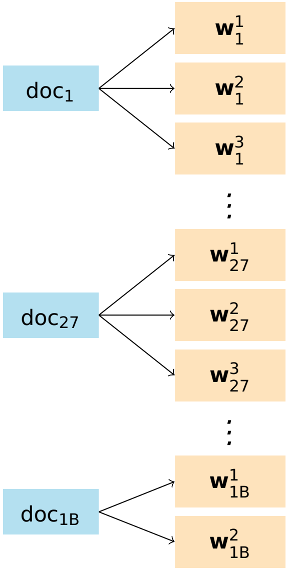
with L is the length of q, M the length of doc.

Khattab and Zaharia 2020

Soft alignment with ColBERT



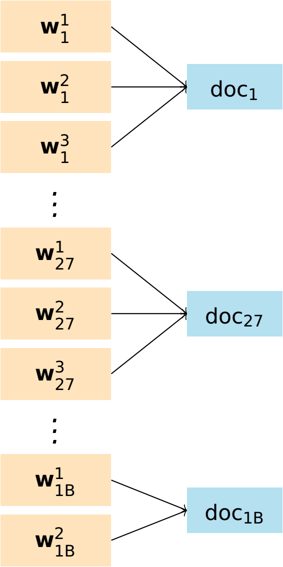
ColBERT as a reranker



Given query $q = [w^1, \ldots, w^M]$:

1. Get the top K documents for q using a fast, term-based model like BM25.
2. Score each of those top K documents using ColBERT.

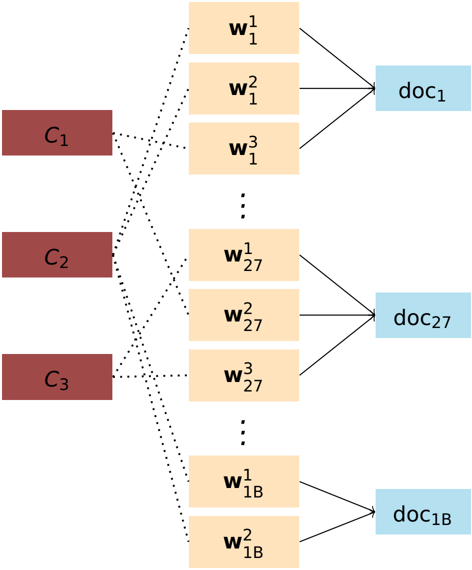
Beyond reranking for ColBERT



Given query q encoded as vectors $[\mathbf{w}^1, \ldots, \mathbf{w}^M]$, for each query vector $\mathbf{w}^i$:

1. Retrieve the p most similar token vectors $\mathbf{w}_j^k$ to $\mathbf{w}^i$.
2. Score each doc_j using ColBERT.

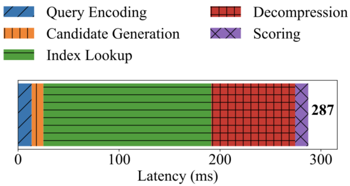
Centroid-based ranking



Given q encoded as $[\mathbf{w}^1, \ldots, \mathbf{w}^M]$, for each vector $\mathbf{w}^i$:

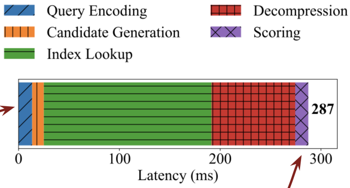
1. Retrieve the p centroids closest to $\mathbf{w}^i$.
2. Retrieve the t most similar token vectors $\mathbf{w}_j^k$ to any of the centroids.
3. Score each doc_j using ColBERT.

ColBERT latency analysis



Santhanam et al. 2022a

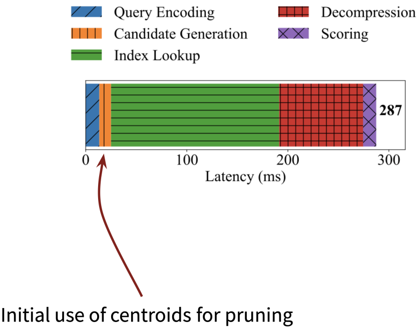
ColBERT latency analysis



Core ColBERT model steps

Santhanam et al. 2022a

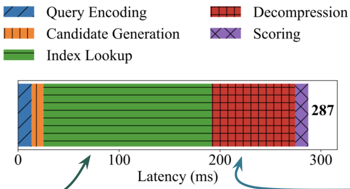
ColBERT latency analysis

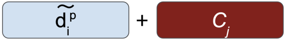


Santhanam et al. 2022a

ColBERT latency analysis

Memory overhead from centroid and residual retrieval over a huge index.



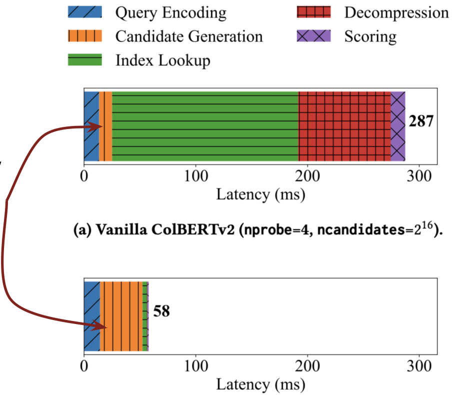


… for up to 40K passages

Santhanam et al. 2022a

Additional ColBERT optimizations

PLAID generates many more candidates and then filters them extremely efficiently.

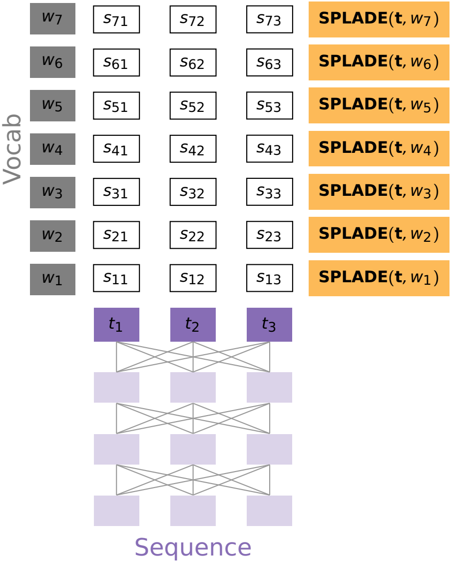


(a) Vanilla ColBERTv2 (nprobe=4, ncandidates=2^{16}).

(b) PLAID ColBERTv2 (k = 1000)

Santhanam et al. 2022a

SPLADE



1. $s_{ij} =$
   $$\textbf{transform}\left(\textbf{Enc}(\mathbf{t})_{N,i}\right)^{\top}\textbf{Emb}(w_j) + b_j\Big)$$
   where
   $$\textbf{transform}(x) = \textbf{LayerNorm}(\textbf{GeLU}(xW + b))$$
   and **Emb**(w) is the embedding for w.
2. $\textbf{SPLADE}(\mathbf{t}, w_j) =$
   $$\sum_{i}^{M} \log\left(1 + \textbf{ReLU}(s_{ij})\right)$$
3. $\textbf{Sim}_{\textbf{SPLADE}}(q, \text{doc}) =$
   $$\textbf{SPLADE}(q)^{\top}\textbf{SPLADE}(\text{doc})$$
4. Loss: Usual negative log-likelihood plus a regularization term that leads to sparse, balanced scores.

Formal et al. 2021

Additional recent developments

This is an incredibly fast-moving field, but here are some selected developments that caught my attention. I confess that these are heavily biased towards ColBERT:

1. CITADEL (Li et al. 2022) is a lightning fast ColBERT-style model.
2. Lassance and Clinchant (2022) developed lightning fast SPLADE variants.
3. DESSERT (Engels et al. 2022) offer ultra-efficient approximate embedding search.
4. Lin et al. (2020) distill ColBERT into a single-vector model akin to DPR.
5. DR.DECR Li et al. (2021) distills multilingual ColBERT models.
6. Choi et al. (2021) distill cross-encoders into ColBERT models.
7. Lee et al. (2023) rework the standard ColBERT objective so that important tokens are retrieved first for blazing fast retrieval.

Multidimensional benchmarking

	Hardware				Performance		
	GPU	CPU	RAM	Instance	Latency	Cost	Success@10
BM25	0	1	4	m6gd.med	11	$0.14	38.6
BM25	0	1	32	x2gd.lrg	10	$0.48	38.6
DPR					146	$6.78	52.1
ColBERTv2-S					206	$9.58	68.8
ColBERTv2-M					321	$14.90	69.6
ColBERTv2-L					459	$21.30	69.7
BT-SPLADE-L					46	$2.15	66.3
BM25	1	16	32	p3.8xl	9	$29.94	38.6
DPR					18	$61.06	52.1
ColBERTv2-S					27	$90.41	68.8
ColBERTv2-M					36	$123.35	69.6
ColBERTv2-L					55	$187.24	69.7
BT-SPLADE-L					33	$112.87	66.3

Selected MS MARCO results form Santhanam et al. 2022c

Multidimensional benchmarking

	Hardware				Performance		
	GPU	CPU	RAM	Instance	Latency	Cost	Success@10
BM25	0	1	4	m6gd.med	11	$0.14	38.6
BM25	0	1	32	x2gd.lrg	10	$0.48	38.6
DPR					146	$6.78	52.1
ColBERTv2-S					206	$9.58	68.8
ColBERTv2-M					321	$14.90	69.6
ColBERTv2-L					459	$21.30	69.7
BT-SPLADE-L					46	$2.15	66.3
BM25	1	16	32	p3.8xl	9	$29.94	38.6
DPR					18	$61.06	52.1
ColBERTv2-S					27	$90.41	68.8
ColBERTv2-M					36	$123.35	69.6
ColBERTv2-L					55	$187.24	69.7
BT-SPLADE-L					33	$112.87	66.3

Selected MS MARCO results form Santhanam et al. 2022c

Multidimensional benchmarking

	Hardware				Performance		
	GPU	CPU	RAM	Instance	Latency	Cost	Success@10
BM25	0	1	4	m6gd.med	11	$0.14	38.6
BM25	0	1	32	x2gd.lrg	10	$0.48	38.6
DPR					146	$6.78	52.1
ColBERTv2-S					206	$9.58	68.8
ColBERTv2-M					321	$14.90	69.6
ColBERTv2-L					459	$21.30	69.7
BT-SPLADE-L					46	$2.15	66.3
BM25	1	16	32	p3.8xl	9	$29.94	38.6
DPR					18	$61.06	52.1
ColBERTv2-S					27	$90.41	68.8
ColBERTv2-M					36	$123.35	69.6
ColBERTv2-L					55	$187.24	69.7
BT-SPLADE-L					33	$112.87	66.3

Selected MS MARCO results form Santhanam et al. 2022c

Multidimensional benchmarking

	Hardware				Performance		
	GPU	CPU	RAM	Instance	Latency	Cost	Success@10
BM25	0	1	4	m6gd.med	11	$0.14	38.6
BM25	0	1	32	x2gd.lrg	10	$0.48	38.6
DPR					146	$6.78	52.1
ColBERTv2-S					206	$9.58	68.8
ColBERTv2-M					321	$14.90	69.6
ColBERTv2-L					459	$21.30	69.7
BT-SPLADE-L					46	$2.15	66.3
BM25	1	16	32	p3.8xl	9	$29.94	38.6
DPR					18	$61.06	52.1
ColBERTv2-S					27	$90.41	68.8
ColBERTv2-M					36	$123.35	69.6
ColBERTv2-L					55	$187.24	69.7
BT-SPLADE-L					33	$112.87	66.3

Selected MS MARCO results form Santhanam et al. 2022c

Multidimensional benchmarking

	Hardware				Performance		
	GPU	CPU	RAM	Instance	Latency	Cost	Success@10
BM25	0	1	4	m6gd.med	11	$0.14	38.6
BM25	0	1	32	x2gd.lrg	10	$0.48	38.6
DPR					146	$6.78	52.1
ColBERTv2-S					206	$9.58	68.8
ColBERTv2-M					321	$14.90	69.6
ColBERTv2-L					459	$21.30	69.7
BT-SPLADE-L					46	$2.15	66.3
BM25	1	16	32	p3.8xl	9	$29.94	38.6
DPR					18	$61.06	52.1
ColBERTv2-S					27	$90.41	68.8
ColBERTv2-M					36	$123.35	69.6
ColBERTv2-L					55	$187.24	69.7
BT-SPLADE-L					33	$112.87	66.3

Selected MS MARCO results form Santhanam et al. 2022c

Datasets

TREC

1. **T**ext **RE**trieval **C**onference (TREC) has annual competitions for comparing IR systems.
2. The 2023 iteration has a number of tracks: `https://trec.nist.gov/pubs/call2023.html`
3. TREC tends to emphasize careful evaluation with a very small set of queries (e.g., 50 queries, each with >100 annotated documents).
4. Having few test queries does not imply few documents!

MS MARCO ranking tasks

1. MS MARCO Ranking is the largest public IR benchmark.
2. It is adapted from a Question Answering dataset
3. It consists of more than 500k Bing search queries
4. Sparse labels: approx. one relevance label per query!
5. Fantastic for training IR models!
6. Passage Ranking: 9M short passages; sparse labels
7. Document Ranking: 3M long documents; sparse labels

BEIR: Benchmarking IR

For testing models in zero-shot scenarios:

Split (→)					Train	Dev	Test			Avg. Word Lengths	
Task (↓)	Domain (↓)	Dataset (↓)	Title	Relevancy	#Pairs	#Query	#Query	#Corpus	Avg. D / Q	Query	Document
Passage-Retrieval	Misc.	MS MARCO [45]	✗	Binary	532,761	—	6,980	8,841,823	1.1	5.96	55.98
Bio-Medical Information Retrieval (IR)	Bio-Medical	TREC-COVID [65]	✓	3-level	—	—	50	171,332	493.5	10.60	160.77
	Bio-Medical	NFCorpus [7]	✓	3-level	110,575	324	323	3,633	38.2	3.30	232.26
	Bio-Medical	BioASQ [61]	✓	Binary	32,916	—	500	14,914,602	4.7	8.05	202.61
Question Answering (QA)	Wikipedia	NQ [34]	✓	Binary	132,803	—	3,452	2,681,468	1.2	9.16	78.88
	Wikipedia	HotpotQA [76]	✓	Binary	170,000	5,447	7,405	5,233,329	2.0	17.61	46.30
	Finance	FiQA-2018 [44]	✗	Binary	14,166	500	648	57,638	2.6	10.77	132.32
Tweet-Retrieval	Twitter	Signal-1M (RT) [59]	✗	3-level	—	—	97	2,866,316	19.6	9.30	13.93
News Retrieval	News	TREC-NEWS [58]	✓	5-level	—	—	57	594,977	19.6	11.14	634.79
	News	Robust04 [64]	✗	3-level	—	—	249	528,155	69.9	15.27	466.40
Argument Retrieval	Misc.	ArguAna [67]	✓	Binary	—	—	1,406	8,674	1.0	192.98	166.80
	Misc.	Touché-2020 [6]	✓	3-level	—	—	49	382,545	19.0	6.55	292.37
Duplicate-Question Retrieval	StackEx.	CQADupStack [25]	✓	Binary	—	—	13,145	457,199	1.4	8.59	129.09
	Quora	Quora	✗	Binary	—	5,000	10,000	522,931	1.6	9.53	11.44
Entity-Retrieval	Wikipedia	DBPedia [21]	✓	3-level	—	67	400	4,635,922	38.2	5.39	49.68
Citation-Prediction	Scientific	SCIDOCS [9]	✓	Binary	—	—	1,000	25,657	4.9	9.38	176.19
Fact Checking	Wikipedia	FEVER [60]	✓	Binary	140,085	6,666	6,666	5,416,568	1.2	8.13	84.76
	Wikipedia	Climate-FEVER [14]	✓	Binary	—	—	1,535	5,416,593	3.0	20.13	84.76
	Scientific	SciFact [68]	✓	Binary	920	—	300	5,183	1.1	12.37	213.63

Thakur et al. 2021

LoTTE: Long-Tail, Topic-stratified Evaluation

Topic	Question Set	Dev			Test		
		# Questions	# Passages	Subtopics	# Questions	# Passages	Subtopics
Writing	Search Forum	497 2003	277k	ESL, Linguistics, Worldbuilding	1071 2000	200k	English
Recreation	Search Forum	563 2002	263k	Sci-Fi, RPGs, Photography	924 2002	167k	Gaming, Anime, Movies
Science	Search Forum	538 2013	344k	Chemistry, Statistics, Academia	617 2017	1.694M	Math, Physics, Biology
Technology	Search Forum	916 2003	1.276M	Web Apps, Ubuntu, SysAdmin	596 2004	639k	Apple, Android, UNIX, Security
Lifestyle	Search Forum	496 2076	269k	DIY, Music, Bicycles, Car Maintenance	661 2002	119k	Cooking, Sports, Travel

Topic-aligned dev-test pairings

Search queries are from GooAQ linked to StackExchange.
Forum queries are from questions-like StackExchange titles

Santhanam et al. 2022b

XOR-TyDI

Information-seeking QA, OpenQA, and multilingual QA

XOR-TyDi v1.1 Leaderboard

Task 1: XOR-Retrieve

XOR-Retrieve is a cross-lingual retrieval task where a question is written in a target language (e.g., Japanese) and a system is required to retrieve English paragraphs that answer the question. The scores are macro-average over the 7 target languages. Although we see the effectiveness of blackbox systems (e.g., Google Translate), **we encourage the community to use white-box systems so that all experimental details can be understood**. The systems using external blackbox APIs are highlighted in gray and ranked in the table of **"Systems using external APIs"** for reference.

Metrics: **R@5kt, R@2kt** (the recall by computing the fraction of the questions for which the minimal answer is contained in the top 5,000 / 2,000 tokens selected.)

Rank	Model	R@5kt	R@2kt
1 October 28, 2022	PrimeQA (DrDecr-large with PLAID + Colbert V2) *IBM Research AI*	**74.7**	**69.2**

https://nlp.cs.washington.edu/xorqa/

Other topics

1. There is a large literature on different techniques for sampling negatives.
2. Weak supervision can often create effective retrieval labels. For example, Khattab et al. (2021) say a passage is relevant in a QA context if it contains the answer as a substring anywhere in the passage.
3. Santhanam et al. (2022c) use Dynascores (Ma et al. 2021) to create unified leaderboards measuring diverse IR metrics, including cost, latency and performance. We will discuss Dynascores in detail later in the course.

Conclusion

NLU and IR are back together again, with profound implications for research and technology development!

References I

Jaekeol Choi, Euna Jung, Jangwon Suh, and Wonjong Rhee. 2021. Improving bi-encoder document ranking models with two rankers and multi-teacher distillation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2192–2196.

Joshua Engels, Benjamin Coleman, Vihan Lakshman, and Anshumali Shrivastava. 2022. DESSERT: An Efficient Algorithm for Vector Set Search with Vector Set Queries. *arXiv preprint arXiv:2210.15748*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *arXiv preprint arXiv:2004.12832*.

Carlos Lassance and Stéphane Clinchant. 2022. An Efficiency Study for SPLADE Models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2220–2226.

Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Y Zhao. 2023. Rethinking the role of token retrieval in multi-vector retrieval. *arXiv preprint arXiv:2304.01982*.

Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2022. CITADEL: Conditional Token Interaction via Dynamic Lexical Routing for Efficient and Effective Multi-Vector Retrieval. *arXiv preprint arXiv:2211.10411*.

Yulong Li, Martin Franz, Md Arafat Sultan, Bhavani Iyer, Young-Suk Lee, and Avirup Sil. 2021. Learning cross-lingual ir from an english retriever. *arXiv preprint arXiv:2112.08185*.

References II

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386*.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In *Advances in Neural Information Processing Systems*, volume 34, pages 10351–10367.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. PLAID: An efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, page 17471756, New York, NY, USA. Association for Computing Machinery.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Keshav Santhanam, Jon Saad-Falcon, Martin Franz, Omar Khattab, Avirup Sil, Radu Florian, Md Arafat Sultan, Salim Roukos, Matei Zaharia, and Christopher Potts. 2022c. Moving beyond downstream task accuracy for information retrieval benchmarking. Ms., Stanford University and IBM Research AI.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.