Pretrained Transformers for Text Ranking:
BERT and Beyond

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin

@andrewyates @rodrigfnogueira @lintool

max planck institut informatik

UNIVERSITY OF WATERLOO

Outline

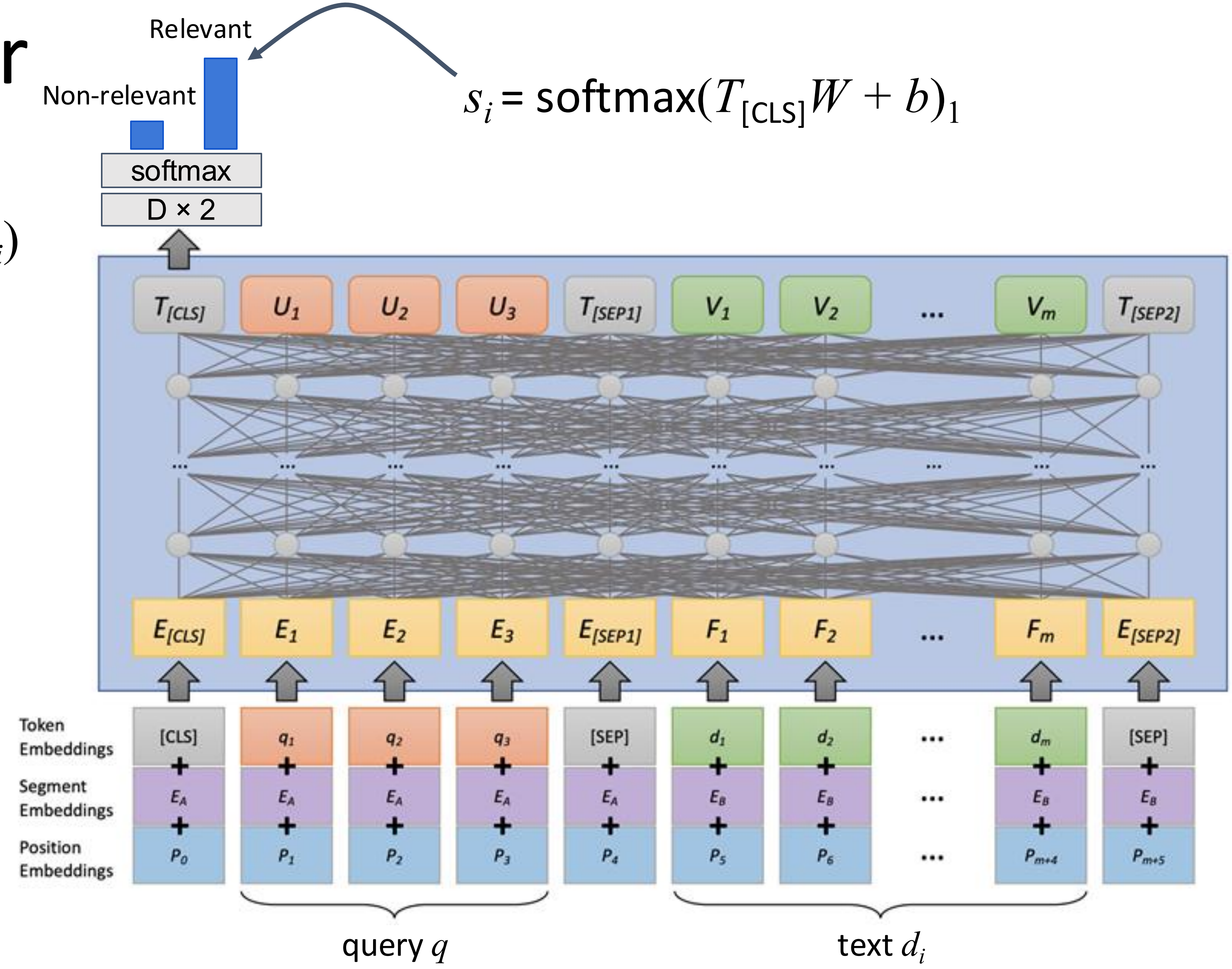
- Part 1: Background
  (text ranking, IR, ML)
- **Part 2: Ranking with relevance classification**
- Part 3: Ranking with dense representations
- Part 4: Conclusion & future directions

monoBERT: BERT reranker

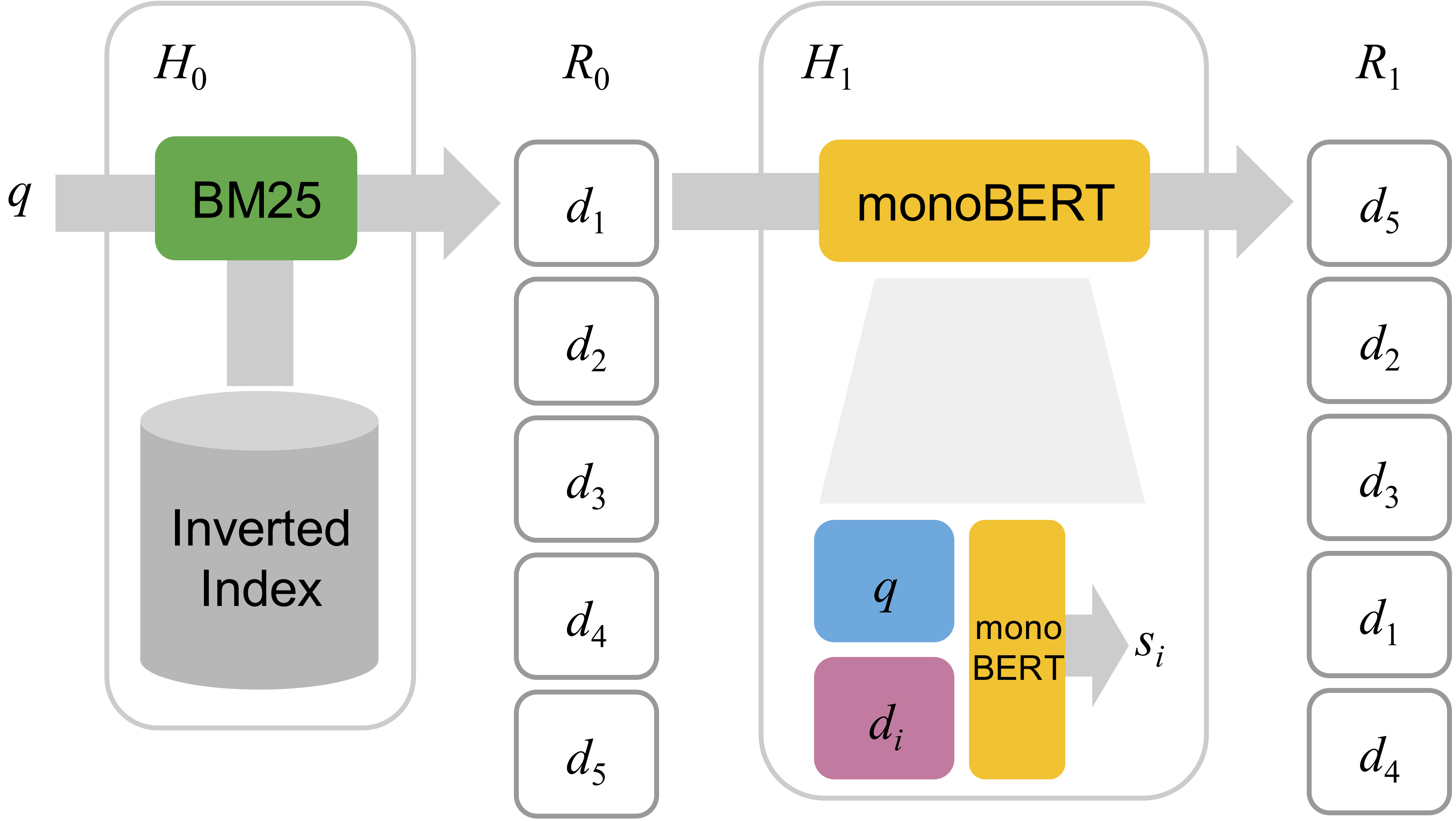
We want:

$s_i = P(\text{Relevant} = 1 | q, d_i)$

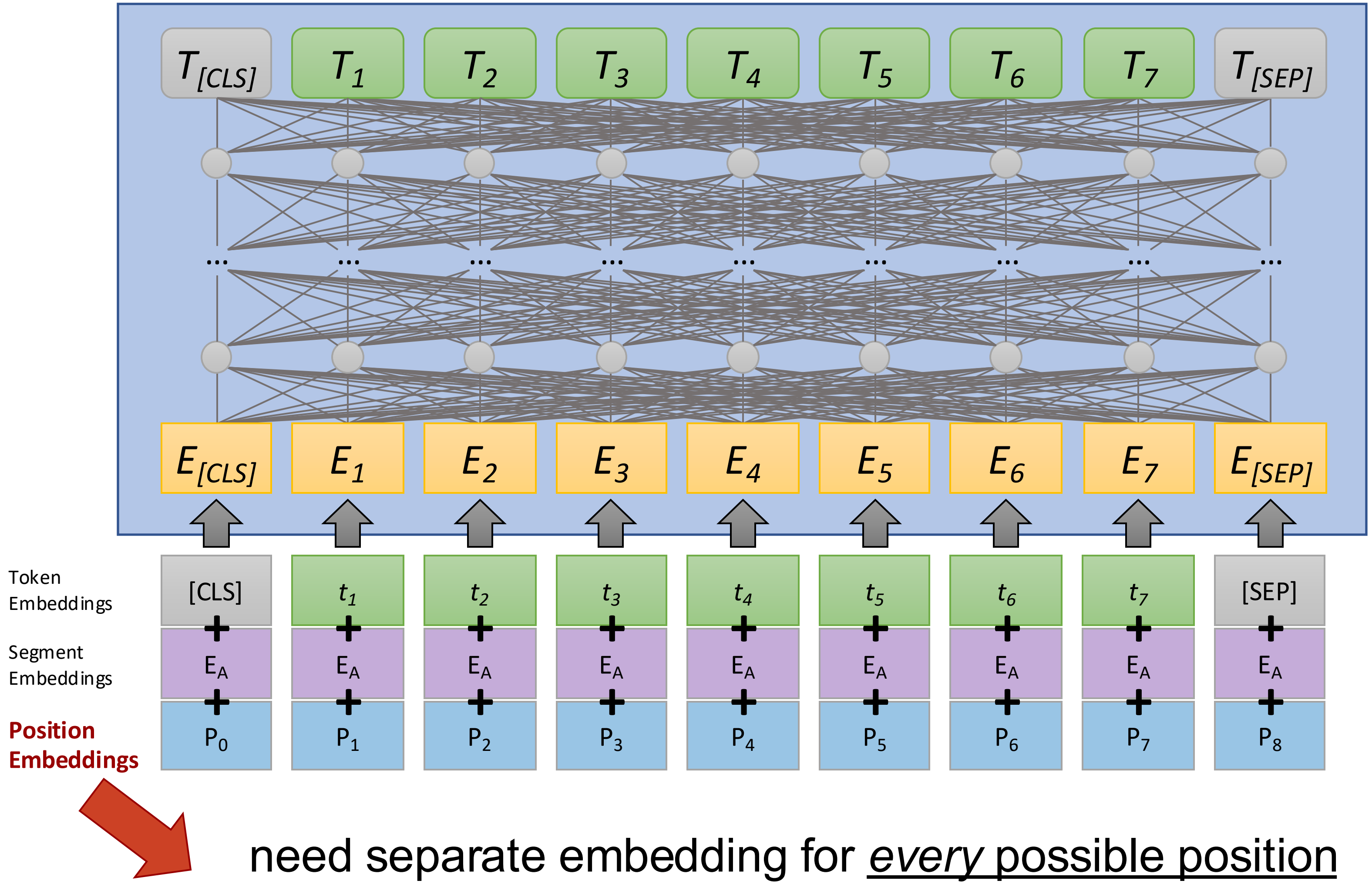
$s_i = \text{softmax}(T_{[\text{CLS}]}W + b)_1$



Once monoBERT is trained...



BERT’s Limitations



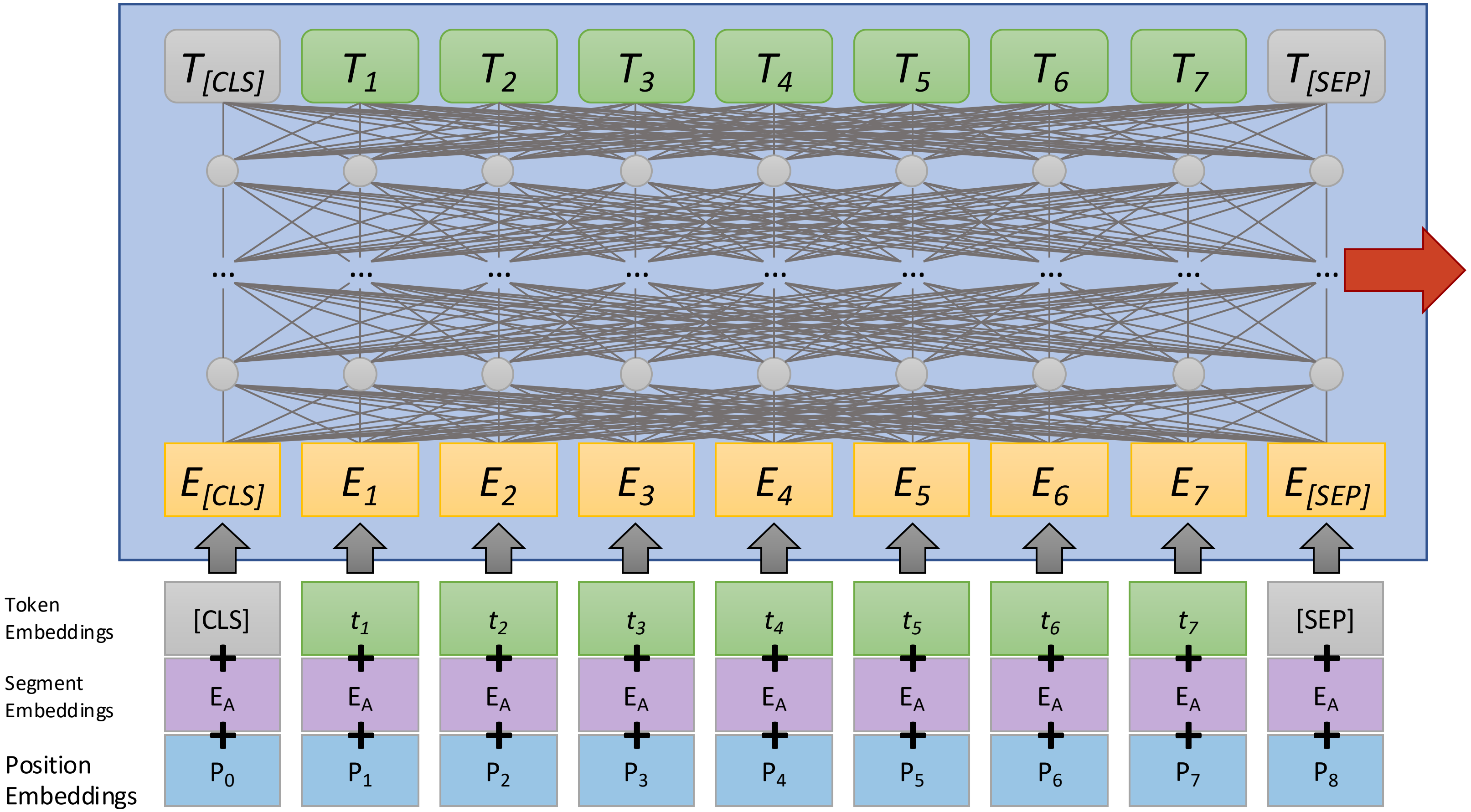
Cannot input entire documents

- what do we input?
- & how do we label it?

need separate embedding for <u>*every* possible position</u>

➔ restricted to indices 0-511

BERT’s Limitations



<u>computationally expensive</u> layers

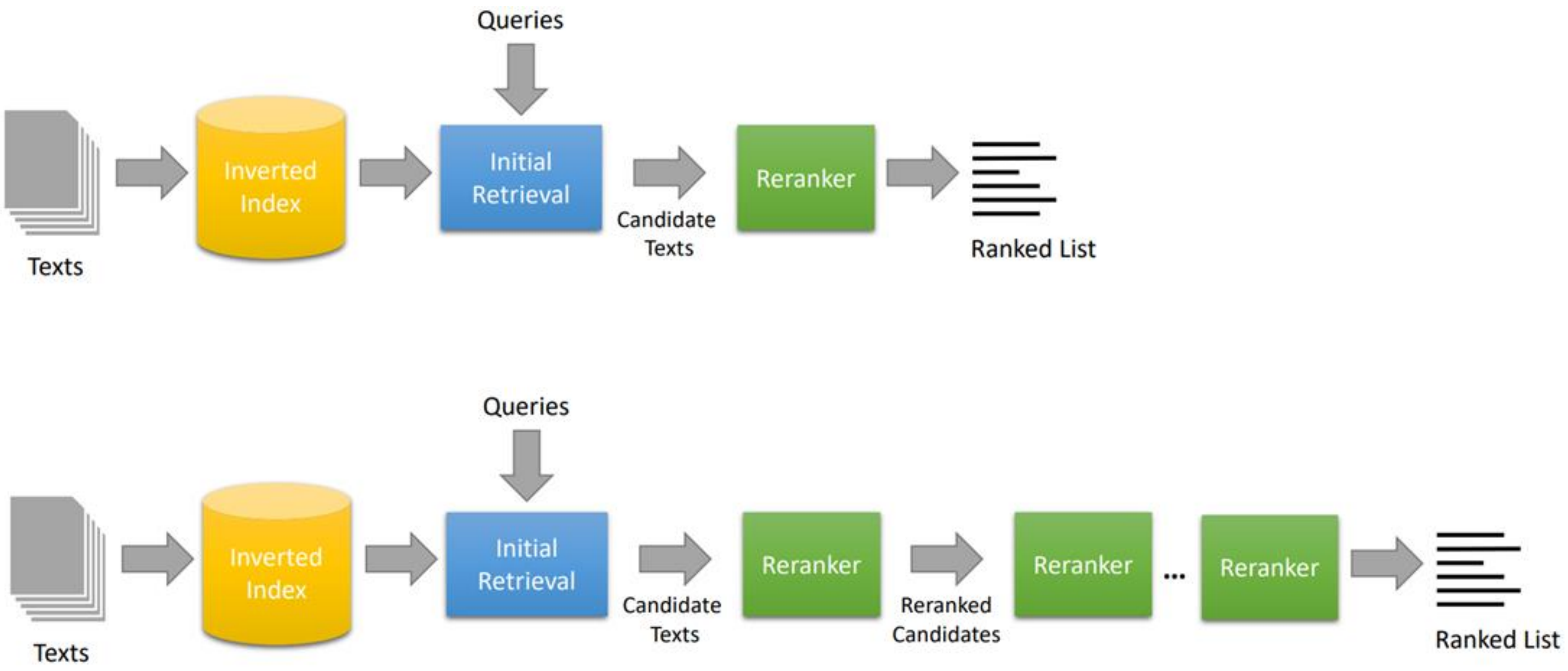
➔ e.g., 110+ *million* learned weights

(later: *Beyond BERT* & *Dense Representations*)

Multi-stage ranking pipeline

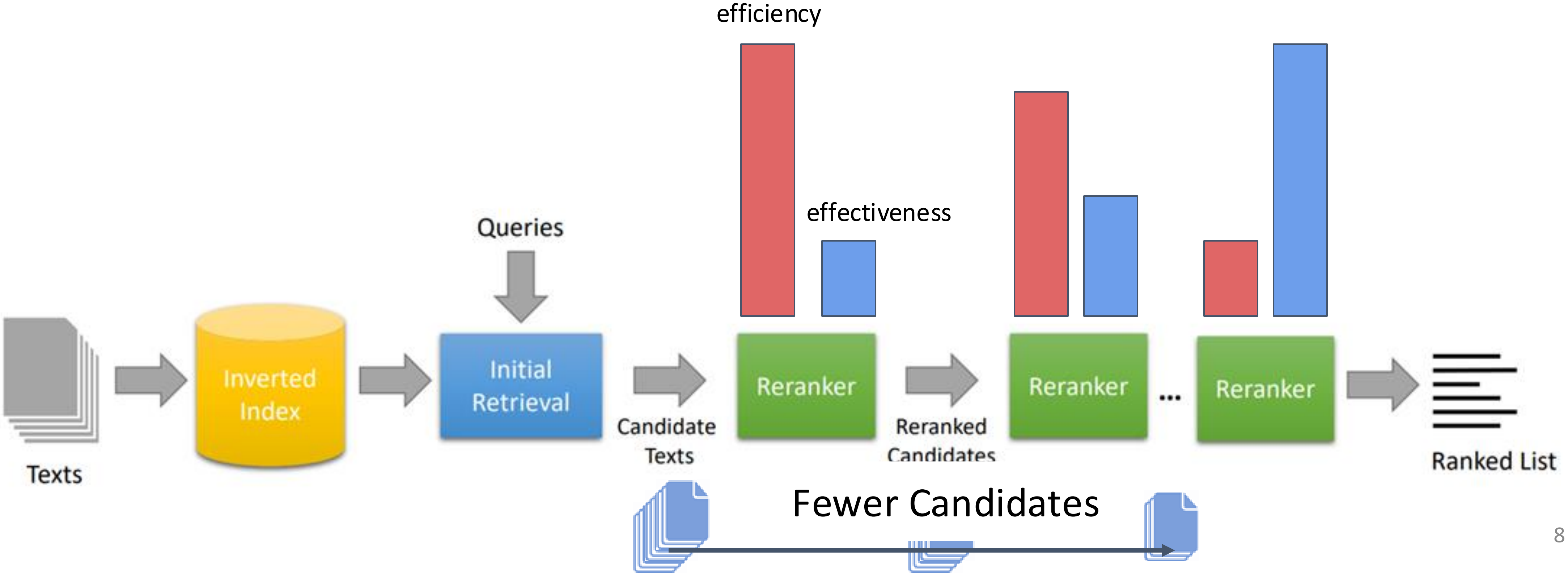
- Identify candidate documents
- Rerank

From Single to Multiple Rerankers

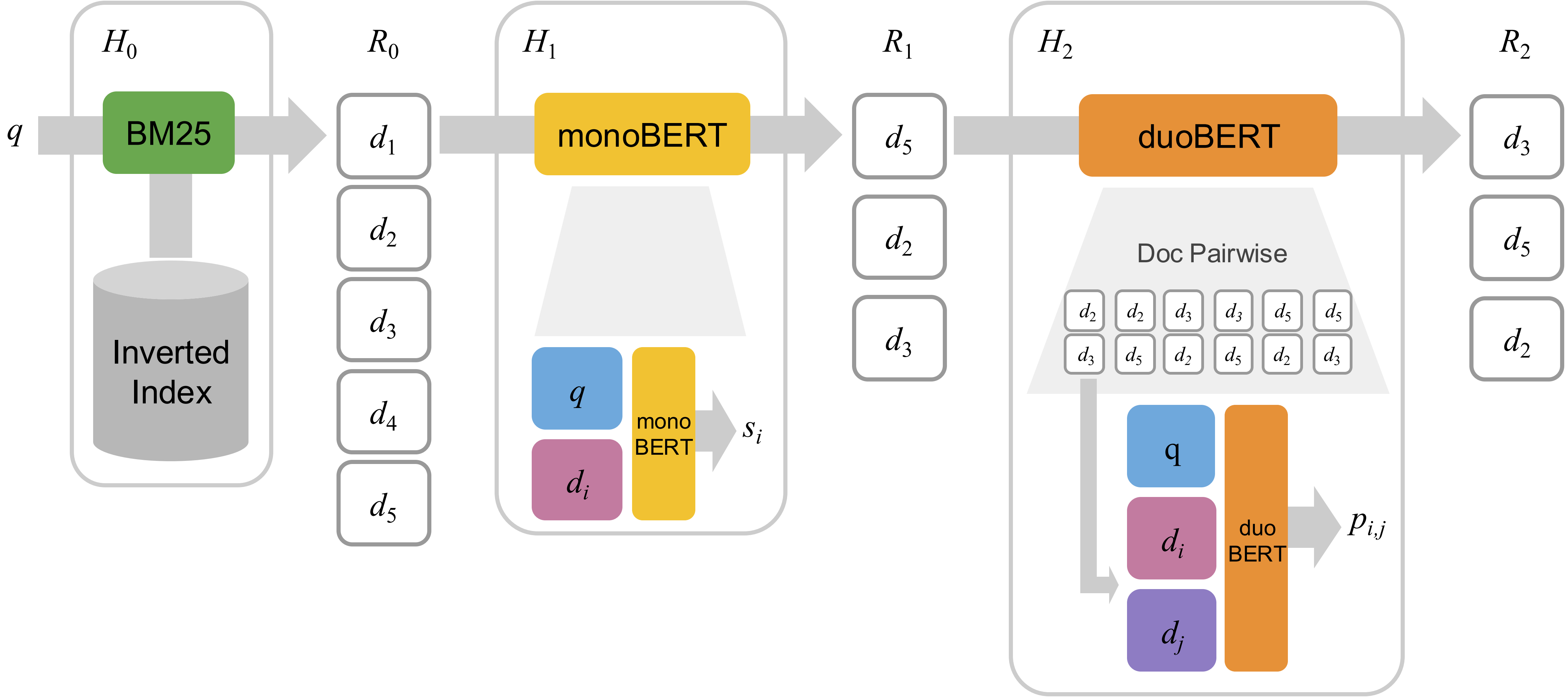


Why Multi-stage?

- Trade-off between effectiveness (quality of the ranked lists) and efficiency (retrieval latency)

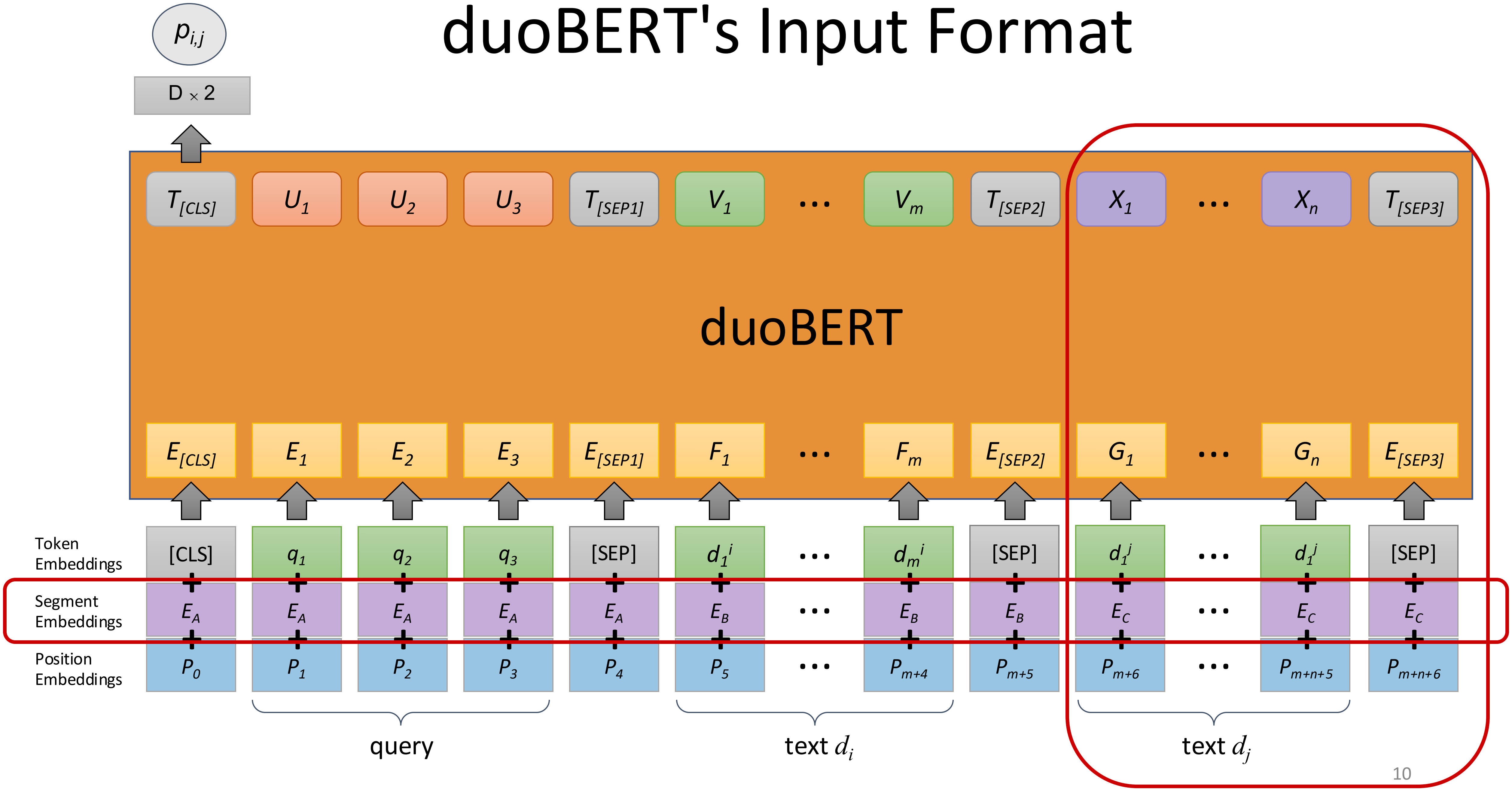


Multi-stage with duoBERT



Nogueira, Yang, Cho, Lin. Multi-stage document ranking with bert. 2019.

duoBERT's Input Format



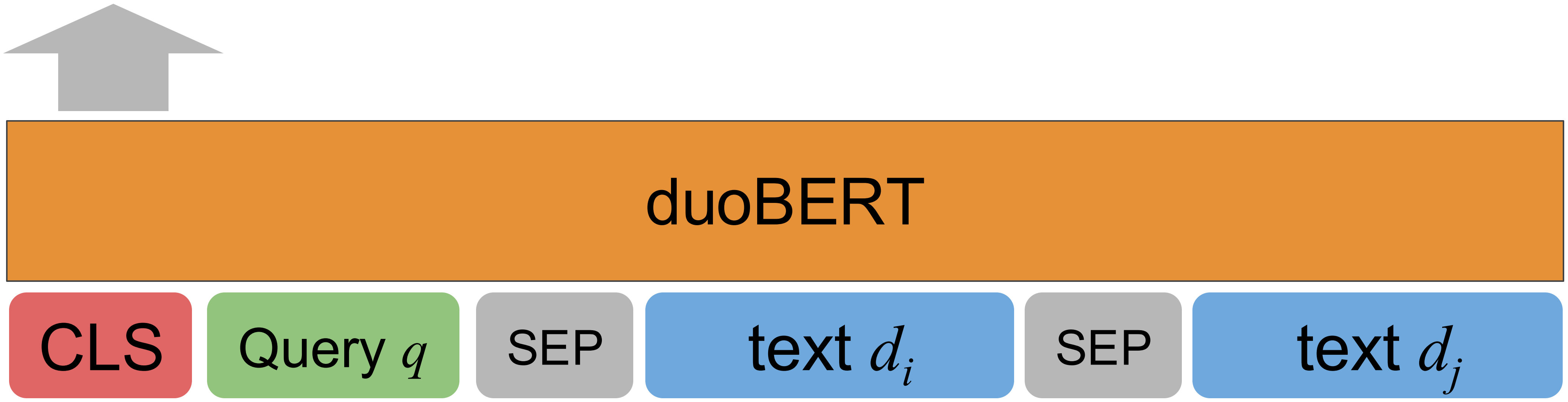
Training duoBERT

Is doc d_i more relevant than doc d_j to the query q?

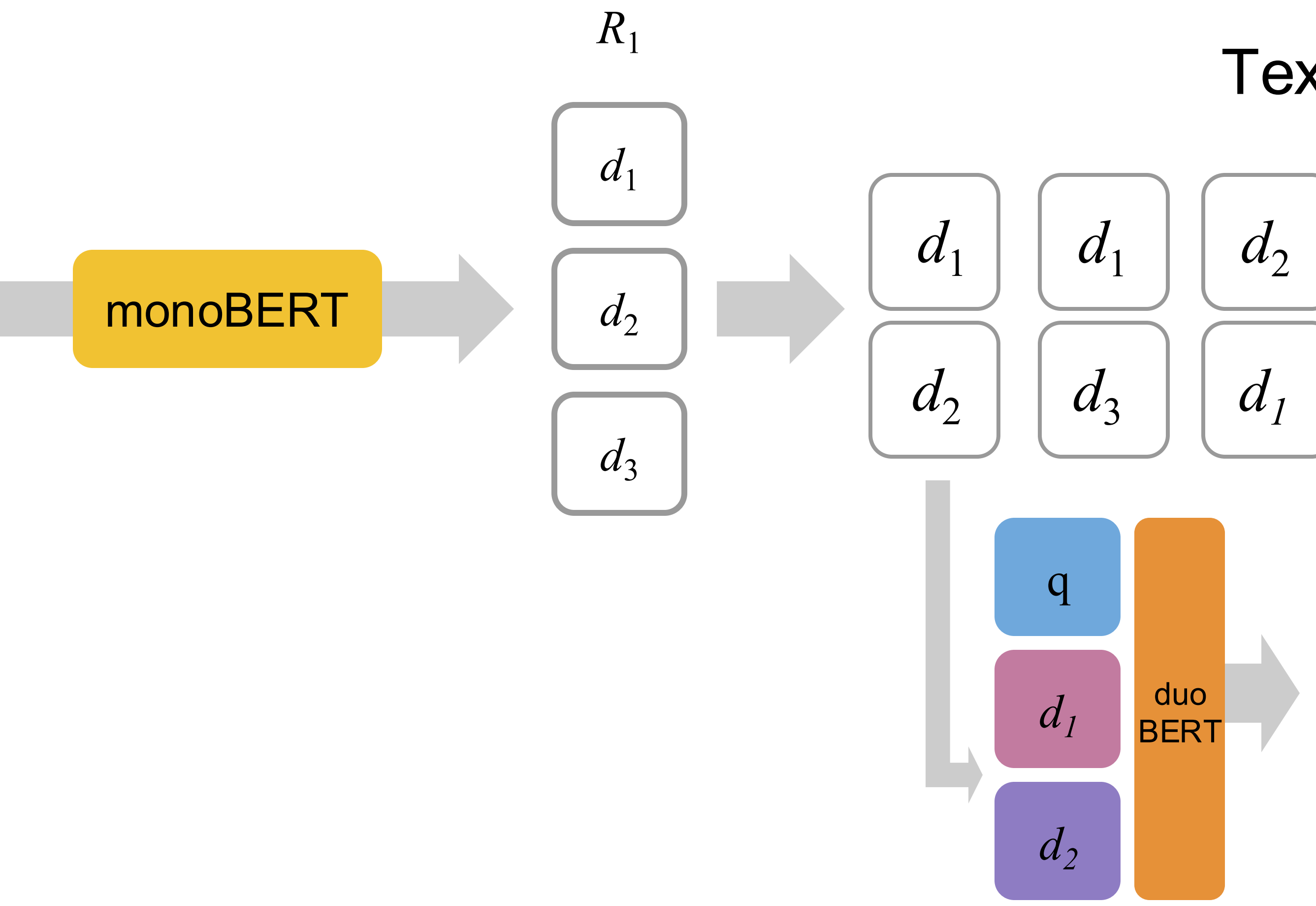
$$p_{i,j} = p(d_i > d_j \mid q)$$

Loss:

$$L_{\text{duo}} = -\sum_{i \in J_{\text{pos}}, j \in J_{\text{neg}}} \log(p_{i,j}) - \sum_{i \in J_{\text{neg}}, j \in J_{\text{pos}}} \log(1 - p_{i,j})$$



Inference with duoBERT



$$p_{1,2} = p(d_1 > d_2 \mid q)$$

Pairwise aggregation:

$$s_1 = p_{1,2} + p_{1,3}$$

$$s_2 = p_{2,1} + p_{2,3}$$

$$s_3 = p_{3,1} + p_{3,2}$$

Outline

- Part 1: Background
  (text ranking, IR, ML)

- Part 2: Ranking with relevance classification

- **Part 3: Ranking with dense representations**

- Part 4: Conclusion & future directions

Sparse Representations

Task: Estimate the relevance of text d to a query q:

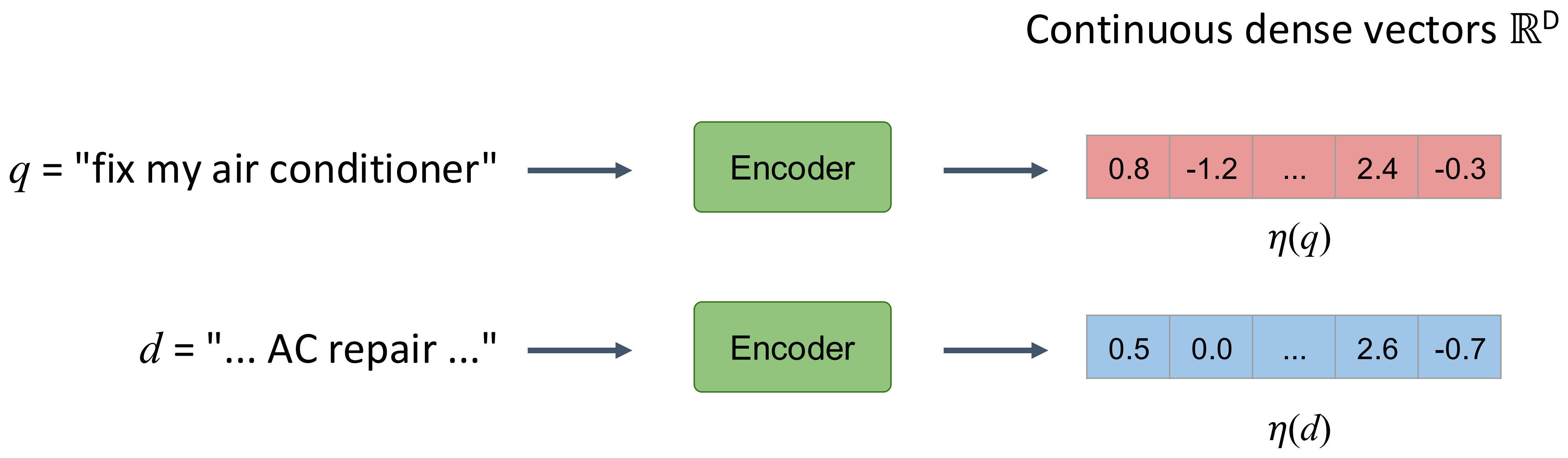
q = "fix my air conditioner"
d = "... AC repair ..."

$$\mathrm{BM25}(q,d) = \sum_{\boxed{t \in q \cap d}} \log \frac{N - \mathrm{df}(t) + 0.5}{\mathrm{df}(t) + 0.5} \cdot \frac{\mathrm{tf}(t,d) \cdot (k_1 + 1)}{\mathrm{tf}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{l_d}{L}\right)}$$

Advantages: 1) Fast to retrieve candidates from a inverted index because q is usually short. 2) Fast to compute because $q \cap d$ is usually small

Disadvantage: Terms need to match exactly

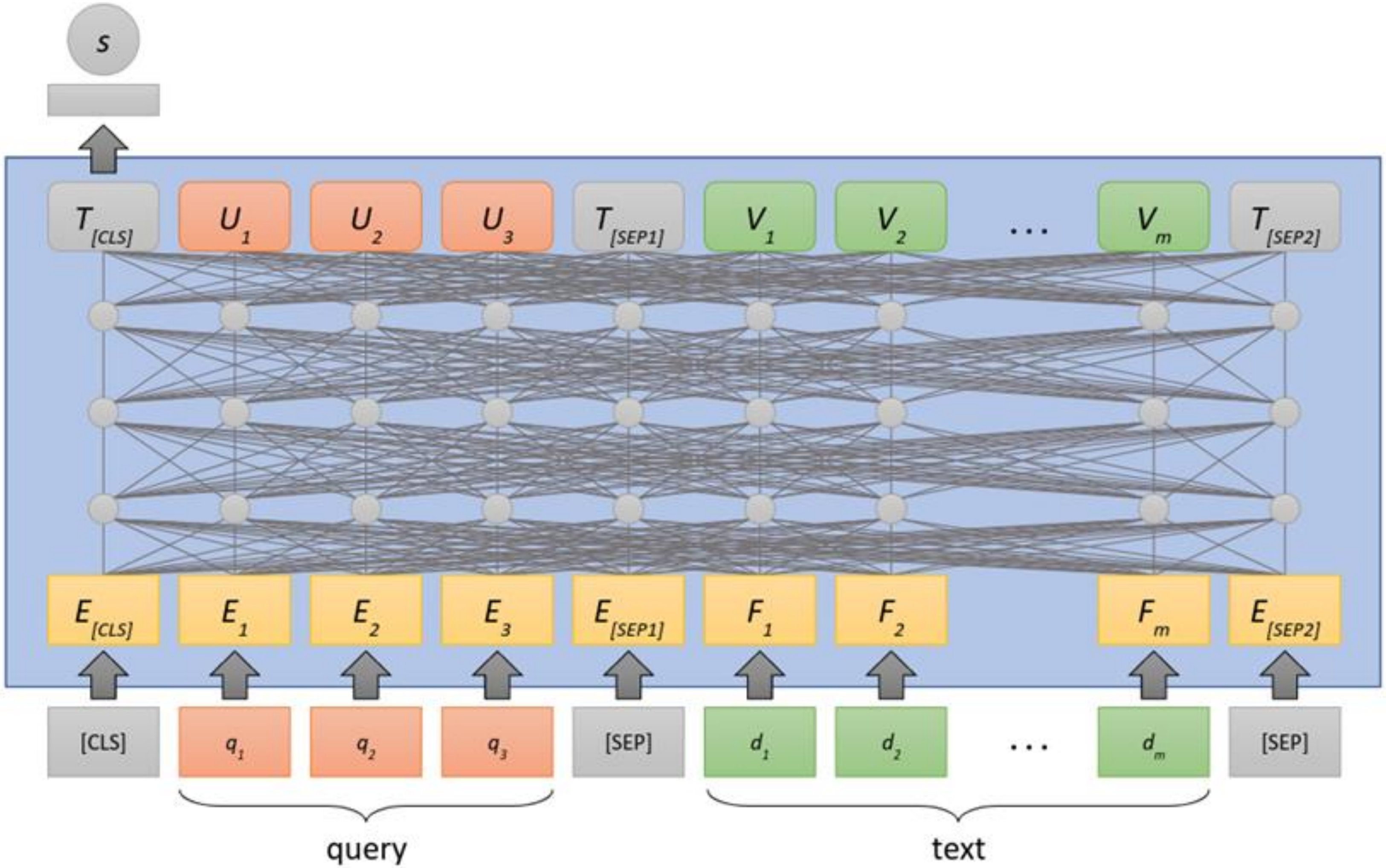
Dense Representations



ϕ is a similarity function (e.g., inner product or cosine similarity)

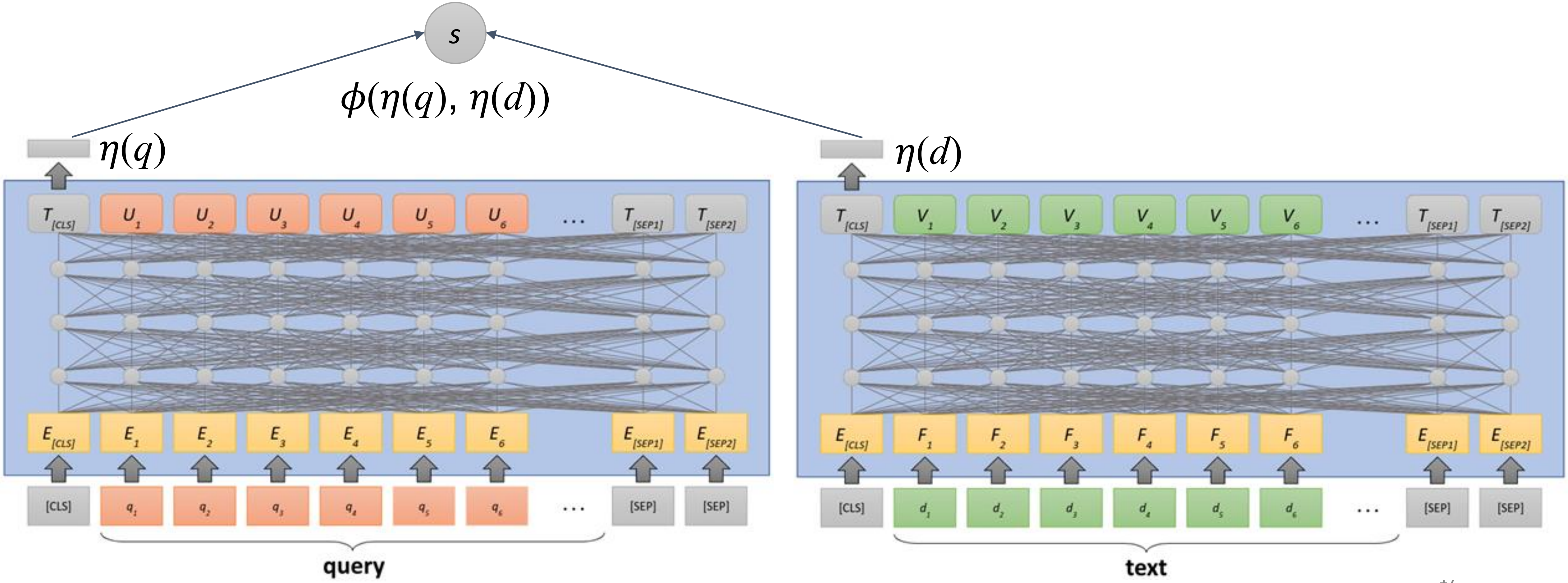
$\phi(\eta(q), \eta(d)) \rightarrow$ ideally measures how relevant q and d are to each other

Types of Encoders: Cross-encoder



Source

Types of Encoders: Bi-encoder

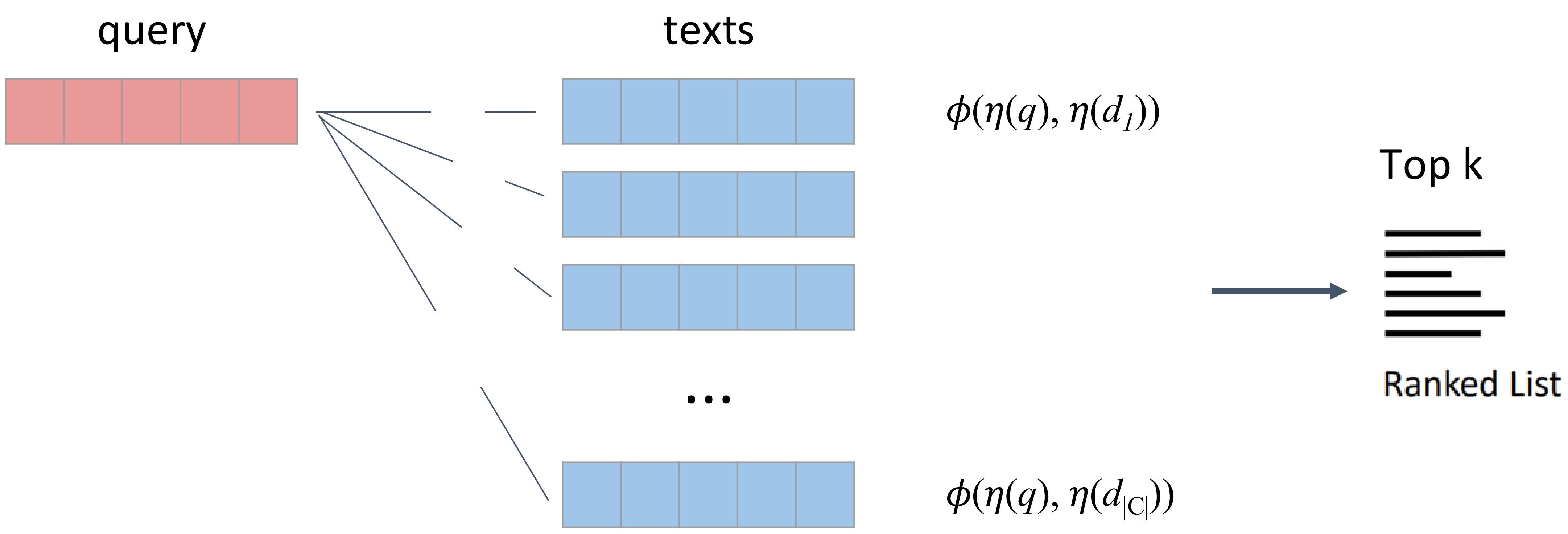


Source

Nearest Neighbor Search

Task: find the top k most relevant texts to a query

Brute-force search:



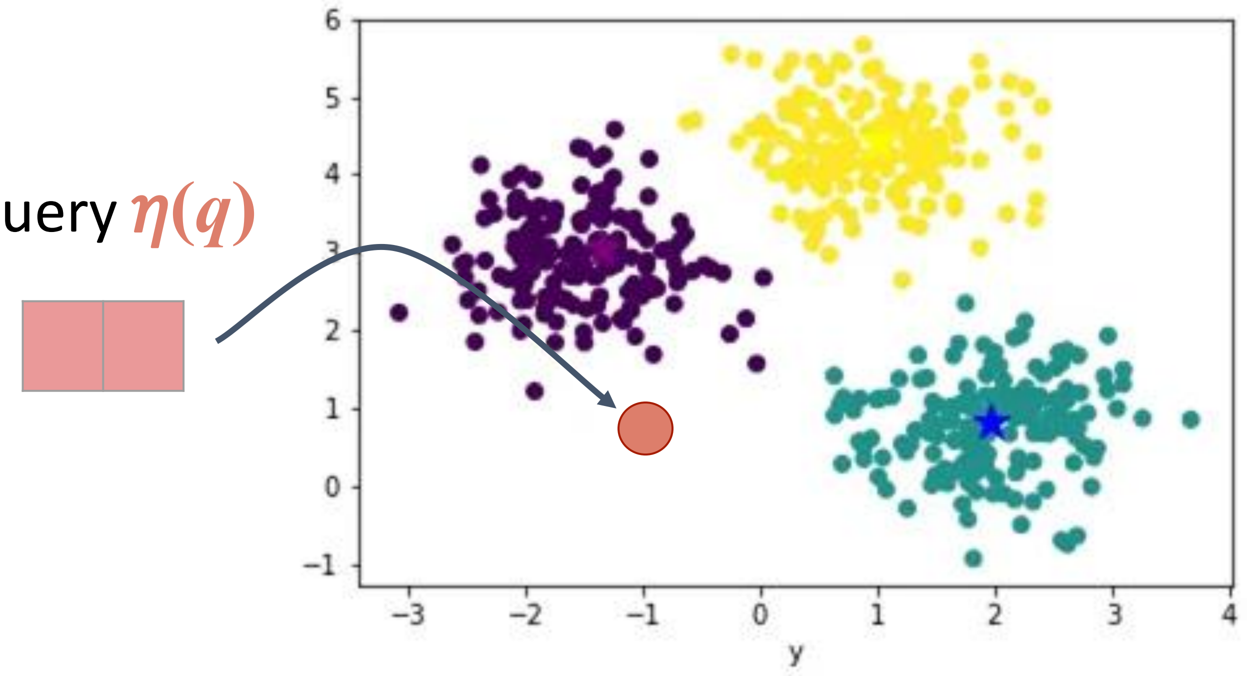
We often need to search many (e.g.: billions) of texts

- Brute-force won't scale

Approximate Nearest Neighbor Search

- Exchange accuracy for speed
- E.g.: k-means:

query $\eta(q)$



Centroids

$\phi\,(\eta(q),\, \eta(c_1))$

$\phi\,(\eta(q),\, \eta(c_2))$

$\phi\,(\eta(q),\, \eta(c_3))$

$\phi\,(\eta(q),\, \eta(d_1))$

$\dots$

$\phi\,(\eta(q),\, \eta(d_m))$

- In practice, ANN implementations are more complicated
- We assume a fast dense retrieval library is available (e.g.: Faiss, Annoy, ScaNN)

Distance-based Transformer Representations

Distance-based Representations

Key characteristic

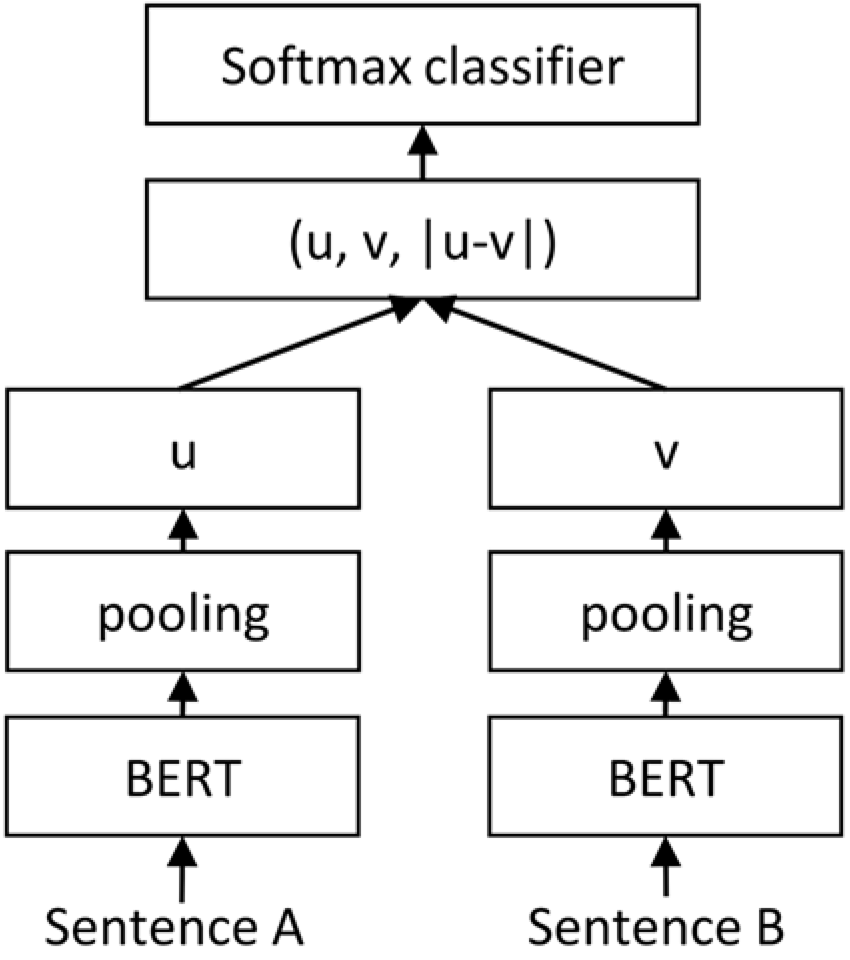
Simple similarity function $\rightarrow$ inner (dot) product, cosine similarity, …

$$\phi(u, v) = \eta(u) \cdot \eta(v)$$

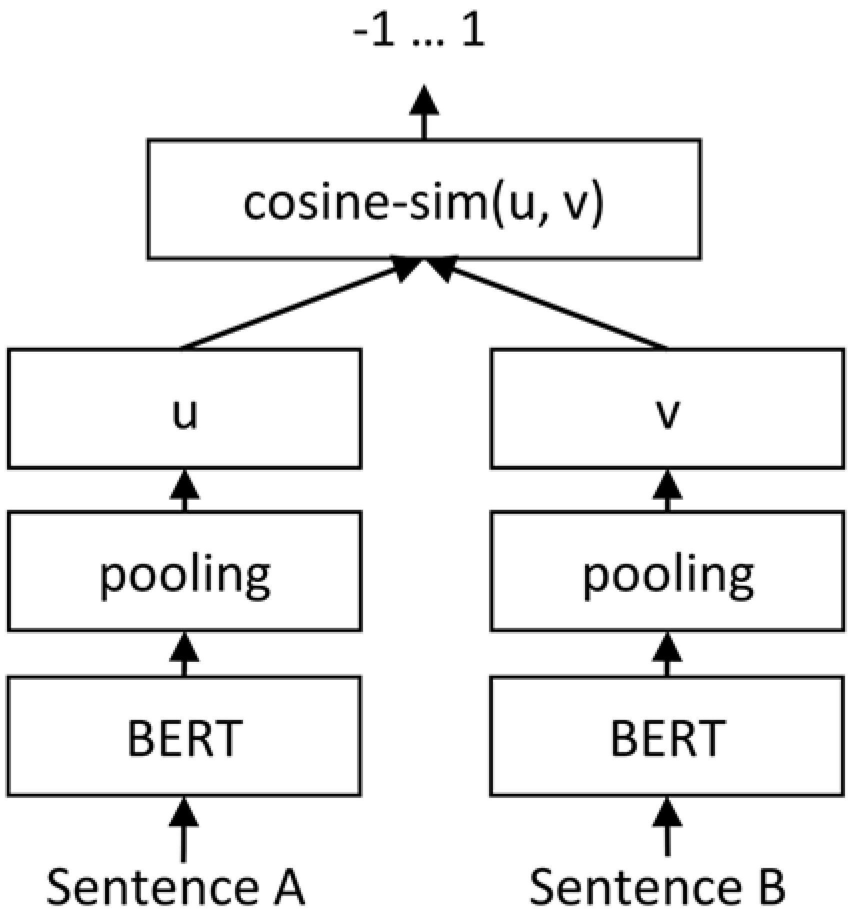
Compatible with ANN search

Johnson, Douze, Jégou. Billion-scale similarity search with GPUs. arXiv 2017.

Distance-based: SentenceBERT



Classification

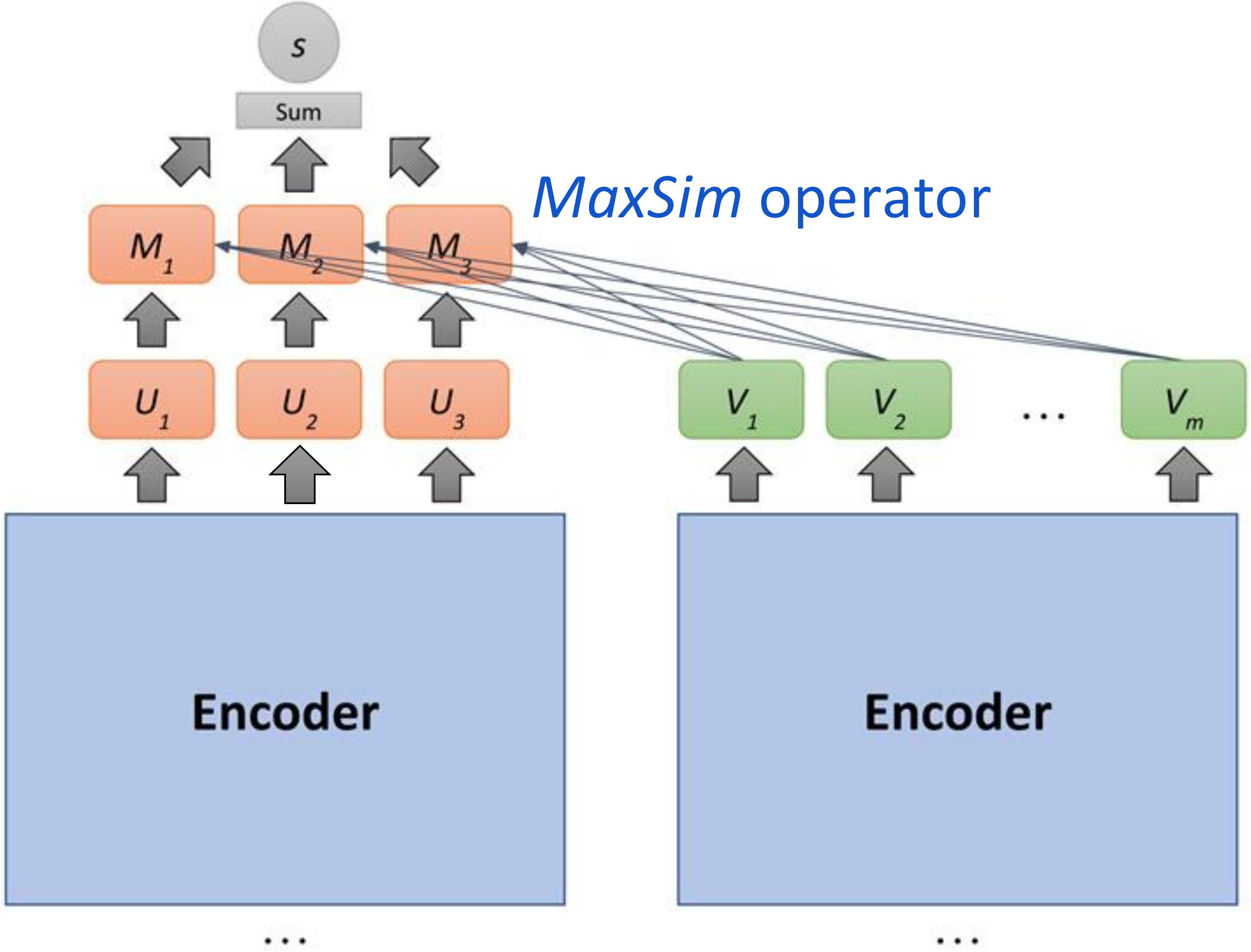


Regression

Reimers, Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP 2019.

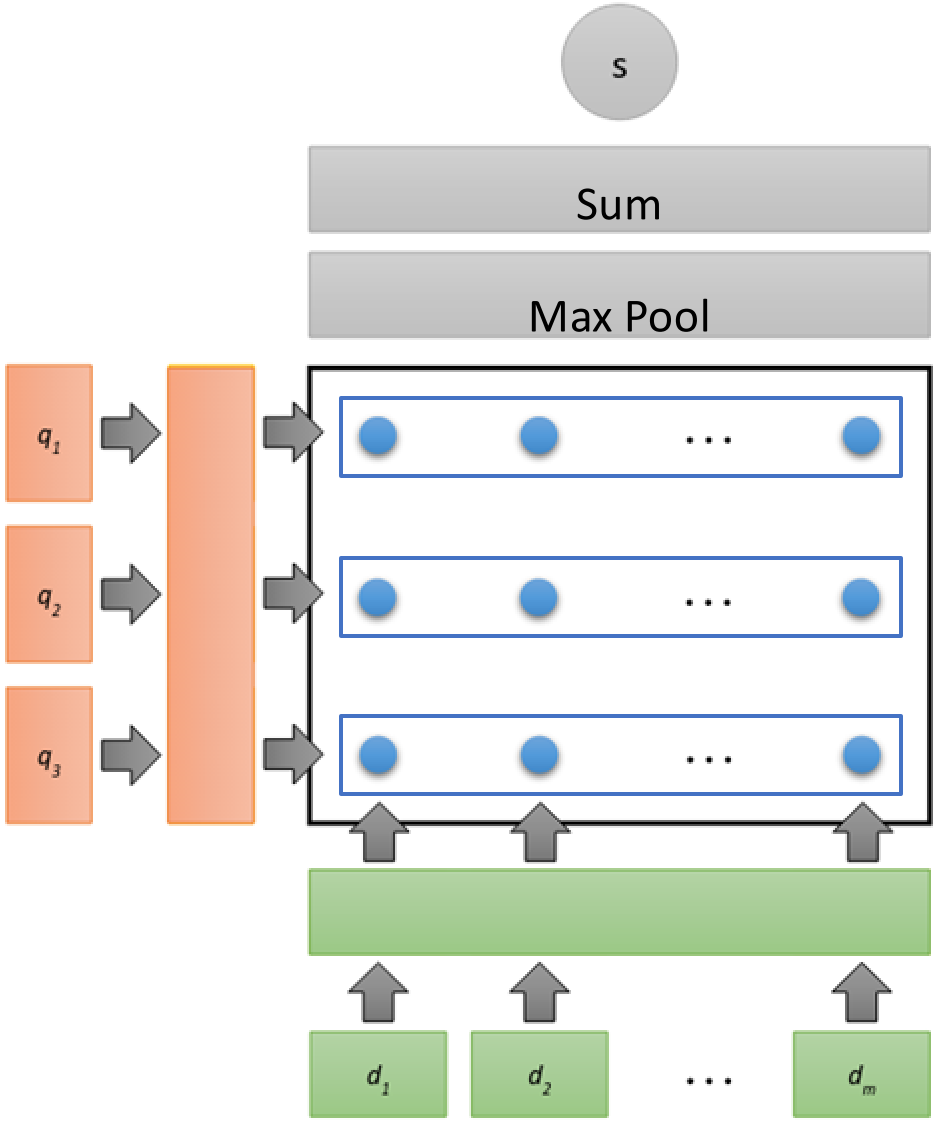
Comparison-based Transformer Representations

Comparison-based: ColBERT



Khattab, Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. SIGIR 2020.

Comparison-based: ColBERT



MaxSim:
Sim-mat max pooling
(along query dimension)

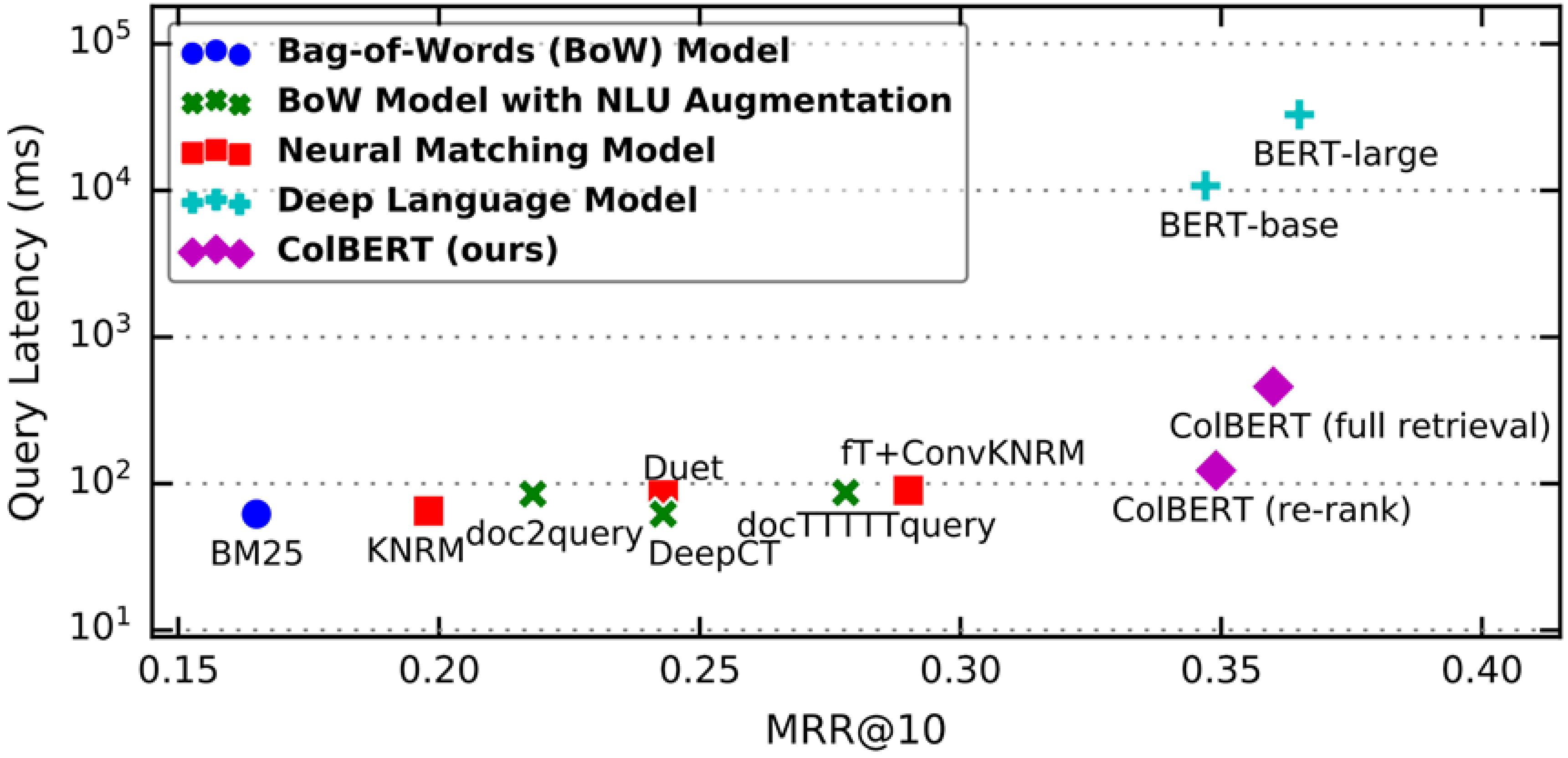
$$s_{q,d} = \sum_{i \in \eta(q)} \max_{j \in \eta(d)} \eta(q)_i \cdot \eta(d)_j^{\mathrm{T}}$$

Khattab, Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. SIGIR 2020.

Comparison-based: ColBERT

Compatible with ANN?

- Unclear
- Data-dependent
- 70x faster than BERT-large



Khattab, Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. SIGIR 2020.

Comparison-based: ColBERT

Method		MS MARCO Passage		
		Development		Latency
		MRR@10	Recall@1k	(ms)
(1a)	BM25 (Anserini, top 1000)	0.187	0.861	62
(1b)	+ monoBERT$_{\text{Large}}$	0.374	0.861	32,900
(2)	FastText + ConvKNRM	0.290	-	90
(3)	doc2query–T5	0.277	0.947	87
(4)	ColBERT (over BERT$_{\text{Base}}$)	0.360	0.968	458

Document Preprocessing Techniques

Query vs document expansion
doc2query
DeepCT
DeepImpact

Query reformulation as a translation task

Query Language → Query Reformulator → Document Language

Hard: Input has little information

Query Language ← Document Translator ← Document Language

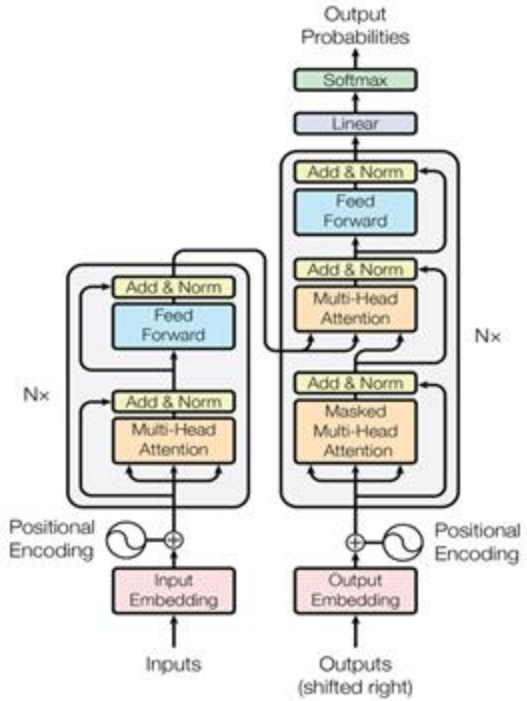
Easier: Input has a lot of information

doc2query

Document → seq2seq Transformer → Query

Supervised training:

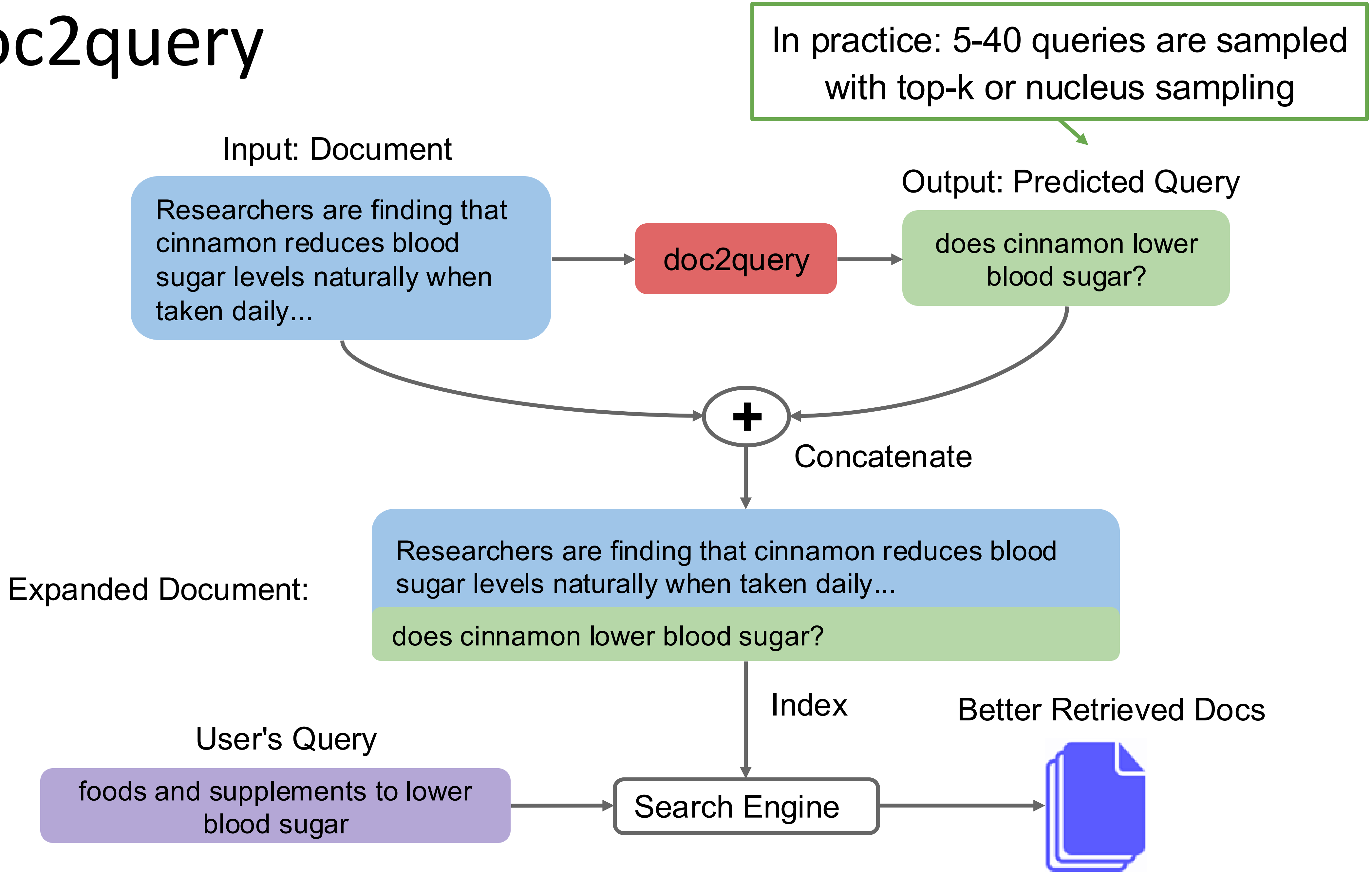
pairs of <query, relevant document>



Source: Vaswani et al., 2017

Nogueira, Yang, Lin, Cho. Document expansion by query prediction. 2019.

doc2query


In practice: 5-40 queries are sampled with top-k or nucleus sampling
Input: Document
Researchers are finding that cinnamon reduces blood sugar levels naturally when taken daily...
doc2query
Output: Predicted Query
does cinnamon lower blood sugar?
+
Concatenate
Expanded Document:
Researchers are finding that cinnamon reduces blood sugar levels naturally when taken daily...
does cinnamon lower blood sugar?
Index
Better Retrieved Docs
User's Query
foods and supplements to lower blood sugar
Search Engine

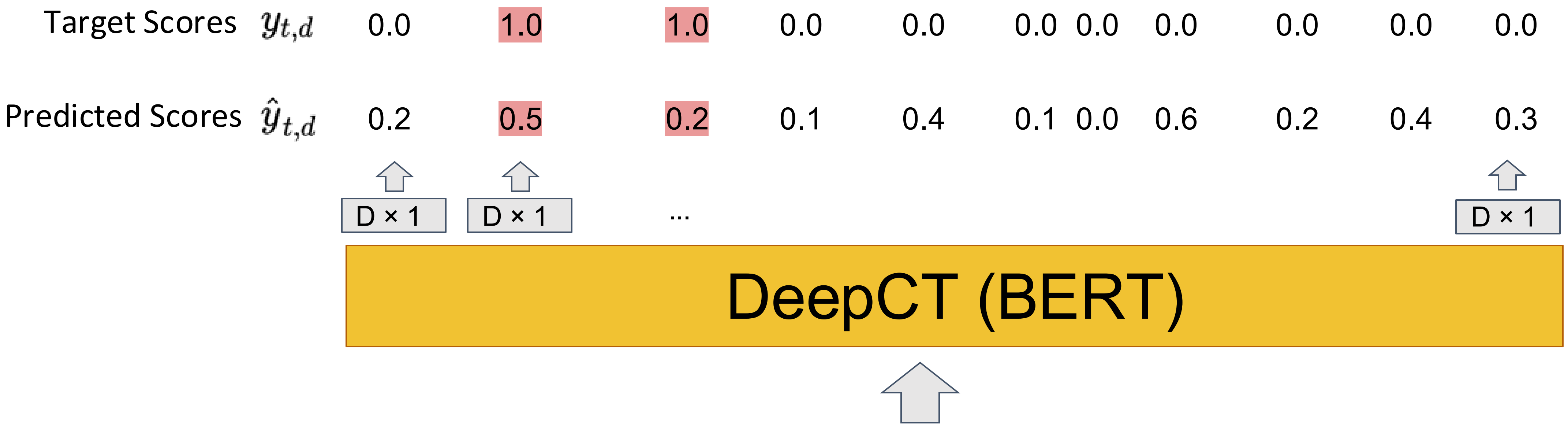
Results

	MARCO Passage (MRR@10)	TREC-DL 19 (nDCG@10)	TREC-COVID (nDCG@20)	Robust04 (nDCG@20)
BM25	0.184	0.506	**0.659**	0.428
+ doc2query	**0.277**	**0.642**	0.6375	**0.446**

zero-shot: doc2query was trained only on MS MARCO

DeepCT

$$\text{loss} = \sum_t (\hat{y}_{t,d} - y_{t,d})^2$$

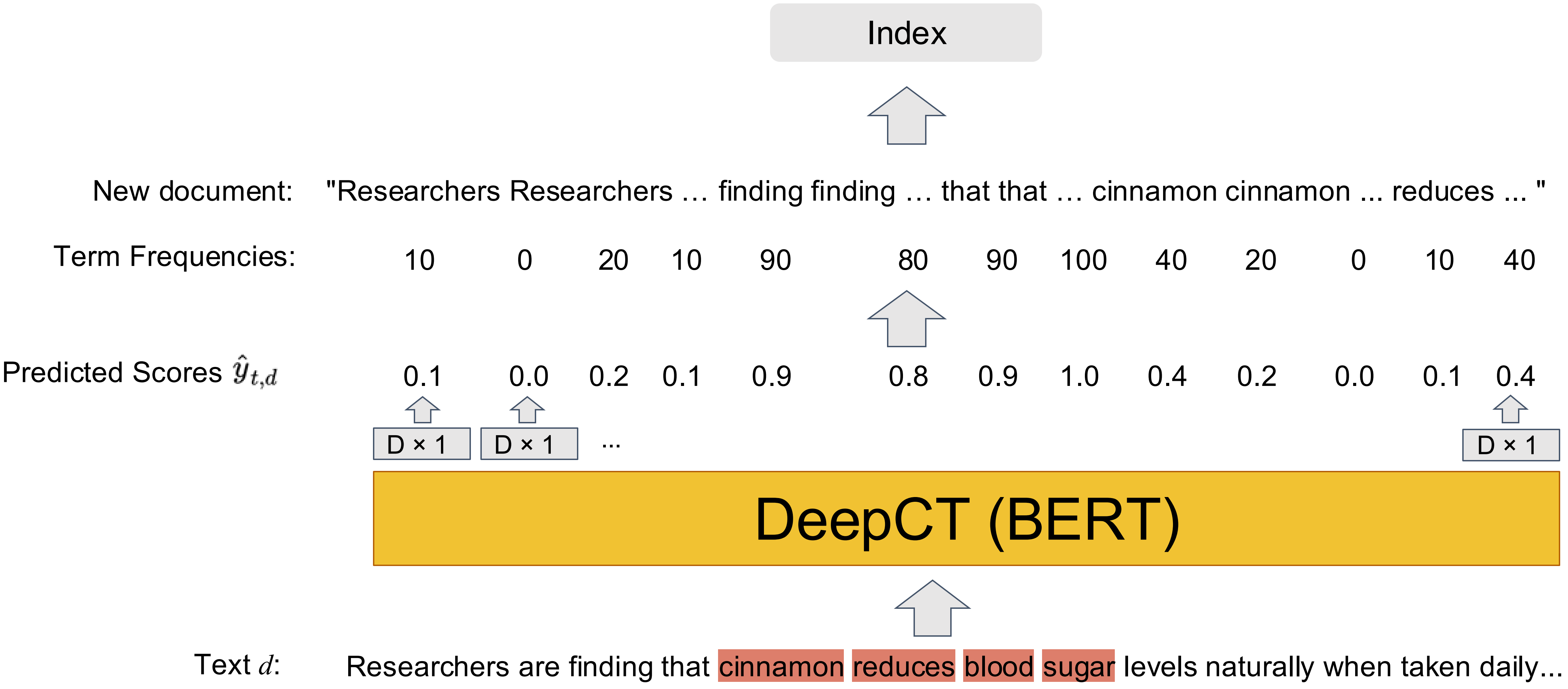


Text d: The Geocentric Theory was proposed by the greeks under the guidance...

Relevant query q: "who proposed the geocentric theory"

Dai, Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. 2019.

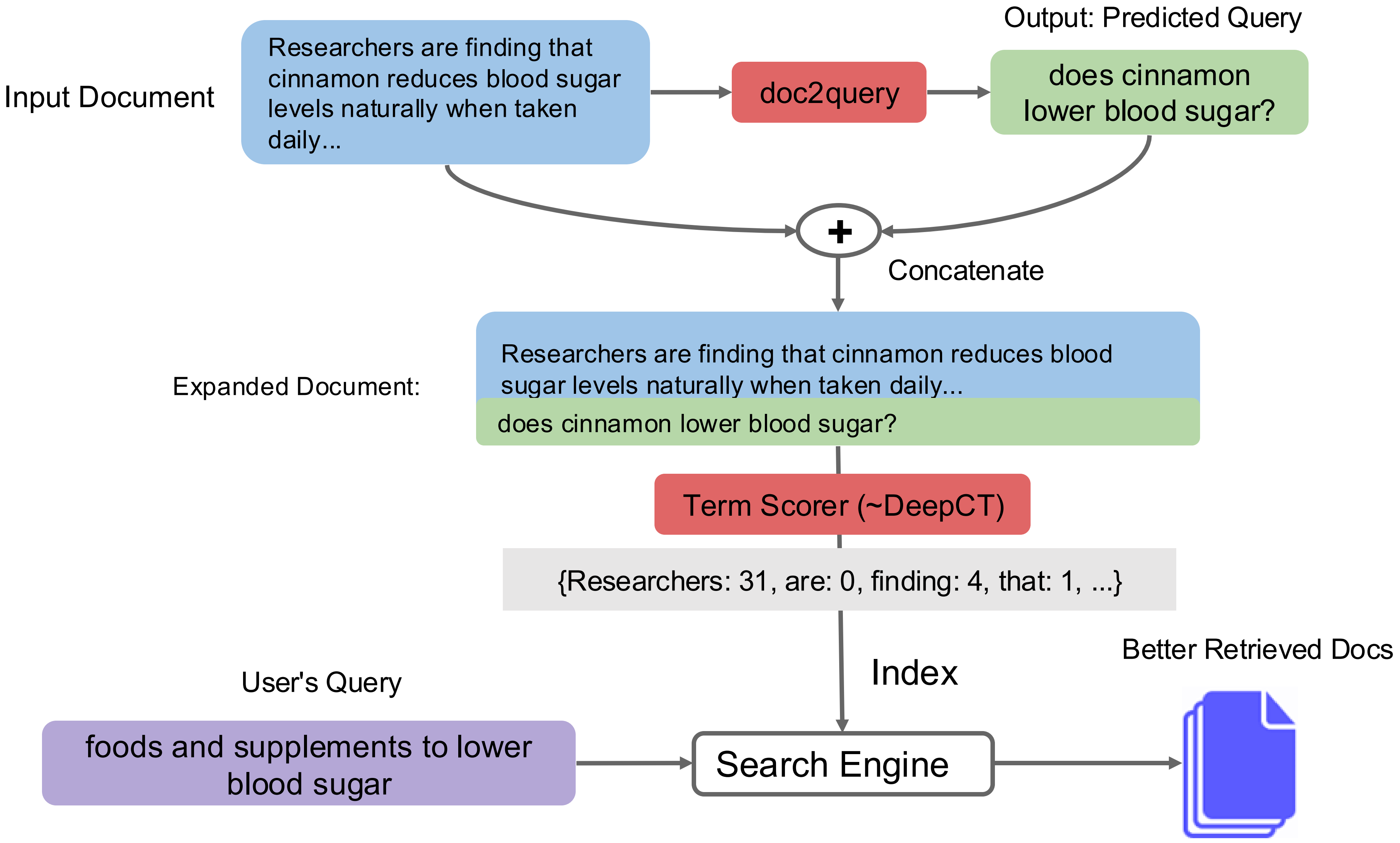
Once DeepCT is trained...



Results on MS MARCO Passage Dev Set

Model	MRR@10	R@1000	BERT Inferences per doc
BM25	0.184	0.853	-
+ doc2query	0.229	0.907	1
+ doc2query	0.277	0.944	40
DeepCT	0.243	0.913	1

DeepImpact: combining doc2query with DeepCT



Mallia, Khattab, Tonellotto, and Suel. Learning Passage Impacts for Inverted Indexes. 2021

Results on MS MARCO Passage Dev Set

Model	MRR@10	R@1000	Latency (ms/query)
BM25	0.184	0.853	13
DeepCT	0.243	0.913	11
doc2query	0.278	0.947	12
DeepImpact	0.326	0.948	58
BM25 + monoBERT	0.355	0.853	(GPU) 10,700

Takeaways of Document Expansion

Advantages:

- Documents have more context than queries →easy prediction task
- Documents can be processed offline *and* in parallel
- Run on CPU at query time

Disadvantages:

- Have to iterate over the entire collection
- Not as effective as rerankers (yet)

Conclusions and Future Directions

Conclusions

- Pretrained Transformers showed significant improvements in various IR benchmarks
- Reproduced and adopted by many in academia and industry
- No doubt we are in the age of BERT and Transformers

Learn more in survey (& upcoming book):

Pretrained Transformers for Text Ranking:
BERT and Beyond

by Jimmy Lin, Rodrigo Nogueira, and Andrew Yates
https://arxiv.org/abs/2010.06467

Thanks!