UMassAmherst | College of Information & Computer Sciences
Center for Intelligent Information Retrieval

Efficient Neural Models for Representing, Indexing, and Retrieving Documents

Hamed Zamani

Center for Intelligent Information Retrieval
University of Massachusetts Amherst

zamani@cs.umass.edu

These efficient contextual models have delivered the **largest quality improvement** to the Bing search results in **the last 6 years**!

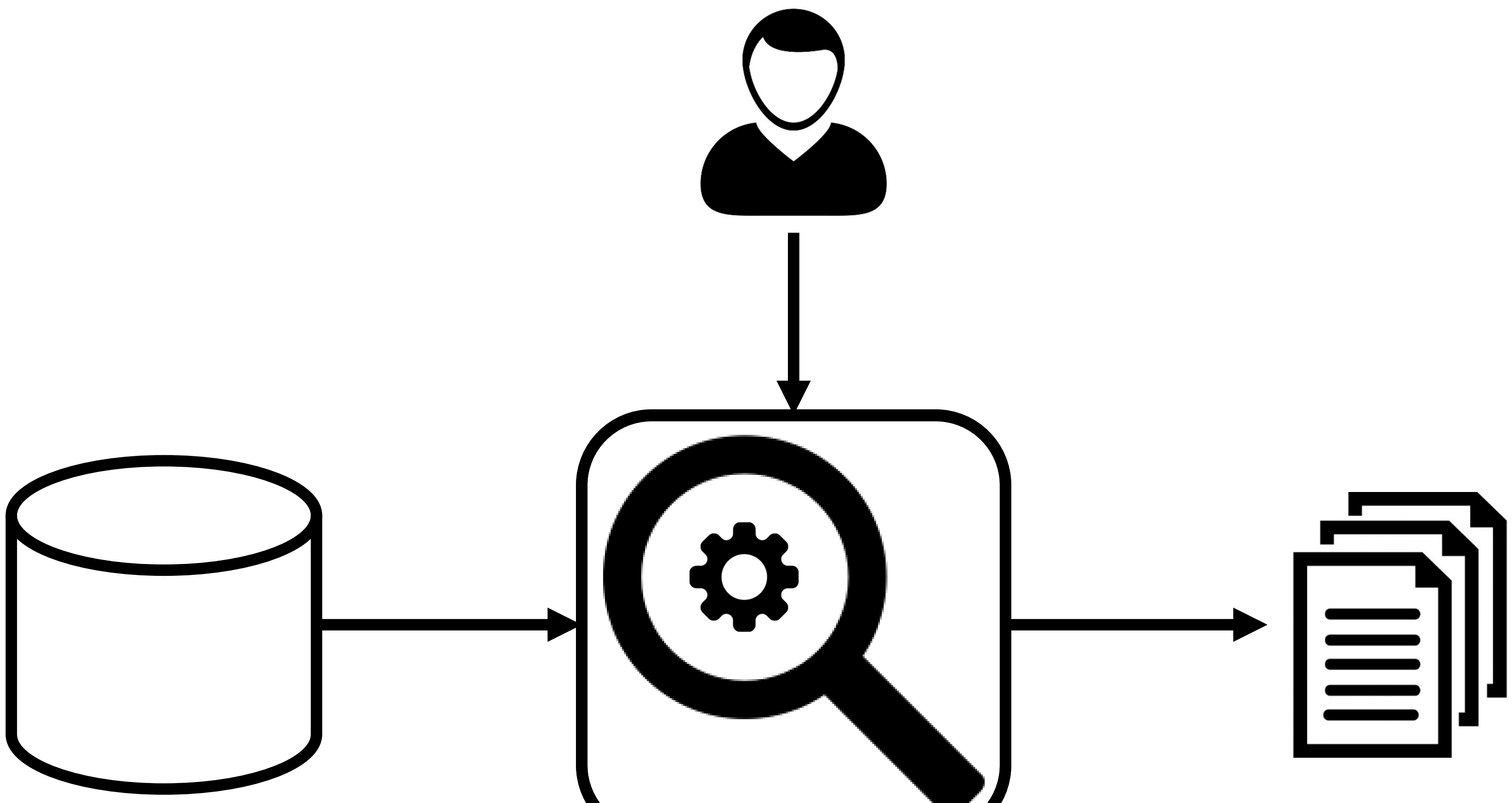
https://tinyurl.com/tzhj3o8

In this talk…

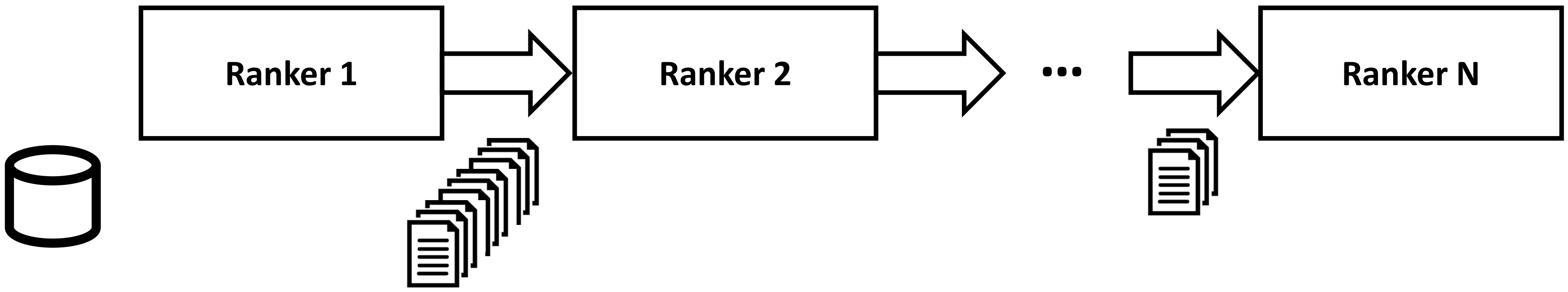
How to efficiently represent long documents?

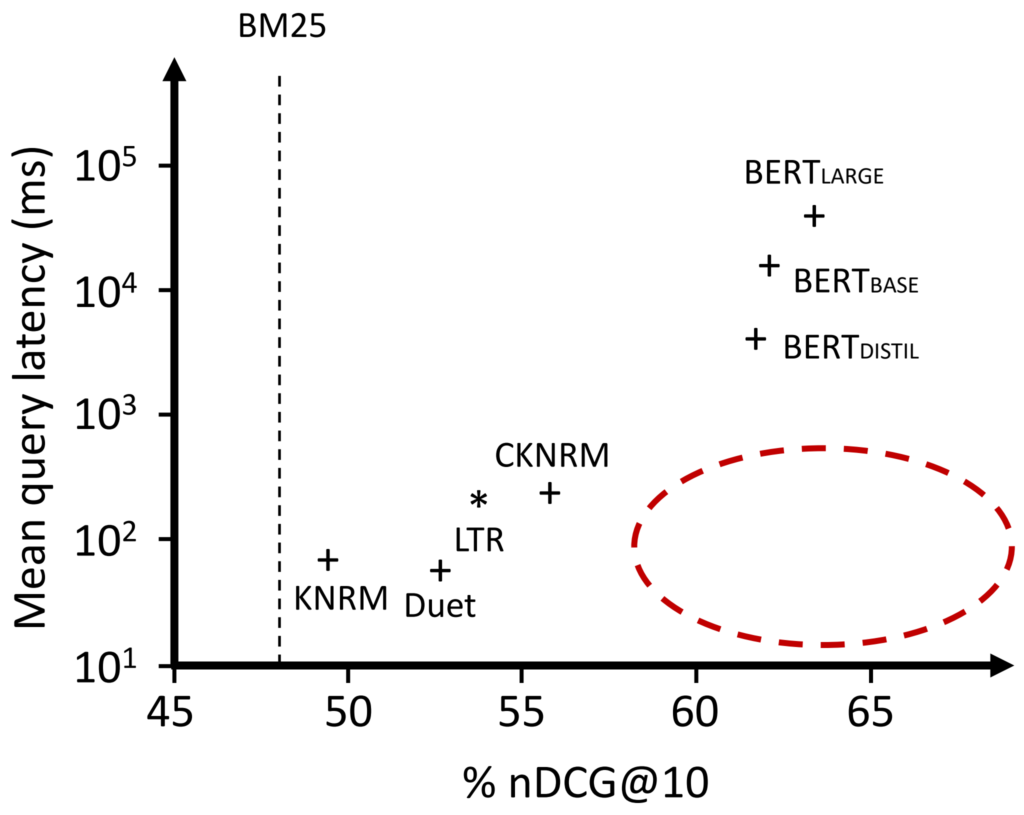
How to efficiently retrieve documents?

How can efficient neural IR advance machine learning research?



Multi-Stage Cascaded Retrieval



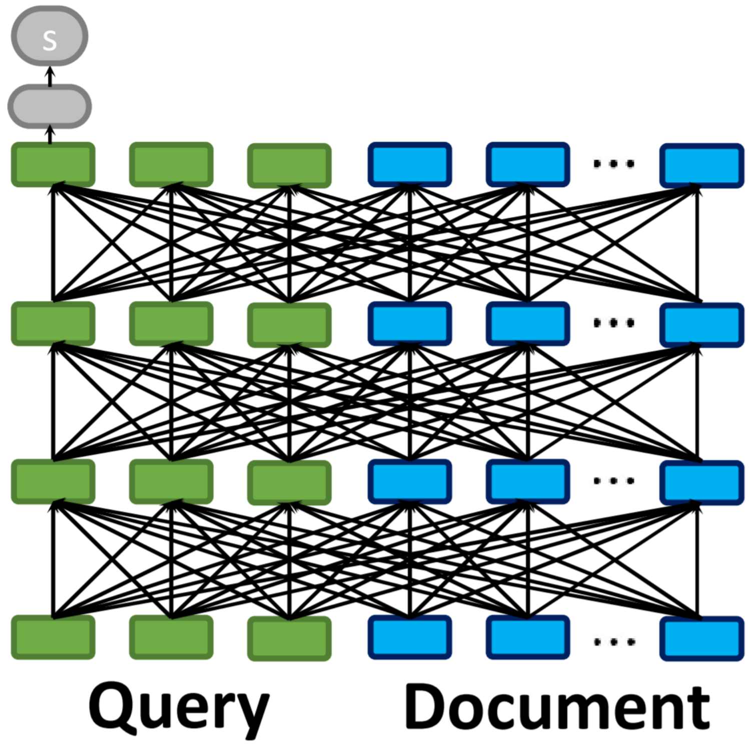


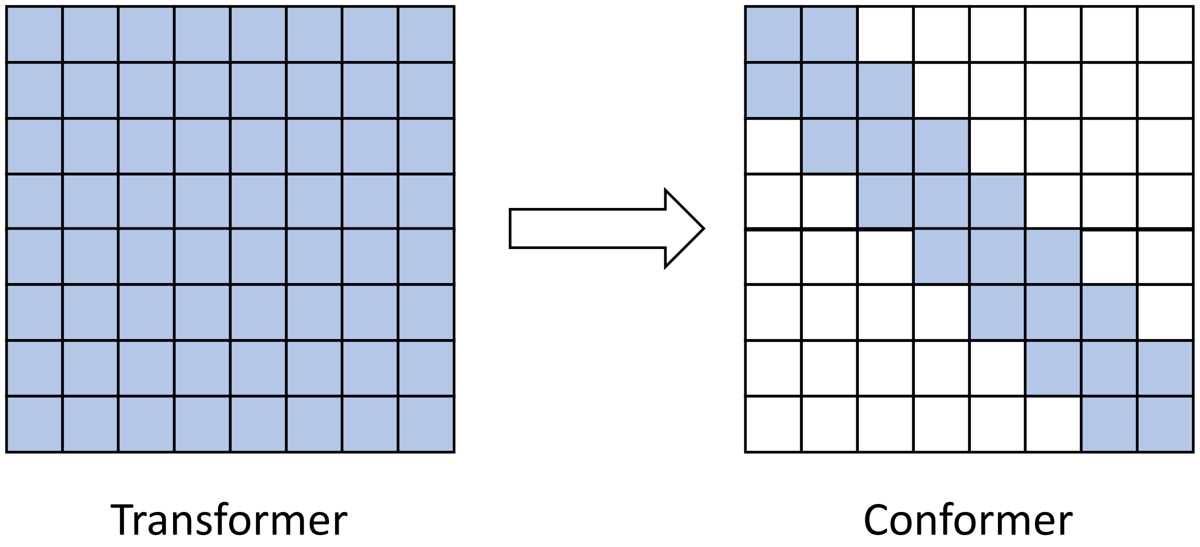
TREC Deep Learning Track 2019 – Document Ranking

BERT for Document Ranking

Reduce online computations

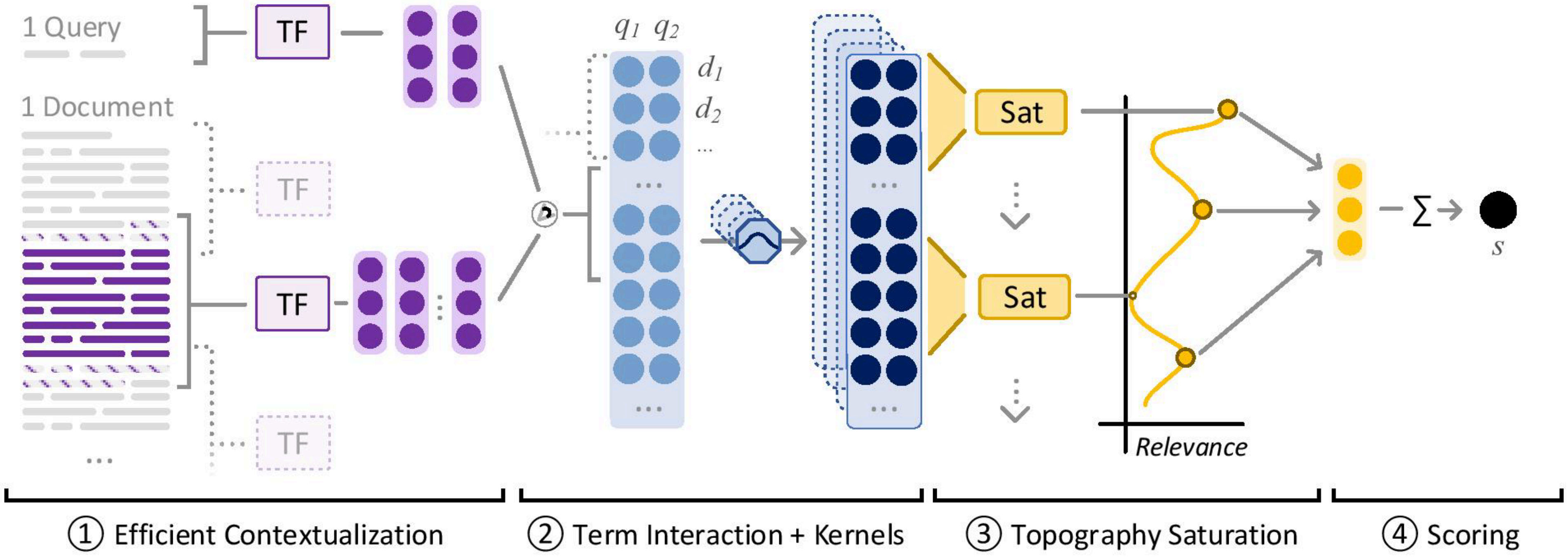
Reduce the quadratic complexity of self-attention in Transformer





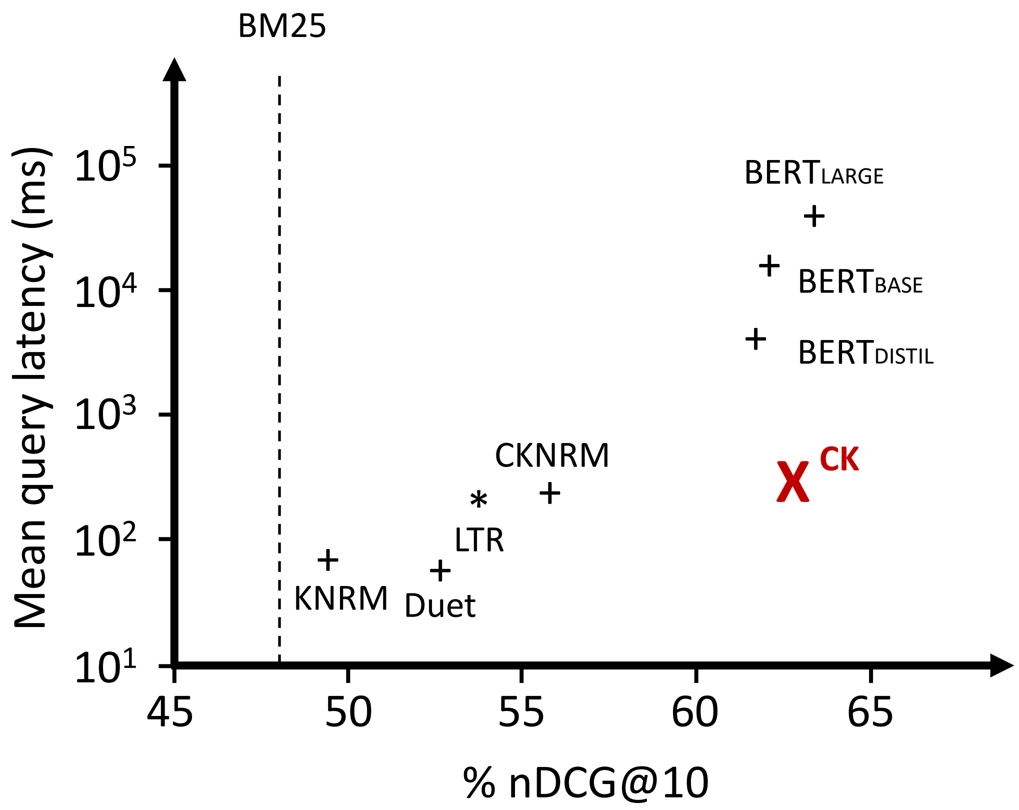
Linformer, Longformer, Sparse Transformer, etc.

Conformer-Kernel



[Hofstätter, Zamani, Mitra, Craswell, Hanbury; SIGIR 2020]
[Mitra, Hofstätter, Zamani, Craswell; SIGIR 2021]

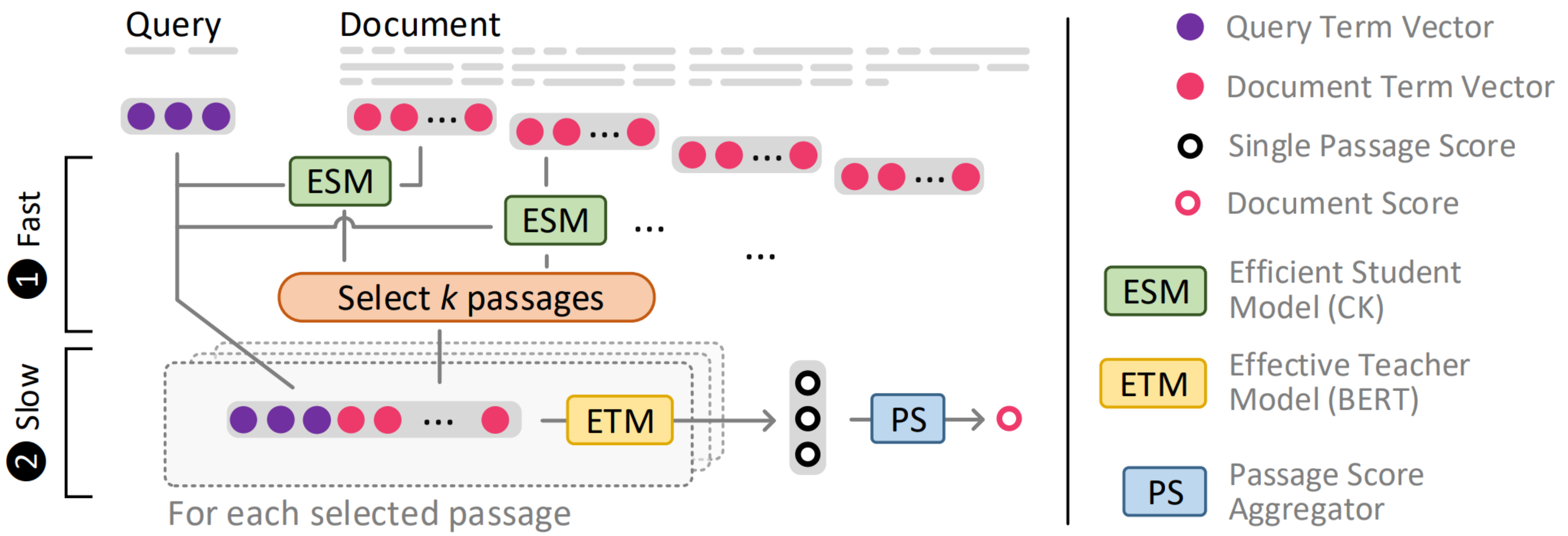
Results



2020

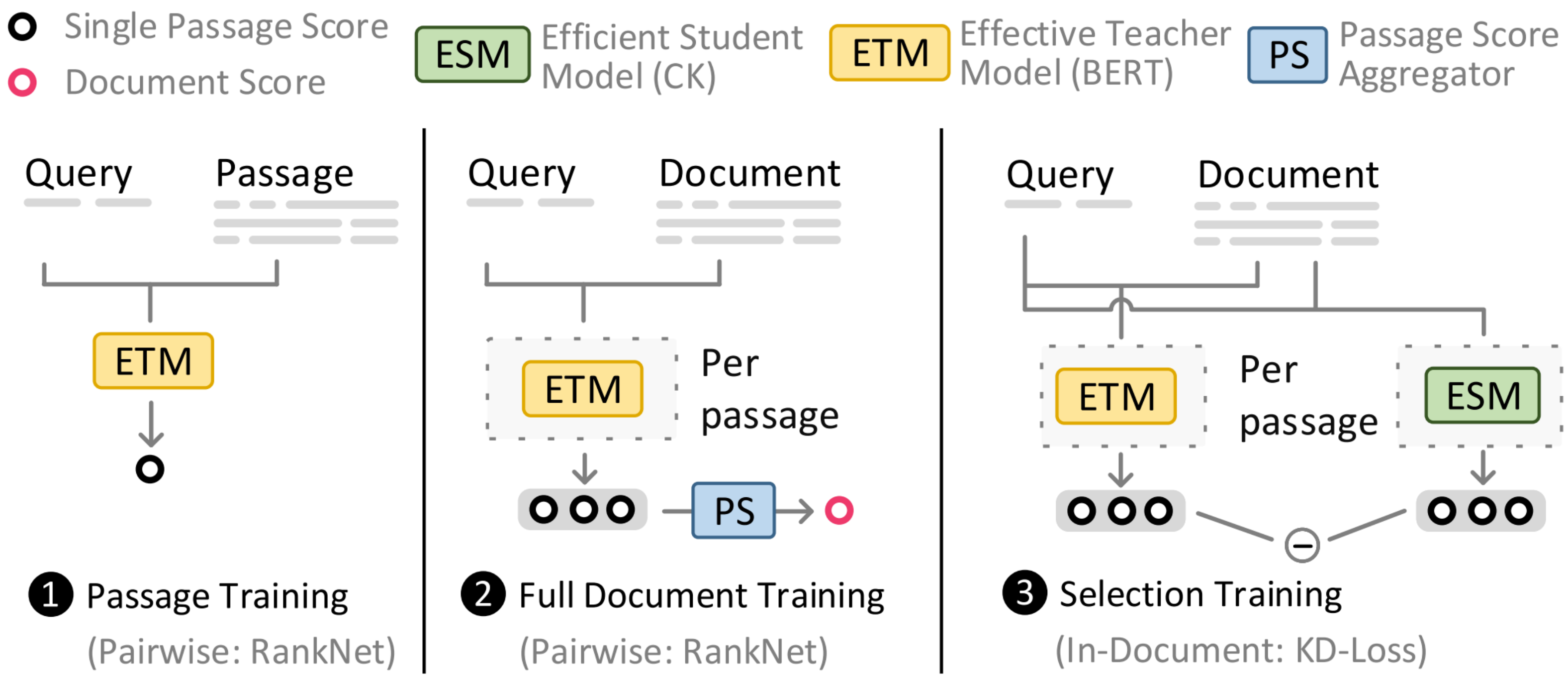
TREC Deep Learning Track ~~2019~~ – Document Ranking

Intra-Document Cascading Model (IDCM)



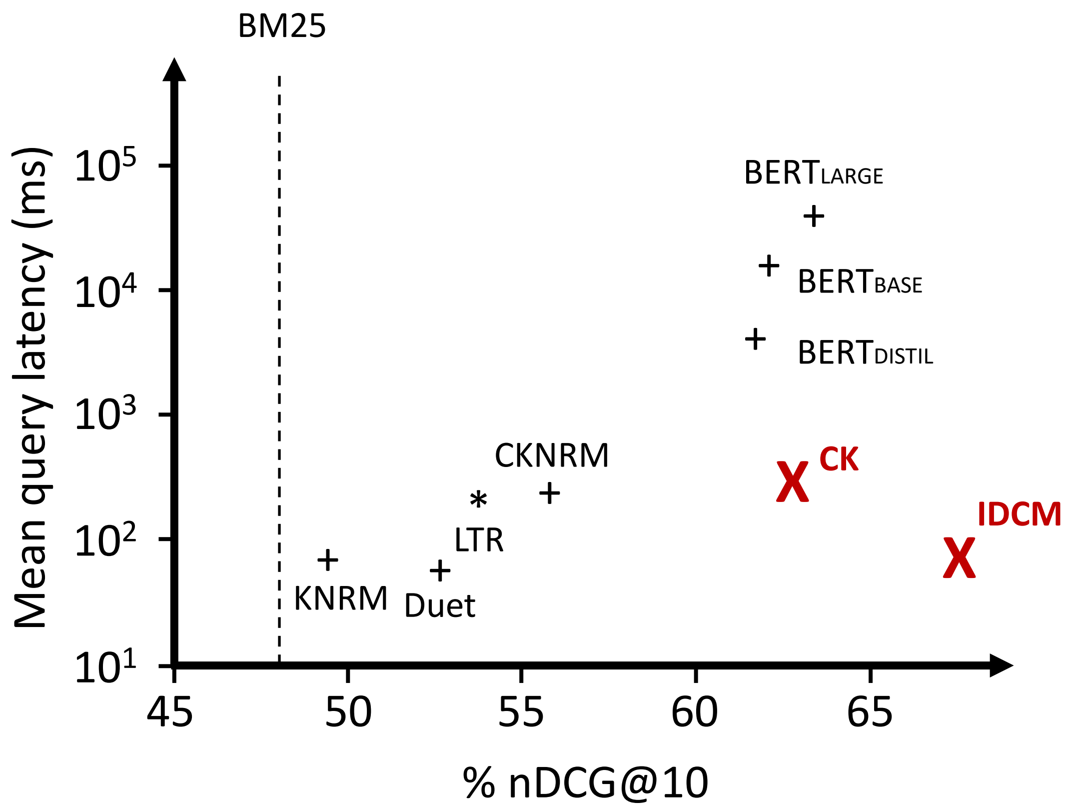
[Hofstätter, Mitra, Zamani, Craswell, Hanbury; SIGIR 2021]

Training IDCM



[Hofstätter, Mitra, Zamani, Craswell, Hanbury; SIGIR 2021]

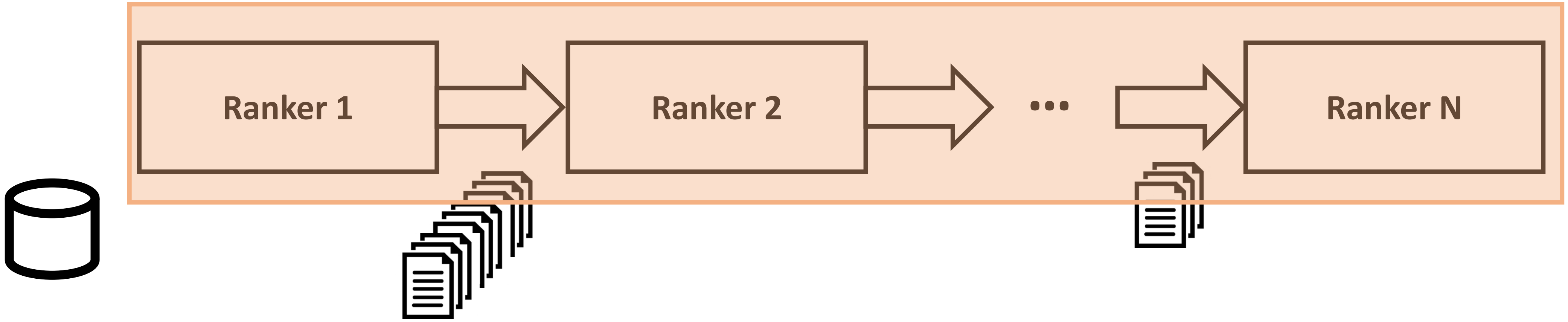
Results



TREC Deep Learning Track 2019 – Document Ranking

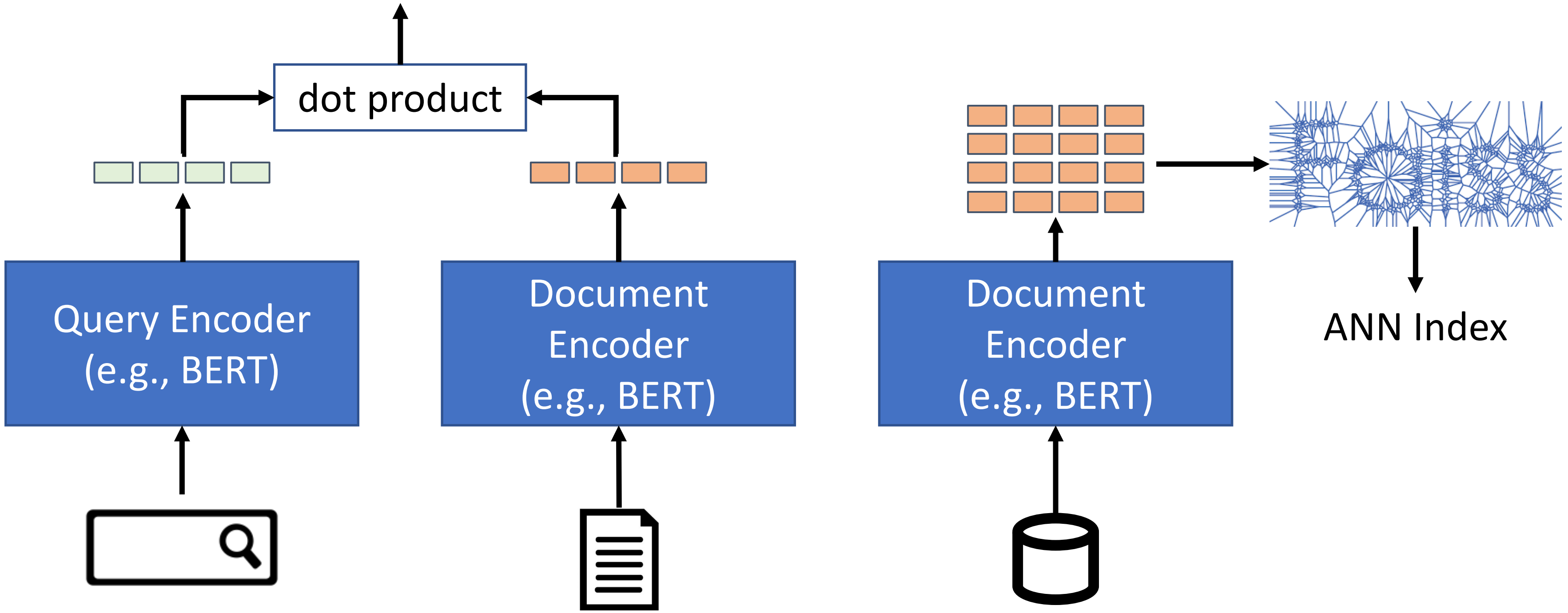
How to efficiently retrieve documents?

Multi-Stage Cascaded Retrieval

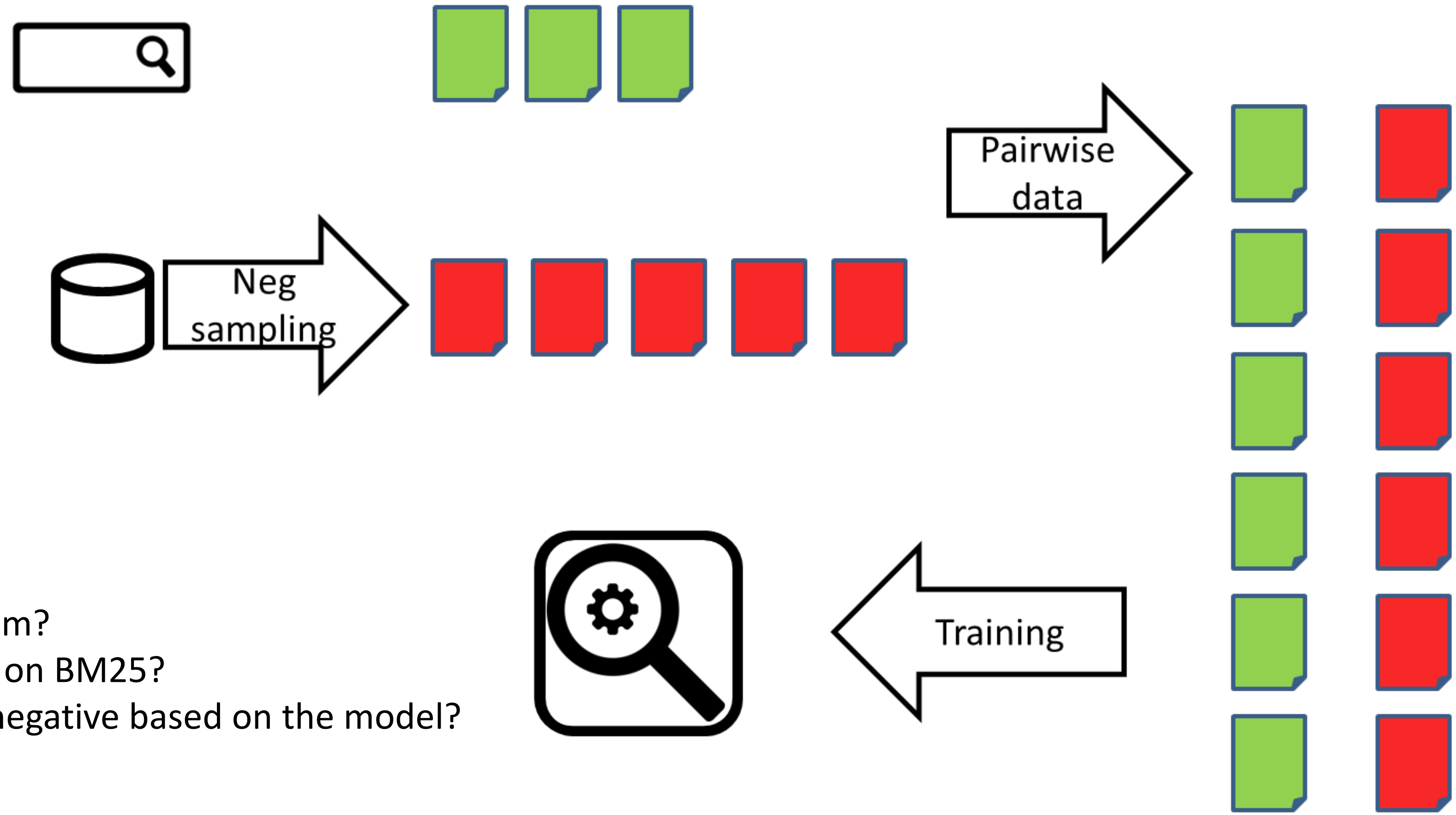


Can we replace multi-stage cascaded architectures with standalone retrieval models?

Dense Retrieval

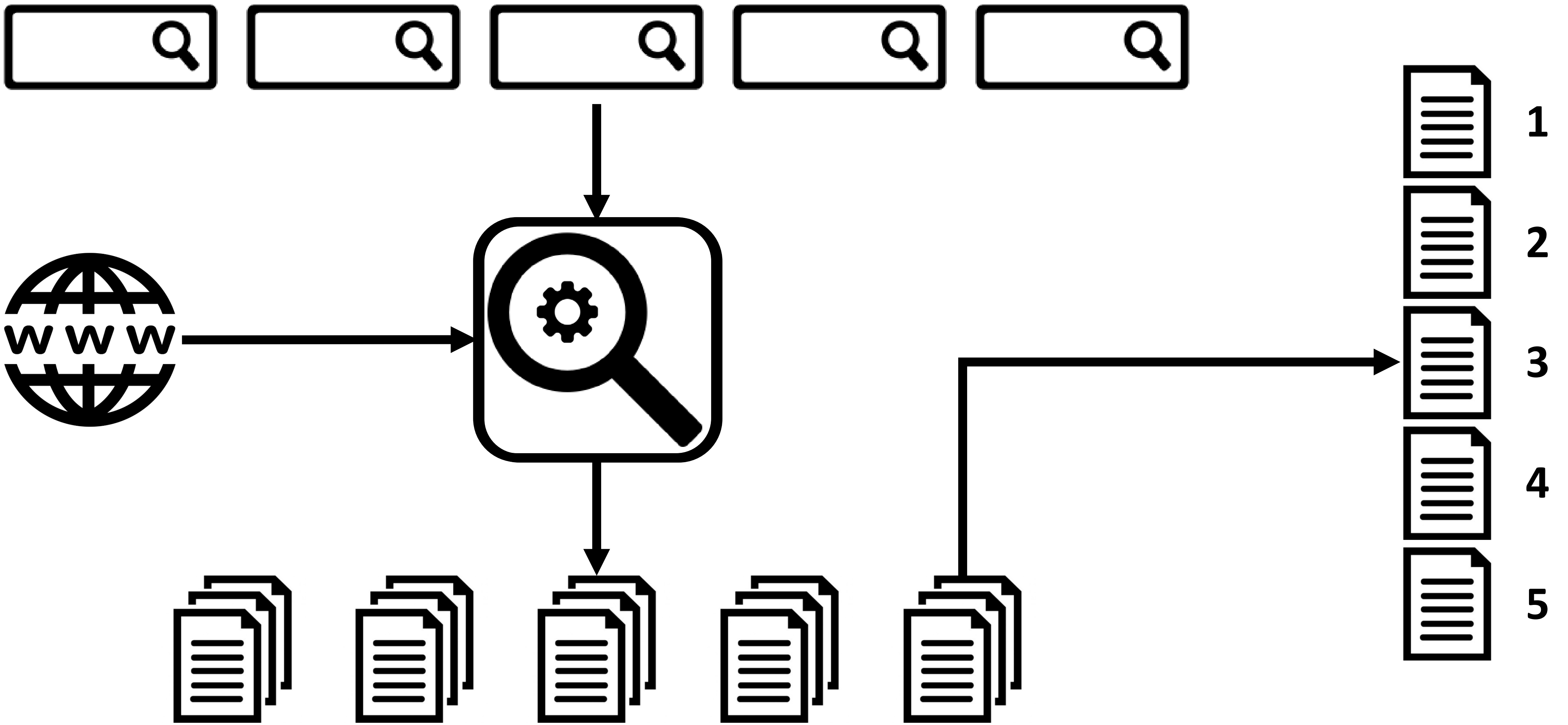


Negative Sampling for Neural IR



- Random?
- Based on BM25?
- Hard negative based on the model?

Teacher-Student Learning:
A Common Recipe for Training Dense Retrieval Models



[Dehghani, Zamani, Severyn, Kamps, Croft, SIGIR ’17]

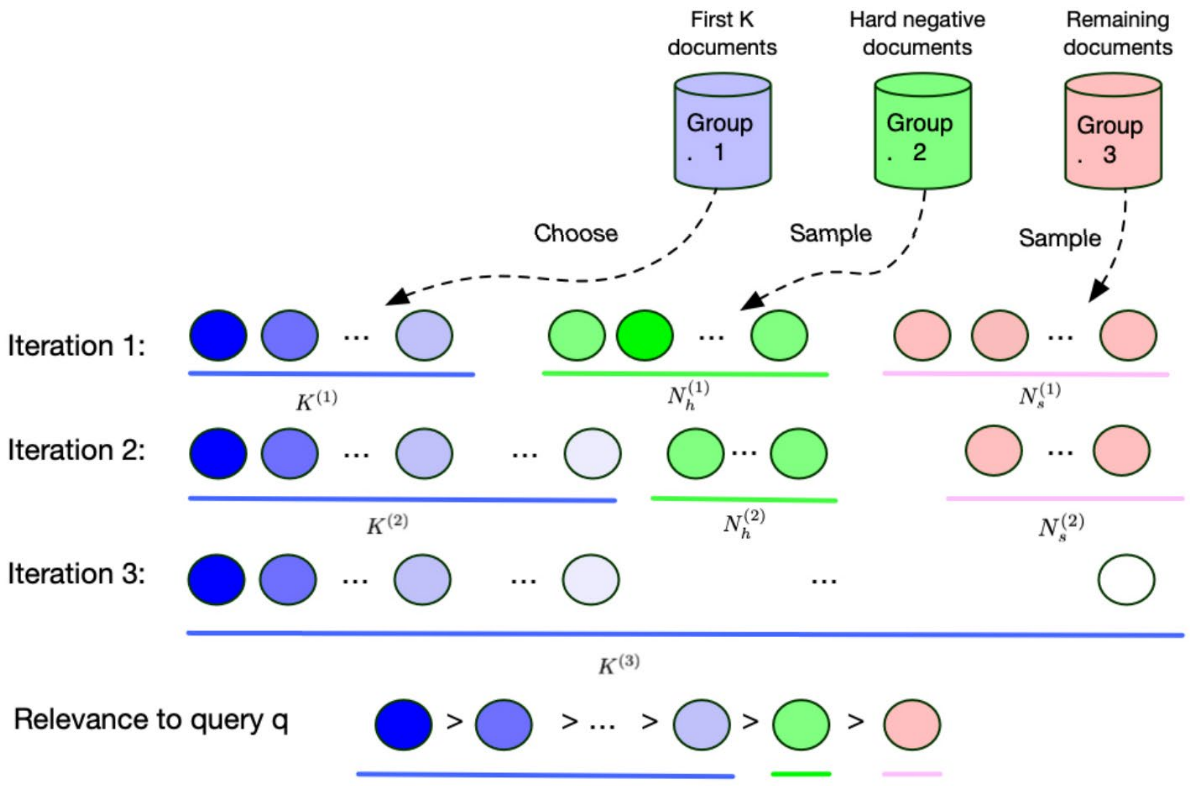
Curriculum Learning for Dense Retrieval Distillation

Algorithm 1 The Iterative Optimization Process in CL-DRD.

Input (a) training query set Q; (b) document collection C; (c) *optional* relevance judgment set R; (d) teacher model $\widehat{M}$.
Output dense retrieval model M_θ.
Initialize θ.
$\delta \leftarrow 1$ ▷ δ denotes the difficulty level in CL data
repeat
    $D_\delta \leftarrow \text{RankingDataGeneration}(\delta; \widehat{M}, Q, C, R)$
    $\theta^* \leftarrow \arg\min_\theta \mathcal{L}_{KD}(M_\theta, D_\delta)$
    $\delta \leftarrow \delta + 1$
until stopping criterion is met
return M_{θ^*}

[Zeng, Zamani, Vinay; SIGIR 2022]

Curriculum Learning for Dense Retrieval Distillation

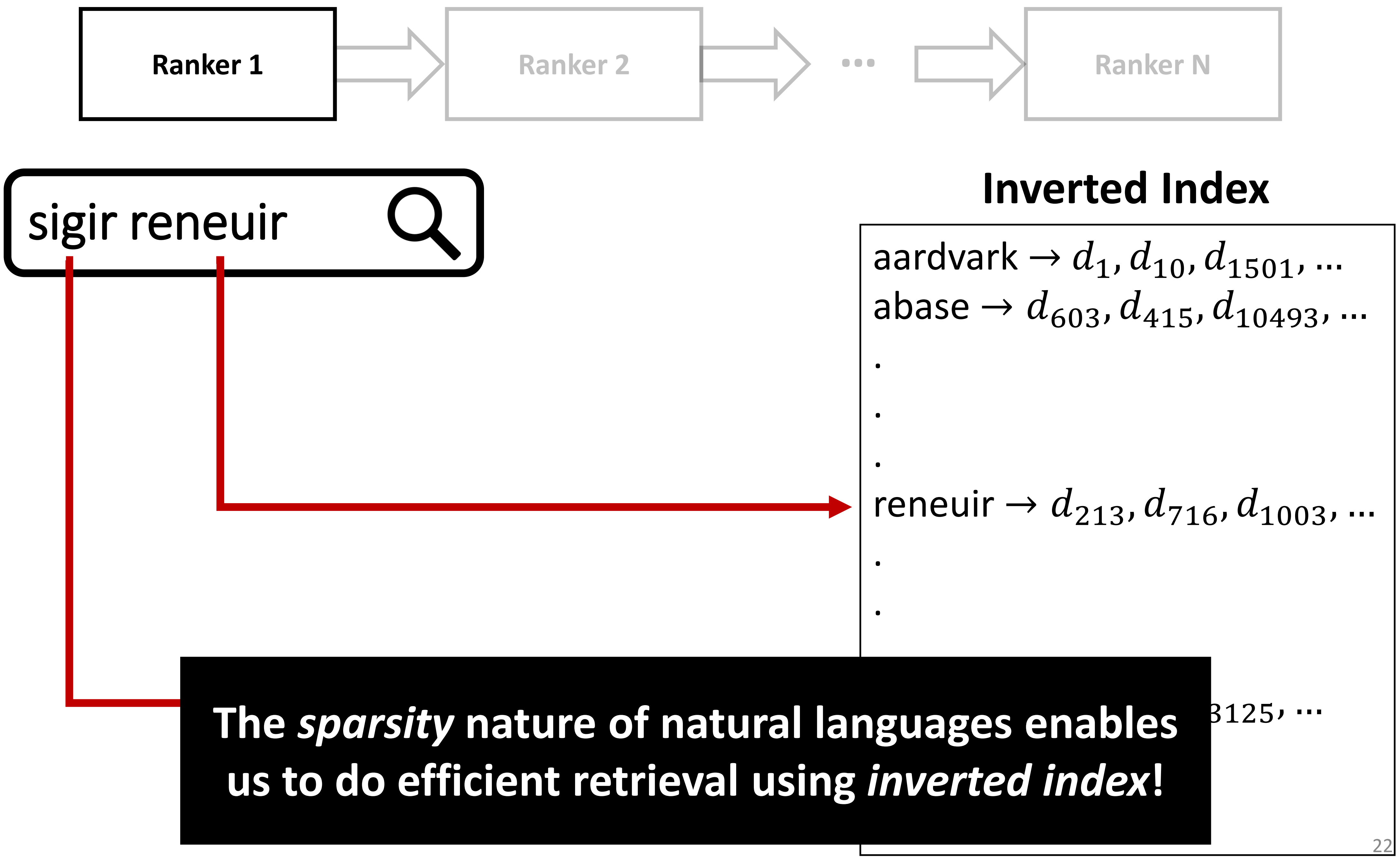


[Zeng, Zamani, Vinay; SIGIR 2022]

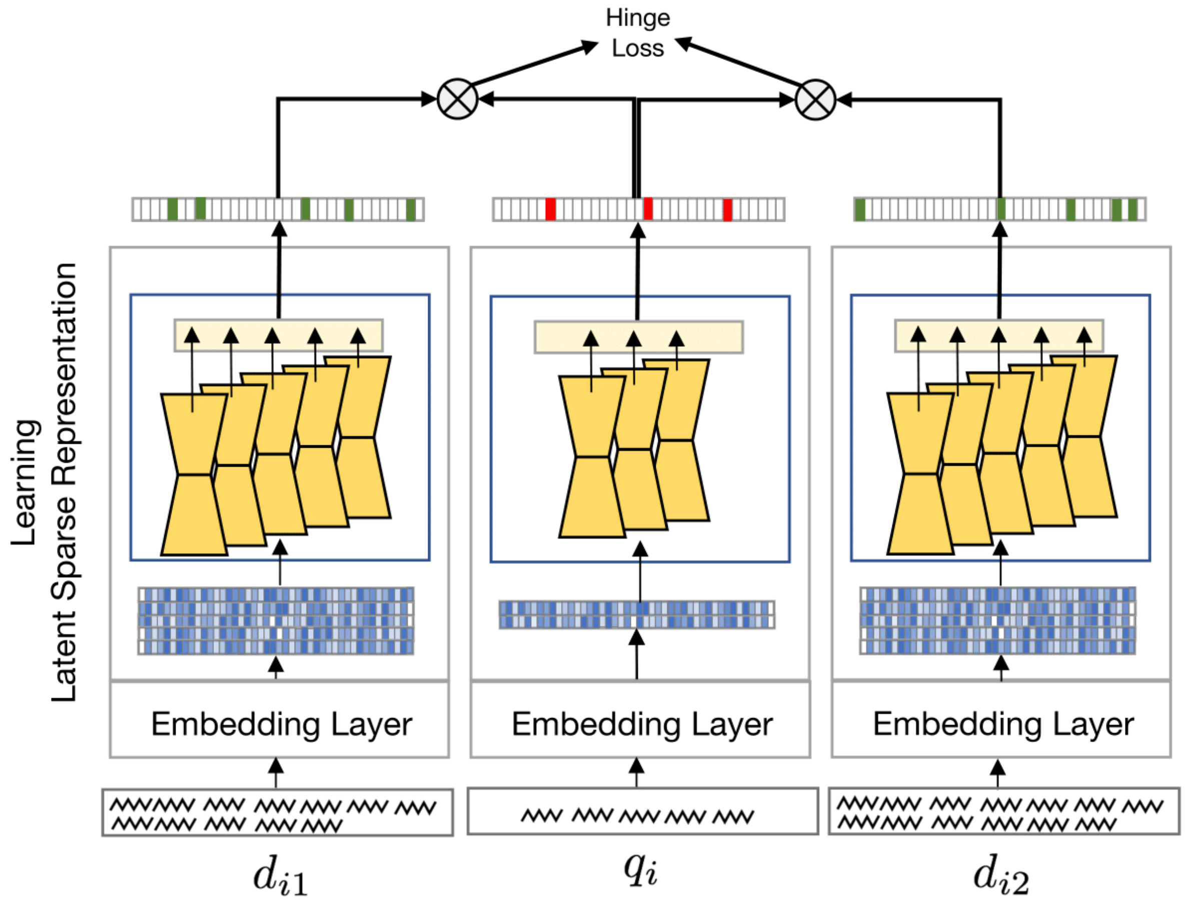
Results

Model	KD	Encoder	#params	MS MARCO DEV MRR@10	MS MARCO DEV MAP@1k	TREC-DL'19 nDCG@10	TREC-DL'19 MAP@1k	TREC-DL'20 nDCG@10	TREC-DL'20 MAP@1k
Sparse Retrieval									
BM25 [27]	-	-	-	.187	.196	.497	.290	.487	.288
DeepCT [6]	-	-	-	.243	.250	.550	.341	.556	.343
docT5query [20]	-	-	-	.272	.281	.642	.403	.619	.407
Multi-Vector Dense Retrieval									
ColBERT [13]	✗	BERT-Base	110M	.360	-	-	-	-	-
ColBERTv2 [30]	✓	BERT-Base	110M	**.397**	-	-	-	-	-
ColBERTv2 [30]	✓	DistilBERT	66M	.384	.389	**.733**	.464	.712	.473
ColBERTv2 + CL-DRD (Ours)	✓	DistilBERT	66M	.394*†‡§	**.398***†‡§¶	.727*†‡§	**.472***†‡§¶	**.717***†‡§	**.487***†‡§¶
Single-Vector Dense Retrieval									
ANCE [32]	✗	BERT-Base	110M	.330	.336	.648	.371	.646	.408
ADORE [33]	✗	BERT-Base	110M	.347	.352	.683	.419	.666	.442
RocketQA [25]	✓	ERNIE-Base	110M	.370	-	-	-	-	-
TCT-ColBERT [16]	✓	BERT-Base	110M	.335	.342	.670	.391	.668	.430
Margin-MSE [10]	✓	DistilBERT	66M	.325	.331	.699	.405	.645	.416
TAS-B [11]	✓	DistilBERT	66M	.344	.351	.717	.447	.685	.455
TAS-B + CL-DRD (Ours)	✓	DistilBERT	66M	**.382***†‡§	**.386***†‡§	**.725***†‡§	**.453***†‡	**.687***†‡	**.465***†‡§

[Zeng, Zamani, Vinay; SIGIR 2022]


Ranker 1
Ranker 2
…
Ranker N
sigir reneuir
Inverted Index
aardvark → $d_1, d_{10}, d_{1501}, \ldots$
abase → $d_{603}, d_{415}, d_{10493}, \ldots$
.
.
.
reneuir → $d_{213}, d_{716}, d_{1003}, \ldots$
.
.
$3125, \ldots$
The *sparsity* nature of natural languages enables us to do efficient retrieval using *inverted index*!

Standalone Neural Retrieval Model

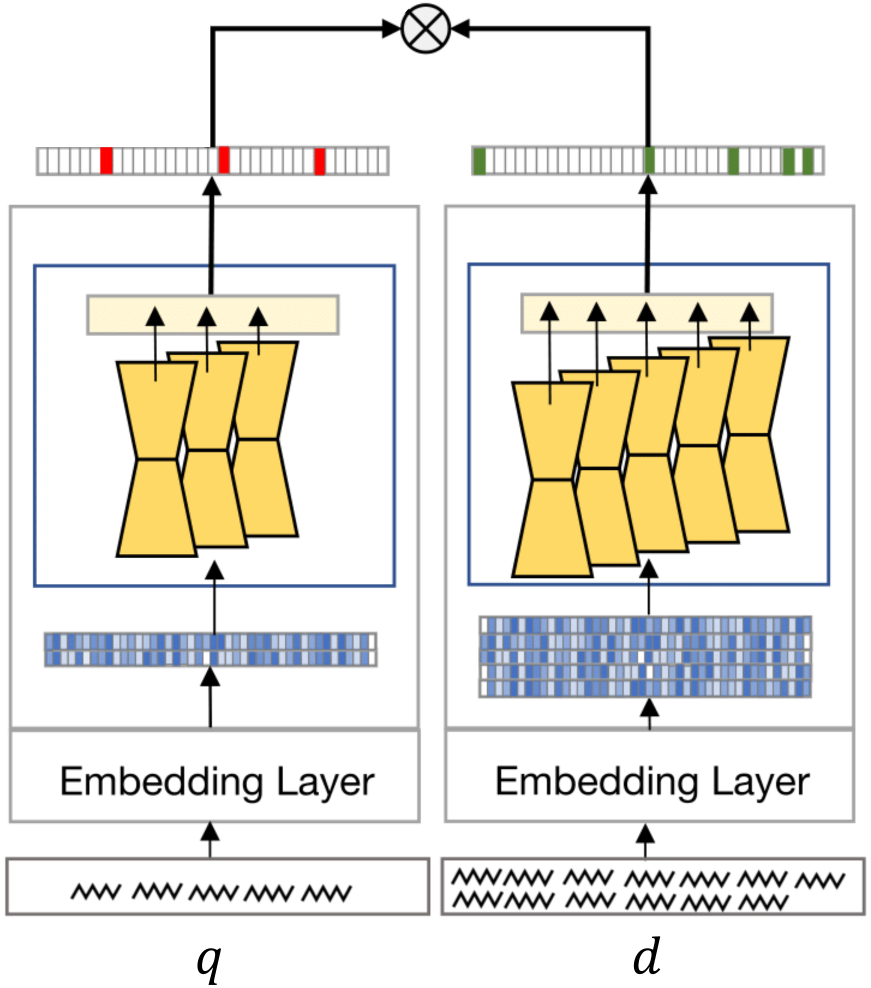


[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

Training

Efficiency -> Sparsity

- Maximizing sparsity ratio, i.e., equivalent to minimizing $L_0(\vec{v}) = \sum_{i=1}^{|\vec{v}|} |\overrightarrow{v_i}|^0$.
- A differentiable approximator for L_0.



[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

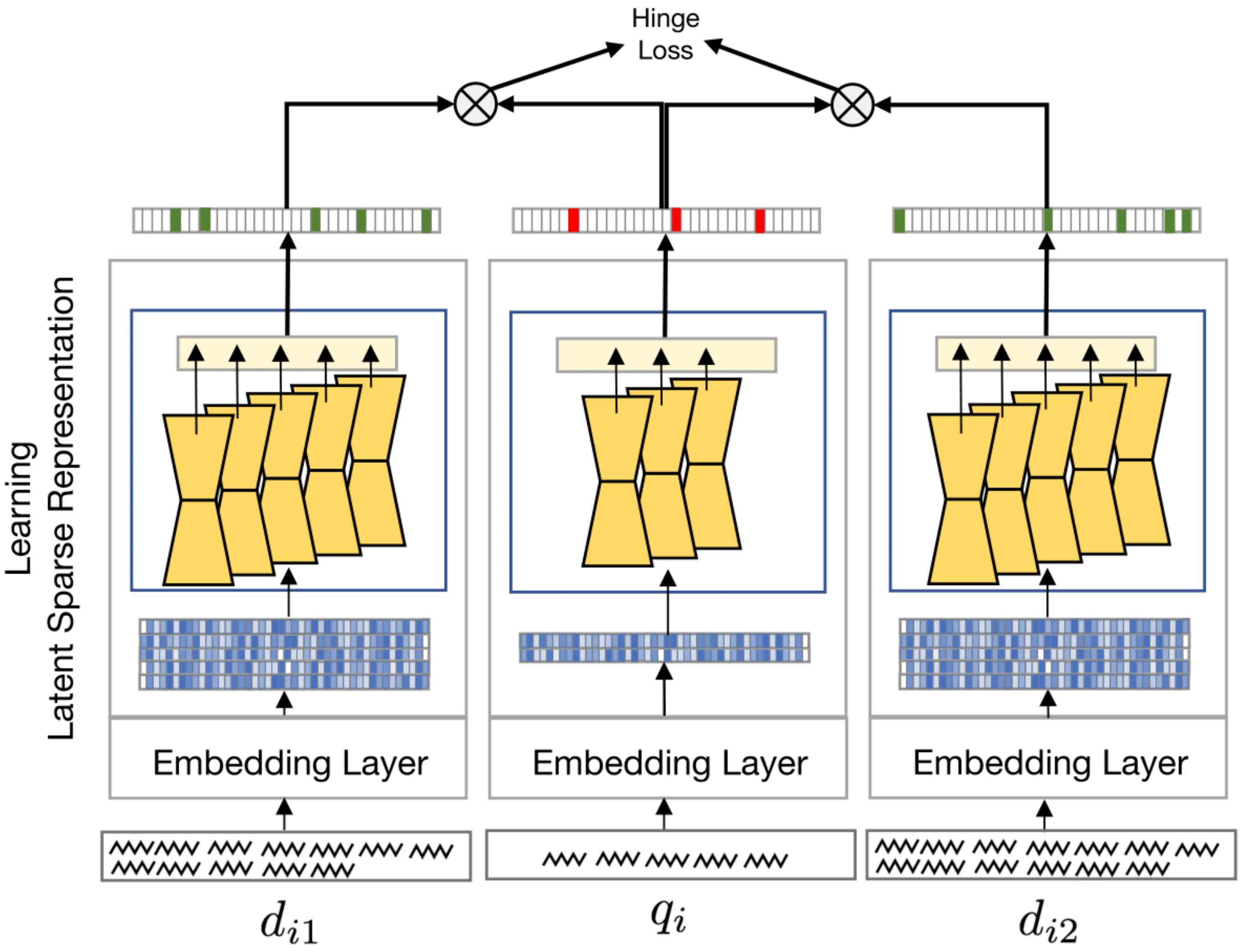
Training

Efficiency -> Sparsity

- Maximizing sparsity ratio, i.e., equivalent to minimizing $L_0(\vec{v}) = \sum_{i=1}^{|\vec{v}|} |\vec{v_i}|^0$.
- A differentiable approximator for L_0.

Effectiveness

- A learning-to-rank loss function (e.g., Hinge loss)



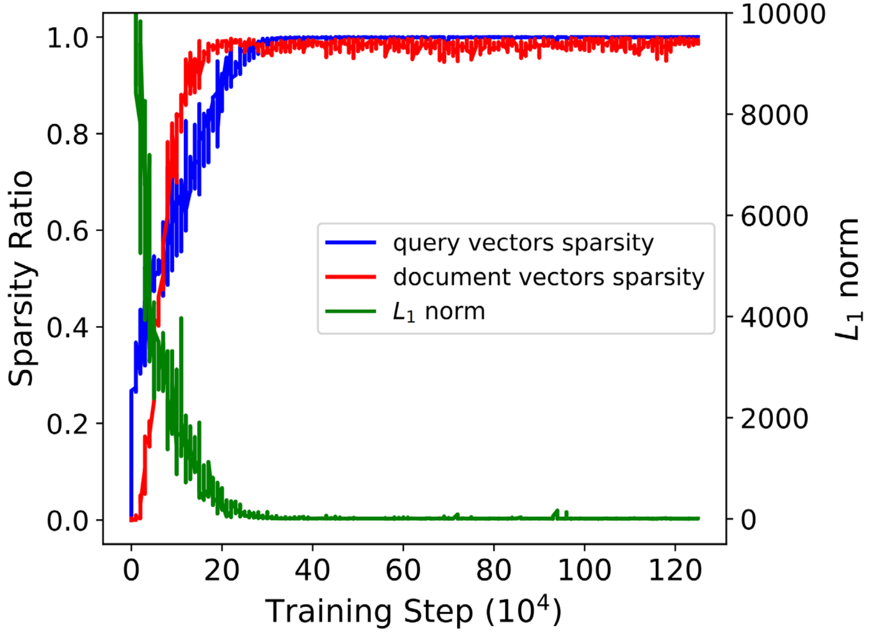
[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

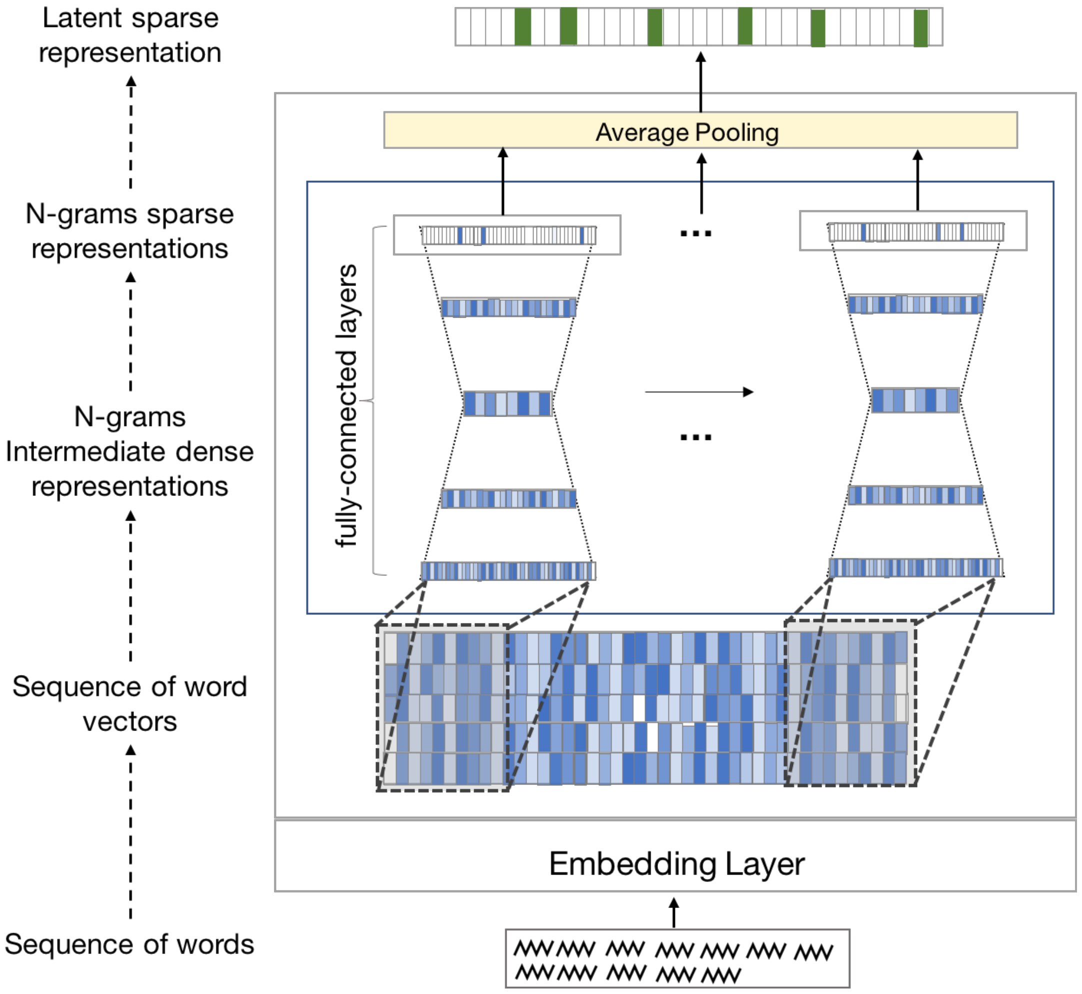
$|q|-n+1 << |d|-n+1$ and ϕ_{ngram} is shared.

$$\vec{d} = \frac{1}{|d|-n+1} \sum_{i=1}^{\boxed{|d|-n+1}} \phi_{\mathrm{ngram}}(w_i, w_{i+1}, \ldots, w_{i+n-1})$$

# of non-zero dimensions (out of 10,000) per document.

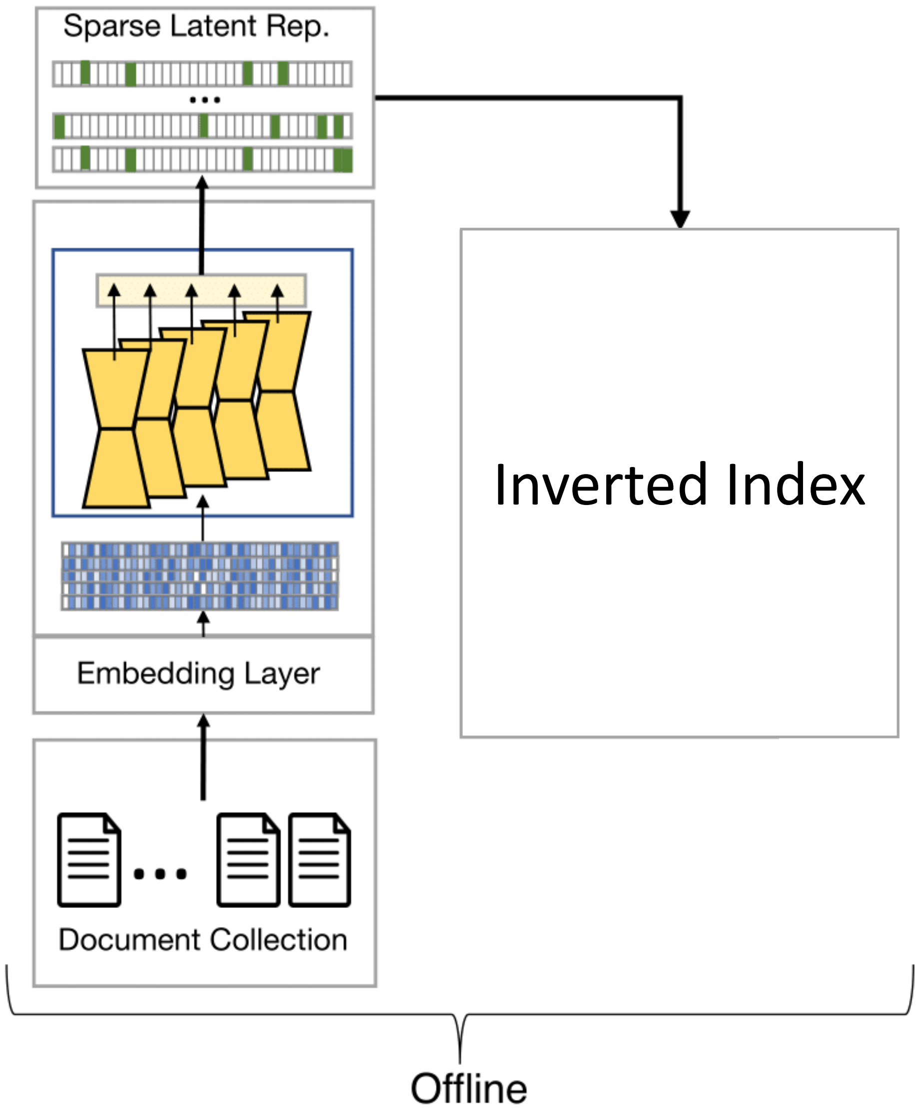
# Unique latent terms…	Robust		ClueWeb	
	Mean	Std. dev.	Mean	Std. dev.
per document	97.96	447.57	130.24	561.53
per query	3.37	3.04	3.87	4.51





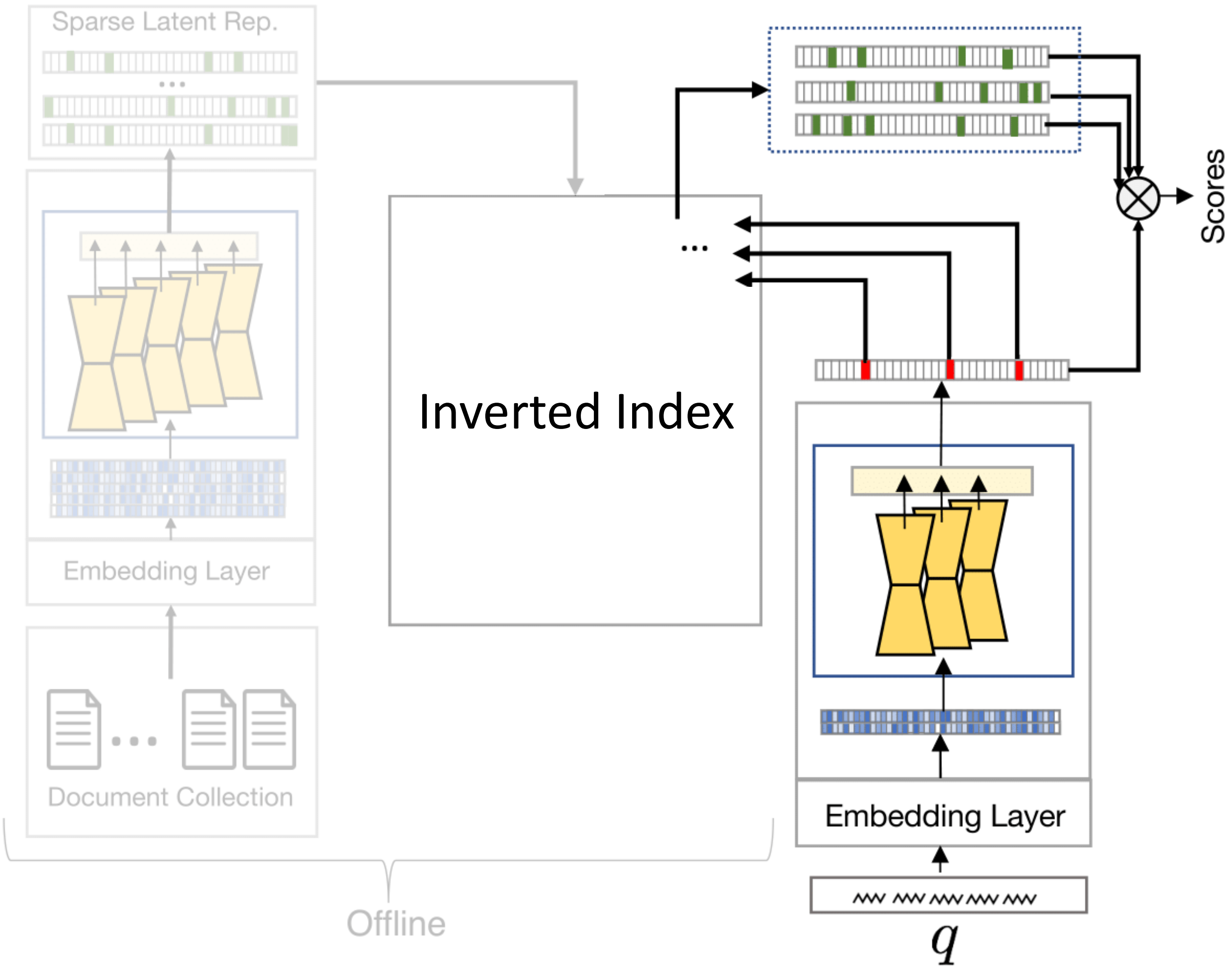
[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

Inverted Index Construction



[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

Query Time



[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

Effectiveness

Method	Robust				ClueWeb			
	MAP	P@20	nDCG@20	Recall	MAP	P@20	nDCG@20	Recall
QL	0.2499	0.3556	0.4143	0.6820	0.1044	0.3139	0.2294	0.3286
SDM	0.2524	0.3679^{1}	0.4242^{1}	0.6858	0.1078	0.3141	0.2320	0.3385^{1}
RM3	0.2865^{12}	0.3773^{12}	0.4295^{12}	0.7494^{12}	0.1068	0.3157	0.2309	0.3298
FNRM	0.2815^{12}	0.3752^{12}	0.4327^{12}	0.7234^{12}	0.1329^{123}	0.3351^{123}	0.2392^{13}	0.3426^{123}
CNRM	0.2801^{12}	0.3764^{12}	0.4341^{123}	0.7183^{12}	0.1286^{123}	0.3317^{123}	0.2337^{1}	0.3345^{13}
SNRM	0.2856^{12}	0.3766^{12}	0.4310^{12}	0.7481^{1245}	0.1290^{123}	0.3336^{123}	0.2351^{13}	0.3393^{135}
SNRM with PRF	$\mathbf{0.2971}^{123456}$	$\mathbf{0.3948}^{123456}$	$\mathbf{0.4391}^{123456}$	$\mathbf{0.7716}^{123456}$	$\mathbf{0.1475}^{123456}$	$\mathbf{0.3461}^{123456}$	$\mathbf{0.2482}^{123456}$	$\mathbf{0.3618}^{123456}$

[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

Efficiency

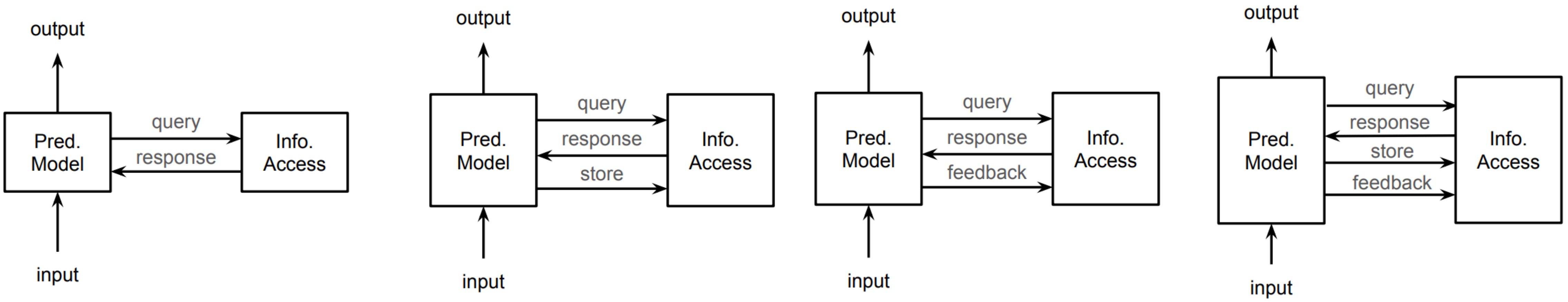
Average running time per query in milliseconds for the ClueWeb collection.

Metric	mean	stdev
Term matching	662 ms	746 ms
SNRM	612 ms	640 ms

[Zamani, Dehghani, Croft, Learned-Miller, Kamps; CIKM 2018]

How can efficient neural IR advance machine learning research?

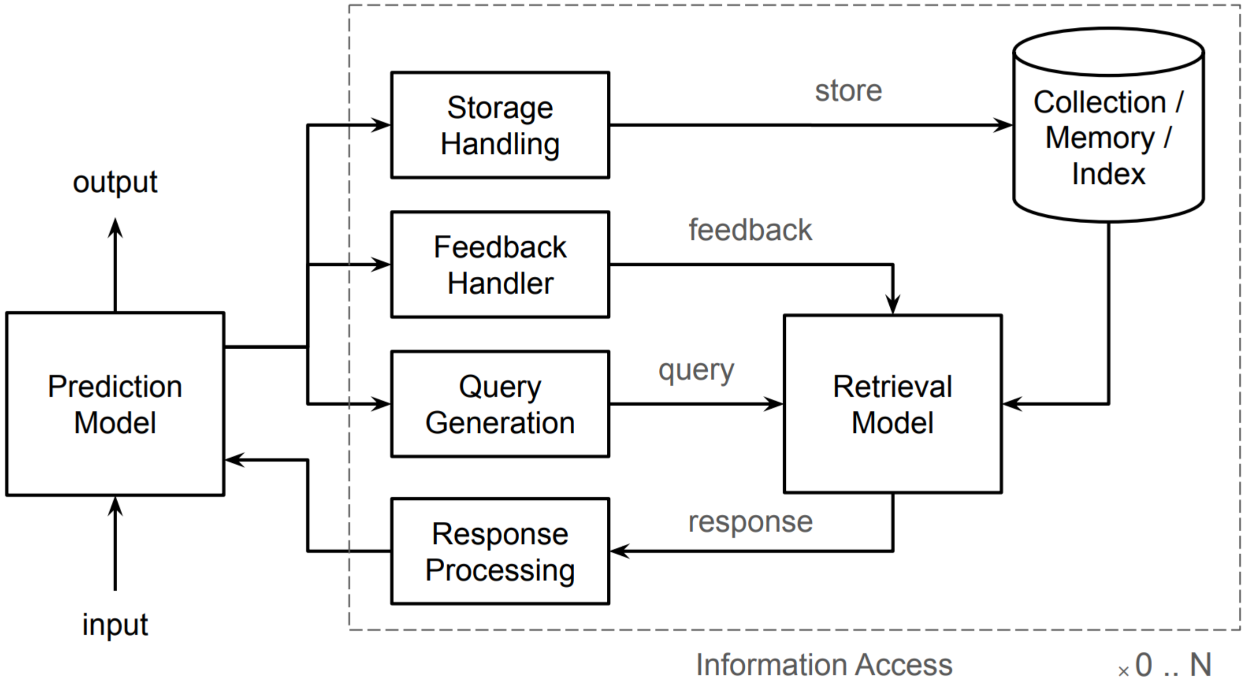
Retrieval-Enhanced Machine Learning



(a) Cat 1: Retrieval-only (b) Cat 2: Retrieval with memory (c) Cat 3: Retrieval with feedback (d) Cat 4: Retrieval with memory & feedback

[Zamani*, Diaz*, Dehghani, Metzler, Bendersky; SIGIR 2022]

A Generic Framework for REML



[Zamani*, Diaz*, Dehghani, Metzler, Bendersky; SIGIR 2022]

Some Case Studies

- Knowledge Grounding
  - REALM [Guu et al., ICML 2020]
  - RetGen [Zhang et al., AAAI 2020]
  - Entities-as-Expert [Févry et al., EMNLP 2020] or Facts-as-Expert [Verga et al., NAACL 2021]
- Memory-Augmented ML
  - Memory network [Sukhbaatar et al., NeurIPS 2015]
  - Neural Turing Machine [Graves et al., arXiv 2014]
- Retrieval-Enhanced Input Representation
  - Pseudo-Relevance Feedback [Attar and Fraenkel, JACM 1977][Croft and Harper, Jdoc 1978]
  - Guided Transformer [Hashemi et al., SIGIR 2020]
  - RETRO [Borgeaud et al., arXiv 2021]

[Zamani*, Diaz*, Dehghani, Metzler, Bendersky; SIGIR 2022]

More Case Studies!

- Generalization through Memorization
  - KNN-LM [Khandelwal et al., ICLR 2020]
  - BERT-KNN [Kassner and Schütze, EMNLP 2020]
  - HybridNCM [Yang et al., CIKM 2019]
- Efficient Access to Longer Context
  - MemViT [Wu et al., arXiv 2022]
  - IDCM [Hofstätter et al., SIGIR 2021]
- Retrieval-Enhanced Optimization
  - Weak supervision for IR [Dehghani et al., SIGIR 2017]
  - Hard negatives in IR optimization, e.g., ANCE [Xiong et al., ICLR 2021]
  - Unsupervised machine translation [Wu et al., NAACL 2019]
  - CLIP [Radford et al., ICML 2021] and VideoCLIP [Xu et al., EMNLP 2021]

[Zamani*, Diaz*, Dehghani, Metzler, Bendersky; SIGIR 2022]

Efficient and effective Neural IR can potentially revolutionize machine learning research through retrieval-enhanced models.

Multi-Task Retrieval-Augmented Text Generation with Relevance Sampling

Sebastian Hofstätter [1] **Jiecao Chen** [2] **Karthik Raman** [2] **Hamed Zamani** [3]

Abstract

This paper studies multi-task training of retrieval-augmented generation models for knowledge-intensive tasks. We propose to clean the training set by utilizing a distinct property of knowledge-intensive generation: The connection of query-answer pairs to items in the knowledge base. We filter training examples via a threshold of confidence on the relevance labels, whether a pair is answerable by the knowledge base or not. We train a single Fusion-in-Decoder (FiD) generator on seven combined tasks of the KILT benchmark. The experimental results suggest that our simple yet effective approach substantially improves competitive baselines on two strongly imbalanced tasks; and shows either smaller improvements or no significant regression on the remaining tasks. Furthermore, we demonstrate our multi-task training with relevance label sampling scales well with increased model capacity and achieves state-of-the-art results in five out of seven KILT tasks.

bated when tasks are retroactively expanded (Kwiatkowski et al., 2019), re-purposed (Bajaj et al., 2016) or adapt the collection (Petroni et al., 2021).

We propose a simple yet effective approach for training retrieval-augmented models for knowledge-intensive tasks with noisy labels. We use a confidence score for query-answer pairs and items in the knowledge base. This confidence can be sourced from manually annotated, heuristic, or model generated aspects. We filter training examples via a threshold of confidence on the relevance labels, whether a pair is answerable by the knowledge base or not. With this we aim to reduce noise in the training process, and produce better results with fewer training examples.

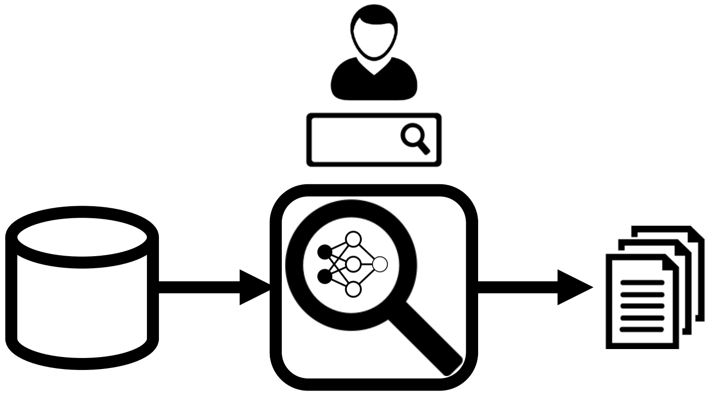
To study our training approach, we use a fixed T5-based dense retrieval module (Ni et al., 2021) and train a Fusion-in-Decoder (FiD) generator (Izacard & Grave, 2020) on multiple tasks of the KILT benchmark (Petroni et al., 2021). KILT aggregates and heuristically maps many different English Wikipedia-based generation tasks to a single Wikipedia snapshot, which introduces considerable noise in the label quality, due to the time-shifted nature of the task creations.

Results on the KILT Benchmark

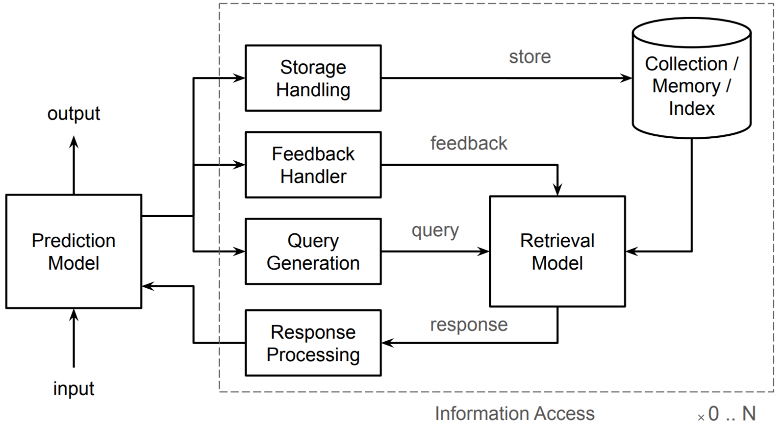
	Model	Generator	Open Domain QA *EM* NQ	HotpotQA	TriviaQA	Fact *Acc.* FEVER	Slot Filling *Accuracy* T-REx	zsRE	Dialog *F1* WOW
	Top Leaderboard Entries								
1	RAG (Petroni et al., 2021)	BART-Large	44.4	27.0	71.3	86.3	59.2	44.7	13.1
2	DPR + FiD (Piktus et al., 2021)	T5-Base	51.6	38.3	72.7	89.0	82.2	74.0	15.7
3	KGI (Glass et al., 2021)	BART-Large	45.2	–	61.0	85.6	84.4	72.6	18.6
4	Re2G (Anonymous, 2022)	BART-Large	51.7	–	76.3	89.6	**87.7**	–	18.9
5	Hindsight (Paranjape et al., 2021)	BART-Large	–	–	–	–	–	–	19.2
6	SEAL+FiD (Bevilacqua et al., 2022)	T5-?	53.7	**40.5**	70.9	89.5	83.7	74.7	18.3
	Ours (Alt-200 passages)								
7	GTR + FiD with treatment $\hat{T}$	T5-Base	52.4	30.1	78.9	87.1	83.4	81.5	18.4
8		T5-XL	**61.2**	39.1	**84.6**	**92.3**	85.2	**83.7**	**20.6**

[Hofstätter, Cao, Raman, Zamani; 2022]

Thank you!



Efficient Neural Information Retrieval



Retrieval-Enhanced Machine Learning

UMassAmherst
College of Information & Computer Sciences
Center for Intelligent Information Retrieval

Hamed Zamani
zamani@cs.umass.edu
Twitter: @HamedZamani