Large-scale Retrieval & Language Model

Jimblin

Overview

- Context-Aware Document Term Weighting for Ad-Hoc Search （*WWW2020*)
- Pre-training Tasks for Embedding-based Large-scale Retrieval （*ICLR2020*)
- ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT （*SIGIR2012*)

Context-Aware Document Term Weighting for Ad-Hoc Search

Zhuyun Dai
Carnegie Mellon University
zhuyund@cs.cmu.edu

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu

Motivation & Contribution

- Bag-of-words document representations is limited by the shallow frequency-based term weighting scheme. (tf.idf, bm25)
- This paper uses the contextual word representations from BERT to generate more effective document term weights.

Methodology

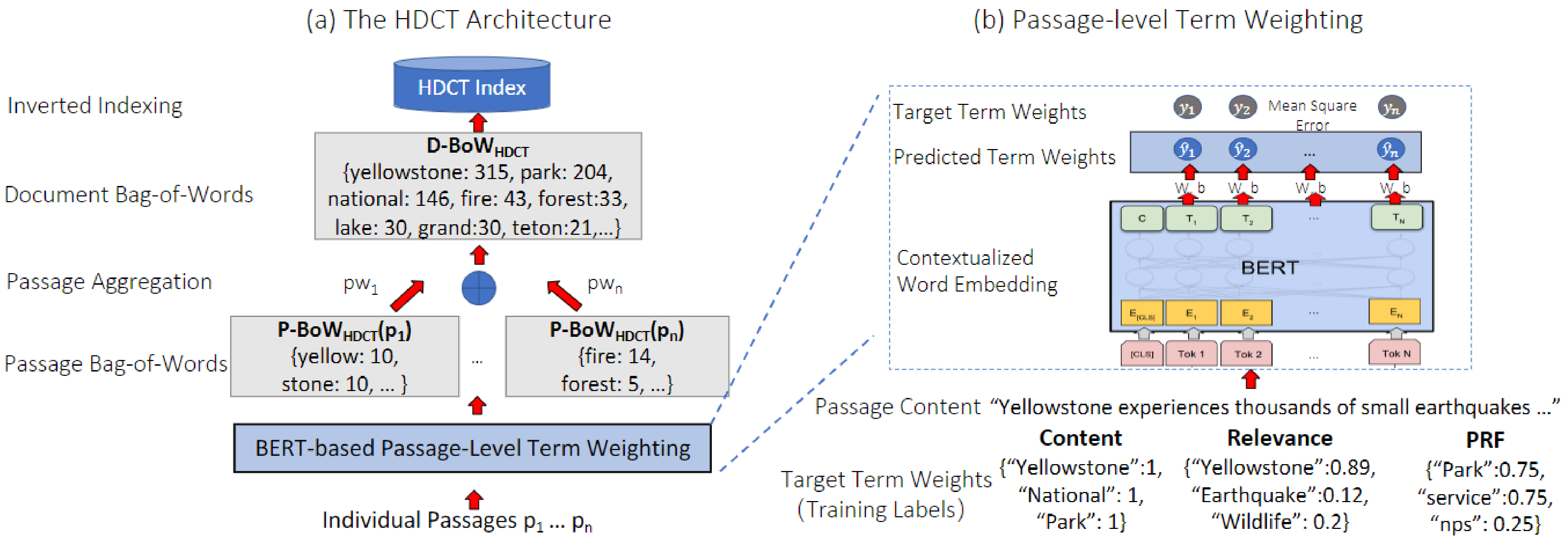
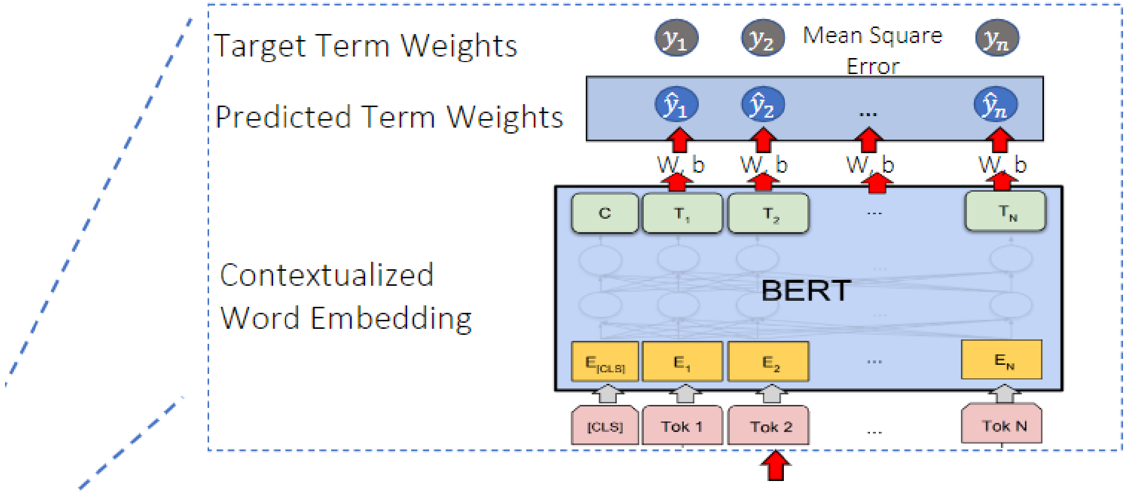

(a) The HDCT Architecture
Inverted Indexing
HDCT Index
Document Bag-of-Words
D-BoW$_{HDCT}$
{yellowstone: 315, park: 204, national: 146, fire: 43, forest:33, lake: 30, grand:30, teton:21,...}
Passage Aggregation
pw$_1$
pw$_n$
Passage Bag-of-Words
P-BoW$_{HDCT}$(p_1)
{yellow: 10, stone: 10, ... }
...
P-BoW$_{HDCT}$(p_n)
{fire: 14, forest: 5, ...}
BERT-based Passage-Level Term Weighting
Individual Passages p_1 ... p_n
(b) Passage-level Term Weighting
Target Term Weights
y_1
y_2
Mean Square Error
y_n
Predicted Term Weights
$\hat{y}_1$
$\hat{y}_2$
...
$\hat{y}_n$
W, b
Contextualized Word Embedding
C
T$_1$
T$_2$
T$_N$
BERT
E$_{[CLS]}$
E$_1$
E$_2$
E$_N$
[CLS]
Tok 1
Tok 2
Tok N
Passage Content “Yellowstone experiences thousands of small earthquakes ...”
Target Term Weights (Training Labels)
Content
{“Yellowstone”:1, “National”: 1, “Park”: 1}
Relevance
{“Yellowstone”:0.89, “Earthquake”:0.12, “Wildlife”: 0.2}
PRF
{“Park”:0.75, “service”:0.75, “nps”: 0.25}

Figure 1: The HDCT architecture.

Passage-Level Term Weighting

(b) Passage-level Term Weighting



Passage Content “Yellowstone experiences thousands of small earthquakes ...”

一个doc分为多个passage，获取每个passage的向量表示。

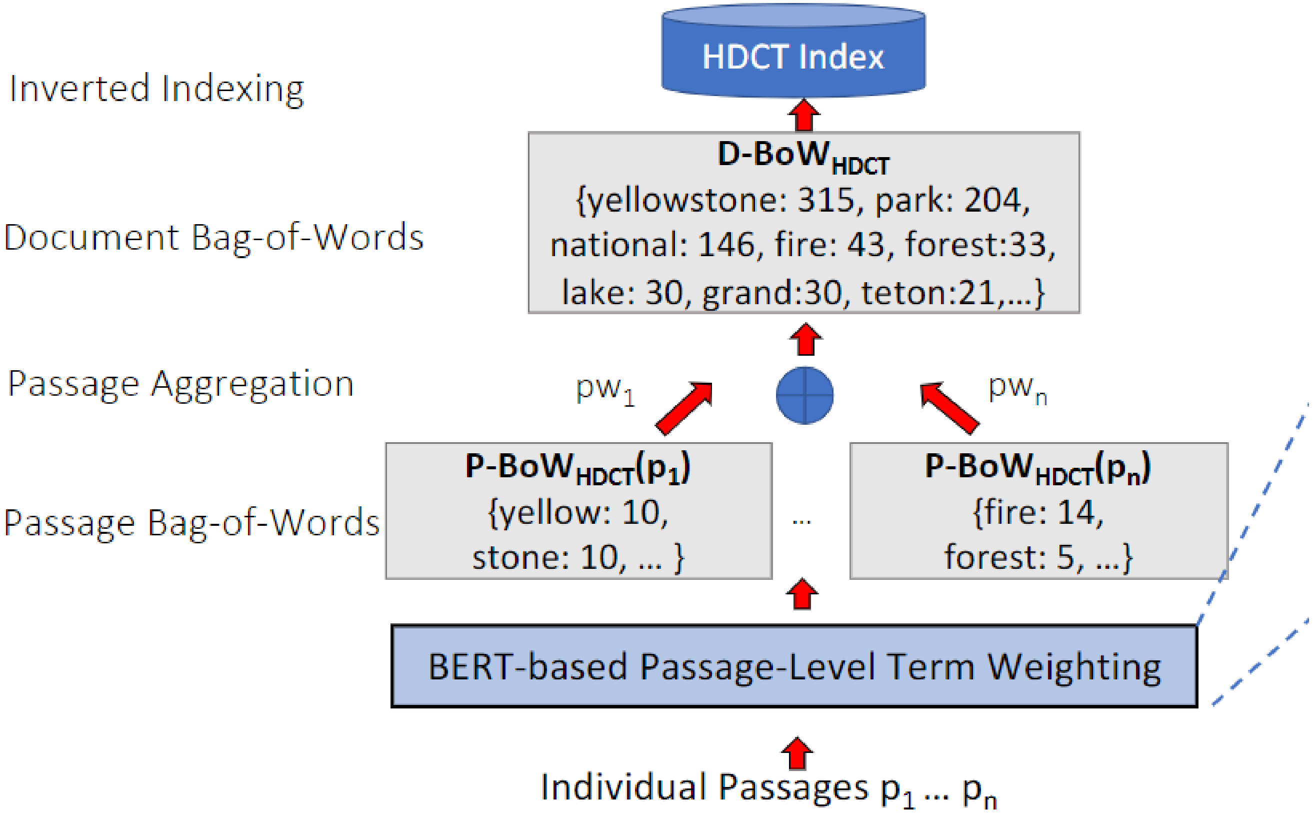
$$\hat{y}_{t,p} = w \cdot T_{\text{BERT}}(t,p) + b.$$

$$tf_{\text{BERT}}(t,p) = round(N * \sqrt{\hat{y}_{t,p}}).$$

$$\text{P-BoW}_{\text{HDCT}}(p) = [tf_{\text{BERT}}(t_1,p),..,tf_{\text{BERT}}(t_m,p)].$$

Document-Level Term Weighting

(a) The HDCT Architecture


HDCT Index
Inverted Indexing
D-BoW$_{HDCT}$
{yellowstone: 315, park: 204, national: 146, fire: 43, forest:33, lake: 30, grand:30, teton:21,…}
Document Bag-of-Words
Passage Aggregation
pw$_1$
pw$_n$
P-BoW$_{HDCT}$(p$_1$)
{yellow: 10, stone: 10, … }
…
P-BoW$_{HDCT}$(p$_n$)
{fire: 14, forest: 5, …}
Passage Bag-of-Words
BERT-based Passage-Level Term Weighting
Individual Passages p$_1$ … p$_n$

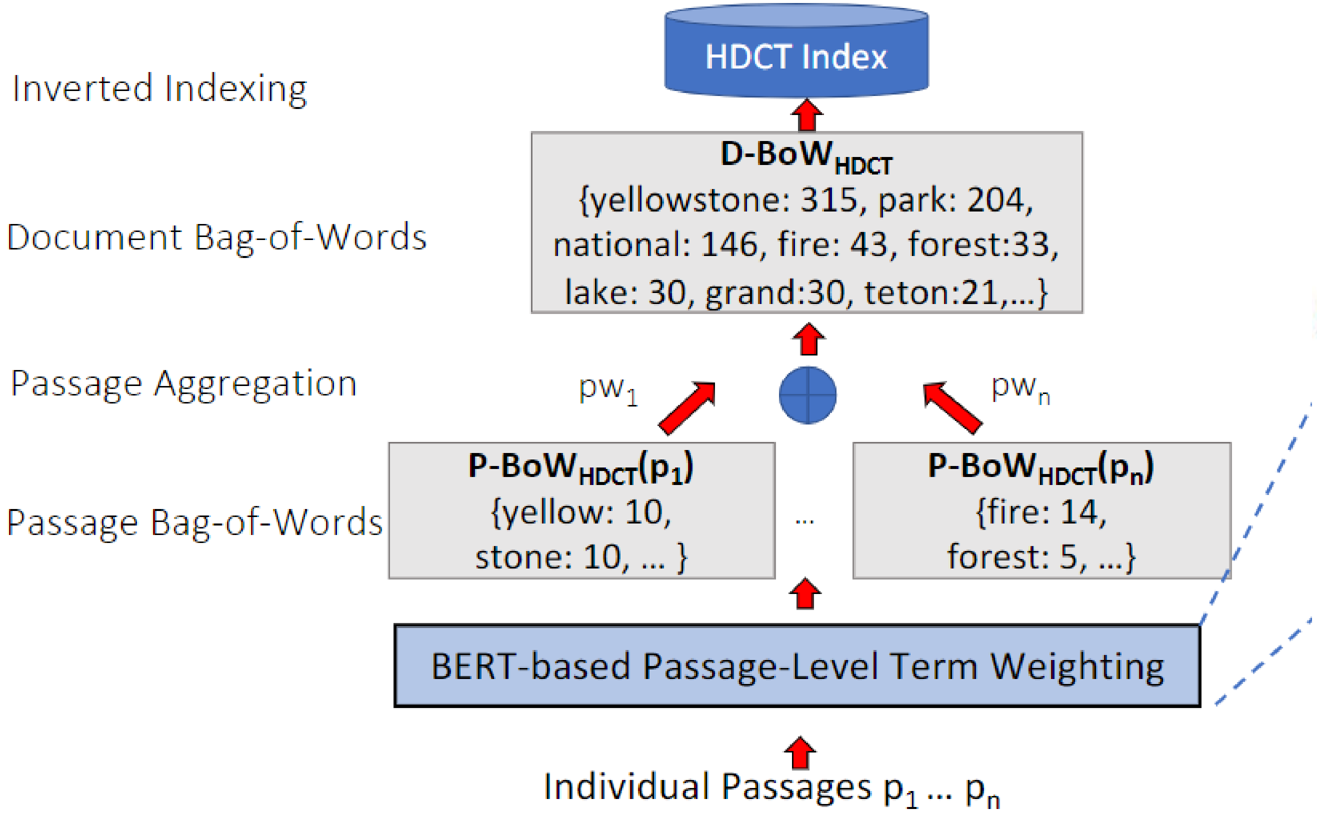
$$\{\text{P-BoW}_{\text{HDCT}}(p_1), \ldots, \text{P-BoW}_{\text{HDCT}}(p_n)\}.$$

$$\text{D-BoW}_{\text{HDCT}}(d) = \sum_{i=1}^{n} pw_i \times \text{P-BoW}_{\text{HDCT}}(p_i).$$

pw_i=1 ： 表示每个段落的重要性一致；
pw_i=1/i：表示段落的重要性在递减，第一个段落比较重要，最后一个段落比较不重要。

Retrieval with HDCT Index

(a) The HDCT Architecture



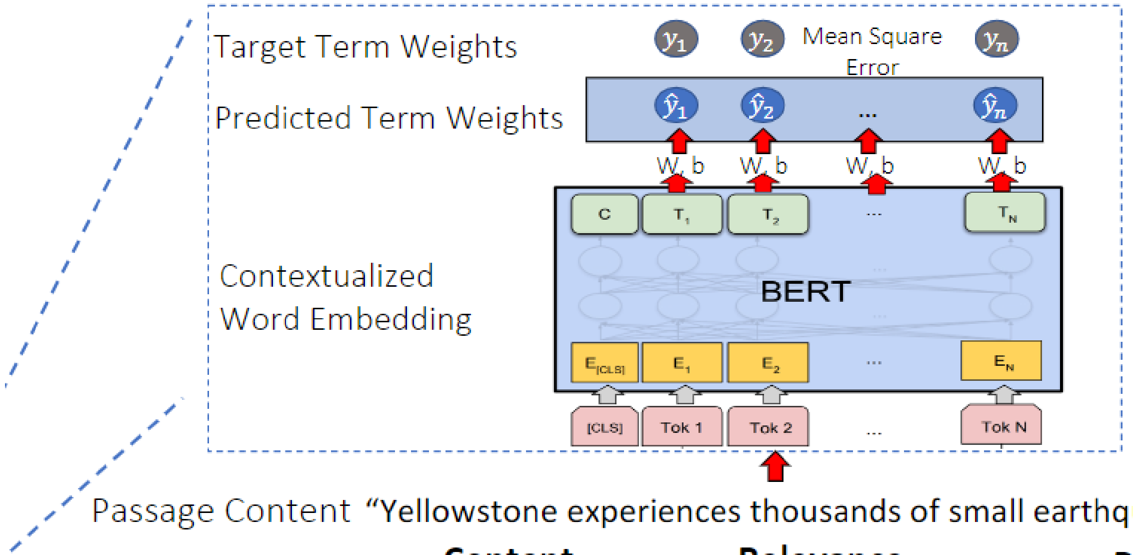
HDCT Index
Inverted Indexing
D-BoW$_{HDCT}$
{yellowstone: 315, park: 204, national: 146, fire: 43, forest:33, lake: 30, grand:30, teton:21,...}
Document Bag-of-Words
Passage Aggregation
pw$_1$
pw$_n$
P-BoW$_{HDCT}$(p$_1$)
{yellow: 10, stone: 10, ... }
...
P-BoW$_{HDCT}$(p$_n$)
{fire: 14, forest: 5, ...}
Passage Bag-of-Words
BERT-based Passage-Level Term Weighting
Individual Passages p$_1$ ... p$_n$

将BM25算法中的idf权重替换为HDCT权重

$$Score(Q,d) = \sum_{i}^{n} IDF(q_i) \cdot \frac{f_i \cdot (k_1 + 1)}{f_i + k_1 \cdot (1 - b + b \cdot \frac{dl}{avgdl})}$$

Training Strategy For HDCT

(b) Passage-level Term Weighting



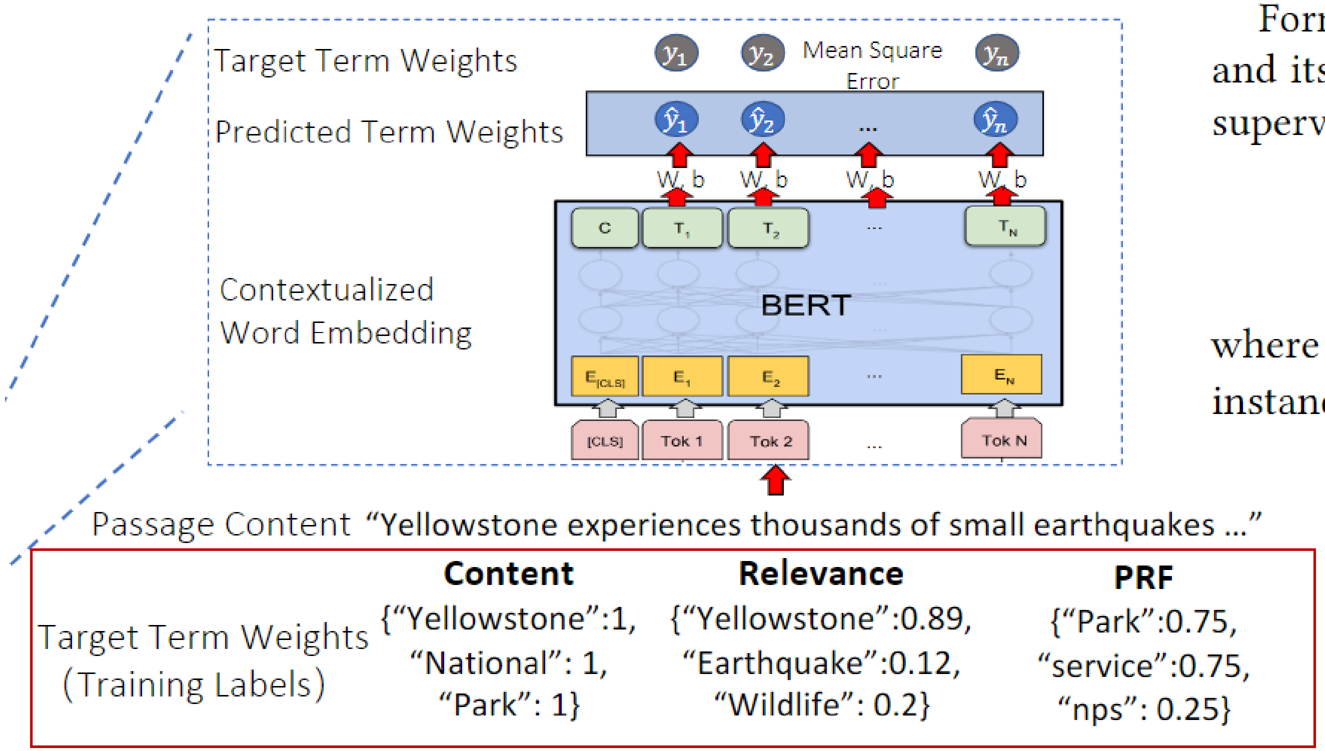
Passage Content “Yellowstone experiences thousands of small earthquakes ...”

	Content	Relevance	PRF
Target Term Weights (Training Labels)	{“Yellowstone”:1, “National”: 1, “Park”: 1}	{“Yellowstone”:0.89, “Earthquake”:0.12, “Wildlife”: 0.2}	{“Park”:0.75, “service”:0.75, “nps”: 0.25}

$$MSE = \sum_{p} \sum_{t \in p} (y_{t,p} - \hat{y}_{t,p})^2.$$

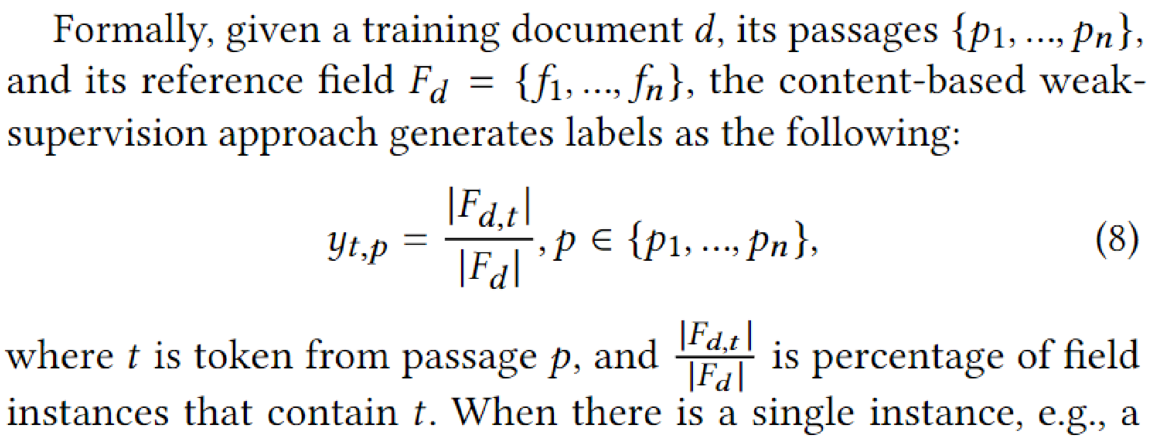
Training Strategy For HDCT

(b) Passage-level Term Weighting



	Content	Relevance	PRF
Target Term Weights (Training Labels)	{"Yellowstone":1, "National": 1, "Park": 1}	{"Yellowstone":0.89, "Earthquake":0.12, "Wildlife": 0.2}	{"Park":0.75, "service":0.75, "nps": 0.25}

1、Supervision from Document Content



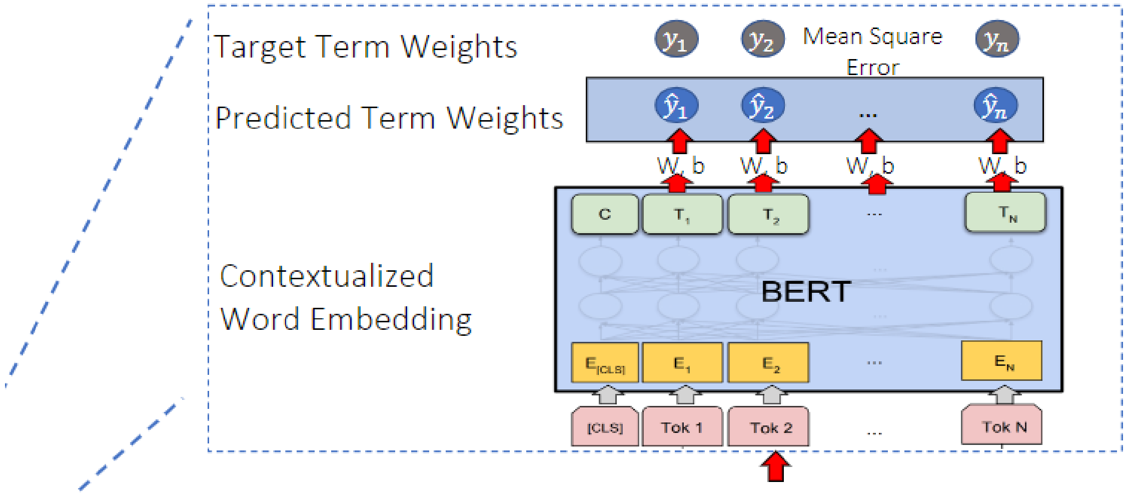
Formally, given a training document d, its passages $\{p_1, ..., p_n\}$, and its reference field $F_d = \{f_1, ..., f_n\}$, the content-based weak-supervision approach generates labels as the following:

$$y_{t,p} = \frac{|F_{d,t}|}{|F_d|}, p \in \{p_1, ..., p_n\}, \tag{8}$$

where t is token from passage p, and $\frac{|F_{d,t}|}{|F_d|}$ is percentage of field instances that contain t. When there is a single instance, e.g., a

Training Strategy For HDCT

(b) Passage-level Term Weighting



Passage Content "Yellowstone experiences thousands of small earthquakes ..."

	Content	Relevance	PRF
Target Term Weights (Training Labels)	{"Yellowstone":1, "National": 1, "Park": 1}	{"Yellowstone":0.89, "Earthquake":0.12, "Wildlife": 0.2}	{"Park":0.75, "service":0.75, "nps": 0.25}

2、Supervision from Relevance and Pseudo-Relevance Feedback

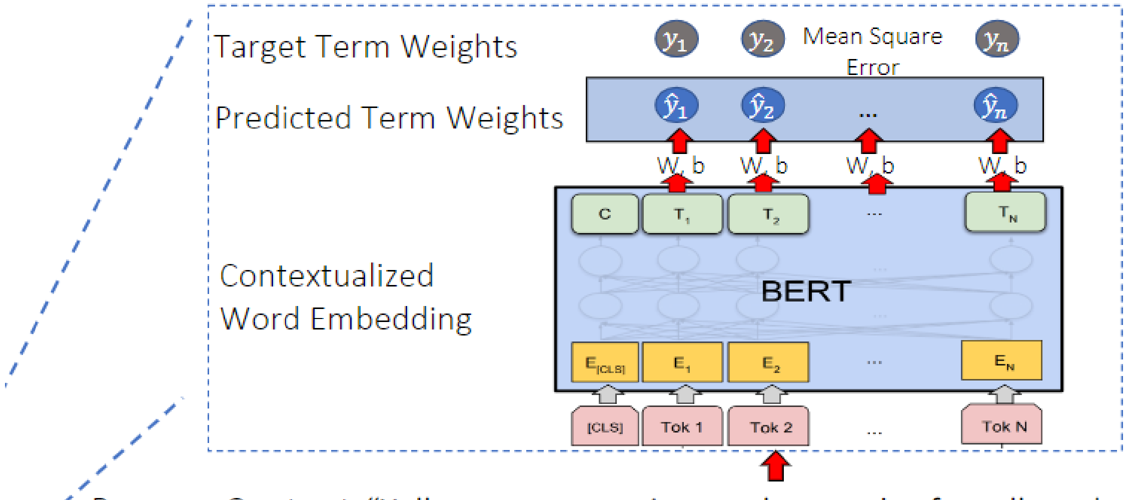
Given a training document d, its passages $P_d = \{p_1, ..., p_n\}$, and its relevant queries $Q_d = \{q_1, ..., q_b\}$, we generate the *relevance-based* training labels as follows:

$$y_{t,p} = \frac{|Q_{d,t}|}{|Q_d|}, p \in \{p_1, ..., p_n\}. \tag{9}$$

t is a term from passage p in document d. $\frac{|Q_{d,t}|}{|Q_d|}$ is the percentage of d's relevant queries that mention term t. If most of d's queries all

Training Strategy For HDCT

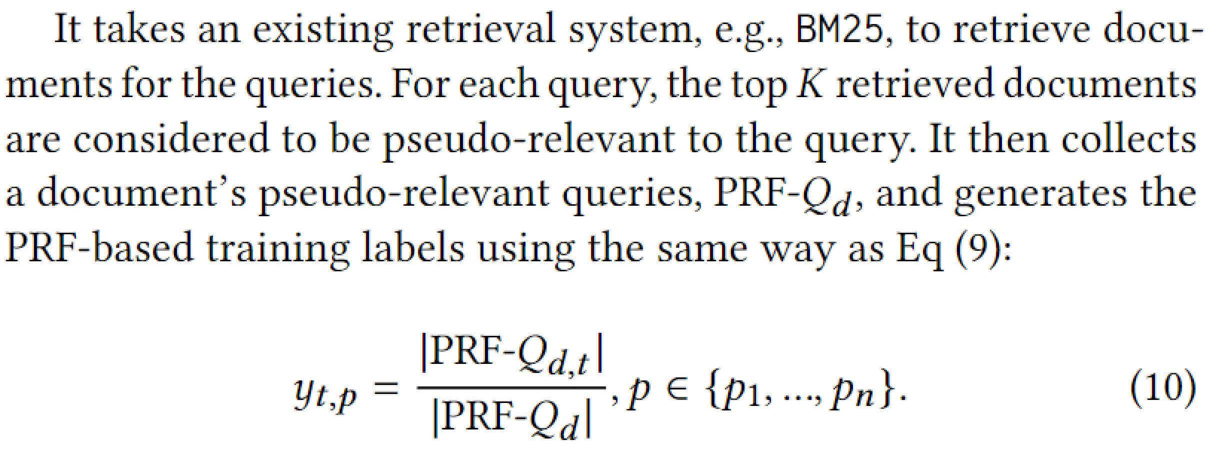
(b) Passage-level Term Weighting



Passage Content “Yellowstone experiences thousands of small earthquakes ...”

	Content	**Relevance**	**PRF**
Target Term Weights (Training Labels)	{“Yellowstone”:1, “National”: 1, “Park”: 1}	{“Yellowstone”:0.89, “Earthquake”:0.12, “Wildlife”: 0.2}	{“Park”:0.75, “service”:0.75, “nps”: 0.25}

3、Global Labels for Local Term Weighting



It takes an existing retrieval system, e.g., BM25, to retrieve documents for the queries. For each query, the top K retrieved documents are considered to be pseudo-relevant to the query. It then collects a document’s pseudo-relevant queries, PRF-Q_d, and generates the PRF-based training labels using the same way as Eq (9):

$$y_{t,p} = \frac{|\text{PRF-}Q_{d,t}|}{|\text{PRF-}Q_d|}, p \in \{p_1, ..., p_n\}. \tag{10}$$

Experiment

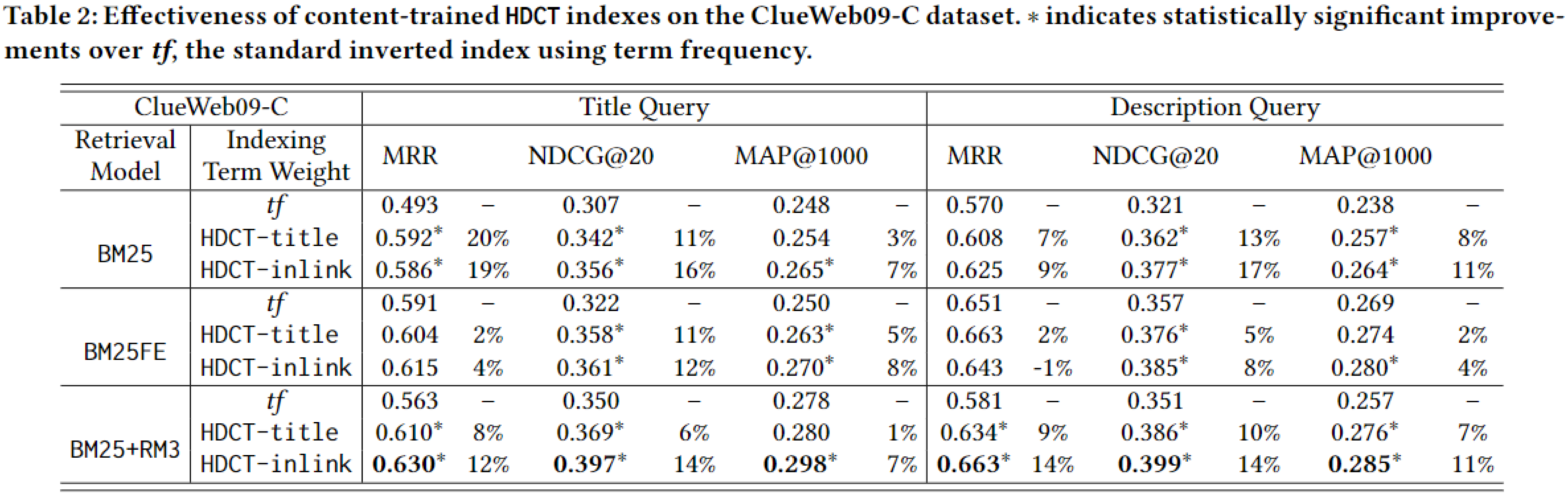
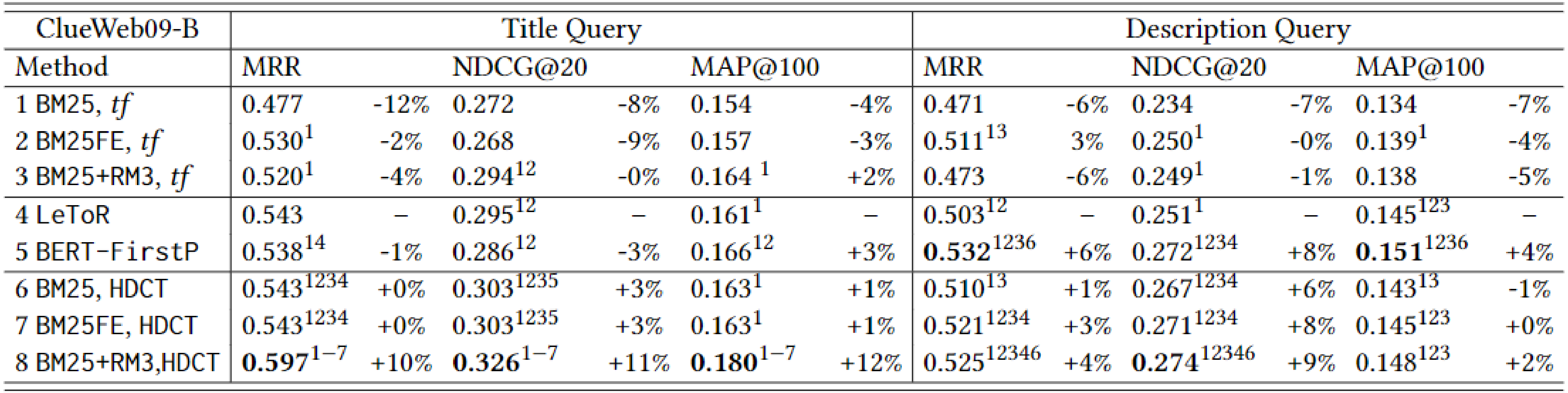


Table 2: Effectiveness of content-trained HDCT indexes on the ClueWeb09-C dataset. * indicates statistically significant improvements over *tf*, the standard inverted index using term frequency.

ClueWeb09-C		Title Query						Description Query					
Retrieval Model	Indexing Term Weight	MRR		NDCG@20		MAP@1000		MRR		NDCG@20		MAP@1000	
BM25	*tf*	0.493	–	0.307	–	0.248	–	0.570	–	0.321	–	0.238	–
	HDCT-title	0.592*	20%	0.342*	11%	0.254	3%	0.608	7%	0.362*	13%	0.257*	8%
	HDCT-inlink	0.586*	19%	0.356*	16%	0.265*	7%	0.625	9%	0.377*	17%	0.264*	11%
BM25FE	*tf*	0.591	–	0.322	–	0.250	–	0.651	–	0.357	–	0.269	–
	HDCT-title	0.604	2%	0.358*	11%	0.263*	5%	0.663	2%	0.376*	5%	0.274	2%
	HDCT-inlink	0.615	4%	0.361*	12%	0.270*	8%	0.643	-1%	0.385*	8%	0.280*	4%
BM25+RM3	*tf*	0.563	–	0.350	–	0.278	–	0.581	–	0.351	–	0.257	–
	HDCT-title	0.610*	8%	0.369*	6%	0.280	1%	0.634*	9%	0.386*	10%	0.276*	7%
	HDCT-inlink	**0.630***	12%	**0.397***	14%	**0.298***	7%	**0.663***	14%	**0.399***	14%	**0.285***	11%

BM25FE is an ensemble of BM25 rankers on different document fields.
BM25+RM3 is a popular query expansion technique

Experiment



ClueWeb09-B	Title Query						Description Query					
Method	MRR		NDCG@20		MAP@100		MRR		NDCG@20		MAP@100	
1 BM25, *tf*	0.477	-12%	0.272	-8%	0.154	-4%	0.471	-6%	0.234	-7%	0.134	-7%
2 BM25FE, *tf*	0.530^{1}	-2%	0.268	-9%	0.157	-3%	0.511^{13}	3%	0.250^{1}	-0%	0.139^{1}	-4%
3 BM25+RM3, *tf*	0.520^{1}	-4%	0.294^{12}	-0%	0.164^{1}	+2%	0.473	-6%	0.249^{1}	-1%	0.138	-5%
4 LeToR	0.543	–	0.295^{12}	–	0.161^{1}	–	0.503^{12}	–	0.251^{1}	–	0.145^{123}	–
5 BERT-FirstP	0.538^{14}	-1%	0.286^{12}	-3%	0.166^{12}	+3%	**0.532**^{1236}	+6%	0.272^{1234}	+8%	**0.151**^{1236}	+4%
6 BM25, HDCT	0.543^{1234}	+0%	0.303^{1235}	+3%	0.163^{1}	+1%	0.510^{13}	+1%	0.267^{1234}	+6%	0.143^{13}	-1%
7 BM25FE, HDCT	0.543^{1234}	+0%	0.303^{1235}	+3%	0.163^{1}	+1%	0.521^{1234}	+3%	0.271^{1234}	+8%	0.145^{123}	+0%
8 BM25+RM3,HDCT	**0.597**$^{1-7}$	+10%	**0.326**$^{1-7}$	+11%	**0.180**$^{1-7}$	+12%	0.525^{12346}	+4%	**0.274**^{12346}	+9%	0.148^{123}	+2%

LeToR is a popular feature-based learning-to-rank method
BERT-FirstP is a neural BERT-based re-ranker

Experiment

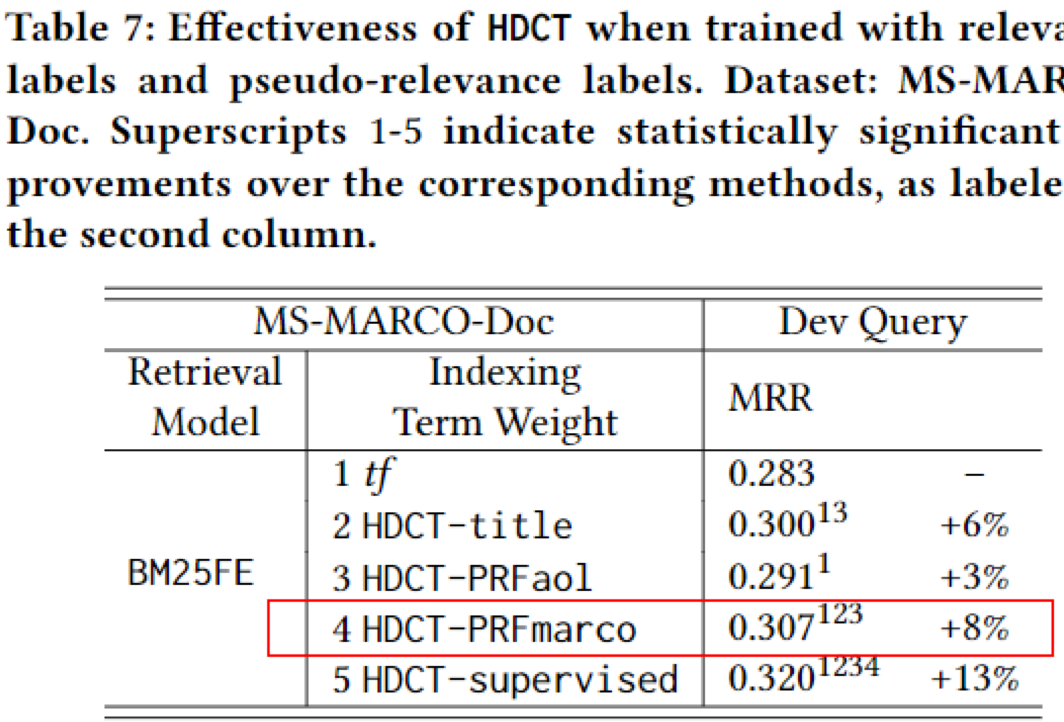


Table 7: Effectiveness of HDCT when trained with relevance labels and pseudo-relevance labels. Dataset: MS-MARCO-Doc. Superscripts 1-5 indicate statistically significant improvements over the corresponding methods, as labeled in the second column.

MS-MARCO-Doc		Dev Query	
Retrieval Model	Indexing Term Weight	MRR	
BM25FE	1 *tf*	0.283	–
	2 HDCT-title	0.300^{13}	+6%
	3 HDCT-PRFaol	0.291^{1}	+3%
	4 HDCT-PRFmarco	0.307^{123}	+8%
	5 HDCT-supervised	0.320^{1234}	+13%

HDCT-PRFmarco was trained with the PRF-based weak supervision strategy using BM25FE.

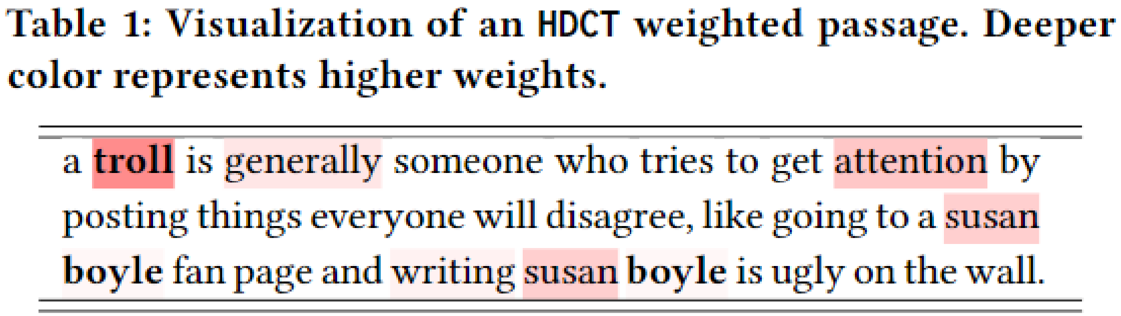


Table 1: Visualization of an HDCT weighted passage. Deeper color represents higher weights.

a **troll** is generally someone who tries to get attention by posting things everyone will disagree, like going to a susan **boyle** fan page and writing susan **boyle** is ugly on the wall.

Conclusion

- HDCT better captures key terms in a passage than tf.
- HDCT allows efficient and effective retrieval from an inverted index.
- A content-based weak-supervision strategy is presented to train HDCT without using relevance labels.
- Widely used in online system.

PRE-TRAINING TASKS FOR EMBEDDING-BASED LARGE-SCALE RETRIEVAL

Wei-Cheng Chang*, Felix X. Yu, Yin-Wen Chang, Yiming Yang, Sanjiv Kumar
Carnegie Mellon University & Google
{wchang2,yiming}@cs.cmu.edu, {felixyu,yinwen,sanjivk}@google.com

Motivation & Contribution

- BERT-style model has succeeded in re-ranking the retrieved documents.
- Using BERT-style model in the retrieval phase remains less well studied.

Methodology

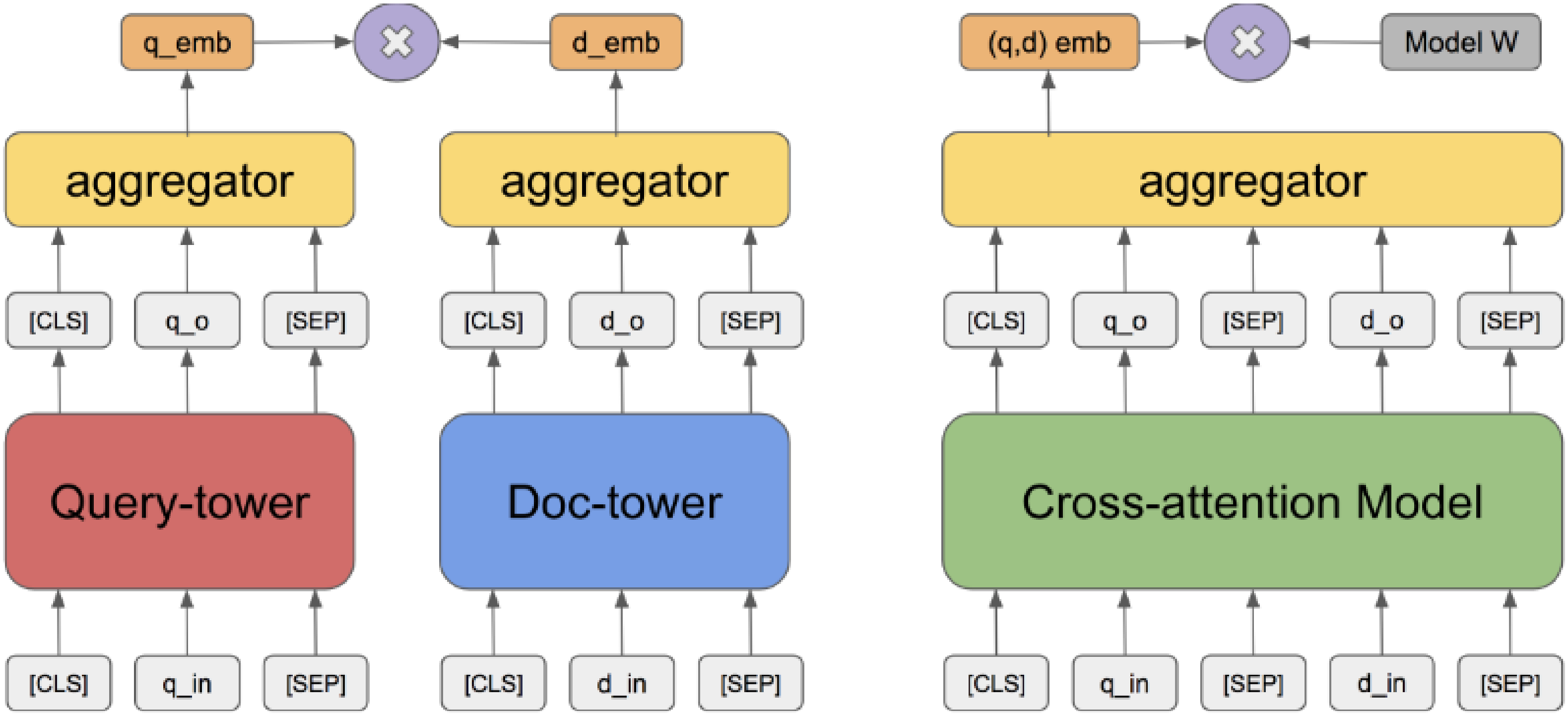
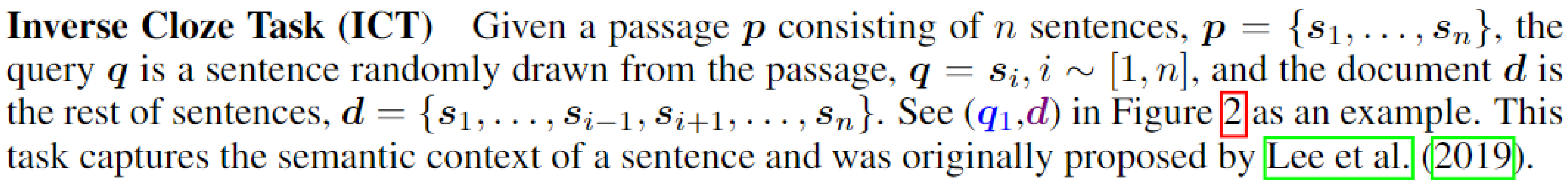


Figure 1: Difference between two-tower models and cross-attention models. Following previous works, we consider [CLS] embedding and average pooling as the aggregator's output for the two-tower Transformer model and the two-tower MLP model, respectively.

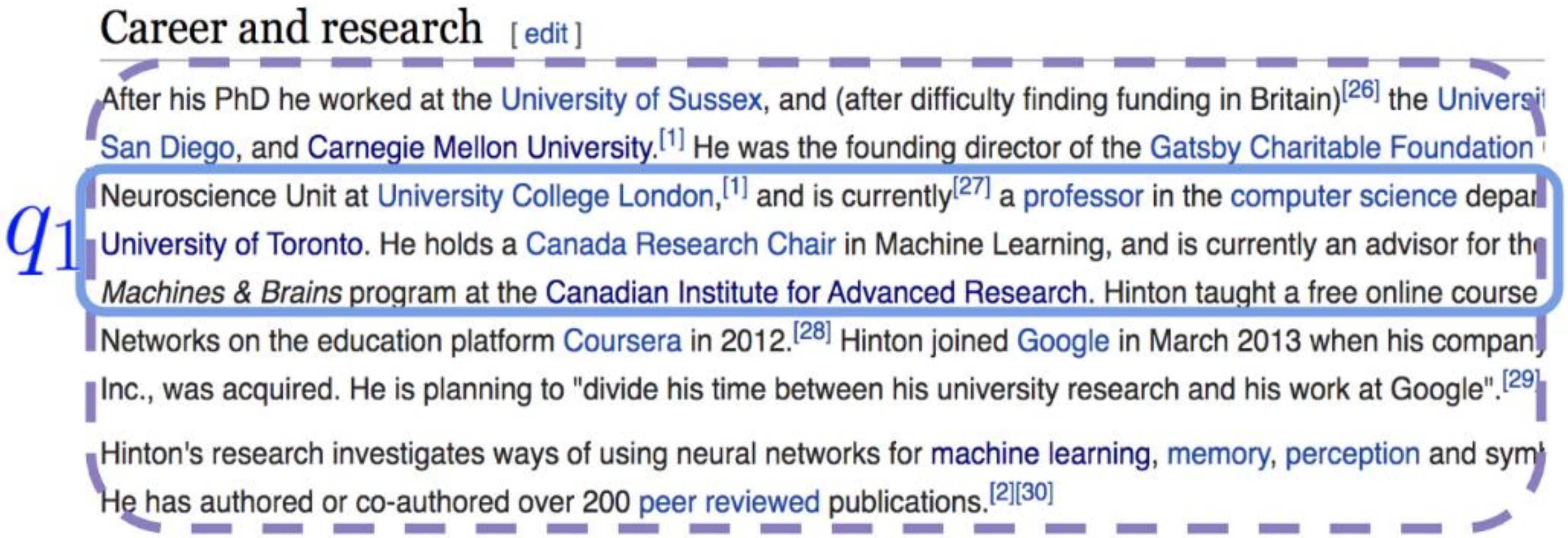
Inference $f_{\theta,\boldsymbol{w}}(\boldsymbol{q},\boldsymbol{d}) = \psi_\theta(\boldsymbol{q} \oplus \boldsymbol{d})^T \boldsymbol{w},$

Learning $p_\theta(\boldsymbol{d}|\boldsymbol{q}) = \dfrac{\exp\left(f_\theta(\boldsymbol{q},\boldsymbol{d})\right)}{\sum_{\boldsymbol{d}' \in \mathcal{D}} \exp\left(f_\theta(\boldsymbol{q},\boldsymbol{d}')\right)},$

three pre-training tasks



Inverse Cloze Task (ICT) Given a passage $\boldsymbol{p}$ consisting of n sentences, $\boldsymbol{p} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$, the query $\boldsymbol{q}$ is a sentence randomly drawn from the passage, $\boldsymbol{q} = \boldsymbol{s}_i, i \sim [1, n]$, and the document $\boldsymbol{d}$ is the rest of sentences, $\boldsymbol{d} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{i-1}, \boldsymbol{s}_{i+1}, \ldots, \boldsymbol{s}_n\}$. See ($\boldsymbol{q}_1$,$\boldsymbol{d}$) in Figure 2 as an example. This task captures the semantic context of a sentence and was originally proposed by Lee et al. (2019).



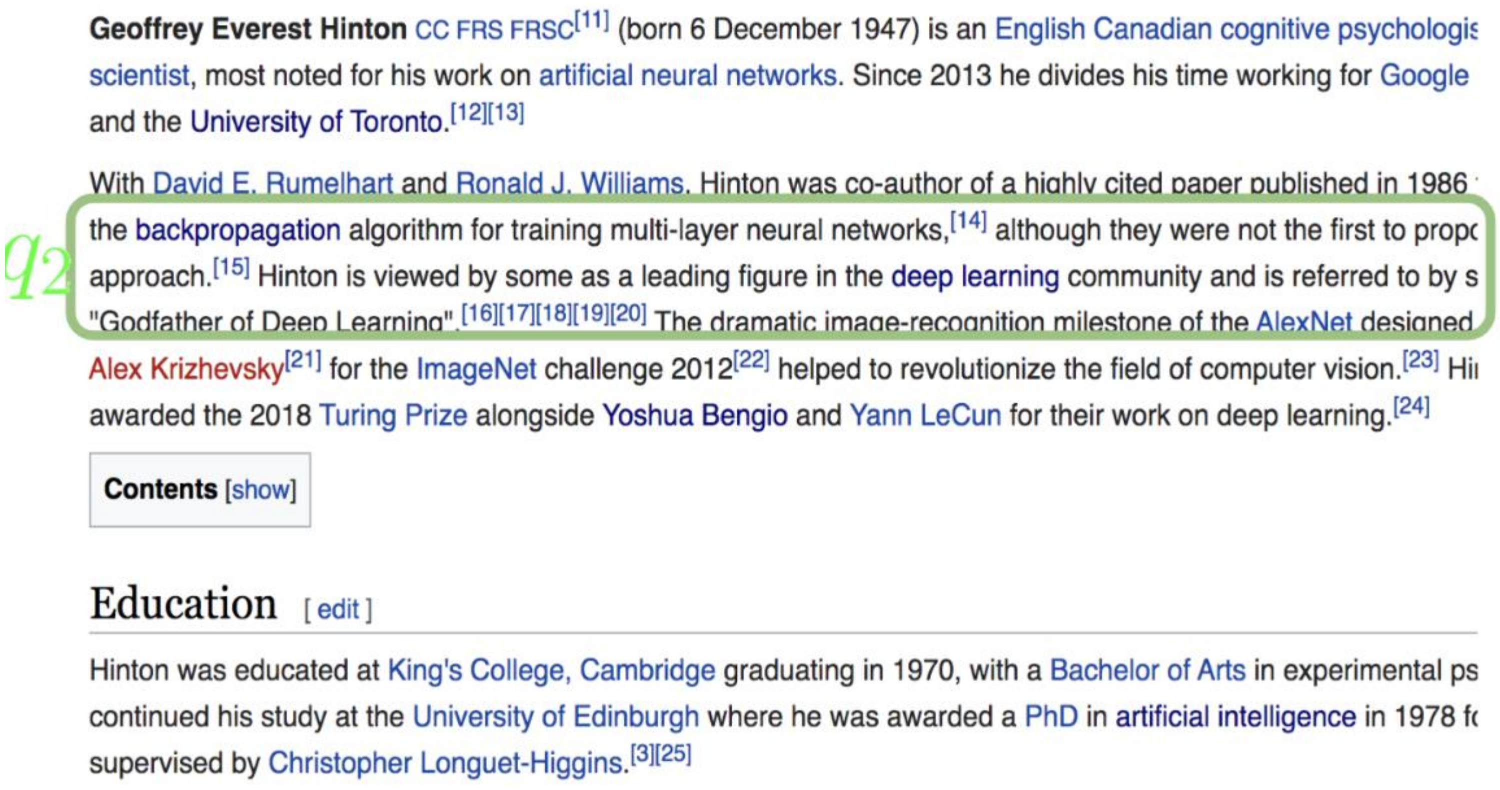
Career and research [edit]

After his PhD he worked at the University of Sussex, and (after difficulty finding funding in Britain)[26] the Universi
San Diego, and Carnegie Mellon University.[1] He was the founding director of the Gatsby Charitable Foundation
Neuroscience Unit at University College London,[1] and is currently[27] a professor in the computer science depar
University of Toronto. He holds a Canada Research Chair in Machine Learning, and is currently an advisor for th
Machines & Brains program at the Canadian Institute for Advanced Research. Hinton taught a free online course
Networks on the education platform Coursera in 2012.[28] Hinton joined Google in March 2013 when his compan
Inc., was acquired. He is planning to "divide his time between his university research and his work at Google".[29]

Hinton's research investigates ways of using neural networks for machine learning, memory, perception and sym
He has authored or co-authored over 200 peer reviewed publications.[2][30]

three pre-training tasks

Body First Selection (BFS) We propose BFS to capture semantic relationship outside of the local paragraph. Here, the query q_2 is a random sentence in the first section of a Wikipedia page, and the document d is a random passage from the same page (Figure 2). Since the first section of a Wikipedia article is often the description or summary of the whole page, we expect it to contain information central to the topic.



Geoffrey Everest Hinton CC FRS FRSC[11] (born 6 December 1947) is an English Canadian cognitive psychologis
scientist, most noted for his work on artificial neural networks. Since 2013 he divides his time working for Google
and the University of Toronto.[12][13]

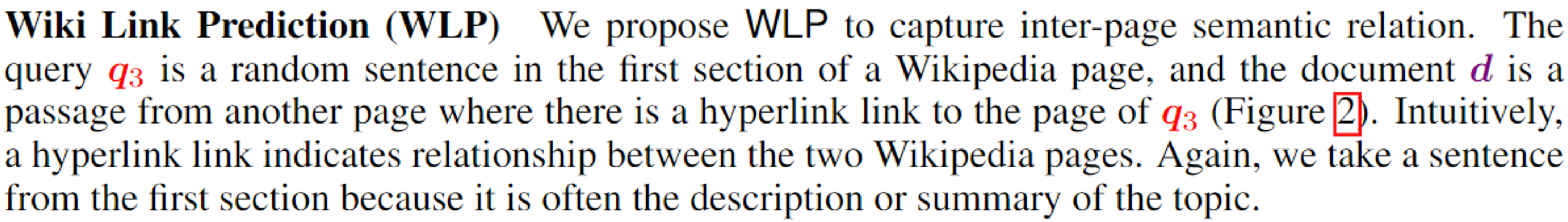
With David E. Rumelhart and Ronald J. Williams, Hinton was co-author of a highly cited paper published in 1986
the backpropagation algorithm for training multi-layer neural networks,[14] although they were not the first to propc
approach.[15] Hinton is viewed by some as a leading figure in the deep learning community and is referred to by s
"Godfather of Deep Learning".[16][17][18][19][20] The dramatic image-recognition milestone of the AlexNet designed
Alex Krizhevsky[21] for the ImageNet challenge 2012[22] helped to revolutionize the field of computer vision.[23] Hi
awarded the 2018 Turing Prize alongside Yoshua Bengio and Yann LeCun for their work on deep learning.[24]

Contents [show]

Education [edit]

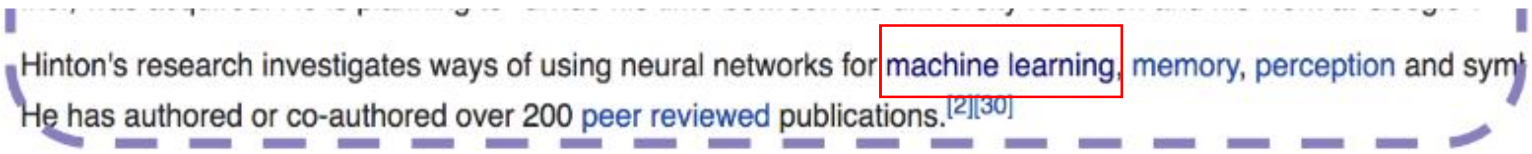
Hinton was educated at King's College, Cambridge graduating in 1970, with a Bachelor of Arts in experimental ps
continued his study at the University of Edinburgh where he was awarded a PhD in artificial intelligence in 1978 fc
supervised by Christopher Longuet-Higgins.[3][25]

three pre-training tasks



Wiki Link Prediction (WLP) We propose WLP to capture inter-page semantic relation. The query q_3 is a random sentence in the first section of a Wikipedia page, and the document d is a passage from another page where there is a hyperlink link to the page of q_3 (Figure 2). Intuitively, a hyperlink link indicates relationship between the two Wikipedia pages. Again, we take a sentence from the first section because it is often the description or summary of the topic.

d1

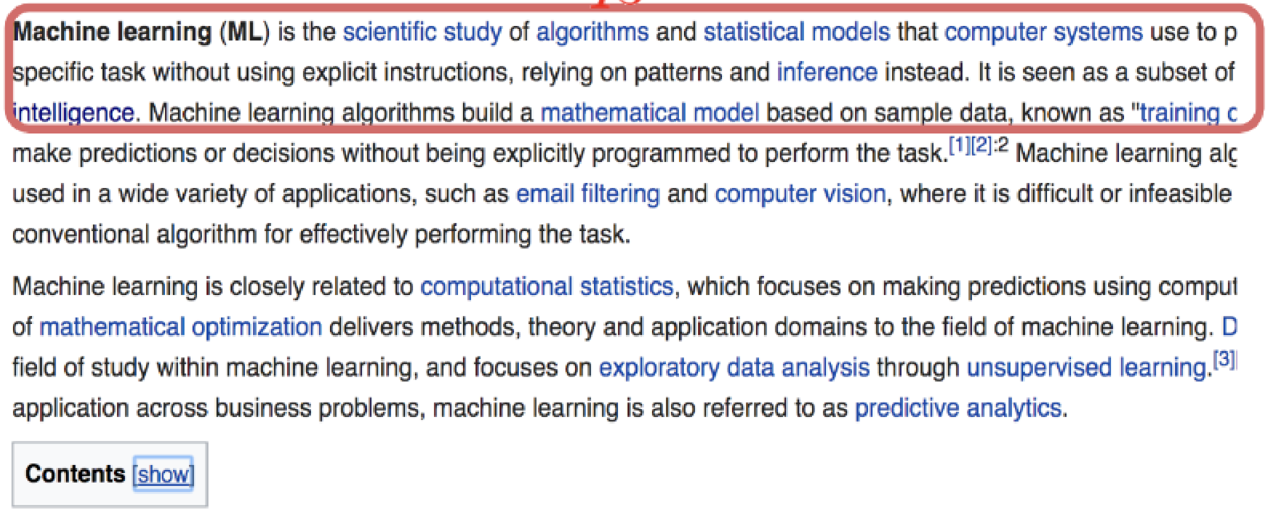


Hinton's research investigates ways of using neural networks for machine learning, memory, perception and sym

He has authored or co-authored over 200 peer reviewed publications.[2][30]

q_3

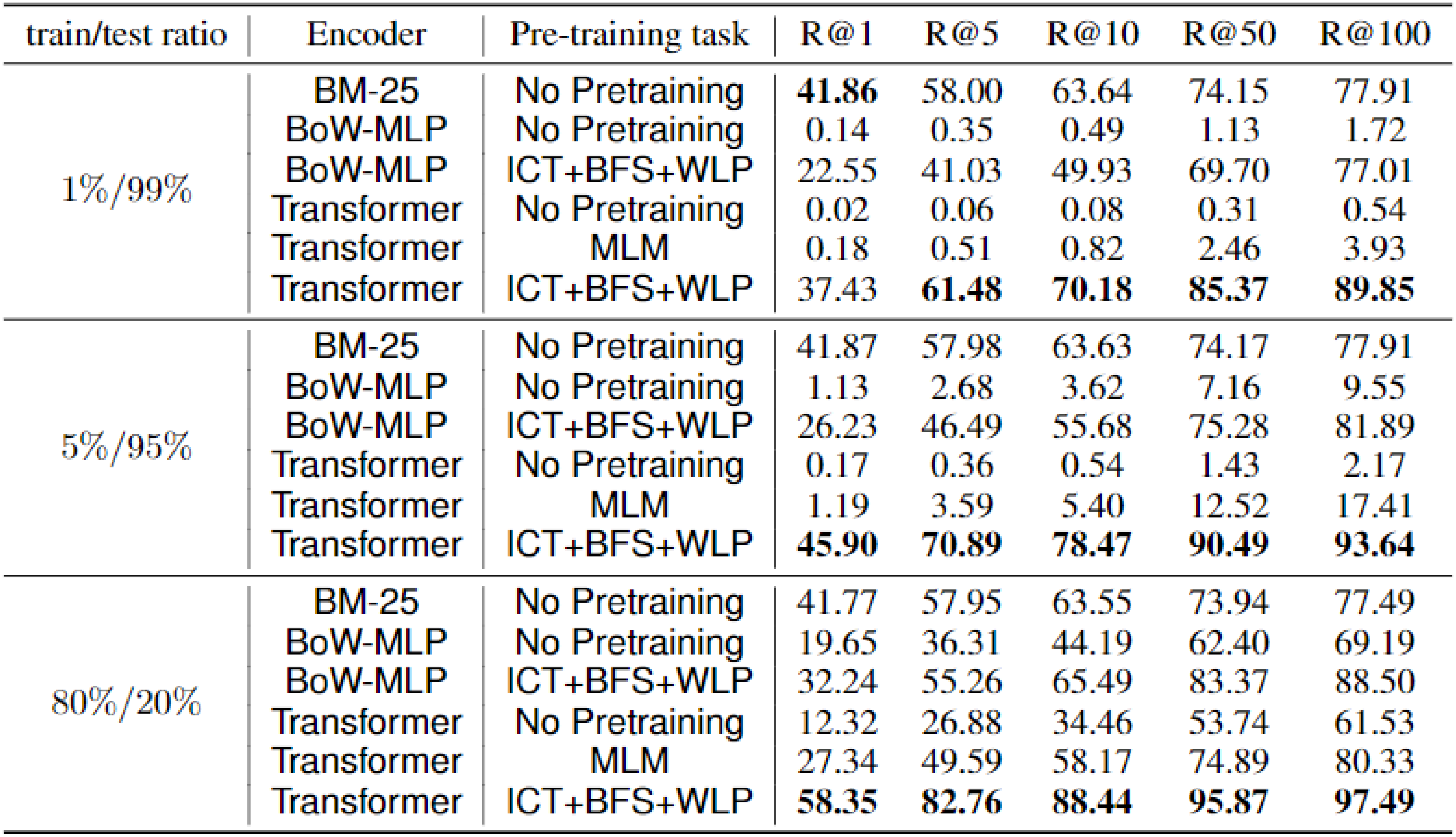
d2



Machine learning (**ML**) is the scientific study of algorithms and statistical models that computer systems use to p specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training c make predictions or decisions without being explicitly programmed to perform the task.[1][2]:2 Machine learning alg used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using comput of mathematical optimization delivers methods, theory and application domains to the field of machine learning. D field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning.[3] application across business problems, machine learning is also referred to as predictive analytics.

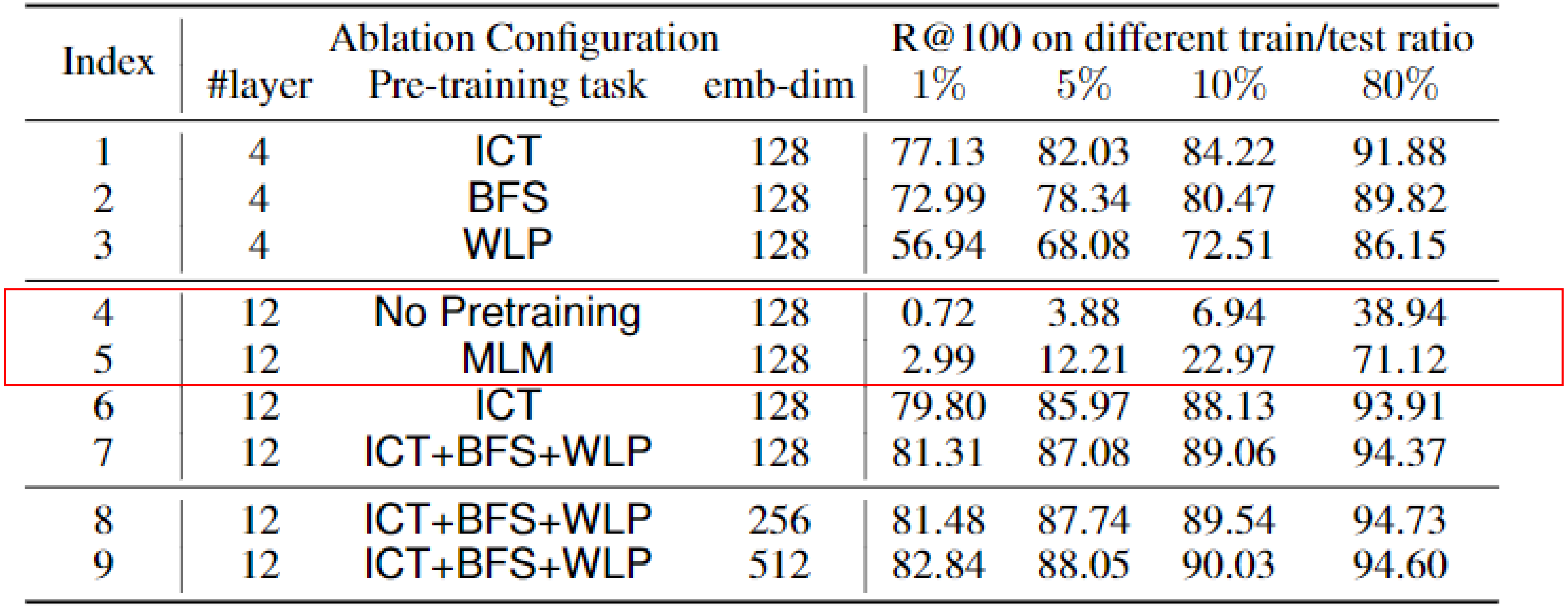
Experiment



train/test ratio	Encoder	Pre-training task	R@1	R@5	R@10	R@50	R@100
1%/99%	BM-25	No Pretraining	**41.86**	58.00	63.64	74.15	77.91
	BoW-MLP	No Pretraining	0.14	0.35	0.49	1.13	1.72
	BoW-MLP	ICT+BFS+WLP	22.55	41.03	49.93	69.70	77.01
	Transformer	No Pretraining	0.02	0.06	0.08	0.31	0.54
	Transformer	MLM	0.18	0.51	0.82	2.46	3.93
	Transformer	ICT+BFS+WLP	37.43	**61.48**	**70.18**	**85.37**	**89.85**
5%/95%	BM-25	No Pretraining	41.87	57.98	63.63	74.17	77.91
	BoW-MLP	No Pretraining	1.13	2.68	3.62	7.16	9.55
	BoW-MLP	ICT+BFS+WLP	26.23	46.49	55.68	75.28	81.89
	Transformer	No Pretraining	0.17	0.36	0.54	1.43	2.17
	Transformer	MLM	1.19	3.59	5.40	12.52	17.41
	Transformer	ICT+BFS+WLP	**45.90**	**70.89**	**78.47**	**90.49**	**93.64**
80%/20%	BM-25	No Pretraining	41.77	57.95	63.55	73.94	77.49
	BoW-MLP	No Pretraining	19.65	36.31	44.19	62.40	69.19
	BoW-MLP	ICT+BFS+WLP	32.24	55.26	65.49	83.37	88.50
	Transformer	No Pretraining	12.32	26.88	34.46	53.74	61.53
	Transformer	MLM	27.34	49.59	58.17	74.89	80.33
	Transformer	ICT+BFS+WLP	**58.35**	**82.76**	**88.44**	**95.87**	**97.49**

Table 3: Recall@k on SQuAD. Numbers are in percentage (%).

Experiment



Index	Ablation Configuration			R@100 on different train/test ratio			
	#layer	Pre-training task	emb-dim	1%	5%	10%	80%
1	4	ICT	128	77.13	82.03	84.22	91.88
2	4	BFS	128	72.99	78.34	80.47	89.82
3	4	WLP	128	56.94	68.08	72.51	86.15
4	12	No Pretraining	128	0.72	3.88	6.94	38.94
5	12	MLM	128	2.99	12.21	22.97	71.12
6	12	ICT	128	79.80	85.97	88.13	93.91
7	12	ICT+BFS+WLP	128	81.31	87.08	89.06	94.37
8	12	ICT+BFS+WLP	256	81.48	87.74	89.54	94.73
9	12	ICT+BFS+WLP	512	82.84	88.05	90.03	94.60

Conclusion

- Models with random initialization (No Pretraining) or the unsuitable token-level pre-training task (MLM) are no better than the robust IR baseline BM-25 in most cases.

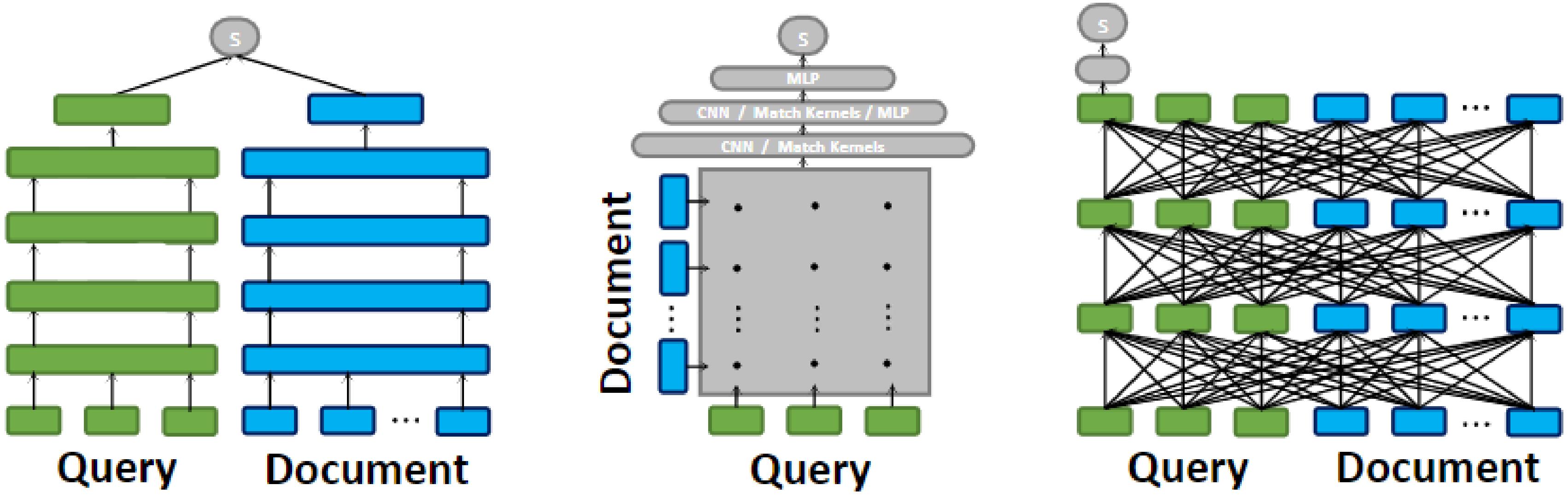
ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Omar Khattab
Stanford University
okhattab@stanford.edu

Matei Zaharia
Stanford University
matei@cs.stanford.edu

Motivation & Contribution

- A novel ***late interaction*** paradigm for estimating relevance between a query and a document



(a) Representation-based Similarity
(e.g., DSSM, SNRM)

(b) Query-Document Interaction
(e.g., DRMM, KNRM, Conv-KNRM)

(c) All-to-all Interaction
(e.g., BERT)

Methodology

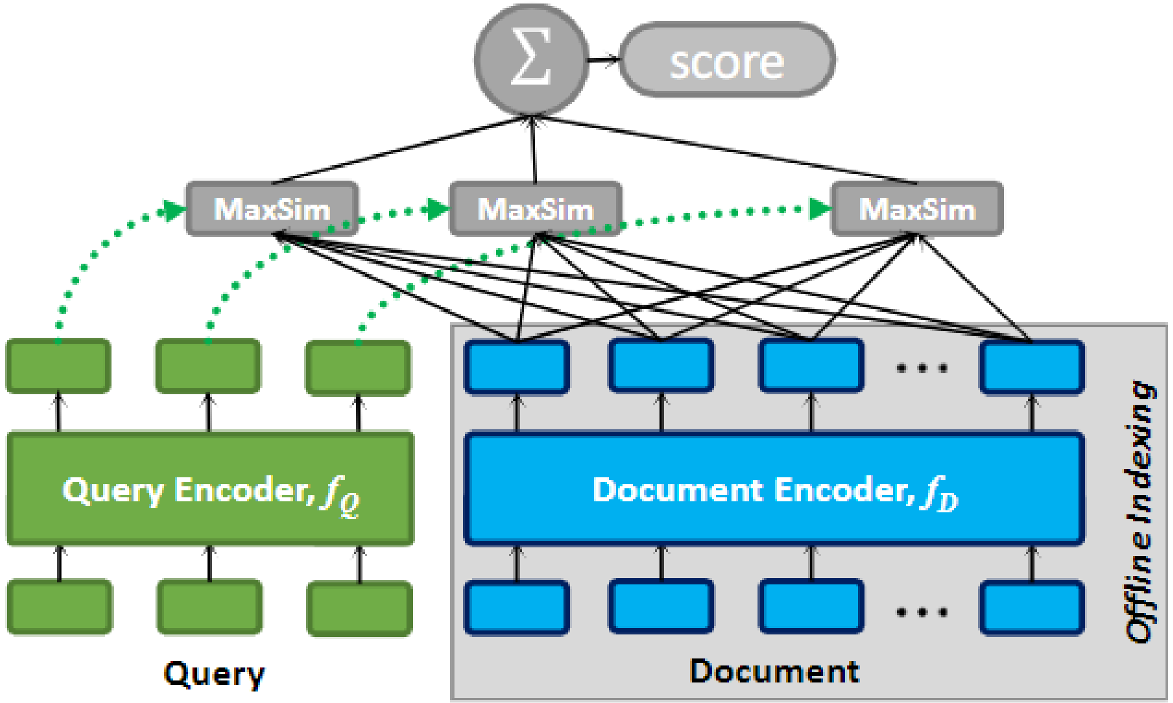


Figure 3: The general architecture of ColBERT given a query q and a document d.

Late Interaction

$$S_{q,d} := \sum_{i\in[|E_q|]} \max_{j\in[|E_d|]} E_{q_i} \cdot E_{d_j}^T$$

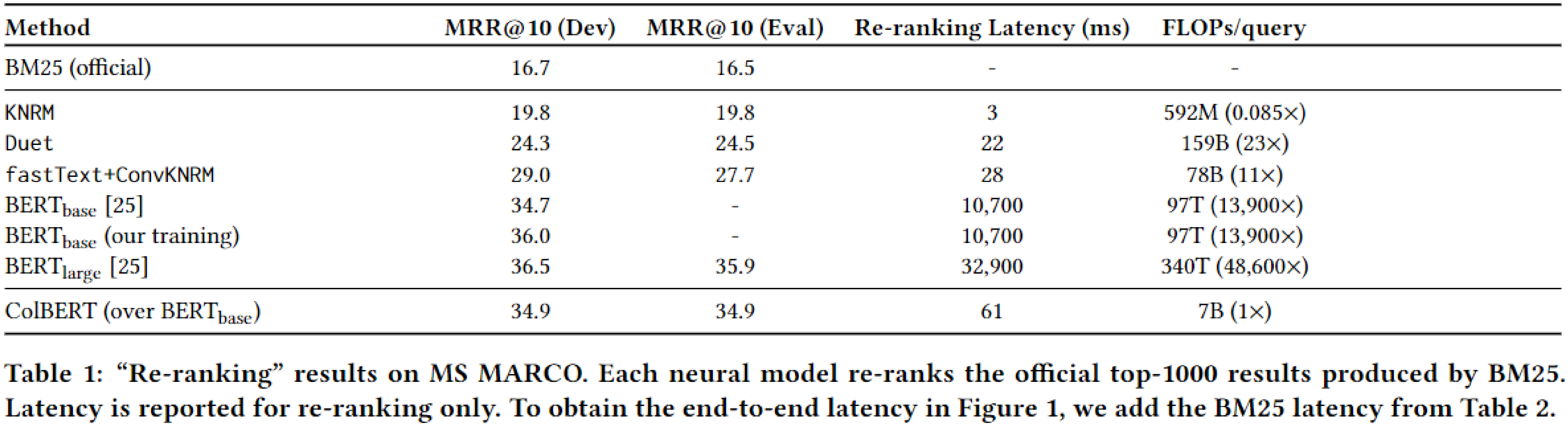
Learning

$$p_\theta(\boldsymbol{d}|\boldsymbol{q}) = \frac{\exp\left(f_\theta(\boldsymbol{q},\boldsymbol{d})\right)}{\sum_{\boldsymbol{d}'\in\mathcal{D}} \exp\left(f_\theta(\boldsymbol{q},\boldsymbol{d}')\right)},$$

two-stage to retrieve the top-k documents

- the first is an approximate stage aimed at filtering (faiss, facebook)
- the second is a refinement stage

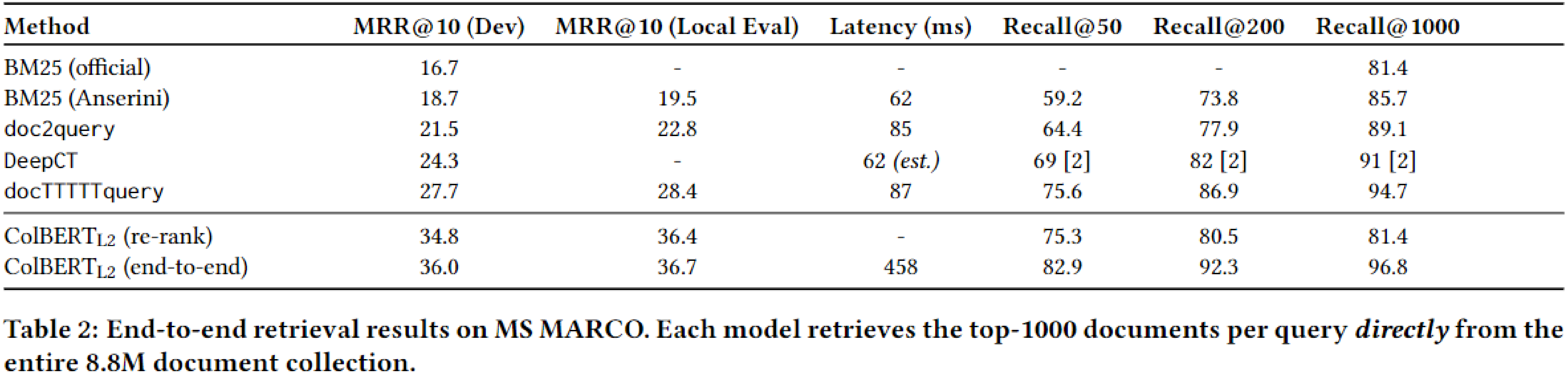
Experiment



Method	MRR@10 (Dev)	MRR@10 (Eval)	Re-ranking Latency (ms)	FLOPs/query
BM25 (official)	16.7	16.5	-	-
KNRM	19.8	19.8	3	592M (0.085×)
Duet	24.3	24.5	22	159B (23×)
fastText+ConvKNRM	29.0	27.7	28	78B (11×)
BERT$_{\text{base}}$ [25]	34.7	-	10,700	97T (13,900×)
BERT$_{\text{base}}$ (our training)	36.0	-	10,700	97T (13,900×)
BERT$_{\text{large}}$ [25]	36.5	35.9	32,900	340T (48,600×)
ColBERT (over BERT$_{\text{base}}$)	34.9	34.9	61	7B (1×)

Table 1: “Re-ranking” results on MS MARCO. Each neural model re-ranks the official top-1000 results produced by BM25. Latency is reported for re-ranking only. To obtain the end-to-end latency in Figure 1, we add the BM25 latency from Table 2.

Experiment



Method	MRR@10 (Dev)	MRR@10 (Local Eval)	Latency (ms)	Recall@50	Recall@200	Recall@1000
BM25 (official)	16.7	-	-	-	-	81.4
BM25 (Anserini)	18.7	19.5	62	59.2	73.8	85.7
doc2query	21.5	22.8	85	64.4	77.9	89.1
DeepCT	24.3	-	62 *(est.)*	69 [2]	82 [2]	91 [2]
docTTTTTquery	27.7	28.4	87	75.6	86.9	94.7
ColBERT$_{\text{L2}}$ (re-rank)	34.8	36.4	-	75.3	80.5	81.4
ColBERT$_{\text{L2}}$ (end-to-end)	36.0	36.7	458	82.9	92.3	96.8

Table 2: End-to-end retrieval results on MS MARCO. Each model retrieves the top-1000 documents per query *directly* from the entire 8.8M document collection.

Experiment

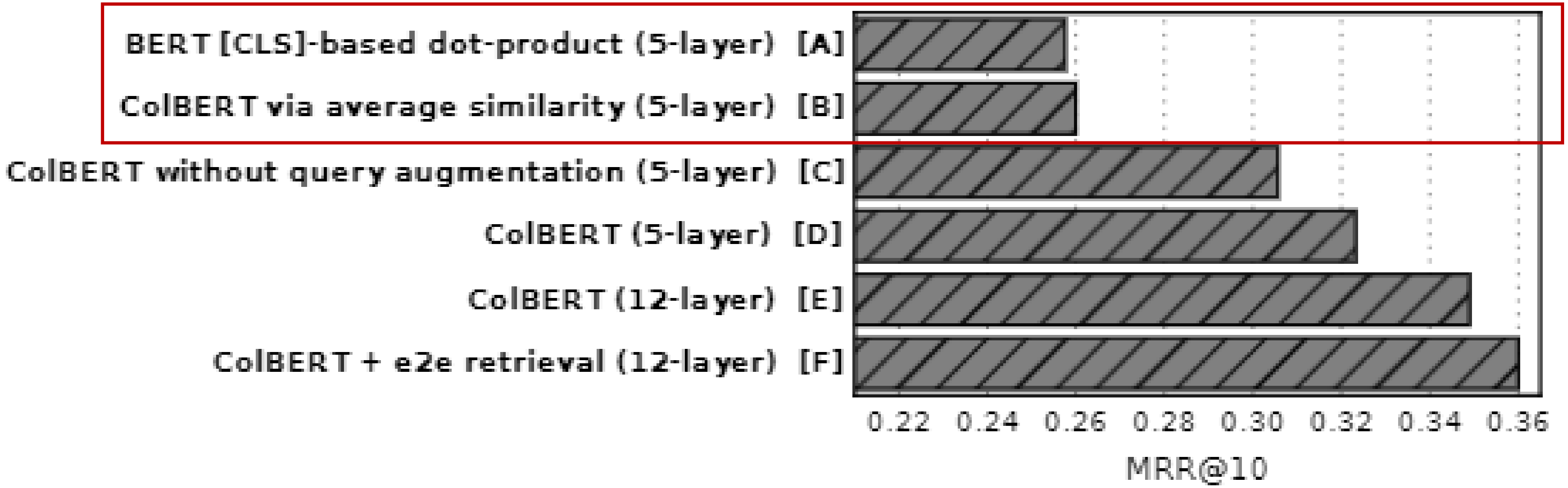


Figure 5: Ablation results on MS MARCO (Dev). Between brackets is the number of BERT layers used in each model.

Conclusion

- ColBERT can leverage the expressiveness of deep LMs while greatly speeding up query processing.
- Easy to implement
- SOTA

END!