Lightweight Composite Re-ranking for Efficient Keyword Search with BERT

Authors: Yingrui Yang, Yifan Qiao, Jinjin Shao, Xifeng Yan, Tao Yang
Speaker: Michael Peet

https://arxiv.org/pdf/2103.06499.pdf

Document Ranking Introduction

Problem:

Given a query q (multiple keywords) and candidate documents, form text features for each document and rank documents based on these features.

Solution:

First-stage retrieval model uses a simple algorithm (ex. BM25) to collect top documents.

The second or later stages use a more advanced algorithm to further rank top documents.

Motivation

- BERT based architectures have been shown to deliver high relevance for document search, but the computational complexity during inference is a bottleneck
- Simplified BERT based architectures allow document encoding to be precomputed, but still need to do transformer computation for queries during inference

Can BERT computation be fully alleviated for fast reranking at inference while maintaining reasonable relevance?

BECR (**BE**RT-based **C**omposite **R**eranking)

- Time Efficiency
  - Dual-Encoder framework to leverage precomputed documents (Similar to DC-BERT[1])
  - Compose query token representations with precomputed uni-gram and skip-n-gram embeddings
- Space Efficiency
  - Do not use all BERT layers for document embeddings
  - Compress embeddings with locality-sensitive hashing [2]
- Relevance
  - Offset approximation relevance with non-neural signals (ex. BM25)

[1] Y. Zhang, Ping Nie, Xiubo Geng, A. Ramamurthy, L. Song, and Daxin Jiang. 2020. DC-BERT: Decoupling Question and Document for Efficient Contextual Encoding. In SIGIR.
[2] Shiyu Ji, Jinjin Shao, and Tao Yang. 2019. Efficient Interaction-based Neural Ranking with Locality Sensitive Hashing. In WWW (San Francisco, CA, USA). ACM, New York, NY, USA, 2858–2864

Runtime Embedding Composition for Query Tokens

Pre-computed uni-grams/token pairs
(window size=3)

…
neural
ranking
model
(neural, ranking)
(neural, model)
(ranking, model)
…

Example Query
“Neural ranking model”

Embed: “Neural”

Fetch Pre-Computed embeddings

$$E(neural) = \frac{\frac{1}{4}e(neural) + \frac{1}{1}e(neural, ranking) + \frac{1}{2}e(neural, model)}{\frac{1}{4} + \frac{1}{1} + \frac{1}{2}}$$

Bernard J Jansen and Amanda Spink. 2006. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information processing & management 42, 1 (2006), 248–263

Online Composite Reranking

Strategy: Linear combination of deep and non-neural ranking signals

Benefits: Offset relevance loss due to previous approximations

$$S = S_{deep} + S_{lexi} + S_{others}$$

- Deep soft matching component
  - Token-token similarity between query and document terms in embedding space similar to in CEDR-KNRM [1]
- Lexical matching
  - Linear combination of BM25, word proximity [2, 3, 4], etc.
- Other Features
  - [CLS] Vector Hidden State (BERT)
  - pageRank score (Clueweb)

[1] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. SIGIR (2019).
[2] Jing Bai, Yi Chang, Hang Cui, Zhaohui Zheng, Gordon Sun, and Xin Li. 2008. Investigation of partial query proximity in web search. In WWW
[3] Jinjin Shao, Shiyu Ji, and Tao Yang. 2019. Privacy-aware Document Ranking with Neural Signals. SIGIR (2019). Jinjin Shao, Shiyu Ji, and Tao Yang. 2019. Privacy-aware Document Ranking with Neural Signals. SIGIR (2019).
[4] Jiashu Zhao and Jimmy Xiangji Huang. 2014. An Enhanced Context-sensitive Proximity Model for Probabilistic Information Retrieval. In SIGIR. 1131–1134

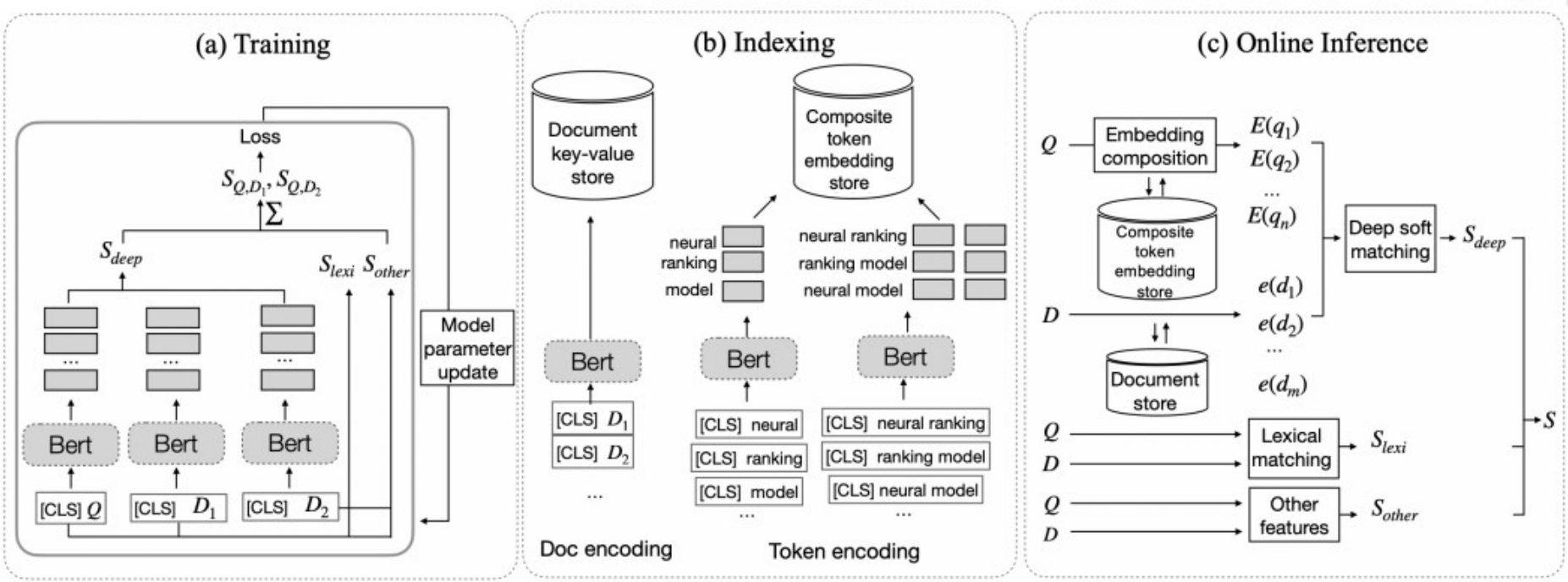
Storage Cost Reduction

- Embedding approximation using LSH compression [1]
- Document Embedding simplification using select layers of BERT
  - Took first 4 layers and last layer in BERT

50M ClueWeb pages with 857 (chars?) doc length	Document Embeddings	Unigram/word pair embeddings
Uncompressed	1711TB	37.9TB
Compressed	7TB (244x smaller)	152GB (249x smaller)

[1] Shiyu Ji, Jinjin Shao, and Tao Yang. 2019. Efficient Interaction-based Neural Ranking with Locality Sensitive Hashing. In WWW (San Francisco, CA, USA). ACM, New York, NY, USA, 2858–2864

Training, Offline Processing, and Online Inference



Evaluation Setup

- Datasets Used:
  - Robust04
  - ClueWeb09-Cat-B
  - MS MARCO Dev
  - TREC DL19, TREC DL20
- Relevance Metrics
  - NDCG@k
  - P@k
  - MRR
- Model comparison is done using a paired t-test comparing average query-level metrics
  - Results with statistical significance at confidence level 0.95 are marked

Scoring Component Significance

Model	ClueWeb09-Cat-B		
NDCG@k	3 ($\Delta \times 100$)	5 ($\Delta \times 100$)	10 ($\Delta \times 100$)
BECR	0.3754	0.3632	0.3417
No S_{deep}	0.2915 (-8.4)$^{\$}$	0.2948 (-6.8)$^{\$}$	0.2788 (-6.3)$^{\$}$
No S_{lexi}	0.2673 (-10.8)$^{\$}$	0.2643 (-9.9)$^{\$}$	0.2561 (-8.6)$^{\$}$
No S_{others}	0.3099 (-6.5)$^{\$}$	0.3032 (-6.0)$^{\$}$	0.2922 (-5.0)$^{\$}$

Performance Comparison - Relevance

Table 2: Relevance comparison of BECR against baselines. Significant improvements to ‡ConvKNRM, *BERT, §ColBERT, ¶CEDR-KNRM at 95% confidence level.

Model	ClueWeb09-Cat-B			Robust04			MSMARCO	DL19	DL20
	NDCG@5	NDCG@20	P@20	NDCG@5	NDCG@20	P@20	MRR@10 Dev	NDCG@10	NDCG@10
BM25	0.2351	0.2294	0.3310	0.4594	0.4151	0.3548	0.167	0.488	0.480
ColBERT (Ours)	0.2408	0.2400	0.2067	0.3809	0.3498	0.3074	0.355	0.701	0.674
ColBERT (from [5, 25])	0.2273 [5]	0.2365 [5]	0.2507 [5]	0.4031 [5]	0.3754 [5]	0.3254 [5]	0.349 [25]	–	–
CONV-KNRM	0.2869§	0.2735§	0.3698§	0.4742§	0.4501§	0.3349§	–	–	–
BERT-base	0.2853§	0.2612§	0.3764§	0.5160‡§	0.4514§	0.3983§	0.349	0.686	0.672
CEDR-KNRM (Ours)	0.3030‡§	0.2693§	0.3961§	0.5563*‡§	0.4637§	0.4249§	0.344	0.702	0.686
CEDR-KNRM (from [3, 34])	–	–	–	–	0.5381 [34]	0.4667 [34]	–	0.682 [3]	0.675 [3]
BECR⁻	0.3588¶*‡§	0.3066¶*‡§	0.4016§	0.5366*‡§	0.4635§	0.4045§	0.323	0.682	0.655
BECR	0.3632¶*‡§	0.3075¶*‡§	0.3987§	0.5349‡§	0.4656§	0.4005§	0.319	0.658	0.647

Relevance is comparable to other baseline models

Performance Comparison - Inference Speed

Table 3: Inference time and operation counts (FLOPs) to re-rank top 150 documents with average length 857 and query length n for ClueWeb09-Cat-B.

Model Specs.	n	FLOPs (ratio)	Time (ms) (ratio) GPU	CPU
KNRM	3	148M (5×)	1.3 (1×)	123.5 (5×)
	5	246M(5×)	1.6(0.5×)	312.8 (8×)
ColBERT	3	480M (15×)	13.7 (9×)	–
	5	779M (15×)	13.7 (4×)	–
BERT	3	12.2T (234k×)	4359 (2900×)	–
	5	12.2T (580k×)	4431 (1300 ×)	–
CEDR-KNRM	3	12.2T(234k×)	5577 (3700×)	–
	5	12.2T (580k×)	5601 (1700×)	–
BECR–, L=13, LSH	3	337M(11×)	6.3(4×)	–
	5	562M (11×)	10.5 (3×)	–
BECR–, L=5, LSH	3	287M (9×)	3.0(2×)	–
	5	478M (9×)	5.0 (2×)	–
BECR,L=13,LSH	3	81M (2.6×)	2.9 (2×)	65.3 (3×)
	5	136M (2.6×)	5.7 (2×)	117.7 (3×)
BECR,L=5,LSH	3	31M (1×)	1.5 (1×)	25.4 (1×)
	5	52M (1×)	3.3 (1×)	40.7 (1×)

Conclusion

- BERT computation can be fully alleviated for fast reranking at inference while maintaining reasonable relevance
- Paper code will be available at https://github.com/yingrui-yang/BECR