XCS224u: BM25S: Towards Efficient Context-Augmented Neural Information Retrieval System for Cloud Archives

Igor Y. Khomyakov
IKH Software, Inc.
ikh@ikhsoftware.com

Abstract

Inspired by ColBERT we have derived a novel approach information retrieval model BM25S that is based on classic BM25 model that operates in both input and semantic domains.

1 Introduction

In the evolving landscape of semantic information retrieval (IR), cross-encoders have emerged as the state-of-the-art. However, their scalability remains a challenge. On the other hand, bi-encoders, with particular emphasis on the ColBERT approach [2, 3, 4], stand at the forefront, offering a promising direction for semantic IR. Historically, BM25 has been recognized as the benchmark during the pre-semantic era, renowned for its efficiency, performance, and compact footprint.

In this study, we have endeavored to harmoniously integrate the strengths of these three methods.

Most of the components incorporated into our method have been de-risked by the studies we reference in our approach. However, our novel vocabulary in the semantic space still requires additional investigation.

Such a vocabulary can be composed in one of the following ways:

a) Processing extensive text corpora in an unsupervised setting and incrementally building N clusters, where N represents the size of the semantic vocabulary.
b) Selecting the semantic vocabulary randomly or positioning the terms equidistantly (cosine similarity wise).
c) Using a LSH (Locality Sensitive Hashing) function, which, however, doesn't allow for "soft" semantic terms, meaning it doesn't allow us to measure similarity.

In this work we conduct experiments with semantic vocabulary produced by a simple LSH function derived from [20].

2 Prior literature

2.1 Dense Passage Retrieval for Open-Domain Question Answering (Karpukhin et al. 2020) [1]

This study demonstrates that Dense Passage Retrieval (DPR) can significantly outperform traditional methods like TF-IDF and BM25 in open-domain question answering. Using embeddings from a dual-encoder framework on a limited set of question-passage pairs, the Dense Passage Retrieval method significantly exceeded a leading Lucene-BM25 system in accuracy. This approach, requiring less data and simpler training, suggests a potential shift from traditional sparse retrieval methods, setting new benchmarks in various QA datasets.

2.2 ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT (Khattab and Zaharia 2020) [2]

The Information Retrieval (IR) field has seen significant advancements in Natural Language Understanding (NLU) and the use of deep pre-trained language models like BERT for document ranking. However, these models, despite their efficacy, are computationally intensive. Addressing this challenge, the paper introduces "ColBERT," an innovative ranking model optimized for efficiency. By employing a "late interaction" approach, ColBERT encodes queries and documents separately and then quickly computes their relevance. This method drastically speeds up query processing without sacrificing result quality. The model's design allows for efficient indexing and retrieval from large datasets.

2.3 Relevance-guided Supervision for OpenQA with ColBERT (Khattab et al. 2021) [3]

Open-Domain Question Answering (OpenQA) systems aim to answer questions using large text datasets. Current methods face limitations in effectively retrieving relevant passages and in supervision. This work introduces ColBERT-QA, which employs the ColBERT neural retrieval model to enhance interaction between questions and passages. To optimize the training process, they propose relevance-guided supervision (RGS), allowing the retriever to iteratively refine its training approach. In their tests on the Natural Questions, SQuAD, and TriviaQA datasets, ColBERT-QA established new performance benchmarks for OpenQA systems.

2.4 Distilling Dense Representations for Ranking using Tightly-Coupled Teachers (Lin et al. 2020) [4]

This work introduces a method to enhance document ranking using dense representations by applying knowledge distillation to the late-interaction ColBERT model. By distilling ColBERT's advanced MaxSim operator into a simpler dot product, this work achieves single step Approximated Nearest Neighbor (ANN) search. The primary insight of this study is that closely linking the teacher and student models during distillation allows for improved distillation methods and better representation learning. This method boosts query response time, significantly cuts down ColBERT's storage needs, and only slightly compromises effectiveness. By merging the dense representations with sparse, this work almost matches the effectiveness of a much slower standard BERT cross-encoder re-ranker.

2.5 Improving Bi-encoder Document Ranking Models with Two Rankers and Multi-teacher Distillation (Choi et al. 2021) [5]

BERT-based Neural Ranking Models (NRMs) are classified into bi-encoders and cross-encoders. While bi-encoders are efficient, cross-encoders perform better. The study introduces Two Rankers and Multi-teacher Distillation (TRMD), a technique that merges knowledge from both encoders to produce an enhanced bi-encoder. Using TRMD, the bi-encoder, trained with teachers like monoBERT, showed a 6.8% average performance boost compared to baselines like TwinBERT and ColBERT. This research underscores the potential of TRMD in refining bi-encoder neural ranking models, a key area in the Information Retrieval (IR) domain.

2.6 SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking (Formal et al. 2021) [6]

In the field of Neural Information Retrieval (IR), the focus is shifting towards sparse representations to leverage benefits like exact term matching from bag-of-words models. However, these models face issues like vocabulary mismatch. This paper presents the SPLADE model, which combines both dense and sparse representations. Through logarithmic activation and sparse regularization, SPLADE enhances document expansion, offering a competitive alternative to dense models like BERT. With its balance of efficiency and effectiveness, SPLADE emerges as a promising direction for future exploration.

2.7 ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction (Santhanam et al. 2022) [7]

ColBERTv2 is a novel retriever introduced to optimize search functions in Neural Information Retrieval (IR). Unlike traditional models that use large multi-vector representations, ColBERTv2 efficiently captures token-level semantics using cluster centroids, reducing space requirements. This system further refines its performance by adopting enhanced supervision techniques from a cross-encoder system. Tested across 28 datasets, ColBERTv2 sets new benchmarks, delivering high-quality search results with a noticeably smaller space footprint.

2.8 Learning Cross-Lingual IR from an English Retriever (Li et al. 2022) [8]

The study introduces DR.DECR, a cutting-edge cross-lingual information retrieval system. Instead of the traditional two-step process involving query translation and monolingual retrieval, DR.DECR operates in a single step using knowledge distillation. Although machine translation-based methods showed higher initial effectiveness, the researchers combined its strengths with their new system. As a result, DR.DECR significantly outperformed baseline methods and set a new standard on the XOR-TyDi benchmark for cross-lingual retrieval.

2.9 PLAID: An Efficient Engine for Late Interaction Retrieval (Santhanam et al. 2022) [9]

This work introduces Performance-optimized Late Interaction Driver (PLAID) to enhance the late interaction process introduced by the ColBERTv2 model. By simplifying each passage into a bag of centroids and applying novel techniques, PLAID significantly reduces search times, while maintaining top-tier retrieval quality. Even on extensive datasets with 140 million passages, search latencies remain impressively low. Essentially, PLAID offers rapid, scalable search capabilities without sacrificing quality.

2.10 An Efficiency Study for SPLADE Models (Lassance and Clinchant 2022) [10]

The study focuses on enhancing the efficiency of SPLADE, an Information Retrieval (IR) model based on Pretrained Language Models (PLMs). While existing methods to adjust SPLADE's efficiency weren't sufficient, the paper introduces multiple techniques that significantly boost its efficiency and performance. As a result, the improved models closely match the latency of the traditional BM25 system yet maintain high performance. This marks a major advancement in neural ranking models, suggesting potential benefits for other systems and paving the way for future research.

2.11 DESSERT: An Efficient Algorithm for Vector Set Search with Vector Set Queries (Engels et al. 2022) [11]

This work focuses on vector set search using vector set queries. Existing solutions are too slow, especially for web applications. To address this, the study introduces DESSERT, a new search algorithm with promising theoretical and empirical results. When integrated into the ColBERT semantic search method, DESSERT achieves a 2-5x speedup with a slight drop in recall. Remarkably, DESSERT operates within a crucial sub-20ms latency, making it ideal for large-scale online deployment.

2.12 CITADEL: Conditional Token Interaction via Dynamic Lexical Routing for Efficient and Effective Multi-Vector Retrieval (Li et al. 2022) [12]

This work presents CITADEL, an innovative multi-vector retrieval method. Unlike traditional methods which are slow and storage-intensive, CITADEL uses a token routing approach to match query token vectors with similar document token vectors, optimizing computational efficiency. Remarkably, it's 40 times faster than ColBERT-v2 and 17 times faster than PLAID. By addressing redundancy and word-mismatch issues seen in other models, tests confirm CITADEL's superior speed and accuracy across different datasets.

2.13 Moving Beyond Downstream Task Accuracy for Information Retrieval Benchmarking (Santhanam et al. 2022) [13]

Neural information retrieval (IR) systems have seen notable advancements, but current benchmarks mainly focus on accuracy, neglecting real-world concerns like efficiency, latency, and cost. This work advocates for multidimensional leaderboards that assess systems on these parameters, alongside accuracy. Experiments on four IR systems demonstrate the value of such comprehensive evaluations. Despite its advocacy, the paper acknowledges potential limitations in chosen metrics and tested systems. The goal is to encourage the development of more holistic leaderboards that better reflect the diverse needs and values of the scientific community.

2.14 Rethinking the Role of Token Retrieval in Multi-Vector Retrieval (Lee et al. 2023) [14]

Multi-vector retrieval models, such as ColBERT, offer state-of-the-art performance in information retrieval by allowing token-level interactions between queries and documents. However, their complex three-stage inference process and non-linear scoring function, which is applied to all token vectors of candidate documents, makes retrieval slow and intricate. This study introduces XTR (ConteXtualized Token Retriever), aiming to streamline multi-vector retrieval. XTR has a novel objective function that emphasizes retrieving essential document tokens first. This enhanced token retrieval process allows XTR to rank candidates using only the retrieved tokens, resulting in a scoring stage vastly more efficient than ColBERT's.

3 Data

In this study, we utilized the MS MARCO Passage Ranking Dataset [16], specifically the "Train Triples Small" subset (triples.train.small.tar.gz). This subset comprises 39,780,811 triples; however, due to time constraints, we only analyzed the first 10,000 triples. This provided us with a sample of 10,000 queries and 20,000 passages of relatively short lengths, averaging ~80 tokens with a maximum of ~1,500 tokens per passage.

We employed this data to evaluate our model. The omission of training stage was justified as there was no requirement to fine-tune our large language model for the purposes of this study.

Each triple in the dataset consists of a query, a positive passage, and a negative passage. Both positive and negative passages were utilized in our assessment, with the evaluation metric being MRR@10, which focuses on the ranking of the positive document.

The MRR@10 metric specifically measures the positive outcome—that is, the relative ranking position of the positive document.

For our future research, we aim to develop a metric that assesses the ranking based on the negative label, determining the prominence of the negative document in the rankings, in other words a Failure metric. Our objective is to amalgamate the assessments based on both positive and negative labels into a comprehensive metric that reflects the full spectrum of the ranking process.

4 Models

In this study, we employed a novel BM25S model which is a BM25 model enhanced with "semantic" tokens generated using the BERT Base Cased (bert-base-cased) model, which features a 768-dimensional embedding and supports up to 2048 tokens. BERT tokenization was utilized to preprocess the text to feed both the BM25 model and the BERT transformer.

5 Experiments

To run the experiments, we implemented and published under GPLv3 license bm25s.py [21]. At the first stage the code ingests the MS MARCO Passage Ranking Dataset [16] triples (triples.train.small.tar.gz) into a Sqlite3 database.

We limited the experiment to first 10,000 triples due to time constraints. For each triple it tokenized the query and the two passages using BERT tokenizer and created Term Frequency and Inverted Documented Frequency indices to support classic BM25 model in input token domain. One difference is that we used BERT tokenization to feed classic BM25 model.

The Sqlite3 database that holds TF and IDF indices for both input and semantic domains takes 91 MB of disk space.

In parallel we ran the same documents through BERT inference to obtain the embedding vectors that correspond to input tokens (we threw away the vectors that correspond to [CLS] and [SEP] tokens). Then we applied an LSH algorithm that we derived from [20] to the embedding vectors to arrive at a sequence of integer tokens in semantic domain that correspond to the input tokens. We employed LSH for 16 bits which potentially can produce a vocabulary of 65K tokens (2 times bigger than BERT token vocabulary).

Then we used the semantic tokens and created Term Frequency and Inverted Documented Frequency indices for semantic domain to support BM25 in semantic domain. In this version we built one set of BM25 indices, however, we separated input from semantic tokens by using positive integer ids for input tokens and negative integer ids for semantic tokens.

Consequently we implemented BM25s function that given a query and parameter K, uses the same method to tokenize the query into input and semantic sets of tokens, and runs classic BM25 algorithm on both sets simultaneously using pre-built indices to retrieve K documents with the highest BM25 scores, and also returns RR@K metric for the positive label.

The current implementation gives equal weight to input and semantic domains, however, in our future work we plan to introduce a parameter S that will allow the users to shift the weight to input or semantic domains. With S=0, the algorithm will work as classic BM25 without semantic tokens. With S=1, the input tokens will be ignored completely. The current experiment uses S=0.5.

Our implementation of BM25 throws away all tokens with non-positive BM25 score.

The whole BM25 retrieval implemented using Sqlite3 as follows:

```
with b as (select tf.did, tf.tid,
tf.tf * (1 + m.k1) / (tf.tf + m.k1 * (1 -
m.b + m.b * d.dl / m.avgdl)) * ln((m.n -
t.nw + 0.5) / (t.nw + 0.5)) bm25 from qtf
join t using (tid) join tf using(tid)
join d using(did) join m where qtf.did in
(qid)) select did, sum(bm25), text from b
join d using (did) where bm25 > 0 group
by did order by 2 desc limit (K)
```

We ran 9,009 labeled queries and yilded MRR@10 = 0.4511860273765051.

6 Analysis, Limitations and Future work

Drawing inspiration from ColBERT, we have developed a novel method which we have termed BM25S, with 'S' denoting the semantic domain. However, we have not conducted a thorough comparison of our approach to ColBERT's methodology or to other related methods. To create the semantic vocabulary, we implemented a basic Locality Sensitive Hashing (LSH) algorithm using random planes and cosine similarity [20], but the properties of this algorithm have yet to be fully investigated. As an alternative, we could consider a soft matching vocabulary in which each semantic token is incorporated into the BM25 algorithm with a weight proportional to its cosine similarity. Additionally, we could explore the use of a sequence-to-sequence model to transform input tokens into semantic tokens. Due to time constraints, our study did not include an analysis of longer documents.

For our future work, we have reserved the following topics:

a) Conduct a more comprehensive investigation to determine the novelty of our BM25S approach and to ascertain whether a similar approach has been previously implemented and if there is accumulated experience with it.
b) Conclude the efficiency analysis of our approach BM25S approach.
c) Implement parameter S to allow to control contribution of input vs. semantic domain.
d) Consider augmenting BM25 in semantic domain using “soft” vocabulary based on cosine similarity.
e) The influence of overlapping passages on efficiency, and how easily such overlaps can be discounted from BM25 indices?
f) The potential for improved ranking efficiency when considering relationships between documents. For instance, if one document references another, or if two documents share the same author, there may be a higher likelihood of relevance between them.
g) The potential for improved ranking efficiency when accounting for the significance of different document sections, such as the title, abstract, introduction, conclusion, and body. It's possible, for example, that the title and abstract may carry more weight.
h) The potential use of passages from the same document in self-supervised training, e.g., to discern the relevance of passages in an unsupervised setting. This could aid in constructing a relevance graph between documents, further enhancing ranking.
i) Identifying other unsupervised sources of relevance information. Analogous to how transformer language models leverage vast text corpora, we must ask where we might find extensive corpora of "relevance" relationships—or a proxy thereof—to harness in an unsupervised context.
j) Whether utilizing sequence-to-sequence encoder-decoder transformers, such as T5, UniLM, BART, and PEGASUS, could heighten efficiency. The underlying rationale is their ability to directly translate from the input token domain into our semantic vocabulary without taking similarity weights into account, however.
k) Assessing the benefits of representing semantic sentences differently than a token sequence. For instance, could a linguistic sentence be depicted as a graph?
l) Assessing the benefits of distilling a cross-encoder to further fine-tune a bi-encoder in an unsupervised environment.
m) Evaluating various language models, such as ROBERTA and ELECTRA, for suitability in ranking tasks.
n) Assessing the benefits of employing distinct models for indexing and the initial stage (bi-encoder) versus the secondary stage (cross-encoder).

o) Investigating relationship of our approach and ColBERT's approach.
p) Include implementation for long documents and evaluate efficiency of the current method.
q) Consider implementing the second stage based on cross-encoder to re-rank the small set of the documents retrieved by the first BM25S stage.
r) We aim to develop a metric that assesses the ranking based on the negative label, determining the prominence of the negative document in the rankings, in other words a Failure metric. Our objective is to amalgamate the assessments based on both positive and negative labels into a comprehensive metric that reflects the full spectrum of the ranking process.

7 Conclusion

Inspired by ColBERT, we have developed a novel information retrieval model, BM25S, which builds on the classic BM25 model and operates across both input and semantic domains.

Our contributions are as follows:

a) We introduced a novel BM25S model that operates in dual input and semantic domains.
b) We introduce a novel vocabulary in the semantic domain and subsequently integrate the retrieval stage with the BM25 method across both input and semantic domains.
c) We devised a method how to address the long document issue by splitting the document into short passages for encoding, and yet we maintain compact BM25 indices by aggregating them at the document level (was not implemented in this study).
d) Our BM25S method runs BM25 across two domains (input and semantic) in parallel, granting users the flexibility to prioritize either exact keyword matches or a more semantic search approach by adjusting weight coefficients (this parameter S is set at 0.5 currently).
e) We argue that our method, which employs a large pre-trained language model like BERT, does not necessitate fine-tuning of said model; this should not substantially impact the efficiency of our retrieval system. The underlying hypothesis is that the language model, having already been pre-trained on a vast corpus of text, has acquired a context-augmented semantic understanding of tokens, which is all that our system requires.
f) In the second retrieval stage, we plan to employ a cross-encoder. Given the semantically-rich nature of our first stage, reinforced by dual-domain (input and semantic) BM25 retrieval, we argue that retrieving a limited number of documents during the first stage should increase second-stage performance without compromising efficiency. We aim to rigorously evaluate this hypothesis and validate it with empirical evidence.
g) We propose the idea and algorithm for "soft" BM25 where the vocabulary is represented in embedding space by vectors, and we use cosine similarity to implement "soft" weighted matching when we build BM25 indices.
h) We propose to derive a metric that assesses the ranking based on the negative label, determining the prominence of the negative document in the rankings, in other words a Failure metric.

Authorship Statement

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

1 Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, Wen-tau Yih. 2020. *Dense Passage Retrieval for Open-Domain Question Answering.* arXiv:2004.04906

2 Omar Khattab, Matei Zaharia. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.* arXiv:2004.12832

3 Omar Khattab, Christopher Potts, Matei Zaharia. 2021. *Relevance-guided Supervision for OpenQA with ColBERT*. arXiv:2007.00814

4 Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin. 2020. *Distilling Dense Representations for Ranking using Tightly-Coupled Teachers.* arXiv:2010.11386

5 Jaekeol Choi, Euna Jung, Jangwon Suh, Wonjong Rhee. 2021. *Improving Bi-encoder Document*

Ranking Models with Two Rankers and Multi-teacher Distillation. arXiv:2103.06523

6 Thibault Formal, Benjamin Piwowarski, Stéphane Clinchant. 2021. *SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking.* arXiv:2107.05720

7 Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, Matei Zaharia. 2022. *ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction.* arXiv:2112.01488

8 Yulong Li, Martin Franz, Md Arafat Sultan, Bhavani Iyer, Young-Suk Lee, Avirup Sil. 2022. *Learning Cross-Lingual IR from an English Retriever.* arXiv:2112.08185

9 Keshav Santhanam, Omar Khattab, Christopher Potts, Matei Zaharia. 2022. *PLAID: An Efficient Engine for Late Interaction Retrieval.* arXiv:2205.09707

10 Carlos Lassance, Stéphane Clinchant. 2022. *An Efficiency Study for SPLADE Models.* arXiv:2207.03834

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

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

13 Keshav Santhanam, Jon Saad-Falcon, Martin Franz, Omar Khattab, Avirup Sil, Radu Florian, Md Arafat Sultan, Salim Roukos, Matei Zaharia, Christopher Potts. 2022. *Moving Beyond Downstream Task Accuracy for Information Retrieval Benchmarking*. arXiv:2212.01340

14 Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, Vincent Y. Zhao. 2023. *Rethinking the Role of Token Retrieval in Multi-Vector Retrieval.* arXiv:2304.019821 Lin, J., Nogueira, R., & Yates, A. (2021). *Pretrained Transformers for Text Ranking: BERT and Beyond*. arXiv:2010.06467

2 Omar Khattab, Matei Zaharia. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.* arXiv:2004.12832

3 Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, Matei Zaharia. 2022. *ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction.* arXiv:2112.01488

4 Omar Khattab, Christopher Potts, Matei Zaharia. 2021. *Relevance-guided Supervision for OpenQA with ColBERT*. arXiv:2007.00814

5 Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, Wen-tau Yih. 2020. *Dense Passage Retrieval for Open-Domain Question Answering.* arXiv:2004.04906

6 Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin. 2020. *Distilling Dense Representations for Ranking using Tightly-Coupled Teachers.* arXiv:2010.11386

7 Jaekeol Choi, Euna Jung, Jangwon Suh, Wonjong Rhee. 2021. *Improving Bi-encoder Document Ranking Models with Two Rankers and Multi-teacher Distillation.* arXiv:2103.06523

8 Thibault Formal, Benjamin Piwowarski, Stéphane Clinchant. 2021. *SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking.* arXiv:2107.05720

9 Keshav Santhanam, Omar Khattab, Christopher Potts, Matei Zaharia. 2022. *PLAID: An Efficient Engine for Late Interaction Retrieval.* arXiv:2205.09707

10 Carlos Lassance, Stéphane Clinchant. 2022. *An Efficiency Study for SPLADE Models.* arXiv:2207.03834

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

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

13 Keshav Santhanam, Jon Saad-Falcon, Martin Franz, Omar Khattab, Avirup Sil, Radu Florian, Md Arafat Sultan, Salim Roukos, Matei Zaharia, Christopher Potts. 2022. *Moving Beyond Downstream Task Accuracy for Information Retrieval Benchmarking*. arXiv:2212.01340

14 Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, Vincent Y. Zhao. 2023. *Rethinking the Role of Token Retrieval in Multi-Vector Retrieval.* arXiv:2304.01982

15 Yulong Li, Martin Franz, Md Arafat Sultan, Bhavani Iyer, Young-Suk Lee, Avirup Sil. 2022. *Learning Cross-Lingual IR from an English Retriever.* arXiv:2112.08185

16 Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B.,

Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., & Wang, T. (2018). *MS MARCO: A Human Generated MAchine Reading COmprehension Dataset*.arXiv:1611.09268

17 Dietz, L., Verma, M., Radlinski, F. and Craswell, N., 2017. TREC Complex Answer Retrieval Overview. In *TREC*.

18 Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. arXiv:1606.05250

19 Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*. arXiv:2104.08663

20 MingYu (Ethen) Liu (@ethen8181), Locality Sensitive Hashing (LSH) - Cosine Distance, http://ethen8181.github.io/machine-learning/recsys/content_based/lsh_text.html

21 Igor Y. Khomyakov (@ikhomyakov), BM25 model in token and semantic domains, https://github.com/ikhomyakov/bm25s