Resources for Brewing BEIR: Reproducible Reference Models and Statistical Analyses

Ehsan Kamalloo
ekamalloo@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Nandan Thakur
nandan.thakur@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Carlos Lassance*
carlos@cohere.com
Cohere
Grenoble, France

Xueguang Ma
x93ma@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Jheng-Hong Yang
j587yang@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Jimmy Lin
jimmylin@uwaterloo.ca
University of Waterloo
Waterloo, Canada

ABSTRACT

BEIR is a benchmark dataset originally designed for zero-shot evaluation of retrieval models across 18 different domain/task combinations. In recent years, we have witnessed the growing popularity of models based on representation learning, which naturally begs the question: How effective are these models when presented with queries and documents that differ from the training data? While BEIR was designed to answer this question, our work addresses two shortcomings that prevent the benchmark from achieving its full potential: First, the sophistication of modern neural methods and the complexity of current software infrastructure create barriers to entry for newcomers. To this end, we provide reproducible reference implementations that cover learned dense and sparse models. Second, comparisons on BEIR are performed by reducing scores from heterogeneous datasets into a single average that is difficult to interpret. To remedy this, we present meta-analyses focusing on effect sizes across datasets that are able to accurately quantify model differences. By addressing both shortcomings, our work facilitates future explorations in a range of interesting research questions.

CCS CONCEPTS

• **Information systems** → **Test collections**.

KEYWORDS

Reproducibility, Evaluation, Domain Generalization

ACM Reference Format:
Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2024. Resources for Brewing BEIR: Reproducible Reference Models and Statistical Analyses. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626772.3657862

*Work done while at Naver Labs Europe.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657862

1 INTRODUCTION

One recent conceptual innovation in information retrieval is the recognition that the "classic" task of *ad hoc* retrieval can be framed as a representation learning problem. Karpukhin et al. [21] demonstrated that transformers can be used to encode queries and documents into a dense vector space, where top-k retrieval translates into the problem of nearest neighbor search. This led to the development of many so-called dense retrieval models [17, 21, 22, 34, 35, 45]. Separately, Zamani et al. [48] showed that neural networks can be used to learn sparse representations of queries and documents that are amenable to retrieval using standard inverted indexes. Later, researchers applied transformers for learning these sparse lexical representations, which led to a long line of so-called sparse retrieval models [1, 6, 10, 11, 19, 24, 27].

Lin [25] pointed out that learned dense representations, learned sparse representations, and even traditional lexical retrieval models such as BM25 can be viewed as parametric variations of a bi-encoder architecture. In this design, both queries and documents are fed to "encoders" that generate vector representations. Retrieval boils down to the problem of efficiently finding the top-k most similar document representations given a query representation and a similarity function, usually the inner product.

The design of encoders in such a bi-encoder architecture is dictated primarily by two choices: (1) the basis of the vector space and (2) how the vector weights are assigned. For example, both dense models such as DPR and sparse models such as SPLADE use pretrained transformers to encode queries and documents into vectors; both take advantage of large amounts of manually labeled data. However, the critical difference is the representational basis of their vectors—DPR generates dense vectors, whereas SPLADE "projects" the scalar weights of each dimension back into the input vocabulary space, generating bag-of-tokens vectors. BM25 can be understood in this bi-encoder architecture as having a document "encoder" that was heuristically designed (the BM25 scoring function) and a query "encoder" that generates multi-hot vectors.

Viewing retrieval as representation learning not only helps us understand the relationship between different models, but immediately illuminates open research questions. The dominant approach today is based on supervised learning with (manually) labeled datasets such as the MS MARCO test collections [2]. This naturally begs the question: What happens when models are applied to out-of-distribution data? Examples include applying retrieval

models trained on one type of text (e.g., passages from the web) to another type of text (e.g., text from the medical or legal domain), or differences between training and test queries (e.g., keyword queries vs. well-formed natural language questions).

This is where BEIR (*Be*nchmarking *IR*) [40] comes in. BEIR is a benchmark for zero-shot evaluation of information retrieval models that enables exactly the types of explorations outlined above. For example, BEIR has shown that traditional lexical retrieval models such as BM25 remain competitive baselines—and in fact, the first dense retrieval models evaluated on BEIR were worse overall than BM25 in a zero-shot setting. With BEIR, researchers have discovered that sparse retrieval models appear to achieve better cross-domain generalization than dense retrieval models. The dataset has spurred entirely new lines of research—for example, on *unsupervised* representation learning [18], where BEIR served as the benchmark for demonstrating model effectiveness.

Nevertheless, we can identify two shortcomings in the state of the current BEIR ecosystem. First, the variety and complexity of modern neural retrieval models create barriers to entry for researchers who wish to explore the research questions that BEIR enables. Successfully executing an end-to-end retrieval run requires coordinating heterogeneous software components that differ depending on the model type. It would be desirable to have reproducible implementations of retrieval models that are easily accessible to everyone, particularly newcomers.

Second, there is no agreed-upon methodology to perform significance testing for BEIR, since it is organized as an aggregation of diverse, heterogeneous individual datasets. Although one could perform tests on each dataset, significance testing across the final macro-average would not be meaningful since the datasets diverge so much in corpus size, number of queries, number of judgments, and many other dimensions [7, 32, 38]. We feel that there is currently no good way to answer the question: Is one retrieval model significantly better than another across different domains?

The good news is that meta-analyses [38], borrowed from statistics, offer a robust solution to integrate information from multiple sources with the objective of deriving a holistic conclusion. Meta-analyses allow us to estimate the effect size for each task and then to compute a summary statistic by aggregating the individual effects. This approach has been previously proven effective in reporting results across various IR tasks [16, 37, 38]. It would be desirable if we could apply similar techniques to assess statistical significance on BEIR.

Contributions. This work builds on BEIR and aims to address the two main shortcomings discussed above. We make the following contributions:

- We share with the community reproducible implementations of five popular retrieval models for BEIR in the open-source Pyserini IR toolkit [28]. From our extensive documentation pages, an end-to-end retrieval run can be reproduced with only two clicks: copy and paste of a command-line invocation.
- We describe the methodological innovation of using radar charts to visualize the effectiveness of different retrieval models across the BEIR datasets. These visualizations allow a researcher to quickly pinpoint the source of gains and losses with respect to a baseline, providing an entry point for error analyses.

Dataset	#Q	#J	#Passages	Task	Domain
TREC-COVID	50	66,336	171,332	Bio-Medical IR	Bio-Medical
BioASQ	500	2,359	14,914,602		
NFCorpus	323	12,334	3,633		
NQ	3,452	4,201	2,681,468	QA	Wikipedia
HotpotQA	7,405	14,810	5,233,329		Wikipedia
FiQA-2018	648	1,706	57,638		Finance
Signal-1M (RT)	97	1,899	2,866,316	Tweet-Retrieval	Twitter
TREC-NEWS	57	15,655	594,977	News-Retrieval	News
Robust04	249	311,410	528,155		
ArguAna	1,406	1,406	8,674	Argument-Retrieval	Misc.
Touché 2020	49	2,214	382,545		
CQADupStack	13,145	23,703	457,199	Dup. Ques.-Retrieval	StackExc.
Quora	10,000	15,675	522,931		Quora
DBPedia	400	43,515	4,635,922	Entity-Retrieval	Wikipedia
SCIDOCS	1,000	29,928	25,657	Citation-Prediction	Scientific
FEVER	6,666	7,937	5,416,568	Fact Checking	Wikipedia
Climate-FEVER	4,681	4,682	5,416,593		Wikipedia
SciFact	300	339	5,183		Scientific

Table 1: Summary of the 18 datasets that comprise the BEIR benchmark. #Q and #J denote the total counts of queries and relevance judgments in the test split of each dataset.

- We present a robust comparison of baselines by conducting meta-analyses on BEIR. Our results are visualized in forest plots to highlight on what tasks models excel and where they falter.
- We explore variations of existing retrieval models that examine field indexing, wordpiece tokenization, sliding window techniques for handling long documents, and hybrid fusion. Analyses of these variants with radar charts provide additional insights into model effectiveness.

2 BEIR OVERVIEW

The BEIR benchmark, introduced by Thakur et al. [40], evaluates information retrieval systems across diverse combinations of tasks and domains. It originally targets the "zero-shot" retrieval setting, where evaluation occurs on tasks and domains without any training data or supervision signals. This benchmark drives innovation in more robust and adaptable retrieval methods, enabling researchers to explore their out-of-domain generalization capabilities.

BEIR encompasses a wide range of tasks, from traditional *ad hoc* retrieval tasks like the TREC 2004 Robust Track to more specialized tasks such as Natural Questions (NQ) [23], which involves retrieving English Wikipedia passages to answer natural language questions. Additionally, BEIR tasks include argument retrieval (e.g., ArguAna, Touché 2020) and fact checking (e.g., FEVER, SciFact), which are related to, but distinct from, traditional *ad hoc* retrieval. BEIR also spans various domains, including scientific articles, news, Wikipedia, tweets, and more. Furthermore, queries in BEIR vary widely in form and length, ranging from a few keywords [4] to paragraphs [41]. Table 1 summarizes the 18 datasets that comprise BEIR. The datasets range in the amount of relevance judgments available. A few datasets have "dense" judgments, such as TREC-COVID [36], with 66k judgments for 50 test queries, but many have "sparse" judgments, such as SciFact, with only 339 judgments. The corpora associated with the datasets also vary in size, some containing millions of passages, whereas others have only a few thousand. BEIR also standardizes its evaluation metric and uses nDCG@10 and

Recall@100 across all datasets to compare the effectiveness of each system on an equal footing. Individual scores are macro-averaged across all datasets for a final cumulative score.

The BEIR authors provide additional resources with the benchmark. They share a GitHub repository[1] that contains source code for the evaluation framework along with example usage. The code is written in Python and is available on PyPI (`pip install beir`).

3 RETRIEVAL MODELS

This work provides reproducible reference implementations of five different retrieval models for BEIR. These comprise a "bag-of-words" BM25 baseline, two learned dense retrieval models (TAS-B and Contriever), and two learned sparse retrieval models (uniCOIL without expansion and SPLADE). In this section, we provide an overview of these models as they are presented in the literature, but explore different model variants in Section 6.

3.1 Multi-Field BM25

Despite tremendous progress in neural retrieval, ranking using traditional lexical "bag-of-words" models such as BM25 remains a strong baseline.

The original BEIR paper presented a BM25 baseline using Elasticsearch. We refer to this as "multi-field" BM25 because it ingested the title and body of documents into separate fields (called "title" and "contents", respectively) in cases where the original corpus provided this information. For corpora that didn't, all content was ingested into the default "contents" field. Search was performed by generating a Lucene multi-field query that assigned both fields equal weight. For corpora that did not explicitly have titles, the multi-field queries yielded the same ranking as if only the main "contents" field had been indexed and queried.

Building a baseline using Elasticsearch has the disadvantage in that it exists as an out-of-process retriever, which creates additional friction for researchers who desire a simple development/evaluation cycle. This was discussed by Devins et al. [9], who noted that since Elasticsearch is built on the open-source Lucene search library, researchers could "bypass" the features offered by Elasticsearch to directly gain in-process access to retrieval capabilities. From the perspective of batch IR evaluations such as BEIR, this was expedient because the additional layers that Elasticsearch builds on top of Lucene provide little value to researchers.

A later iteration of the BEIR evaluation resources moved from Elasticsearch to the Pyserini IR toolkit, but this feature was never refined into a reproducible baseline that could be easily invoked by researchers. In this work, we complete this "packaging" and explore additional BM25 variants (see Section 6).

3.2 Learned Dense Retrieval Models

We examine two learned dense retrieval models. This class of models exhibits two key characteristics: use of dense semantic representations for retrieval and encoders for generating these representations that are trained with labeled datasets.

TAS-B. This is a BERT-based dense retrieval model proposed by Hofstätter et al. [17], where the primary innovation is a Balanced Topic Aware Sampling (TAS-B) strategy to assemble training batches for optimizing retrieval effectiveness in a data-efficient manner. It was one of the earliest dense retrieval models to successfully exploit knowledge distillation, using dual supervision from a cross-encoder model and ColBERT. TAS-B was one of the first dense retrieval models to be applied to BEIR, and was discussed in the original paper by Thakur et al. [40].

Contriever. This is a dense retrieval model proposed by Izacard et al. [18] that first applies retrieval-specific pretraining in an unsupervised manner (an Inverse Cloze Task variant) before fine-tuning with the MS MARCO passage dataset to optimize for retrieval effectiveness. Contriever also builds on a BERT backbone and was specifically designed to explore zero-shot domain transfer capabilities. At its introduction, it was among the most effective dense retrieval models available on the BEIR benchmark.

3.3 Learned Sparse Retrieval Models

We examine two learned sparse retrieval models. Like their dense counterparts, these models rely on an approach to retrieval based on representation learning that exploits labeled datasets. However, these models generate sparse lexical representations instead of dense semantic ones.

uniCOIL (noexp). This model, originally proposed by Lin and Ma [27], is a variant of COIL [13], where BERT is trained to assign scalar weights to document tokens based on manually labeled relevance data (the MS MARCO passage dataset) to optimize retrieval effectiveness. In the full setting, uniCOIL depends on a separate document expansion model [30], but here we use the "no expansion" (noexp) variant, which allows us to examine the domain transfer capabilities of a "basic" learned term weighting function.

SPLADE. This refers to a family of sparse retrieval models [10, 11] that learns both document/query expansion and term weighting with the help of a regularization factor to induce sparsity. More precisely, we use the SPLADE++ (CoCondenser-EnsembleDistil) model [11],[2] which as the name implies, uses distillation and the pretrained CoCondenser model [12]. Today, this model remains highly effective, particularly in a zero-shot setting.

4 MAIN RESULTS

The effectiveness of the five models presented in the previous section is shown in Table 2, with nDCG@10 in the left group of columns and Recall@100 in the right group of columns. Each row corresponds to one of the BEIR datasets, and the rows are ordered in the same manner as Thakur et al. [40].

4.1 Reporting Best Practices

We have seen previous papers report BEIR results using slightly different layouts and organizations, which make comparisons difficult. Moving forward, we offer a few best practices to promote consistency in how results are shared: We feel that presenting the datasets in rows and effectiveness metrics in columns feels more

[1] https://github.com/beir-cellar/beir

[2] https://huggingface.co/naver/splade-cocondenser-ensembledistil

Dataset	nDCG@10					Recall@100				
	BM25	uniCOIL	SPLADE	TAS-B	Contriever	BM25	uniCOIL	SPLADE	TAS-B	Contriever
TREC-COVID	0.656	0.640	0.727	0.505	0.596	0.114	0.111	0.128	0.090	0.091
BioASQ	0.465	0.477	0.498	0.371	0.383	0.715	0.731	0.739	0.598	0.607
NFCorpus	0.325	0.333	0.347	0.324	0.328	0.250	0.257	0.284	0.284	0.301
NQ	0.329	0.425	0.538	0.465	0.498	0.760	0.833	0.930	0.904	0.925
HotpotQA	0.603	0.667	0.687	0.584	0.638	0.740	0.798	0.818	0.728	0.777
FiQA-2018	0.236	0.289	0.347	0.296	0.329	0.539	0.553	0.631	0.582	0.656
Signal-1M	0.330	0.275	0.301	0.288	0.278	0.370	0.313	0.340	0.304	0.322
TREC-NEWS	0.398	0.374	0.415	0.394	0.428	0.422	0.357	0.441	0.454	0.492
Robust04	0.407	0.403	0.468	0.461	0.473	0.375	0.317	0.385	0.411	0.392
ArguAna	0.414	0.396	0.520	0.436	0.446	0.943	0.923	0.974	0.945	0.977
Touché 2020	0.367	0.298	0.247	0.222	0.204	0.538	0.485	0.471	0.526	0.442
CQADupStack	0.299	0.301	0.334	0.309	0.345	0.606	0.569	0.650	0.612	0.663
Quora	0.789	0.662	0.834	0.835	0.865	0.973	0.948	0.986	0.986	0.994
DBPedia	0.313	0.338	0.437	0.384	0.413	0.398	0.441	0.562	0.499	0.541
SCIDOCS	0.158	0.144	0.159	0.146	0.165	0.356	0.328	0.373	0.332	0.378
FEVER	0.753	0.812	0.788	0.733	0.758	0.931	0.955	0.946	0.945	0.949
Climate-FEVER	0.213	0.182	0.230	0.237	0.237	0.436	0.418	0.521	0.553	0.575
SciFact	0.665	0.686	0.704	0.644	0.677	0.908	0.912	0.935	0.894	0.947
Avg. nDCG@10	0.429	0.428	0.477	0.424	0.448	0.576	0.569	0.618	0.591	0.613

Table 2: Effectiveness results of five retrieval models across all 18 datasets in BEIR: nDCG@10 (left) and Recall@100 (right).

natural, and urge the community to also adopt this layout. The alternative of showing the different datasets in columns feels more awkward to us. Furthermore, we recommend that researchers order the rows exactly as Thakur et al. [40], which we have done here. Other reasonable alternatives, for example, alphabetical sorting, discard the "semantic grouping" of the datasets. Finally, it would be preferable if researchers evaluating on BEIR report results on *all* 18 datasets, as opposed to slightly different subsets that make results difficult to compare.

Encouraging consistency in the presentation of BEIR results is an important first step to gaining insight when comparing retrieval models. However, there is no hiding the fact that BEIR scores comprise a complex aggregation of diverse datasets, and the standard approach of comparing macro-averaged nDCG@10 scores (as we have done in the final row of Table 2) is deficient in many ways.

It is well known that averages often hide important individual differences, but teasing apart these differences from a large table of numbers such as Table 2 can be difficult. For example, the results show that uniCOIL and BM25 achieve a similar level of effectiveness overall (0.428 vs. 0.429), but what can we say about effectiveness on individual datasets? Glancing down the rows, we see many differences—some large, some small—so it is possible to conclude that although uniCOIL and BM25 are "about the same" averaged across the BEIR datasets, effectiveness on individual datasets differ. How can we gain more insight easily? The same question applies when comparing BM25 and TAS-B, where the average nDCG@10 scores are comparable. Consider the SPLADE and Contriever results: we see that both achieve a higher average across all the datasets, but is this due to consistent gains across many datasets or a few big gains? It's difficult to tell from Table 2.

4.2 Radar Charts

We present a potential solution to these challenges in terms of radar charts: Figure 1 shows visualizations comparing the effectiveness of the five retrieval models, with nDCG@10 and Recall@100, respectively. Each radar chart comprises 18 axes, arranged radially, in the same order as the rows in Table 2. The effectiveness of a model is plotted on each of the 18 axes and connected by line segments to form a polygon. The effectiveness of BM25, which serves as a baseline, is scaled to half of the radius of the entire chart area, so the effectiveness of BM25 is captured by the dotted polygon. The effectiveness of the other models on each dataset is scaled relative to BM25. That is, points further away from the center represent higher scores and points closer to the center represent lower scores, where the distance to the midpoint of the axis is proportional to the score difference with respect to BM25.

The radar charts allow us to easily compare the effectiveness of the models across all datasets, and differences that are obscured by averages come readily to light. Focusing on nDCG@10, consider the question above about BM25 vs. uniCOIL (orange): We can see that uniCOIL excels on HotpotQA and NQ in terms of nDCG@10, but otherwise achieves effectiveness that is either on par with BM25 or worse. In particular, on Signal-1M, Quora, and Touché 2020, uniCOIL is substantially worse. We see a similar situation with TAS-B, which is more effective on some datasets but performs terribly on others, most notably TREC-COVID, BioASQ, and Touché 2020. Inconsistent effectiveness is similarly observed with Contriever as well, even though on average the model scores higher than BM25. It appears that both dense retrieval models perform rather poorly on BioASQ and TREC-COVID, two datasets that focus on biomedical retrieval. For all the models examined, it appears that SPLADE

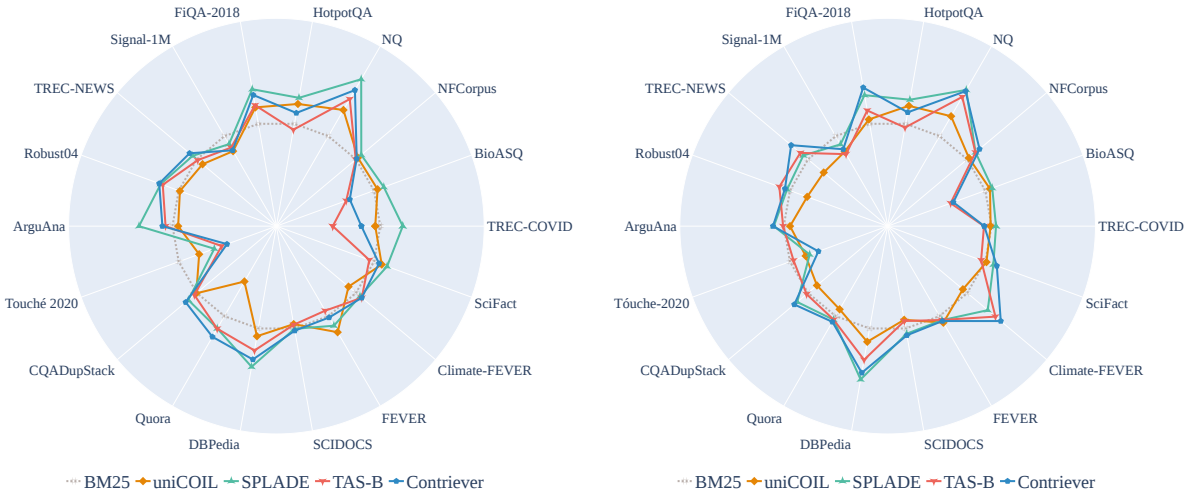

Figure 1: Radar charts comparing nDCG@10 (left) and Recall@100 (right) of five retrieval models across all 18 datasets in BEIR. The effectiveness of BM25 (dotted line) is scaled to half of the radius of the chart area, and the effectiveness of the other models is scaled accordingly.

exhibits the most consistent gains, with only a few datasets where nDCG@10 is worse than BM25.

Focusing on Recall@100 (Figure 1, right), we believe that the radar chart is similarly helpful in highlighting effectiveness differences between the models that are obscured by only looking at the means. In particular, conclusions drawn from the nDCG@10 scores differ from those based on Recall@100: both metrics are important, but for different reasons. Early precision metrics such as nDCG@10 directly quantify output quality if results from a first-stage retriever are directly presented to the user. On the other hand, Recall@100 quantifies the upper bound of reranking effectiveness. Based on the radar chart, we clearly see that neural models do *not* consistently increase effectiveness. We observe large gains—for example, all four models perform very well on NQ and DBPedia—but we also observe many cases where at least some of the neural models perform poorly—for example, on Signal-1M and BioASQ.

We have some explanations for these findings. First, let us consider the poor effectiveness of dense retrieval models on TREC-COVID and BioASQ. BioASQ involves scientific paper retrieval given a biomedical query that often involves specialized terminology (and similarly for TREC-COVID as well). As an example, consider the BioASQ query, "Is AZD5153 active in prostate cancer?" Here, "AZD5153" is a specialized biomedical term, which the model most likely has seen only rarely during training. This would cause issues for dense retrievers, as they are unable to represent rare terms well within the embedding space, and hence retrieval effectiveness would suffer. That is, poor effectiveness can be explained by domain shifts between training data (general web) and the test data (biomedical texts). On the other hand, sparse models such as SPLADE or uniCOIL produce representations that retain lexical matching, and thus are less impacted by domain shifts involving vocabulary differences. The relatively poor effectiveness of models on Signal-1M (tweets) can be explained similarly.

Second, we observe that all transformer models achieve big gains on NQ. Previous work has found that effectiveness on NQ and effectiveness on the TREC 2019 Deep Learning (DL) track have the highest correlation among all BEIR datasets in terms of scores [49] (that is, models that perform well on TREC DL also perform well on NQ). The close connections between the MS MARCO datasets and the TREC Deep Learning tracks suggest that transfer from MS MARCO is particularly advantageous for the NQ dataset. Since all models examined in this paper take advantage of MS MARCO, the big gains on NQ might be a data artifact and not a demonstration of robust domain transfer capabilities.

Finally, it appears that all transformer models perform worse than the BM25 baseline for Touché 2020, sometimes by sizeable margins. This, in fact, led to an in-depth exploration described in Thakur et al. [39], which untangled a number of interacting factors, but still concludes that BM25 outperforms many neural retrieval models.

Our point here is not to exhaustively explain all the effectiveness differences observed, although we do offer some analyses above. Instead, our contribution is a methodological tool (i.e., visualizations using radar charts) that provides a starting point for further analyses. Beyond support for error analyses, we believe that these radar chart visualizations are helpful for practitioners who might be interested in deploying neural models. From the perspective of real-world applications, it would make sense to ensure that a deployed model "performs no worse" than BM25, and thus these results lead us to conclude that none of the models are viable as a replacement yet, given that there are clearly situations where effectiveness is substantially lower.

5 REPRODUCIBLE IMPLEMENTATIONS

We provide reproducible implementations of the retrieval models discussed in this paper. At a high level, our goal is to make it as easy as possible for researchers to reproduce the results in Table 2. To be precise, here we are using reproducibility in the sense articulated by the ACM in its Artifact Review and Badging Policy,[3] characterized

[3]https://www.acm.org/publications/policies/artifact-review-and-badging-current

as "different team, same experimental setup". Specifically, "this means that an independent group can obtain the same result using the author's own artifacts."

Our reproducible implementations conform to the aspirational ideal of "two-click reproductions" described by Lin [26]. The motivation is that a user should be able to reproduce an experimental result with only two clicks: a "copy" and a "paste" from a documentation page. That is, the user will arrive at the same nDCG@10 and Recall@100 reported in Table 2.

To accomplish this, we leverage previous efforts and infrastructure investments in the Pyserini IR toolkit [28], which is built on Anserini [47]. Anserini is an IR toolkit built on the open-source Lucene search library, and like Lucene, it was written in Java. To provide compatibility with Python, the dominant language for building neural retrieval models today, we developed Pyserini, which provides Python bindings for Anserini as well as many other non-Java capabilities.

In our design, BM25 baselines and sparse retrieval models are directly implemented in Anserini with Lucene inverted indexes, exposed in Pyserini in Python. The dense retrieval models are implemented using the Faiss library [20] for efficient similarity search and clustering of dense vectors by Meta Research; for simplicity, we used flat indexes. Pyserini provides a uniform API to support retrieval using all the models, for example, abstracting over the Java-based implementation of retrieval using BM25 and the sparse retrieval models.

To provide a concrete example, performing a BM25 retrieval run over the test queries in the BioASQ corpus in BEIR can be accomplished by the following command:

```
python -m pyserini.search.lucene \
  --index beir-v1.0.0-bioasq.multifield \
  --topics beir-v1.0.0-bioasq-test \
  --output run.beir-multifield.bioasq.txt \
  --output-format trec \
  --batch 36 --threads 12 \
  --hits 1000 --bm25 --fields contents=1.0 title=1.0
```

The main driver program for searching Lucene inverted indexes is `pyserini.search.lucene`. In this example, we are using multi-field BM25, with equal weights set to both the "title" and "contents" fields (by default), specified using the `--fields` command-line argument. The remaining arguments are mostly self-explanatory, but we provide additional commentary:

The `--index` argument specifies a prebuilt inverted index for the BioASQ corpus that is stored on our group's servers. On the first invocation of the above command, the driver automatically downloads the index and caches it on the local machine. The `--topics` argument specifies the BioASQ test queries, which are already included as part of Pyserini. With this design, the user does not need to separately figure out where to download the indexes and queries to successfully reproduce a result.

Pyserini also includes all the components necessary to evaluate the retrieval results. In this case, the nDCG@10 score can be computed as follows:

```
python -m pyserini.eval.trec_eval \
  -c -m ndcg_cut.10 beir-v1.0.0-bioasq-test \
  run.beir-multifield.bioasq.txt
```

We provide a wrapper around the `trec_eval` package, and relevance judgments are included in Pyserini. Once again, this saves the user additional effort in needing to track down evaluation tools and relevance judgments from various web sources.

Modern IR evaluation methodology can be quite complex, but with our "two click reproductions", the two commands above will produce the results in Table 2. We have built a landing page[4] in the Pyserini documentation that provides an entry point to a "reproduction matrix" comprising all models and all datasets.

6 MODEL VARIANTS

The models presented in Section 3 cover the major approaches to neural retrieval today. In this section, we further examine variants that help us better understand some of the strengths and limitations of those models.

6.1 Multi-Field Indexing

A baseline "as simple as BM25" still presents a number of design decisions that may impact effectiveness in substantive ways. One such choice made by Thakur et al. [40] in the initial BEIR release is the use of multi-field indexing in the BM25 baseline. That is, the title and main body of each document were separately indexed, inserted into the "title" and "contents" fields, respectively. At search time, a multi-field (Lucene) query combines evidence from both fields (with equal weights).

What is the impact of this document structure on effectiveness? We can answer this question with a variant that we call "flat" BM25, where the title and the main body of each document are concatenated together and indexed in a single field. These results are shown in Table 3 under the "flat" column; the "multifield" column refers to the default BM25 configuration from Table 2. Following the analyses in Section 4, the radar chart visualization comparing "flat" to "multifield" BM25 is shown in Figure 2, with the latter configuration as the reference. The radar chart shows that in some cases "flat" is better (e.g., BioASQ, HotpotQA, and Touché 2020) and in other cases, it is worse (e.g., TREC-COVID, FEVER, and Climate-FEVER), but overall the differences are relatively small. It is hard to draw reliable conclusions, as these differences primarily stem from corpus organization, which obviously varies across the datasets.

Another important decision in building a BM25 baseline is the choice of tokenization and stemming. In Lucene, an abstraction called the analyzer is responsible for converting a sequence of bytes into a sequence of tokens. In all the BM25 variants discussed above, we used Lucene's default analyzer for English. In contrast, the sparse representation models use BERT's wordpiece vocabulary. Since the two vocabulary spaces are different, one might argue that comparisons are not fair. To examine these effects, we applied the wordpiece tokenizer to the "flat" BM25 condition.

The results are shown in the "flat-wp" column of Table 2, and visualized in the radar chart shown in Figure 2 (left). Once again, the differences are relatively small, but it does appear that wordpiece tokenization consistently degrades effectiveness. This occurs because wordpiece tokenization often chops long content words into shorter subwords that are polysemous, hence introducing noise.

[4] https://castorini.github.io/pyserini/2cr/beir.html

Task	BM25			TASB			Dense–Sparse Hybrid		
	multifield	flat	flat-wp	FirstP	MaxP (10/5)	MaxP (8/4)	Contriever	SPLADE	Hybrid
TREC-COVID	0.656	0.595	0.565	0.481	0.491	0.505	0.596	0.727	0.723
BioASQ	0.465	0.522	0.419	0.360	0.367	0.371	0.383	0.498	0.457
NFCorpus	0.325	0.322	0.314	0.319	0.321	0.324	0.328	0.347	0.348
NQ	0.329	0.305	0.305	0.463	0.463	0.465	0.498	0.538	0.552
HotpotQA	0.603	0.633	0.593	0.584	0.584	0.584	0.638	0.687	0.684
FiQA-2018	0.236	0.236	0.218	0.300	0.295	0.296	0.329	0.347	0.361
Signal-1M	0.330	0.330	0.350	0.288	0.288	0.289	0.278	0.301	0.296
TREC-NEWS	0.398	0.395	0.361	0.377	0.398	0.394	0.428	0.415	0.468
Robust04	0.407	0.407	0.377	0.428	0.455	0.461	0.473	0.468	0.493
ArguAna	0.414	0.397	0.364	0.427	0.433	0.436	0.446	0.520	0.517
Touché 2020	0.367	0.442	0.466	0.163	0.215	0.222	0.204	0.247	0.233
CQADupStack	0.299	0.302	0.295	0.314	0.309	0.309	0.345	0.334	0.354
Quora	0.789	0.789	0.730	0.835	0.835	0.835	0.865	0.834	0.858
DBPedia	0.313	0.318	0.284	0.384	0.384	0.384	0.413	0.437	0.449
SCIDOCS	0.158	0.149	0.138	0.149	0.147	0.146	0.165	0.159	0.172
FEVER	0.753	0.651	0.658	0.700	0.724	0.733	0.758	0.788	0.791
Climate-FEVER	0.213	0.165	0.158	0.228	0.241	0.237	0.237	0.230	0.265
SciFact	0.665	0.679	0.672	0.643	0.645	0.644	0.677	0.704	0.715
Avg. nDCG@10	0.429	0.424	0.404	0.414	0.422	0.424	0.448	0.477	0.485

Table 3: Effectiveness (in terms of nDCG@10) of model variants across all 18 datasets in BEIR.

6.2 Searching Long Documents

One well-known issue with retrieval methods built on pretrained transformers is that the underlying models have length restrictions in input text; see Lin et al. [29] for extensive discussions of this topic. The two commonly adopted solutions are to either encode only the first N tokens in each document or to segment a longer document into passages and encode each passage independently. In the terminology of Dai and Callan [8], these approaches are known as FirstP and MaxP, respectively. With MaxP, multiple representations are generated per document, and at retrieval time, the maximum of the passage scores is taken as the score of the document; this heuristic itself dates back to at least the 1990s [5, 15].

Curiously, most papers that report evaluations on BEIR contain no explicit discussions about how long documents were processed. Based on informal communications with model developers and examination of available open-source implementations, it appears that most researchers apply the FirstP approach. That is, they simply truncate each document to the first N tokens (where N varies by model). This, of course, begs the question of whether different techniques for searching long documents "make a difference".

We conducted experiments to answer this question. Given the vast design space of options for segmenting longer documents into shorter passages, we built on previous work that explored some of the design choices. Following Pradeep et al. [33] and later work by Ma et al. [30], we decided to segment documents into sliding windows of n sentences. Based on their previous explorations, we examined two configurations: a sliding window of 10 sentences with a stride of 5 sentences, and an 8/4 combination. Passages based on sentences yield variable-length passages, in contrast to the obvious alternative of using fixed-length windows. However, sentence-based windows preserve natural discourse units and better encapsulate context that might be useful for determining relevance. For these experiments, we used the sentence chunker in spaCy (version 3.4.4).

Experimental results are presented in Table 3, shown for the dense retrieval model TAS-B; the corresponding radar chart is shown in Figure 2 (middle). The visualization makes it clear that MaxP does yield some gains—in 15 out of the 18 datasets—but the overall differences are small. Based on these results, we would argue that to evaluate future models, FirstP is "good enough", since methodological consistency is likely more important. That is, comparing FirstP on dense model A with MaxP on dense model B would introduce confusion and conflate unrelated factors.

6.3 Hybrid Fusion

One clear takeaway from Section 4 is that the effectiveness of zero-shot transfer for both learned dense and learned sparse representations is inconsistent across the 18 BEIR datasets. From the radar charts, we see some clear gains, for example, on NQ, but also cases where some models underperform, most notably, dense retrieval models on BioASQ. In such cases, hybrid fusion techniques can perhaps be helpful in combining evidence from different sources. While this general idea dates back several decades at least [3], more recent work has demonstrated that fusion between lexical and semantic representations work particularly well [14, 31]. Furthermore, the fusion of BM25 with dense [46] and sparse [10] representations has already been shown to be effective.

This work explored fusion techniques further, primarily to see whether we can combine multiple sources of evidence to achieve *consistent* gains across all BEIR datasets. We applied the simple dense–sparse hybrid fusion techniques described by Ma et al. [31] to combine Contriever and SPLADE, the most effective dense and

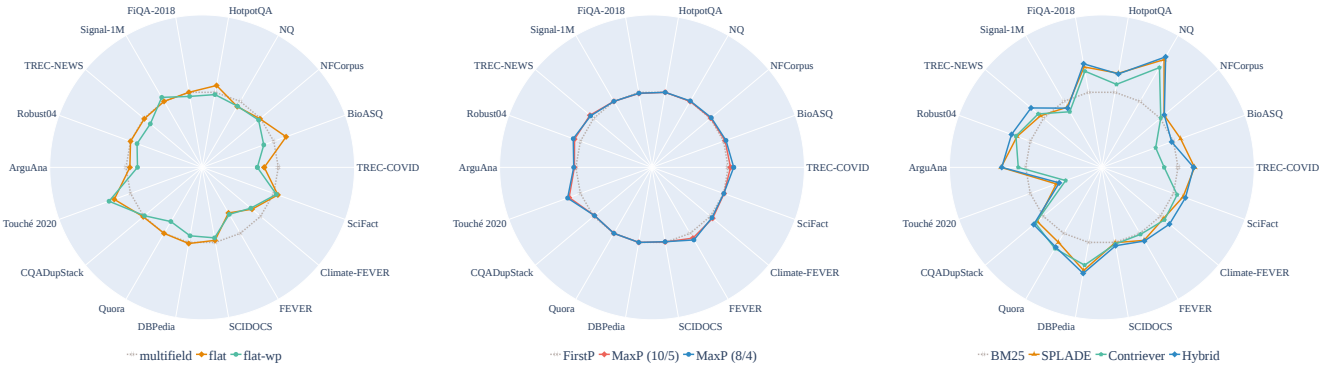

Figure 2: Radar charts visualizing the effectiveness (in terms of nDCG@10) of different model variants: BM25 variants (left), techniques for searching long documents (middle), and dense–sparse hybrids (right).

sparse retrieval models, respectively. Specifically, we first retrieved the top 1000 documents separately using each model. We then normalized the relevance scores from each source into the range [0, 1] and computed the final hybrid score as the average of the two scores to produce new rankings for evaluation.

Experimental results are presented in Table 3 in the rightmost column, "Hybrid". The nDCG@10 figures for the two sources, Contriever and SPLADE, are copied from Table 2 for convenience. We see that, in general, the hybrid approach improves over the best individual model, with four exceptions: BioASQ, Touché 2020, ArguAna, and Quora, although for the last two, the differences are quite small.

How does this fusion run compare to BM25? The radar chart visualization that answers this question is shown in Figure 2 (right), where we plot the effectiveness of Contriever, SPLADE, and our hybrid approach with BM25 as the reference. Table 3 shows more than a six-point gain on average, but more importantly, the radar chart visualization shows that the gains are consistent. We see that the hybrid beats BM25 on all but two datasets: Signal-1M and Touché 2020. In particular, the visualization makes it clear that SPLADE is able to compensate for the poor effectiveness of Contriever on BioASQ and TREC-COVID, and in cases where Contriever is more effective than SPLADE, the hybrid approach further boosts effectiveness. Simple score averaging seems to achieve the best of both worlds, and this dense–sparse hybrid appears to attain a level of robustness that none of the other models exhibit.

7 META-ANALYSES

The final contribution of this work is to illustrate the use of meta-analyses for comparing evaluation results on BEIR. The common practice for reporting results is to provide nDCG@10 scores for each task along with a macro-average that aggregates the individual scores, as we have done in Table 2. Models that achieve higher average scores are considered to be "better" in possessing out-of-domain generalization capabilities. Although easy-to-compute and prevalent (e.g., [42–44]), averaging scores across different datasets is inherently flawed because (1) the scores are not comparable [16, 37], (2) simple averages are susceptible to outliers [32], and (3) such an approach overlooks the intrinsic difficulty of individual datasets by ignoring effect sizes [7, 38]. Thus, simple averages fail to capture how well models perform in reality.

As a remedy to these issues, we turn to meta-analyses, which are designed to integrate evidence from multiple sources in order to arrive at a holistic conclusion [38]. To this end, an effect size is first determined for each task, before combining them using the random-effects model that assumes the true effect size varies across datasets. We opted for the raw mean difference to estimate effect sizes, following Soboroff [38] and Sertkan et al. [37]. Specifically, we used the Ranger toolkit [37] to compute confidence intervals for significance testing.

Here, we present a case study that illustrates an application of meta-analysis. In particular, our goal is to explore the impact of dense–sparse hybrid models on BEIR, since according to Table 3, the model appears to be the most effective. However, can we make stronger statements about the significance of effectiveness differences compared to the base SPLADE and Contriever models? Figure 3 presents forest plots visualizing the effect size and the confidence interval for each dataset, comparing the hybrid model against Contriever (on the left) and against SPLADE (on the right). The summary effect row shows that overall, the hybrid model significantly outperforms Contriever and SPLADE. However, this analysis shows that the hybrid approach is only significantly better than SPLADE alone in 14 out of the 29 cases, and 16 out of the 29 cases for Contriever alone.

While this specific exploration is only focused on comparing three models, our meta-analyses are able to reveal more insights than comparison of simple macro-averages. We advocate the use of such approaches to more definitively answer the question: Is this model really better than that one?

8 CONCLUSIONS AND FUTURE WORK

The BEIR benchmark provides an important instrument for evaluating the cross-domain robustness of retrieval models and has gained traction due to the growing recognition of retrieval as a form of representation learning. The efforts described in this paper address two shortcomings that we have identified with BEIR: challenges in reproducibility and in comparing results. Reproducible reference implementations in the Pyserini IR toolkit tackle the first challenge.

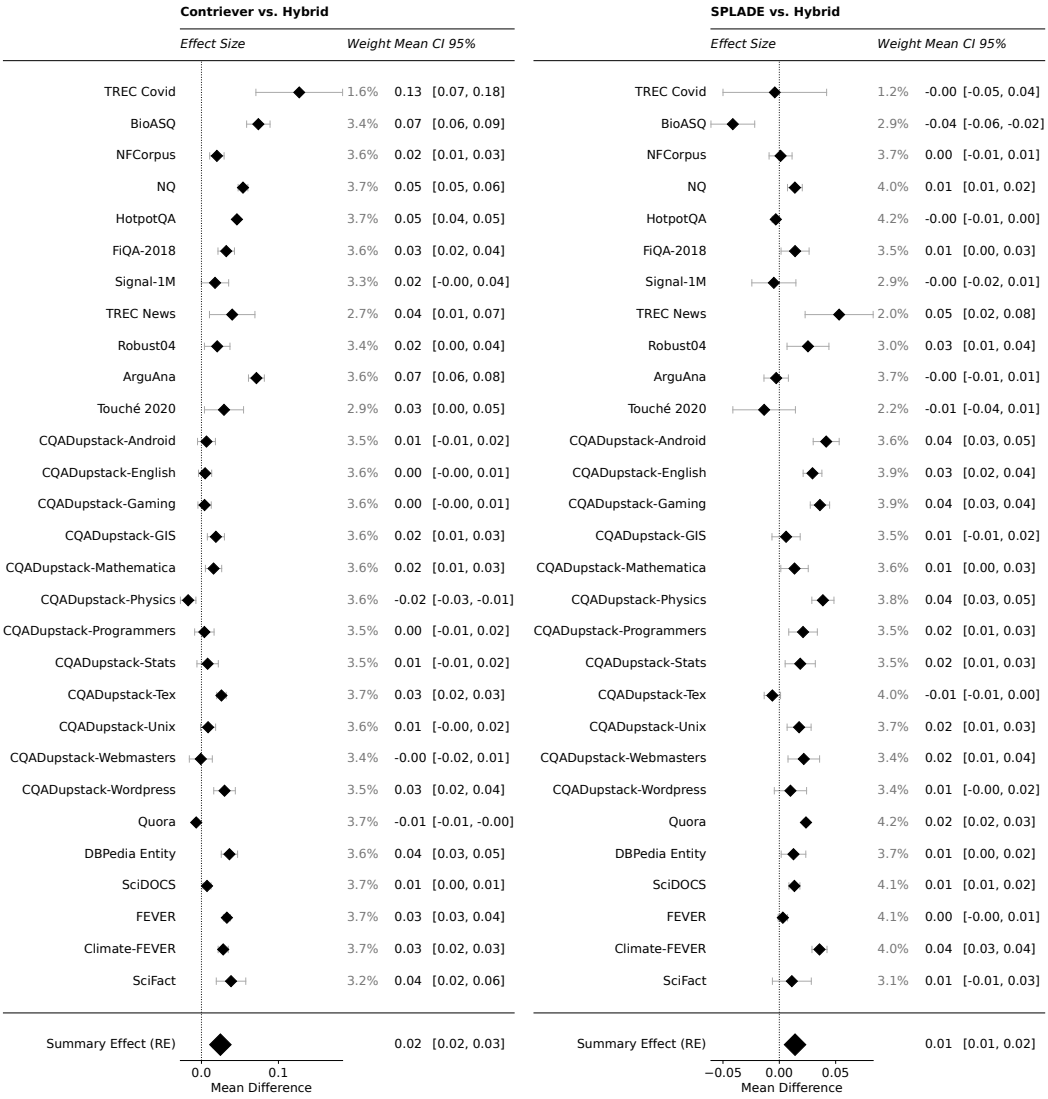

Figure 3: Forest plots visualizing meta-analyses (in terms of nDCG@10) of different model variants: Contriever vs. dense–sparse hybrid (left) and SPLADE vs. dense–sparse hybrid (right).

Meta-analyses for robust comparisons beyond macro-averages with statistical rigor target the second challenge.

We are optimistic about the future of BEIR. It has become an important evaluation instrument for the community to address a number of important research questions. This work mitigates two existing shortcomings, and while there remain more challenges ahead, BEIR has already and will continue to help advance the state of the art.

ACKNOWLEDGEMENTS

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada. We'd like to acknowledge computational resources provided by Compute Canada and Microsoft Azure.

REFERENCES

[1] Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval. *arXiv:2010.00768* (2020).

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v3* (2018).

[3] Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. 1994. Automatic Combination of Multiple Ranked Retrieval Systems. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 173–181.

[4] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. In *Proceedings of the 38th European Conference on Information Retrieval*.

[5] James P. Callan. 1994. Passage-Level Evidence in Document Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 302–310.

[6] Eunseong Choi, Sunkyung Lee, Minijn Choi, Hyeseon Ko, Young-In Song, and Jongwuk Lee. 2022. SpaDE: Improving Sparse Representations Using a Dual Document Encoder for First-Stage Retrieval. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 272–282.

[7] Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stéphan Clémençon. 2022. What Are the Best Systems? New Perspectives on NLP Benchmarking. In *Advances in Neural Information Processing Systems*, Vol. 35. 26915–26932.

[8] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 985–988.

[9] Josh Devins, Julie Tibshirani, and Jimmy Lin. 2022. Aligning the Research and Practice of Building Search Applications: Elasticsearch and Pyserini. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*.

[10] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. *arXiv:2109.10086* (2021).

[11] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2353–2359.

[12] Luyu Gao and Jamie Callan. 2021. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. *arXiv:2108.05540* (2021).

[13] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3030–3042.

[14] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complementing Lexical Retrieval with Semantic Residual Embedding. In *Proceedings of the 43rd European Conference on Information Retrieval*.

[15] Marti A. Hearst and Christian Plaunt. 1993. Subtopic Structuring for Full-Length Document Access. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 56–68.

[16] Sebastian Hofstätter, Omar Khattab, Sophia Althammer, Mete Sertkan, and Allan Hanbury. 2022. Introducing Neural Bag of Whole-Words with ColBERTer: Contextualized Late Interactions using Enhanced Reduction. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 737–747.

[17] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[18] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards Unsupervised Dense Information Retrieval with Contrastive Learning. *arXiv:2112.09118* (2021).

[19] Kyoung-Rok Jang, Junmo Kang, Giwon Hong, Sung-Hyon Myaeng, Joohee Park, Taewon Yoon, and Heecheol Seo. 2021. Ultra-High Dimensional Sparse Representations with Binarization for Efficient Text Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 1016–1029.

[20] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2021), 535–547.

[21] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 6769–6781.

[22] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 39–48.

[23] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *TACL* (2019).

[24] Minghan Li, Sheng-Chieh Lin, Xueguang Ma, and Jimmy Lin. 2023. SLIM: Sparsified Late Interaction for Multi-Vector Retrieval with Inverted Indexes. *arXiv:2302.06587* (2023).

[25] Jimmy Lin. 2021. A Proposed Conceptual Framework for a Representational Approach to Information Retrieval. *SIGIR Forum* 55, 2 (2021), 4:1–29.

[26] Jimmy Lin. 2022. Building a Culture of Reproducibility in Academic Research. *arXiv:2212.13534* (2022).

[27] Jimmy Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *arXiv:2106.14807* (2021).

[28] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2356–2362.

[29] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Morgan & Claypool Publishers.

[30] Xueguang Ma, Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2022. Document Expansions and Learned Sparse Lexical Representations for MS MARCO V1 and V2. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3187–3197.

[31] Xueguang Ma, Kai Sun, Ronak Pradeep, Minghan Li, and Jimmy Lin. 2022. Another Look at DPR: Reproduction of Training and Replication of Retrieval. In *Proceedings of the 44th European Conference on Information Retrieval*.

[32] Maxime Peyrard, Wei Zhao, Steffen Eger, and Robert West. 2021. Better than Average: Paired Evaluation of NLP systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. 2301–2315.

[33] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *arXiv:2101.05667* (2021).

[34] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5835–5847.

[35] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2825–2835.

[36] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R. Hersh. 2020. TREC-COVID: Rationale and Structure of an Information Retrieval Shared Task for COVID-19. *JAMIA* 27, 9 (2020), 1431–1436.

[37] Mete Sertkan, Sophia Althammer, and Sebastian Hofstätter. 2023. Ranger: A Toolkit for Effect-Size Based Multi-Task Evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 581–587.

[38] Ian Soboroff. 2018. Meta-Analysis for Retrieval Experiments Involving Multiple Test Collections. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 713–722.

[39] Nandan Thakur, Luiz Bonifacio, Maik Fröbe, Alexander Bondarenko, Ehsan Kamalloo, Martin Potthast, Matthias Hagen, and Jimmy Lin. 2024. Systematic Evaluation of Neural Retrieval Models on the Touché 2020 Argument Retrieval Subset of BEIR. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[40] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Neural Information Processing Systems Datasets and Benchmarks Track*.

[41] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the Best Counterargument without Prior Topic Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 241–251.

[42] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, Vol. 32.

[43] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *International Conference on Learning Representations*.

[44] Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models. In *Neural Information Processing Systems Datasets and Benchmarks Track*.

[45] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[46] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. LaPraDoR: Unsupervised Pretrained Dense Retriever for Zero-Shot Text Retrieval. *arXiv:2203.06169* (2022).

[47] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. *JDIQ* 10, 4 (2018), Article 16.

[48] Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 497–506.

[49] Jingtao Zhan, Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. Evaluating Interpolation and Extrapolation Performance of Neural Retrieval Models. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 2486–2496.