University of Ottawa

Comparison of Neural Information Retrieval Methods on the BEIR Collection

CSI 4900 (Honours Project)

Author: Mark Perera

Supervisor: Professor Diana Inkpen

Introduction

Neural models for information retrieval have shown significant promise in recent years, outperforming traditional models in many aspects. However, benchmarking these models on diverse datasets is crucial to understand their generalizability and robustness. The BeIR collection provides a wide array of datasets that all use very similar formatting, making it an ideal for evaluating the performance of IR models. This report evaluates benchmarks of various neural IR models using BeIR datasets, comparing their performance to a BM25 baseline so we can see their strengths and weaknesses.

Problem Definition

Evaluating information retrieval models can be challenging due to the variety of datasets and models available, each with its unique characteristics. Traditional models like BM25 have been extensively benchmarked and are known for their reliability in many scenarios. In this report I will be testing other models to see if they can stand as a viable alternative to traditional models.

This report addresses the need for a comprehensive evaluation of neural IR models across a wide range of datasets and models. The main question is how these models perform relative to a well-established baseline like BM25. Through this comparison, this report aims to identify where neural models excel and where they might still need improvement.

Motivation

The motivation for this study stems from the ongoing advancements in information retrieval and the increasing demand for more accurate and context-aware search capabilities. As the complexity and volume of information grow, so does the need for standardized benchmarking that can efficiently handle diverse types of models and datasets. Neural IR models represent a promising development in this field, but without thorough proper evaluation, their potential remains uncertain.

The BeIR collection offers a unique opportunity to test these models in a way that is both rigorous and consistent across datasets. By comparing the performance of neural IR models against a BM25 baseline across multiple datasets, this report aims to provide insights that could inform future use of evaluation in information retrieval technology. The ultimate goal is to contribute to the creation of more robust and effective IR systems that can meet the demands of modern information retrieval tasks.

Methods and Setup

Initial System

The base information retrieval system I made has six files - parser.py, preprocessing.py, indexing.py, utils.py, and main.py.

- **parser.py**: Parses through the documents and queries, to put them into a list of dictionaries I can work with.
- **preprocessing.py**: Takes the parsed documents and queries and performs tokenizing and stemming on them.
- **indexing.py**: Builds an inverted index out of the preprocessed documents.
- **ranking.py**: Performs BM25 using the inverted index and queries, and ranks the results.
- **utils.py**: Has a few miscellaneous functions like a progress bar for better console displays for loading, a function for converting between qrels formats, and functions to save results to a file.
- **main.py**: Calls functions from all the previous classes and manages the overall workflow of the information retrieval system.

The information retrieval system was tested with BM25 using the AP collection available here: https://www.site.uottawa.ca/~diana/csi4107/AP_collection.zip.
By creating this information retrieval system with BM25, it helped to set a baseline to ensure the system is able to produce reasonable results.
Using just BM25 with TREC evaluation, it resulted in a MAP score of 0.2888.
I then ran the information retrieval system on five BeIR datasets. I chose to use TREC formatting so I could run the TREC evaluation on it to see how the results compare against my baseline of 0.2888.

Dataset	TREC MAP Score
trec-covid	0.2129
nfcorpus	0.1371
scifact	0.6012
scidocs	0.0855
webis-touche2020	0.1594

BeIR Setup With BeIR Evaluation

To complete the IR system and allow it to use other models besides BM25, I created beir_ranking.py, benchmarking.py, utils.py, and combine.py.

- **beir_ranking.py**: Uses a selected neural model to perform ranking on the inverted index and queries.

- **benchmarking.py**: Loads BeIR benchmarking information from a selected dataset and compares it against the results produced by my IR system to evaluate metrics.
- **utils.py**: Has a few miscellaneous functions like a progress bar for better console displays for loading, a function for converting between qrels formats, and functions to save results to a file.
- **combine.py**: Combines two selected results files from different models into a new results file that has the possibility of providing improved results.

Dataset Descriptions

TREC-COVID

- **Description**: TREC-COVID is a moderately sized dataset that is designed for evaluating information retrieval systems on COVID-19 related scientific literature.
- **Size**:
    - corpus.jsonl: 216,182 KB
    - queries.jsonl: 17 KB
    - test.tsv: 958 KB

NFCorpus

- **Description**: NFCorpus is one of the smaller datasets in this study, containing nutrition and biomedical information. Its compact size allows for quicker processing and evaluation.
- **Size**:
    - corpus: 6,074 KB
    - queries: 432 KB
    - test.tsv: 274 KB

SciFact

- **Description**: SciFact is slightly larger than NFCorpus, consisting of scientific claims and abstracts.
- **Size**:
    - corpus.jsonl: 7,917 KB
    - queries.jsonl: 205 KB
    - test.tsv: 6 KB

SciDocs

- **Description**: The SciDocs dataset is a moderately sized collection of scientific documents used for various IR tasks. It's the second largest dataset in this report.
- **Size**:
    - corpus.jsonl: 251,304 KB
    - queries.jsonl: 3,093 KB

- test.tsv: 2,485 KB

Webis-Touche2020

- **Description**: Webis-Touche2020 is the largest dataset in this report, focusing on argument retrieval. The extensive corpus size makes it an ideal candidate for testing the scalability and performance of IR models on argumentative texts.
- **Size**:
  - corpus.jsonl: 719,066 KB
  - queries.jsonl: 29 KB
  - test.tsv: 99 KB

Evaluation Metrics Descriptions

NDCG (Normalized Discounted Cumulative Gain)

- **Description**: NDCG is a metric used to evaluate the ranking quality of the results returned by a retrieval model. The metric considers the position of relevant documents in the ranked list, giving higher scores to relevant documents appearing higher in the list. The number after the @ symbol (e.g., @1, @10, @100) indicates the cutoff rank at which the NDCG is calculated, meaning only the top results up to that rank are considered in the metric.

MAP (Mean Average Precision)

- **Description**: MAP is used to evaluate the precision of a model across multiple queries, averaging the precision scores at different cutoff points in the ranked list.

Recall

- **Description**: Recall measures the ability of a model to retrieve all relevant documents within a certain cutoff rank. Higher recall indicates that the model is more effective at finding relevant documents.

Precision (P)

- **Description**: Precision measures the proportion of relevant documents within the retrieved documents at a specified cutoff rank. A higher precision value indicates that the model is returning more relevant documents relative to the total number retrieved.

Results

Trec-covid

Model	NDCG@1	NDCG@5	NDCG@10	NDCG@100	MAP@1	MAP@5	MAP@10	MAP@100	Recall@1	Recall@5	Recall@10	Recall@100	P@1	P@5	P@10	P@100
BM25	0.396	0.321	0.288	0.245	**0.046**	**0.090**	**0.102**	**0.122**	**0.046**	**0.104**	**0.131**	**0.220**	0.411	0.275	0.207	0.060
msmarco-distilbert-base-v3	0.500	0.434	0.392	0.244	0.001	0.004	0.006	0.028	0.001	0.005	0.008	0.053	0.540	0.464	0.406	0.239
BM25 + msmarco-distilbert-base-v3	**0.750**	0.682	**0.666**	**0.479**	0.002	0.008	0.015	0.080	0.002	0.009	0.018	0.105	**0.820**	0.748	**0.736**	**0.497**
universal-sentence-encoder-qa	0.380	0.367	0.363	0.241	0.001	0.003	0.006	0.028	0.000	0.004	0.009	0.052	0.420	0.412	0.408	0.246
BM25 + universal-sentence-encoder-qa	0.720	**0.695**	0.646	0.449	0.002	0.008	0.015	0.071	0.002	0.009	0.017	0.102	0.760	**0.752**	0.686	0.4632
facebook-dpr-ctx_encoder-multiset-base	0.130	0.120	0.112	0.077	0.001	0.000	0.001	0.003	0.000	0.001	0.002	0.017	0.140	0.128	0.122	0.077

Nfcorpus

Model	NDCG@1	NDCG@5	NDCG@10	NDCG@100	MAP@1	MAP@5	MAP@10	MAP@100	Recall@1	Recall@5	Recall@10	Recall@100	P@1	P@5	P@10	P@100
BM25	**0.720**	**0.659**	**0.626**	**0.462**	0.002	0.008	0.014	0.077	0.002	0.009	0.017	0.110	**0.800**	**0.708**	**0.682**	**0.495**
sparta-msmarco-distilbert-base-v1	0.407	0.316	0.281	0.248	0.051	0.091	0.102	0.123	0.051	0.103	0.122	0.230	0.414	0.262	0.199	0.062
BM25+sparta-msmarco-distilbert-base-v1	0.417	0.337	0.301	0.259	**0.052**	**0.095**	**0.108**	**0.130**	**0.052**	0.108	0.131	0.238	0.433	0.286	0.217	0.065
msmarco-distilbert-base-v3	0.306	0.250	0.229	0.205	0.037	0.063	0.073	0.090	0.037	0.081	0.106	0.213	0.315	0.213	0.169	0.053
BM25 + msmarco-distilbert-base-v3	0.408	0.336	0.302	0.262	0.047	0.092	0.105	0.129	0.047	**0.110**	**0.137**	0.245	0.424	0.290	0.221	0.067
msmarco-roberta-base-ance-firstp	0.294	0.226	0.205	0.187	0.034	0.059	0.068	0.082	0.034	0.074	0.101	0.200	0.303	0.190	0.146	0.044
BM25+msmarco-roberta-base-ance-firstp	0.294	0.226	0.205	0.187	0.034	0.059	0.068	0.082	0.034	0.074	0.101	0.200	0.303	0.190	0.146	0.044
universal-sentence-encoder-qa	0.260	0.216	0.194	0.181	0.026	0.050	0.056	0.071	0.026	0.064	0.084	0.210	0.275	0.197	0.147	0.051
BM25 + universal-sentence-encoder-qa	0.400	0.334	0.296	0.263	0.046	0.089	0.103	0.127	0.046	0.107	0.134	**0.260**	0.414	0.291	0.218	0.069
facebook-dpr-ctx_encoder-multiset-base	0.196	0.145	0.126	0.118	0.017	0.031	0.034	0.041	0.017	0.040	0.054	0.141	0.204	0.120	0.088	0.033

Scifact

Model	NDCG@1	NDCG@5	NDCG@10	NDCG@100	MAP@1	MAP@5	MAP@10	MAP@100	Recall@1	Recall@5	Recall@10	Recall@100	P@1	P@5	P@10	P@100
BM25	0.526	0.613	0.635	0.662	0.513	0.585	0.595	0.600	0.513	0.685	0.750	0.876	0.526	0.147	0.082	0.009
sparta-msmarco-distilbert-base-v1	0.456	0.524	0.553	0.586	0.440	0.499	0.512	0.519	0.440	0.583	0.669	0.827	0.456	0.125	0.073	0.009
BM25+sparta-msmarco-distilbert-base-v1	0.476	0.546	0.575	0.609	0.460	0.520	0.534	0.541	0.460	0.606	0.691	0.851	0.476	0.130	0.076	0.009
msmarco-distilbert-base-v3	0.346	0.431	0.454	0.503	0.329	0.399	0.410	0.420	0.329	0.510	0.577	0.797	0.346	0.110	0.064	0.009
BM25 + msmarco-distilbert-base-v3	0.540	0.624	0.645	**0.678**	0.523	0.595	0.604	0.611	0.523	0.697	**0.759**	**0.908**	0.540	0.151	**0.083**	**0.010**
msmarco-roberta-base-ance-firstp	0.346	0.427	0.442	0.481	0.325	0.395	0.402	0.409	0.325	0.506	0.548	0.733	0.346	0.115	0.063	0.008
BM25+msmarco-roberta-base-ance-firstp	0.470	0.537	0.554	0.587	0.441	0.509	0.516	0.524	0.441	0.601	0.648	0.798	0.470	0.134	0.073	0.009
universal-sentence-encoder-qa	0.143	0.2003	0.212	0.253	0.136	0.181	0.186	0.194	0.136	0.249	0.283	0.479	0.143	0.052	0.031	0.005
BM25 + universal-sentence-encoder-qa	**0.543**	**0.629**	**0.646**	0.673	**0.527**	**0.598**	**0.607**	**0.613**	**0.527**	**0.706**	0.754	0.872	**0.543**	**0.152**	0.083	0.009
facebook-dpr-ctx_encoder-multiset-base	0.050	0.101	0.119	0.163	0.046	0.084	0.092	0.100	0.046	0.142	0.194	0.415	0.050	0.032	0.022	0.004

Scidocs

Model	NDCG@1	NDCG@5	NDCG@10	NDCG@100	MAP@1	MAP@5	MAP@10	MAP@100	Recall@1	Recall@5	Recall@10	Recall@100	P@1	P@5	P@10	P@100
BM25	0.156	0.105	0.125	0.181	0.031	0.062	0.071	0.083	0.031	0.092	0.130	0.296	0.156	0.091	0.064	0.014
sparta-msmarco-distilbert-base-v1	0.120	0.083	0.103	0.153	0.024	0.049	0.058	0.067	0.024	0.072	0.108	0.258	0.120	0.071	0.053	0.012
BM25+sparta-msmarco-distilbert-base-v1	0.135	0.093	0.112	0.164	0.027	0.054	0.063	0.073	0.027	0.081	0.116	0.271	0.135	0.081	0.057	0.013
msmarco-distilbert-base-v3	0.144	0.089	0.106	0.150	0.029	0.052	0.059	0.068	0.029	0.076	0.106	0.234	0.144	0.075	0.052	0.011
BM25 +	0.161	**0.115**	**0.140**	**0.198**	0.032	**0.068**	**0.079**	**0.092**	0.032	**0.102**	**0.148**	**0.320**	0.161	0.100	0.073	**0.015**

msmarco-distilbert-base-v3																
msmarco-roberta-base-ance-firstp	0.108	0.077	0.092	0.130	0.022	0.044	0.050	0.058	0.022	0.068	0.097	0.209	0.108	0.067	0.047	0.010
BM25+msmarco-roberta-base-ance-firstp	0.146	0.094	0.113	0.155	0.029	0.056	0.063	0.072	0.029	0.081	0.116	0.240	0.146	0.080	0.057	0.011
universal-sentence-encoder-qa	0.092	0.062	0.072	0.112	0.018	0.035	0.038	0.045	0.018	0.054	0.073	0.193	0.092	0.054	0.036	0.009
BM25 + universal-sentence-encoder-qa	**0.163**	0.114	0.135	0.191	**0.033**	0.068	0.077	0.090	**0.033**	0.101	0.140	0.306	**0.163**	**0.099**	**0.068**	0.015
facebook-dpr-ctx_encoder-multiset-base	0.049	0.028	0.033	0.054	0.009	0.015	0.017	0.020	0.009	0.023	0.031	0.098	0.049	0.023	0.015	0.004

Webis-touche2020

Model	NDCG@1	NDCG@5	NDCG@10	NDCG@100	MAP@1	MAP@5	MAP@10	MAP@100	Recall@1	Recall@5	Recall@10	Recall@100	P@1	P@5	P@10	P@100
BM25	0.346	0.272	0.264	0.353	0.028	0.070	0.098	0.149	0.028	0.091	0.153	0.399	0.408	0.273	0.248	0.071
msmarco-distilbert-base-v3	0.173	0.149	0.142	0.242	0.014	0.043	0.053	0.091	0.014	0.060	0.093	0.328	0.183	0.155	0.129	0.053
BM25 + msmarco-distilbert-base-v3	0.173	0.150	0.143	0.242	0.015	0.043	0.054	0.091	0.015	0.060	0.093	0.329	0.184	0.155	0.129	0.053
universal-sentence-encoder-qa	0.143	0.131	0.137	0.204	0.013	0.036	0.057	0.080	0.013	0.052	0.098	0.277	0.163	0.139	0.129	0.046
BM25 + universal-sentence-encoder-qa	**0.429**	**0.346**	**0.308**	**0.375**	**0.035**	**0.089**	**0.119**	**0.169**	**0.035**	**0.115**	**0.172**	**0.425**	**0.490**	**0.347**	**0.271**	**0.073**
facebook-dpr-ctx_encoder-multiset-base	0.082	0.052	0.044	0.074	0.009	0.012	0.015	0.022	0.009	0.017	0.028	0.102	0.102	0.049	0.037	0.016

Performance of Models

Dataset	Preprocessing	Indexing	BM25 Ranking	msmarco-distilbert-base-v3	BeIR/sparta-msmarco-distilbert-base-v1	facebook-dpr-ctx_encoder-multiset-base	universal-sentence-encoder-qa/3	msmarco-roberta-base-ance-firstp
trec-covid	1545.79 seconds (26 mins)	98.45 seconds	5838.34 seconds (1h 37 mins)	72580.18 seconds (20h)	-	34678.58 seconds (9h 38 mins)	1189.83 seconds (20mins)	-

nfcorpus	22.95 seconds	1.43 seconds	45.44 seconds	1959.47 seconds (33 mins)	2723.54 seconds (45 mins)	592.16 seconds (10 mins)	61.50 seconds	5241.89 seconds (1h 27 mins)
scifact	135.30 seconds	9.51 seconds	661.35 seconds (11 mins)	2141.37 seconds (37 mins)	4168.86 seconds (1h 9 mins)	740.50 seconds (12 mins)	45.58 seconds	6639.11 seconds (1h 51mins)
scidocs	261.14 seconds (4 mins)	16.33 seconds	924.80 seconds (15 mins)	12509.46 seconds (3h 29 mins)	25694.02 seconds (7h 8 mins)	5823.33 seconds (1h 37 mins)	111.47 seconds	17957.56 seconds (5 hours)
webis-touche2020	3893.86 seconds (1h 5 mins)	221.98 seconds (4 mins)	7641.64 seconds (2h 7 mins)	129165.06 seconds (35h 53 mins)	-	74471.44 seconds (20h 41 mins)	1616.55 seconds (27 mins)	-

Model Highlights

msmarco-distilbert-base-v3: Although this model took significantly longer to rank (e.g., 20 hours for trec-covid), its performance was generally inferior to BM25.

sparta-msmarco-distilbert-base-v1: This model showed improvements over BM25 in a few cases, but it took a long time to rank.

facebook-dpr-ctx_encoder-multiset-base: This model took a medium amount of time to rank results, but it performed the worst by far.

universal-sentence-encoder-qa/3: This was the fastest model, with results slightly worse than the average model.

msmarco-roberta-base-ance-firstp: This model took the longest amount of time to rank results. It performed around average, worse than the best model but better than USE-QA.

Issues and Design Decisions

1. **BeIR Model Script Issues**:
   I had to make changes to these two files inside C:\Users\markn\AppData\Local\Programs\Python\Python311\Lib\site-packages\beir\retrieval\models

because they weren't running as is from the BeIR collection.

- **UseQA**: I had to move import tensorflow as tf, and import tensorflow_hub as hub, from inside the if to the top of the script. These imports weren't properly being accessed otherwise.
- **SPARTA**: When using this file through SparseSearch, the console was giving me an error of not knowing what np.int and np.float were. So all instances of dtype=np.float and dtype=np.int had to be changed to dtype=float and dtype=int.

2. **Performance Constraints**:
   - **Reranking with BeIR Models**: The reranking process using BeIR models was time-consuming. I ran a BeIR reranking cross-encoder model with results from another model (UseQA). It ran for about 26 hours, but only made about 14% progress. I decided not to continue with performing reranking with a BeIR reranking model because I didn't want to wait a week to perform reranking for each model.
   - **TREC-COVID and Webis-touche2020**: TREC-COVID took 20 hours and webis-touche2020 took 35 hours to rank with msmarco-distilbert-base-v3. The ance and sparta models took even longer to rank with the other datasets, so I decided not to try ranking these datasets with those models.
3. **Size Management**:
   - **Suboptimal dataset size vs time**: I ran the IR system on the datasets arguana, quora, and fiqa as well, but they took too long to run and produced files that were too big in comparison to the size of the datasets. So I decided not to use these datasets for my report.
   - **Large Results File**: I ran the IR system on the nq dataset, but after running it for 20 hours, with 27% of it ranked, my results file was about 145gb. I decided not to continue with this as well.

Suggestions for Improvement to BeIR

While the BEIR collection provides a robust benchmark for evaluating IR models, improvements can be made in order to maintain a standard for which different models and datasets can be evaluated and compared. There is a lot of similarity in the formatting of the datasets, with each dataset having a corpus.jsonl, a queries.jsonl, and a test.tsv qrels file. Inside each corpus.jsonl, there are consistent labels for _id, title, text, and metadata. Within each queries.jsonl file, there are also consistent labels for _id, text, and metadata. However, within metadata, there is a lack of uniformity. Some entries include url, authors, year, cited_by, references, pubmed_id, etc., while others have no metadata. Inconsistencies in the formatting here can affect the results obtained from different datasets, because this could affect the effectiveness of the parsing and preprocessing done, as well as other aspects of ranking. In addition to this, the number of queries in queries.jsonl is different for each dataset. Even though trec-covid is the third largest dataset I used, its queries.jsonl has only 50 entries in it, whereas some of the others vary from having 1000 queries to 3237 queries. This could also affect benchmarked evaluation information.

Conclusion

The use of BEIR as a heterogeneous benchmark has been effective in evaluating the performance of various neural IR models. The results gathered through the information retrieval system highlight the strengths and weaknesses of different models across diverse datasets, with neural models showing potential in specific scenarios. However, traditional models like BM25 still hold competitive ground in many situations. Future model research should focus on optimizing neural models for efficiency and exploring hybrid approaches that combine the strengths of both neural and traditional methods. While the BeIR collection itself could benefit from increased consistency in the formatting of its datasets.