The Utility of Query Expansion for Semantic Re-ranking Models

An empirical analysis on the performance impact for ad-hoc retrieval

Victor-Filip Ghita

Supervisor(s): Avishek Anand, Jurek Leonhardt

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Victor-Filip Ghita
Final project course: CSE3000 Research Project
Thesis committee: Avishek Anand (Responsible Professor), Jurek Leonhardt (Supervisor), Alan Hanjalic (Examiner)

Abstract

In the past years, data has become increasingly important to more and more domains, leading to more efficient decision-making. As the amount of collected data grows, there is an increased need for tools that help with various Information Retrieval (IR) tasks. One of the most widespread IR tasks is ad-hoc retrieval which, for a given search query, returns a list of relevant documents from a large corpus ordered by their relevance. Initial Ad-hoc retrieval models were based on term matching, which could not overcome vocabulary mismatch. On one hand, initial strategies aiming to overcome semantic limitations were adopting query expansion, augmenting the initial search query with new terms to capture more relevant documents. On the other hand, newer strategies rely on Natural Language Processing (NLP) for ranking documents by semantic similarity. One such example is retrieve-and-re-rank models, which retrieve documents by their lexical similarity and re-rank the retrieved documents based on semantic similarities, by making use of NLP embedding models. This research focuses on analysing the performance of combining RM3, a pseudo-relevance feedback query expansion strategy, with the semantic re-ranking model TCT-ColBERT. This model is compared with the lexical retrieval model BM25 which serves as a benchmark, as well as with its components RM3 and TCT-ColBERT. Results indicate that on certain tasks, the model performs better (up to 7%), while on other tasks it performs worse (up to 3%).

1 Introduction

Information Retrieval (IR) is the field of information science which identifies and retrieves resources relevant to an information need, usually specified as a search query. Ad-hoc retrieval is an IR task which finds a list of relevant documents from a larger corpus, ranking them by their relevance to the search query. One popular example is web searches, which list the most relevant web pages for a user's query.

Initial models proposed for ad-hoc retrieval tasks, like BM25 [1], rely on term matching between search queries and documents. This means that an important metric for ranking documents is how often terms from the query occur within the document. This method is computationally inexpensive, given proper indexing data structures such as Inverted Indices, but they are fairly limited. As these models rely on the exact matching of terms, they ignore semantic similarities between words, such as synonyms, which is illustrated in Figure 1.

One of the solutions proposed to overcome the vocabulary mismatch is query expansion, a technique which augments a search query with new terms, aiming to infer user intent. A popular strategy for query expansion is pseudo-relevance feedback of retrieved documents (PRF QE), which first identifies relevant documents and then adds terms important terms from the relevant retrieved documents to the initial search query and re-runs the retrieval stage with the newly obtained query. One pseudo-relevance feedback query expansion model is RM3 [2]. Although query expansion may perform better than the standard BM25 model, it is still heavily dependent on keyword-based searching and can't take contextual information into account.

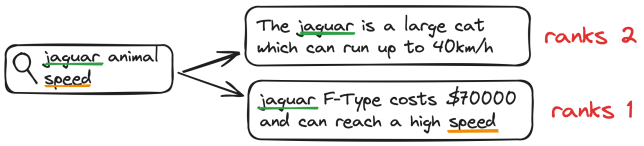


Figure 1: Example of how term matching may lead to undesired ranking.

Another approach to incorporate semantic meaning from search queries proposes using state-of-the-art Neural Language models such as OpenAI's GPT-4 [3] or Google's BERT [4] for document ranking. As recent developments in the field of Natural Language Processing (NLP) lead to improved results of these models, literature analysed the performance of these models for various Information Retrieval tasks through various dense retrieval models. The main idea is to represent both queries and documents as embeddings in a high-dimensional space and rank documents' relevance based on the similarity between their vector representation. These techniques can achieve good results for document ranking, but they increase the computation cost and resource utilisation by orders of magnitude [5] and are not explainable.

To overcome the limitations of dense retrieval models, a combination of both dense and sparse retrieval was proposed through semantic re-ranking. These models combine the two approaches into a two-step process: first, the documents are retrieved using a lexical model. Then, the retrieved documents are re-ranked based on a combination of their semantic similarity and lexical similarity.

This research proposes to analyse the utility of query expansion for semantic re-ranking models. More specifically, the performance difference is measured when adding an RM3 Query Expansion stage to the initial retrieval of semantic re-ranking model TCT-ColBERT [6] - an adaptation of the Bidirectional Encoder Representations from Transformers (BERT) model for Information Retrieval tasks. The research will cover the following sub-questions:

- **RQ1:** Does combining query expansion with semantic re-ranking models lead to better results for ad-hoc retrieval tasks?
- **RQ2:** How does the number of retrieved documents in the first stage impact the performance for query expansion combined with a semantic re-ranking model?
- **RQ3:** What type of queries benefit from a query expansion stage in a retrieve-and-re-rank pipeline?

The motivation to test if the retrieve-and-re-rank benefit from query expansion is that semantic re-ranking models rely on lexical retrieval for the initial stage. However, it may still happen that documents are not retrieved due to lexical models' inability to overcome vocabulary mismatch. For example, if the query is "jaguar animal speed", lexical retrievers may ignore documents containing terms like "cat", as illustrated in Figure 1. The hypothesis is that models like RM3 could help augment the initial query to include important terms which are initially ignored by exact matching.

This paper is structured as follows. The first chapter explains the background knowledge required for understanding this research. The next chapter describes the methodology used, including the experimental setup and datasets. Next, the report will dive into the results, analysing for which tasks it performs better and for which tasks it performs worse. Afterwards, the responsible research considerations are discussed, followed by the limitations and future work. Lastly, the conclusion of the research is presented.

2 Background

This chapter aims to offer readers additional context on how the information retrieval pipelines are structured. Each IR pipeline consists of multiple stages (at least one). Each stage takes an input, applies a transformation to it and returns the resulting output. These stages can be chained together, to allow for consecutive operations. There are three types of transformations, described in the upcoming subsections.

2.1 Retrieval Stage

The retrieval stage takes as an input a set of queries and returns a set of $(document, query)$ pairs. Each $(document, query)$ pair is associated a lexical score, where the higher the score, the more relevant the document is for the query. For this research, the model used for retrieval is Okapi BM25 [1]. BM25 scores $(document, query)$ based on the following formula. Notations are explained in Table 1.

$$score(D,Q) = \sum_{i=1}^{n} \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right) \cdot \cdot \frac{f_{doc}(q_i, D) \cdot (k_1 + 1)}{f_{doc}(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \cdot \frac{(k_3 + 1) \cdot f_q(q_i)}{k_3 \cdot f_q(q_i)}$$

BM25 can be used together with stemming of words. Stemming can be performed to match words that belong into the same lexical family. For example, when indexing the corpus of documents, the term "running" is removed its suffix and is stored as "run". For this research, Porter's stemming algorithm is used, with the aim of ranking documents with terms from the same lexical family as terms in the search query as more relevant.

For an efficient retrieval, the corpus of documents is indexed using an inverted index data structure. For each term of the dictionary, a posting list is stored which contains the frequency of the term in each document, making the score calculation efficient.

Table 1: Summary of notation for BM25 scoring formula.

Symbol	Meaning
D	represents a candidate document
q_i	represents a search term of the query
N	the total number of documents in the collection
$n(q_i)$	is the number of documents containing q_i
$f_q(q_i)$	is the number of times that the keyword q_i occurs in the search query
$\|D\|$	is number of terms in document D
avgdl	is the average number of terms in the collection of documents
k_1, k_3, b	tunable parameters, which for this experiment are set to: $k_1 = 1.2, k_3 = 8, b = 0.75$

2.2 Query Expansion Stage

The PRF QE stage takes as an input a set of $(document, query)$ pairs and returns a set with the rewritten queries. This research uses RM3, a pseudo-relevance feedback model which assumes that good expansion terms will occur frequently in the feedback set (therefore assumed representative), but infrequently in the collection as a whole (therefore assumed sufficiently discriminative). As described by Abdul-Jaleel et al. [2], RM3 uses a modified version of the Lemur language modelling toolkit to perform retrieval with a maximum likelihood query model: $P(w|Q)$, which ranks documents based on Kullback-Leibler divergence with $P(w|Q)$:

$$score(D,Q) = \sum_{w} P(w|Q) \cdot log \frac{P(w|Q)}{P(w|D)} \tag{1}$$

The relevance model requires a prior ranking of the collection, based on the maximum likelihood query model. Let R be the set of ranked documents for a given query. For computational efficiency without noticeable performance decrease, R doesn't include the entire collection of documents, but rather only the highest-scoring k documents are used. Firstly, RM3 computes the relevance models, as described in Equation 2, where R is the set of relevant documents. Then, terms are ordered in decreasing order of probability, out of which the top M documents are chosen, with the weights normalised to sum to 1. Lastly, because the original query is no longer taken into account, the relevance model is interpolated with the original query, based on Equation 3.

$$P(w|R) = \sum_{D \in R} P(w|D) \frac{P(Q|D)}{\sum_{D' \in R} P(Q|D')} \tag{2}$$

$$P'(w|R) = \lambda \cdot P(w|R) + (1 - \lambda) \cdot P(w|Q). \tag{3}$$

2.3 Semantic Re-ranking Stage

The semantic re-ranking stage takes as an input a set of $(document, query)$ pairs and returns a new set of

$(document, query)$ pairs by re-ranking each pair by incorporating a notion of semantic similarity. This stage enhances traditional information retrieval systems by incorporating contextual understanding, leading to more relevant search results. Such methods leverages advanced natural language processing techniques to better match the user's intent with the available documents. This research focuses specifically on interpolation-based re-ranking using BERT-based dual-encoders models.

During this stage, each query q is tokenised and its tokens are encoded into a dense-represented space. For all relevant documents $\{d_1, d_2, ..., d_m\}$ retrieved for q, the vectorial representations are retrieved from the indexed collection, based on the documents' unique identifiers. Next, the semantic score is calculated: $S(Q, D) = \sum_{i=1}^{n} max_{j=1}^{m} sim(E_{q_i}, E_{d_j})$, where $sim(E_{q_i}, E_{d_j})$ represents the similarity between i^{th} token $q_i \in q$ and j^{th} token $d_j \in D$, represented as their cosine similarity.
Interpolation-based re-rankers account for both semantic and lexical similarity, giving more importance to either of the two components through a tunable α parameter. This implies each $(document, query)$ pair is re-ranked based on a linear combination of their semantic and lexical scores:

$$score_{\text{interpolated}} = \alpha \cdot s_{\text{lexical}} + (1 - \alpha) \cdot s_{\text{semantic}} \quad (4)$$

The semantic index is pre-computed by embedding all documents in the corpus beforehand, using fast-forward indices [7]. When indexing, each document is split into tokens. Each token is then encoded into a densely-represented space. In this report, TCT-ColBERT [8] was used for both query and document encoding. This model was pre-trained on MS Marco Passage dataset [9].

3 Experimental Setup

This chapter describes how the experiments are set up, covering the models and datasets used. Next, this section explains the evaluation metrics used for analysing the performance of the models and the parameters choice.

3.1 Models

Throughout the paper, 4 models have been used to compare the performance for ad-hoc retrieval tasks:

1. **BM25** [1] - a simple one-stage retrieval pipeline, which serves as a benchmark for this research.
2. **BM25 → RM3** [2] - a pseudo-relevance feedback query expansion model, with three stages. The first stage retrieves the most relevant k documents using BM25. Next, a query expansion stage is employed, augmenting the initial search query with representative and discriminative terms from the M most relevant documents found in the previous stage. Lastly, another retrieval stage takes place, which uses the same BM25 model, but with the expanded search queries. The scoring function takes into account the weight assigned to each term from the previous stage. For convenience, throughout this paper this model will be referred to as "RM3".
3. **BM25 → TCT-ColBERT** [6] [8] - a model which consists of two stages. First, the retrieval stage retrieves uses BM25 to get the most relevant k documents for a search query and to compute the lexical scores. Next, for each $(document, query)$ pair, the re-ranking stage computes a the interpolated score, based on the similarity between the embedded search query and the embedding of the document and on the lexical score, as described in Section 2.3 and re-ranks all $(document, query)$ pairs. The model used for embedding documents and queries is TCT-ColBERT, which is available on Hugging Face[1]. Fast-forward indices [7] are pre-computed on the entire corpus, making the retrieval process more efficient. For convenience, throughout this paper this model will be referred to as "TCT-ColBERT".
4. **BM25 → RM3 → TCT-ColBERT** - a combination of the previous two models, with the following stages: first, the retrieval stage, which uses a BM25 to retrieve the most relevant k documents for each search query. Next, the RM3 query expansion stage takes place, followed by the second BM25 retrieval stage, which keeps only the most relevant n documents (cutoff rank). Next, because of the issues with embeddings of stemmed words, a transform stage rewrites the expanded query back to the original search query. The last stage is the semantic re-ranking stage, as described for TCT-ColBERT. For convenience, throughout this paper this model will be referred to as "RM3+TCT-ColBERT".

3.2 Evaluation Metrics

For evaluating how IR pipelines perform, there are set of standard scores which are widely used within the TREC (Text REtrieval Conference) community. These scores are calculated based on:

- The qrels (query relevance), which is the score assigned to each $(query, document)$ pair by assessors, prior to running the experiment, which are the expected relevant documents.
- The ranking of documents for a given search query, which is calculated by the tested IR pipeline.

This research focuses on three evaluation metrics, which are the official measurements of the TREC DL conference:

1. Mean Reciprocal Rank (RR): a precision score which is calculated based on the position of the first relevant document for each query[2]. The formula is:

$$RR = \frac{1}{|Q|} \cdot \sum_{q_i \in Q} \frac{1}{rank_{q_i}}$$

   where $rank_{q_i}$ represents the position of the first relevant document in the top ranked documents for query q_i and Q represents the set of queries.
2. The normalized Discounted Cumulative Gain (nDCG): uses a graded labels system and normalises which ranks

[1] https://huggingface.co/castorini/tct_colbert-msmarco
[2] https://ir-measur.es/en/latest/measures.html#rr

the highest graded documents on the top. The discontinued cumulative gain is normalised with regards to an ideal ranking, taking into account both precision of the ranked documents and their position in the ranking [3]. This is calculated based on:

$$NDCG = \frac{1}{|Q|} \cdot \sum_{q_i \in Q} \frac{DCG_{q_i}}{IDCG_{q_i}}$$

$$DCG_{q_i} = \sum_{j=1}^{n} \frac{rel_j}{\log_2(j+1)}$$

where rel_j represents the relevance of the j^{ht} ranked document for q_i and $IDCG_{q_i}$ represents the ideal ranking for q_i.

3. Mean Average Precision (MAP): measures the average precision of a set of queries, where precision is defined as the number of relevant documents in the top k retrieved[4]. Let Q be the set of queries and in:

$$MAP = \frac{1}{|Q|} \sum_{q_i \in Q} AP(q_i)$$

$$AP(q_i) = \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} \frac{P(j)}{j} \cdot rel(j)$$

where $|D_i|$ is the set of ranked documents for query q_i, $P(j)$ is the number of relevant documents for q_i in top j ranked documents and $rel(j)$ is a binary function with value of 1 if the document at rank j is relevant or 0 otherwise.

Each of these measurements can accept a cutoff parameter k. This means that for calculating the metric score, only the highest ranked k documents are used. In the results section, this is denoted as "metric @ k" (eg. RR @ 10)

3.3 Datasets

In order to obtain accurate results, a subset of the following large scale, open evaluation IR benchmark were used:

- **BEIR** (BEnchmarking IR) [10] - a collection of robust and heterogeneous evaluation. Out of 18 available datasets, the following datasets were used: *arguana* (ArguAna [11]: retrieval of counterargument without prior topic knowledge), *cq* (CQADupStack [12]: StackExchange duplicate question retrieval) containing 12 subsets with different topics, *fiqa* (FIQA-2018 [13]: financial opinion question answering), *nfcorpus* (NF Corpus [14]: Nutrition Facts question answering), *scifact* (SciFact [15]: fact verification) and *webis-touche-v2* (Touchè-2020 V2 [16]: argument retrieval).
- **TREC** (Text REtrieval Conference) Deep Learning track - evaluation of web searches based on the MS Marco passage retrieval [9]. Three subsets were used throughout this paper: *trec-dl-2019* [17], *trec-dl-2020* [18] and *trec-dl-hard* [19], a challenging subset of trec-dl-2019 and trec-dl-2020. The queries were retrieved from the Bing search engine's history and the query relevancies were judged by NIST (National Institute of Standards and Technology) assessors.

3.4 Model Parameters

This subsection elaborates on the parameters values used for the models described in section 3.1. For RM3, the value of λ, which controls the weight associated to the expanded terms, was set to 0.6. The number of documents considered in the set R was empirically set to $k = 5$ and M, the number of documents consider for feedback, was set to 3. If k was set to lower values than 5, it would usually lead to higher accuracy, but lower exploration rates, while setting it over the value of 5 would result into the opposite. As such, these values would strict the number of terms added but still leave enough room for exploration of new terms. To make computation faster, the retrieval stage of TCT-ColBERT and the second retrieval of RM3+TCT-ColBERT were limited to 1000 documents. This was done to make computations more efficient, as it didn't impact performance noticeably, but still allowed enough documents to be re-ranked. The interpolation weight α was set to 0.05 on most datasets, which yielded the best results when tuning on the MS Marco train dataset. On datasets with available train sets, the value of α was set to the value for which TCT-ColBERT yielded the best RR score.

Tuning the parameters of RM3 doesn't always lead to improved scores. For finding the parameters which lead to the best performance, an RM3 model was used, for which an exhaustive search took place to measure which parameters would yield the best RR score on the training subsets of *fiqa*, *scifact* and *nfcorpus*. During the search, the combination of two parameters were tested: $M \in \{3, 5, 7, 10\}$, representing the number of documents for feedback, and $n \in \{3, 10, 15\}$, representing the number of terms added (total of 12 combinations). Then, on the evaluation sets of the three datasets mentioned before, two models were compared: RM3+TCT-ColBERT with tuned parameters (the combination which yielded the best results on the train set) and RM3+TCT-ColBERT with the parameters described at the beginning of the section. To test the hypothesis that these two models perform similarly, a significance t-test was run, whose result can be seen in Table 2. Since the p-value was always greater than 0.05, the hypothesis is rejected, which means the models perform similarly.

4 Results

This chapter describes the outcome of these experiments, discussing the results for each research question mentioned in the introduction. The detailed results can be found in the the Appendix, Section A.3.

4.1 Performance evaluation when combining QE with semantic re-ranking models

This subsection explores the performance impact when adding a RM3 query expansion stage to a TCT-ColBERT retrieve-and-re-rank pipeline. The results will be compared

[3] https://ir-measur.es/en/latest/measures.html#ndcg

[4] https://ir-measur.es/en/latest/measures.html#ap

Table 2: Statistical t-test, which examines if tuned RM3 parameters yield better performance for RM3+TCT-ColBERT. Each column contains the p-value for the metrics described in the column header for the t-test. The null hypothesis is that these models perform similarly.

Dataset	RR @ 10 p-value	nDCG @ 10 p-value	MAP @ 100 p-value
scifact	0.392	0.516	0.280
nfcorpus	0.618	0.771	0.807
fiqa	0.895	0.887	0.539

against BM25, as well as against the individual sub-models RM3 and TCT-ColBERT.

4.1.1 Tasks and Datasets

For running the experiments, a wide variety of tasks and datasets have been chosen, described in Section 3.3. These task are different by nature and topic, so the results should be representative as a whole for general ad-hoc retrieval tasks. More details about the used datasets are listed in the Appendix, in Table 4.

4.1.2 Experimental Findings

The results indicate that the performance of the models depends on the nature of the IR task and dataset. As illustrated in Figure 2, when using the official BEIR metric "nDCG" with a cutoff parameter $k = 10$, RM3+TCT-ColBERT performs considerably better than BM25 and RM3 on TREC DL datasets (0.691 compared to 0.493 and 0.504 on *trec-dl-2020*, 0.718 compared to 0.479 and 0.515 on *trec-dl-2019* and 0.404 compared to 0.274 and 0.270 on *trec-dl-hard*). On the BEIR datasets, the performance varies: it is either slightly better or worse, but overall quite close. The datasets where RM3+TCT-ColBERT performs slightly worse are datasets with specialised domains, like statistics (*cq/stats*: 0.265 and 0.264 compared to 0.249) or LaTeX (*cq/tex*: 0.234 and 0.229 compared to 0.221). When taking all datasets into account, the average nDCG score increases from 0.334 (BM25) and 0.330 (RM3) to 0.367 (RM3+TCT-ColBERT). This improvement can also be noticed when using "RR @ 10": from 0.371 (BM25) and 0.354 (RM3) to 0.406 (RM3+TCT-ColBERT), as well as "MAP @ 100": from 0.261 (BM25) and 0.257 (RM3) to 0.289 (RM3+TCT-ColBERT).

When comparing the RM3+TCT-ColBERT to the standalone TCT-ColBERT model, it performs better on most datasets. The best result is obtained on the *trec-dl-hard* dataset, where adding the extra QE resulted in a 7.7% improvement in RR score, a 4% improvement in MAP score and 2.5% increase in nDCG. The biggest negative performance hit was on the *arguana*, where the decrease in RR is about 3.5%, 3.4% in MAP and 2.6% in nDCG. On average, there is an overall increase of approximately 1% in all three scores ("nDCG @ 10" is increased from 0.366 to 0.368, "RR @ 10" from 0.402 to 0.406 and "MAP @ 100" from 0.287 to 0.289). The results when using "RR @ 10" and "MAP @ 100" metrics can be seen in Figures 4 and 5, in the Appendix.

4.2 Influence of the number of retrieved documents on RM3+TCT-ColBERT performance

This section aims to reason about how choosing different number of retrieved documents by BM25 affects the relative performance of RM3+TCT-ColBERT with regards to TCT-ColBERT. Let ΔRR be the difference in RR score of the two models, calculated as: $RR_{RM3+TCT-ColBERT} - RR_{TCT-ColBERT}$. Figure 3 shows how choosing different cutoff values k increases or decreases ΔRR, depending on the nature of the task and dataset. Figure 3a illustrates that retrieving a smaller number of candidate documents yields better relative performance for RM3+TCT-ColBERT, but as k becomes larger $\Delta RR \to 0$. An important thing to note is that the dataset used for Figure 3a, *webis-touche-v2*, has very short queries, which seem to benefit from the extra terms added by RM3 for smaller k values. On the other hand, for *trec-dl-hard* and *arguana*, the opposite happens: ΔRR directly increases as the number of candidates k increases, as seen in Figures 3d and 3c. Such conclusion cannot be drawn for all datasets. For instance, Figure 3b shows that such correlations don't occur on the *fiqa* datasets. As a result, best the number of candidate documents for query expansion depends very much on the nature of the dataset; on some datasets a lower number k achieves better comparative results, on other datasets a higher k leads to a smaller ΔRR, while on the rest of the datasets there is no direct correlation between relative performance and the number of retrieved documents.

4.3 Examples of queries that benefit from QE in a retrieve-and-re-rank pipeline

For an accurate understanding of which queries benefit from query expansion, it is beneficial to look into some examples where query expansion lead to an increased RR score, when comparing RM3+TCT-ColBERT with TCT-ColBERT. Queries from Table 3 lead to the highest individual increase in RR which implies that QE lead to the model ranking a more relevant higher. This is justified by the fact that RM3 yields a higher lexical score, which influences the result of the re-ranking.

The first example is: "should social security be privatized", extracted from *webis-touche-v2*. Adding terms like "invest-", "poor-" and "incom-" lead to the retrieval of a document which accurately captures the intent of the initial search query, even if the added terms have a lower weight. For the given search query, the most relevant document has ID: "2d6f4e75-2019-04-15T20:22:43Z-00007-000". Both RM3 and BM25 rank the document on the third position with lexical scores 28.741 and 26.665, respectively. After re-ranking, RM3+TCT-ColBERT ranks this document of the first position (leading to a RR score of 1) and TCT-ColBERT on the second position (leading to a RR score of 0.5). This difference is due to the higher $\Delta score_{lexical}$ which has a 0.05 weight in the final score: $\Delta score_{interpolated} = (28.741 - 26.665) * 0.05 = 0.104$.

The second example is "anthropological definition of

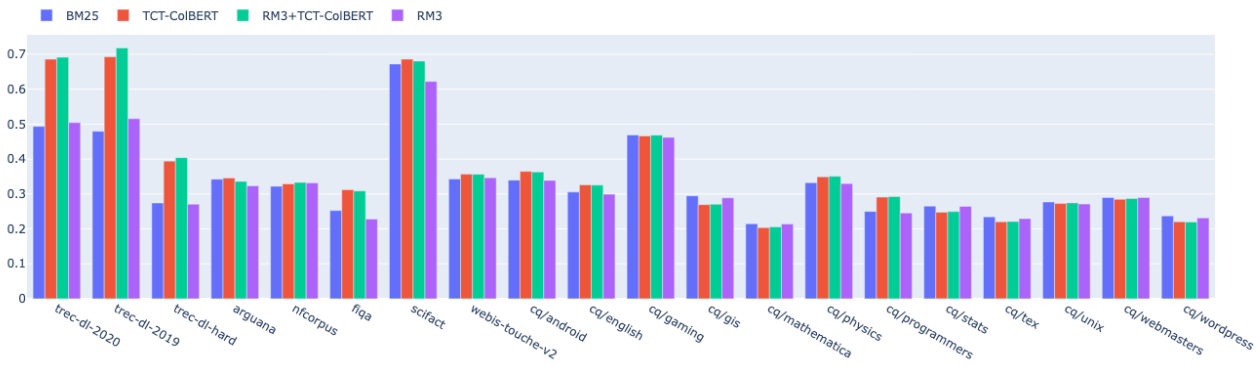
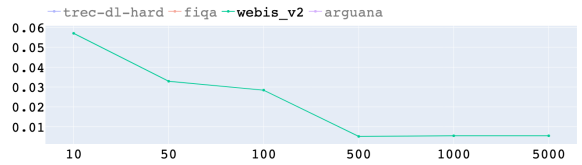
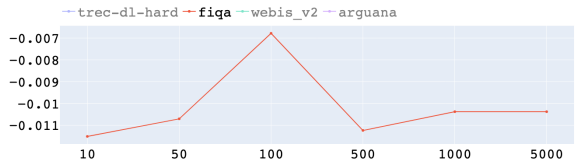


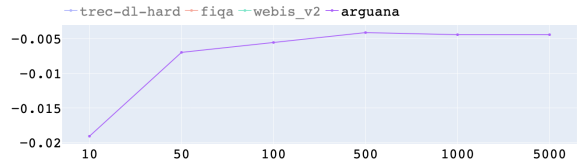
Figure 2: Results of running the experiments from RQ1 using the "nDCG @ 10" metric. The datasets used are described on the X-axis, while the Y-axis contains the nDCG values for each model.



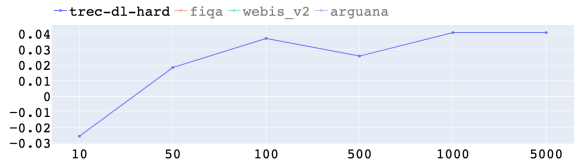
(a) webis-touche-v2 dataset



(b) fiqa dataset



(c) arguana dataset



(d) trec-dl-hard dataset

Figure 3: The difference in RR @ 10 scores, based on different numbers of candidate documents considered for RM3 query expansion. The X-axis shows the number of candidate documents considered, while the Y-axis displays ΔRR.

environment" from *trec-dl-hard*. Because of the newly added terms like "human" or "ecolog-", the document which is most relevant (Document ID: 8412683) has a higher score for RM3+TCT-ColBERT (67.327) than for TCT-ColBERT (67.293). This difference in score is given by the difference in lexical score between RM3 and BM25: $\Delta score_{lexical} = 24.171 - 23.502 = 0.670$. The lexical score is then multiplied by $\alpha = 0.05$ when calculating the overall score, which leads to the difference in ranking of the two models and to a $\Delta RR = 0.8$.

The last example is a longer search query from *fiqa*: "for insurance why should you refuse 4 000 year for only 10 years and prefer 500 year indefinitely". After the query expansion stage, the re-written query contains 15 terms with low individual weights. The relevant document as specified in the QRels is the document ID 462892. Because of the newly added terms by QE, the lexical score before re-ranking is 31.191 for RM3+TCT-ColBERT and 19.616 for TCT-ColBERT, leading to a total of $\Delta score_{interpolated} = (31.191 - 19.616) \cdot \alpha(0.05) = 0.578$. As such, document with ID 462892 is ranked first by RM3+TCT-ColBERT (with a score of: 68.173) and 8^{th} by TCT-ColBERT (with a score of: 67.594), resulting in $\Delta RR = 0.875$.

To sumarise, it is difficult to identify a topology of datasets which would benefit from query expansion in semantic re-ranking models for all queries. The provided examples only illustrate a higher ΔRR, but on some queries, ΔRR can be negative. As such, the performance is highly dependent on the nature of the query and of the dataset, which implies there is no one solution which performs better on all tasks.

Table 3: Examples of query expansions with highest gain in RR per query. Some documents, which were too long, were only cited partially. The numbers in the paranthesis indicate the weight of each term and only new terms were highlighted. The terms have been stemmed.

Augmented Query	Retrieved Document
webis-touche-v2: "should social security be privatized" (Query ID: 5)	
social(0.28) invest(0.016) privat(0.260) elderli(0.031) benefit(0.026) system(0.016) secur(0.279) poor(0.045) retire(0.020) incom(0.022)	Privatizing social security would enable investment of savings. Commentator Alex Schibuola argues that: "If Social Security were privatized, people would deposit their income with a bank. People actually save resources that businesses can invest. We, as true savers, get more resources in the future."[1] As a result private accounts would also increase investments, jobs and wages. Michael Tanner of the think tank the Cato Institute argues: "Social Security drains capital from the poorest areas of the country, leaving less money available for new investment and job creation. Privatization would increase national savings and provide a new pool of capital for investment that would be particularly beneficial to the poor." [...] (Document ID: 2d6f4e75-2019-04-15T20:22:43Z-00007-000)
trec-dl-hard: "anthropological definition of environment" (Query ID: 19335)	
human(0.043) attempt(0.022) ecolog(0.055) environ(0.233) definit(0.200) journal(0.022) understand(0.022) anthropolog(0.316) societi(0.044) rel(0.022) influenc(0.022)	" Ecological anthropology is defined as the study of cultural adaptations to environments. The sub-field is also defined as, the study of relationships between a population of humans and their biophysical environment .he abstract noun anthropology is first attested in reference to history. Its present use first appeared in Renaissance Germany in the works of Magnus Hundt and Otto Casmann. Their New Latin anthropologia derived from the combining forms of the Greek words anthrōpos (ánthrōpos,) human and (logos, lógos). study" (Document no: 8412683)
fiqa: "for insurance why should you refuse 4 000 year for only 10 years and prefer 500 year indefinitely" (Query ID: 5155)	
indefinite(0.100) 10(0.096) salari(0.048) year(0.050) current(0.022) hous(0.036) 000(0.090) onli(0.050) average(0.060) 500(0.091) scenario(0.027) prefer(0.050) refuse(0.05) insur(0.075) 4(0.050)	"The breakeven amount isn't at 8 years. You calculated how many years of paying $500 it would take to break even with one year of paying $4000. 8 x 10 years = 80 years. So by paying $500/year it will take you 80 years to have spent the same amount ($40000 total) as you did in 10 years. At this point it may seem obvious what the better choice is. Consider where you'll be after 10 years: In scenario 1 you've spent $5000 ($500*10) and have to continue spending $500/year indefinitely. In scenario 2 you've spent $40000 ($4000*10) and don't have to pay any more, but you currently have $35000 ($40000 - $5000) less than you did in scenario 1. [...]" (Document no: 462892)

5 Responsible Research

This section describes how the principles of responsible research were implemented throughout the project. First, reproducibility will be discussed, illustrating the steps taken to make it accessible for anyone to reproduce the results. Next, ethical considerations will be presented.

5.1 Open Science and Reproducibility

This research was thought from the beginning to adhere to the principles of the "Netherlands Code of Conduct for Research Integrity 2018": Honesty, Scrupulousness, Transparency, Independence, Responsibility. Firstly, for a transparent and responsible research, all the source code used in this report is made publicly available on Github[5]. The code is built on top of PyTerrier[6], an open-source framework which allows for a declarative implementation of retrieval pipelines. The models used are made available through a PyTerrier plugin[7] in the case of RM3, and through Huggingface[8] in the case of TCT-ColBERT. The datasets[9] and evaluation methods[10] are standard benchmarks for IR tasks, which makes them both popular and publicly available. The datasets have also been

[5]https://github.com/tomighita/ir-query-expansion
[6]https://github.com/terrier-org/pyterrier
[7]https://github.com/terrierteam/terrier-prf/
[8]https://huggingface.co/castorini/tct_colbert-msmarco
[9]https://ir-datasets.com
[10]https://ir-measur.es/en/latest/measures.html

curated to ensure that no sensitive information can be leaked. Secondly, for an honest and scrupulous research, the experiments results are genuine. They present both positive and negative results, without cherry-picking. The variety in the datasets used should be representative enough to support the conclusions drawn. Lastly, the independence was achieved through the way the course was organised. Authors were given freedom to decide the direction of the research, but offering enough support through ethical oversight of the supervisors and through organised peer-review amongst students.

5.2 Ethical Considerations

As recent developments in the field of NLP push for more powerful and resource-intensive models, this report wishes to anaylse a different path for IR tasks. The intended results is to discover models that are both capable and efficient. By combining both efficient lexical retrieval with performant semantical re-ranking models, good performance can be achieved without compromising the efficiency of the model. Focusing the research on more sustainable approaches, the impact of such studies can have positive environmental consequences.

6 Limitations and Future Work

This chapter will discuss the limitations imposed by the time and resource constraints and will suggest recommendations to further elaborate on the topic.

6.1 Limitations

The results shown in this paper are constrained by two factors: time, given the project's nine week duration, and hardware limitations, as most of the results were run on a single machine (Apple M1 Macbook Pro with an M1 processor and 16GB of RAM). With the current setup, some of the experiments, like indexing datasets, were too complex in terms of resource consumption and had to be executed on DHCP [20], to run on dedicated GPUs.

Because of the aforementioned constraints, these experiments only include one query expansion strategy, namely RM3, and one semantic re-ranking model, TCT-ColBERT. There are newer and better performing models [21] [22], but which have an increased implementation complexity or significantly larger sizes. Working with such models would increase the experiments' running time drastically. The constraints also impacted the amount of datasets used, both in number (only 20 datasets were used) and in size (more comprehensive datasets such as *msmarco-passage-v2* exist, but these datasets are order of magnitudes larger - 138M passages vs 8M in *msmarco-passage-v1*).

6.2 Future Recommendations

Following the limitations described in the previous subsection, the recommendations to expand on this study follow two dimensions: models and datasets. More precisely, the recommendation would be to experiment by combining newer and better performing embedding models like *gte_large* [23] or *arctic-embed* [24] as the re-rankers, as well as other query expansion strategies such as *BertQE* [22] which can provide contextualised query expansion. As new embedding models also come in multiple sizes (which vary in the dimension of vectors), another recommendation would be to compare the impact of QE on different-sized models. The final recommendation would be to run these experiments against larger and more comprehensive datasets, like *msmarco-passage-v2*.

7 Conclusion

The goal of this research was to to analyse the utility of query expansion for semantic re-ranking models. To test how adding a QE stage impacts a retreve-and-rerank pipeline, experiments have been run against different ad-hoc retrieval tasks, with datasets ranging from general topics to specialised content like Financial or Health data.

Results indicate that adding a query expansion stage can lead to both better and worse performance, depending on the nature of the dataset. When adding new terms to the initial search query through PRF, the retrieved documents can be aligned with the user intent or far from it. QE adds an exploratory approach to retrieving documents with the intent to overcome vocabulary mismatch, where the results are dependent on how discriminative and representative of the underlying language models are. It is difficult to come up with a model that performs well on all tests, so fine-tuning for each tasks is highly advised for achieving the best performance.

References

[1] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at trec-3," *NIST Special Publication SP*, vol. 109, p. 109, 1995.

[2] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li *et al.*, "Umass at trec 2004: Novelty and hard," *Computer Science Department Faculty Publication Series*, vol. 189, 2004.

[3] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, and S. A. et al., "Gpt-4 technical report," 2024.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[5] J. Leonhardt, K. Rudra, M. Khosla, A. Anand, and A. Anand, "Efficient neural ranking using forward indexes," in *Proceedings of the ACM Web Conference 2022*, ser. WWW '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 266–276. [Online]. Available: https://doi.org/10.1145/3485447.3511955

[6] O. Khattab and M. Zaharia, "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*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 39–48. [Online]. Available: https://doi.org/10.1145/3397271.3401075

[7] J. Leonhardt, K. Rudra, M. Khosla, A. Anand, and A. Anand, "Efficient neural ranking using forward indexes," in *Proceedings of the ACM Web Conference 2022*, ser. WWW '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 266–276. [Online]. Available: https://doi.org/10.1145/3485447.3511955

[8] S.-C. Lin, J.-H. Yang, and J. Lin, "Distilling dense representations for ranking using tightly-coupled teachers," 2020.

[9] N. C. L. D. J. G. X. L. R. M. A. M. B. M. T. N. M. R. X. S. A. S. S. T. T. W. Payal Bajaj, Daniel Campos, "Ms marco: A human generated machine reading comprehension dataset," 2018.

[10] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models," *arXiv preprint arXiv:2104.08663*, 4 2021. [Online]. Available: https://arxiv.org/abs/2104.08663

[11] H. Wachsmuth, S. Syed, and B. Stein, "Retrieval of the best counterargument without prior topic knowledge," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018, pp. 241–251. [Online]. Available: http://aclweb.org/anthology/P18-1023

[12] D. Hoogeveen, K. M. Verspoor, and T. Baldwin, "CQADupStack: A benchmark data set for community question-answering research," *Proceedings of the 20th Australasian Document Computing Symposium*, 2015.

[13] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur, "Www'18 open challenge: Financial opinion mining and question answering," *Companion Proceedings of the The Web Conference 2018*, 2018.

[14] V. Boteva, D. Gholipour, A. Sokolov, and S. Riezler, "A full-text learning to rank dataset for medical information retrieval," in *Proceedings of the European Conference on Information Retrieval (ECIR)*. Springer, 2016.

[15] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or fiction: Verifying scientific claims," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 7534–7550. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.609

[16] A. Bondarenko, M. Fröbe, M. Beloucif, L. Gienapp, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, and M. Hagen, "Overview of touché 2020: Argument retrieval," in *CLEF*, 2020.

[17] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. Voorhees, "Overview of the trec 2019 deep learning track," in *TREC 2019*, 2019.

[18] N. Craswell, B. Mitra, E. Yilmaz, and D. Campos, "Overview of the trec 2020 deep learning track," in *TREC*, 2020.

[19] I. Mackie, J. Dalton, and A. Yates, "How deep is your learning: the dl-hard annotated deep learning dataset," *ArXiv*, vol. abs/2105.07975, 2021.

[20] Delft High Performance Computing Centre (DHPC), "DelftBlue Supercomputer (Phase 2)," https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

[21] X. Wang, C. Macdonald, N. Tonellotto, and I. Ounis, "Pseudo-relevance feedback for multiple representation dense retrieval," in *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ser. ICTIR '21. ACM, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1145/3471158.3472250

[22] Z. Zheng, K. Hui, B. He, X. Han, L. Sun, and A. Yates, "Bert-qe: Contextualized query expansion for document re-ranking," 2020.

[23] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards general text embeddings with multi-stage contrastive learning," *arXiv preprint arXiv:2308.03281*, 2023.

[24] L. Merrick, D. Xu, G. Nuti, and D. Campos, "Arctic-embed: Scalable, efficient, and accurate text embedding models," 2024.

A Appendix

A.1 Datasets

Table 4: Table presenting the datasets used to run the experiment for **RQ1**. The entire list of datasets can be found at: https://ir-datasets.com/beir.html.

Dataset	# Documents	# Queries	# QRels	Task
trec-dl-2020	3213835	43	16258	Queries from TREC Deep Learning 2019, sampled from MS Marco, judged by NIST assesors
trec-dl-2019	3213835	45	9098	Queries from TREC Deep Learning 2020, sampled from MS Marco, judged by NIST assesors
trec-dl-hard	3213835	50	8544	A challenging subset of the MS Marco document dataset
arguana	8674	1406	1406	Argument Retrieval
nfcorpus	3633	323	12334	Medical Information Retrieval (Nutrition Facts)
fiqa	57638	648	1706	Financial Opinion Question Answering
scifact	5183	300	339	Scientifical Fact Verification
webis-touche-v2	3633	323	12334	Argument Retrieval
cq/android	22998	699	1696	Subset of Stack Exchange android sub-forum used for duplicate question retrieval
cq/english	40221	1570	3765	Subset of Stack Exchange english sub-forum used for duplicate question retrieval
cq/gaming	45301	1595	2263	Subset of Stack Exchange gaming sub-forum used for duplicate question retrieval
cq/gis	37637	885	1114	Subset of Stack Exchange Geographic Information Systems sub-forum used for duplicate question retrieval
cq/mathematica	16705	804	1358	Subset of Stack Exchange Mathematics sub-forum used for duplicate question retrieval
cq/physics	38316	1039	1933	Subset of Stack Exchange Physics sub-forum used for duplicate question retrieval
cq/programmers	32176	876	1675	Subset of Stack Exchange Programming sub-forum used for duplicate question retrieval
cq/stats	42269	652	913	Subset of Stack Exchange Statistics sub-forum used for duplicate question retrieval
cq/tex	68184	2906	5154	Subset of Stack Exchange LaTeX sub-forum used for duplicate question retrieval
cq/unix	47382	1072	1693	Subset of Stack Exchange Unix sub-forum used for duplicate question retrieval
cq/webmasters	17405	506	1395	Subset of Stack Exchange webmasters sub-forum used for duplicate question retrieval
cq/wordpress	48605	541	744	Subset of Stack Exchange Wordpress sub-forum used for duplicate question retrieval

A.2 Extra Results

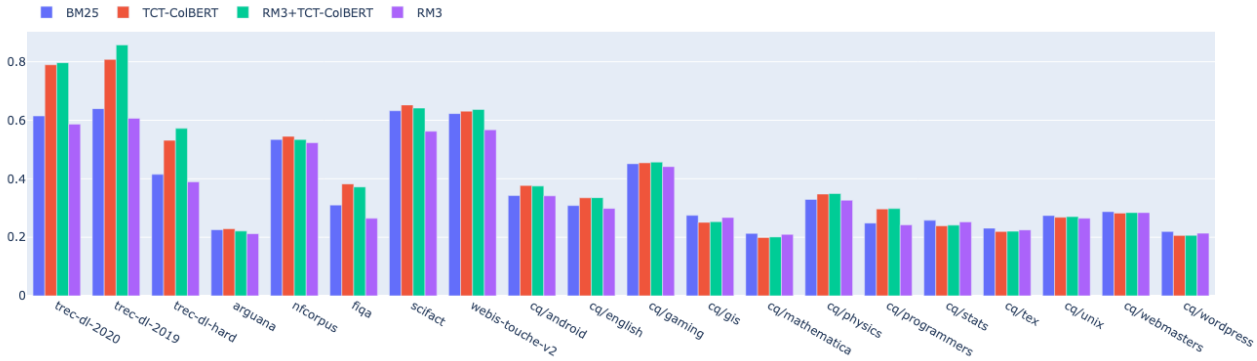


Figure 4: Results of running the experiments from RQ1 using the "RR @ 10" metric. The datasets used are described on the X-axis, while the Y-axis contains the RR values for each model.

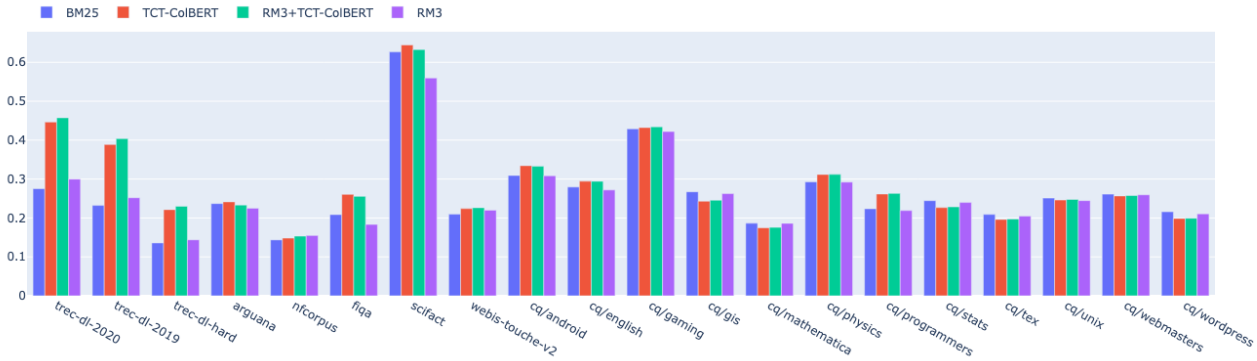


Figure 5: Results of running the experiments from RQ1 using the "MAP @ 100" metric. The datasets used are described on the X-axis, while the Y-axis contains the MAP values for each model.

A.3 Detailed Results

Table 5: Results of the experiment on the msmarco-passage/trec-dl-2020 dataset:

	RR(rel=2)@10	nDCG@10	AP(rel=2)@100
BM25	0.614675	0.493627	0.275282
RM3	0.586464	0.504314	0.299899
TCT-ColBERT	0.789506	0.686044	0.446461
RM3+TCT-ColBERT	0.796649	0.691352	0.457449

Table 6: Results of the experiment on the msmarco-passage/trec-dl-2019/judged dataset:

	RR(rel=2)@10	nDCG@10	AP(rel=2)@100
BM25	0.639655	0.479540	0.232165
RM3	0.606681	0.515595	0.251896
TCT-ColBERT	0.807752	0.692802	0.388712
RM3+TCT-ColBERT	0.857364	0.718088	0.403819

Table 7: Results of the experiment on the msmarco-passage/trec-dl-hard dataset:

	RR(rel=2)@10	nDCG@10	AP(rel=2)@100
BM25	0.415056	0.274333	0.135798
RM3	0.389222	0.270870	0.143734
TCT-ColBERT	0.531222	0.394371	0.221447
RM3+TCT-ColBERT	0.572056	0.404091	0.230071

Table 8: Results of the experiment on the arguana dataset:

	RR@10	nDCG@10	AP@100
BM25	0.225617	0.342442	0.236988
RM3	0.207798	0.308206	0.222128
TCT-ColBERT	0.214414	0.319316	0.227657
RM3+TCT-ColBERT	0.206548	0.296615	0.218786

Table 9: Results of the experiment on the nfcorpus dataset, with $\alpha = 0.5$:

	RR@10	nDCG@10	AP@100
BM25	0.534378	0.322219	0.143582
RM3	0.523259	0.331467	0.154988
TCT-ColBERT	0.544943	0.328974	0.148259
RM3+TCT-ColBERT	0.534144	0.333020	0.153332

Table 10: Results of the experiment on the fiqa dataset, with $\alpha = 0.05$:

	RR@10	nDCG@10	AP@100
BM25	0.310271	0.252589	0.208640
RM3	0.264714	0.228014	0.183207
TCT-ColBERT	0.382270	0.312334	0.260618
RM3+TCT-ColBERT	0.371894	0.308762	0.255465

Table 11: Results of the experiment on the scifact dataset, with $\alpha = 0.1$:

	RR@10	nDCG@10	AP@100
BM25	0.632427	0.672167	0.626749
RM3	0.562431	0.622227	0.559660
TCT-ColBERT	0.652175	0.686199	0.644616
RM3+TCT-ColBERT	0.641512	0.680644	0.632925

Table 12: Results of the experiment on the webis-touche-v2 dataset:

	RR@10	nDCG@10	AP@100
BM25	0.622846	0.342774	0.209593
RM3	0.567282	0.346563	0.219955
TCT-ColBERT	0.630977	0.356660	0.224018
RM3+TCT-ColBERT	0.636451	0.356520	0.226307

Table 13: Results of the experiment on the cq/android dataset:

	RR@10	nDCG@10	AP@100
BM25	0.225617	0.342442	0.236988
RM3	0.207798	0.308206	0.222128
TCT-ColBERT	0.229278	0.345821	0.241444
RM3+TCT-ColBERT	0.221094	0.336209	0.233351

Table 14: Results of the experiment on the cq/english dataset:

	RR@10	nDCG@10	AP@100
BM25	0.308292	0.305562	0.279482
RM3	0.298654	0.299227	0.271920
TCT-ColBERT	0.334736	0.326068	0.294369
RM3+TCT-ColBERT	0.335160	0.325327	0.294291

Table 15: Results of the experiment on the cq/gaming dataset:

	RR@10	nDCG@10	AP@100
BM25	0.451437	0.468988	0.428994
RM3	0.403317	0.435731	0.386717
TCT-ColBERT	0.454453	0.466276	0.432399
RM3+TCT-ColBERT	0.456849	0.468423	0.434427

Table 16: Results of the experiment on the cq/gis dataset:

	RR@10	nDCG@10	AP@100
BM25	0.275023	0.294736	0.267279
RM3	0.245500	0.270527	0.242725
TCT-ColBERT	0.2511052	0.269571	0.242997
RM3+TCT-ColBERT	0.2531759	0.270903	0.245564

Table 17: Results of the experiment on the cq/mathematica dataset:

	RR@10	nDCG@10	AP@100
BM25	0.213123	0.214626	0.186297
RM3	0.190267	0.201326	0.171837
TCT-ColBERT	0.199041	0.203824	0.174633
RM3+TCT-ColBERT	0.201007	0.205585	0.175706

Table 18: Results of the experiment on the cq/physics dataset:

	RR@10	nDCG@10	AP@100
BM25	0.329452	0.332119	0.292656
RM3	0.326568	0.329985	0.292167
TCT-ColBERT	0.347660	0.349216	0.311791
RM3+TCT-ColBERT	0.349334	0.350632	0.311989

Table 19: Results of the experiment on the cq/programmers dataset:

	RR@10	nDCG@10	AP@100
BM25	0.248566	0.250200	0.223630
RM3	0.242422	0.245588	0.219096
TCT-ColBERT	0.296837	0.290960	0.261580
RM3+TCT-ColBERT	0.297944	0.292677	0.262992

Table 20: Results of the experiment on the cq/stats dataset:

	RR@10	nDCG@10	AP@100
BM25	0.257951	0.265192	0.244461
RM3	0.252610	0.264646	0.239736
TCT-ColBERT	0.239209	0.247859	0.226616
RM3+TCT-ColBERT	0.241359	0.249655	0.228460

Table 21: Results of the experiment on the cq/tex dataset:

	RR@10	nDCG@10	AP@100
BM25	0.230964	0.234625	0.209105
RM3	0.225147	0.229494	0.204637
TCT-ColBERT	0.219527	0.220506	0.196252
RM3+TCT-ColBERT	0.220484	0.221305	0.197070

Table 22: Results of the experiment on the cq/unix dataset:

	RR@10	nDCG@10	AP@100
BM25	0.274394	0.277035	0.251039
RM3	0.264915	0.271276	0.244572
TCT-ColBERT	0.268316	0.272659	0.246095
RM3+TCT-ColBERT	0.270348	0.274693	0.247698

Table 23: Results of the experiment on the cq/webmasters dataset:

	RR@10	nDCG@10	AP@100
BM25	0.287533	0.289924	0.261102
RM3	0.284346	0.289954	0.259468
TCT-ColBERT	0.282458	0.284944	0.256601
RM3+TCT-ColBERT	0.284427	0.287200	0.257534

Table 24: Results of the experiment on the cq/wordpress dataset:

	RR@10	nDCG@10	AP@100
BM25	0.219477	0.237277	0.215881
RM3	0.213735	0.231421	0.210678
TCT-ColBERT	0.206146	0.220715	0.198848
RM3+TCT-ColBERT	0.206480	0.220006	0.199380