Generative Relevance Feedback and Convergence of Adaptive Re-Ranking: University of Glasgow Terrier Team at TREC DL 2023

Andrew Parry
a.parry.1@research.gla.ac.uk
University of Glasgow
Glasgow, UK

Thomas Jänich
t.jaenich.1@research.gla.ac.uk
University of Glasgow
Glasgow, UK

Sean MacAvaney
Iadh Ounis
first.last@glasgow.ac.uk
University of Glasgow
Glasgow, UK

ABSTRACT

This paper describes our participation in the TREC 2023 Deep Learning Track. We submitted runs that apply generative relevance feedback from a large language model in both a zero-shot and pseudo-relevance feedback setting over two sparse retrieval approaches, namely BM25 and SPLADE. We couple this first stage with adaptive re-ranking over a BM25 corpus graph scored using a monoELECTRA cross-encoder. We investigate the efficacy of these generative approaches for different query types in first-stage retrieval. In re-ranking, we investigate operating points of adaptive re-ranking with different first stages to find the point in graph traversal where the first stage no longer has an effect on the performance of the overall retrieval pipeline. We find some performance gains from the application of generative query reformulation. However, our strongest run in terms of P@10 and nDCG@10 applied both adaptive re-ranking and generative pseudo-relevance feedback, namely uogtr_b_grf_e_gb.

1 INTRODUCTION

The University of Glasgow Terrier team participated in the TREC 2023 Deep Learning track to further explore new generative approaches to retrieval and validate existing approaches by a deeper exploration of their performance. We investigate generative approaches to relevance feedback utilising both generative query reformulation (Gen-QR) and pseudo-relevance feedback (Gen-PRF) [22] as well as conducting a further evaluation of adaptive re-ranking [13] on the MS MARCO-v2 corpus [14]. Both of these approaches overcome the lexical mismatch problem of many classic retrieval approaches, albeit in different ways. A re-ranker is normally constrained by the texts that could be retrieved by a first-stage model. Generative relevance feedback alleviates this problem by adding expansion terms to the query text attempting to improve the recall of first-stage retrieval models. Adaptive re-ranking instead expands the initial ranking with candidate documents found by traversal of a corpus graph to find the nearest neighbours of highly ranked texts.

We explored the following research questions: (1) Can we validate the performance improvements from generative relevance feedback on a different test set and downstream ranking model? (2) What type of queries are most improved or harmed by generative relevance feedback? and (3) Can the use of a sufficiently large corpus graph allow a lightweight first-stage ranker to converge to identical performance of a first-stage learned retrieval function?

To answer these questions, we used our PyTerrier Information Retrieval (IR) toolkit [15], which allows for the composition of both lexical and neural retrieval components into flexible pipelines. We apply two forms of generative relevance feedback over two sparse first-stage ranking models. Moreover, we apply adaptive re-ranking with monoELECTRA [16] in this setting to assess the need for complex first-stage pipelines (learned models or weighted expansion terms) given a sufficiently large corpus graph. Specifically, we use a large search budget of 5000 over a 32 nearest-neighbour BM25 corpus graph.

The structure of the remainder of this paper is as follows: Section 2 details the notation of retrieval pipelines composed in PyTerrier. Section 3 describes the methods used in our experiments and the particular models used in these methods. We then outline our experiment setup. Section 4 summarises our submitted baseline and group runs. In Section 5 we present our results and analysis before providing some concluding remarks in Section 6.

2 PYTERRIER RETRIEVAL PIPELINES

Our experiments and submitted runs for the TREC 2023 Deep Learning Track have been built upon PyTerrier, Python bindings for the Terrier search engine [15]. The key abstraction of PyTerrier is the *Transformer* object, of which all retrieval components are a subclass. Using Pandas dataframes, a transformer object *transforms* one dataframe to another. To create multi-stage retrieval pipelines, PyTerrier overloads the bitwise shift operators (« and ») to allow the chaining of multiple transformers into a single component where the output of each transformer is directly passed into the next sequentially.

We express all ranking features described in the rest of this paper as pipelines of transformers using this sequential operator. For more information about the PyTerrier platform and the operators' flexibility, we refer the reader to the documentation, which can be found at https://pyterrier.readthedocs.io/en/latest/.

3 METHODS

In this section, we outline the background knowledge of Generative Relevance Feedback (Section 3.1) and Adaptive Re-Ranking (Section 3.2), followed by our experimental setup in Section 3.3.

3.1 Generative Query Reformulation & Pseudo-Relevance Feedback

Generative query reformulation (Gen-QR) and generative pseudo-relevance feedback (Gen-PRF) are mechanisms for the expansion of query terms in ranking pipelines [22]. Originally proposed with both a weakly supervised fine-tuned variant as well as a zero-shot prompted instruction-tuned variant, we only cover the latter as

we do not utilise any fine-tuned checkpoints in our generative expansion stage.

Recent developments in language model pre-training [17, 20, 21] have led to a new paradigm in language modelling known as prompting [5] in which a task is described with some input in a zero-shot setting to a language model. Due to extensive pre-training and additional fine-tuning on instruction style inputs [6], these models can perform a task that has not been observed in training data [5]. This zero-shot generalisation poses some benefits in retrieval as the labelling of ground truth by experts for neural models is an expensive process [1].

Following Wang et al. [22], we define query reformulation as a process $\mathcal{P}$ which, given a user query q^0, reformulates it as q^r potentially conditioned on some additional context to improve the retrieval effectiveness of the downstream retrieval pipeline.

$$q^r = \mathcal{P}(q^0, \dots) \tag{1}$$

In equation $1, \dots$ is any additional context, such as an initial ranking in the case of pseudo-relevance feedback. Generative relevance feedback prompts a large language model (LLM) to suggest expansion terms to improve the performance of a retrieval model [1]. In the case of Gen-QR, no additional context is supplied to the generative model. However, when applying Gen-PRF, a ranking of K documents from a corpus D by some score function s, $\{s(d, q^0); \forall d \in D\}$ is provided as topical context for the expansion and/or re-weighting of query terms.

The authors propose multiple pruning criteria to reduce the context size provided to the Gen-PRF model. We use the approach *Top-P* and outline its method as follows. To select M context pieces from documents in an initial ranking R (this process is simplified for passage ranking as we do not further split passages), each document d is split into passages. The context C of length M is then chosen as follows:

$$C_{Top\text{-}P}(R, q^0) = \underset{p \in R}{\overset{M}{\arg\max}}\ s\ s(p, q^0) \tag{2}$$

Example pipelines are as follows: Gen-QR with FLAN-T5 can be applied before BM25 retrieval, re-ranked with monoELECTRA:

$$\text{QR(FLAN-T5)} >> \text{BM25} >> \text{monoELECTRA} \tag{3}$$

Where in Equation 3, QR(FLAN-T5) is the generative reformulation of a query using FLAN-T5. Alternatively, Top-P context Gen-PRF with FLAN-T5 can be applied to a first-stage retrieval by BM25, re-ranked by monoELECTRA:

$$\text{BM25} >> \text{PRF}_{Top\text{-}P}\text{(FLAN-T5)} >> \text{BM25} >> \text{monoELECTRA} \tag{4}$$

3.2 Graph-based Adaptive Re-Ranking (GAR)

We utilise graph-based adaptive re-ranking (GAR), which is an efficient approach to improving re-ranking performance [13]. GAR works within a cascading retrieval pipeline in which new candidate documents are found by traversing a nearest-neighbour graph $G = (V, E)$ where each node in V represents a document in the corpus. Each edge in E is weighted by some heuristic, either a lexical or semantic similarity score between two nodes. Online latency is unaffected by the chosen number of nearest neighbours. However, due to the graph structure's quadratic time and space complexity, we limit the number of nearest neighbours to a small number and prune by sorting edge weights in descending order. An online traversal of this corpus graph is performed by scoring the nearest neighbours of the initial ranking using some score model S. The highest-ranking document neighbours from the current corpus graph frontier are added to a re-ranking pool. The initial pool is then revisited to update the graph frontier. This process alternates between scoring pools until a predefined compute budget is met.

An example pipeline is as follows: A first-stage BM25 retrieval is adaptively re-ranked using GAR with a BM25 corpus graph and a monoELECTRA scoring model:

$$\text{BM25} >> \text{GAR}_{\text{BM25}}\text{(monoELECTRA)} \tag{5}$$

Where in Equation 5, GAR_{BM25}(monoELECTRA) represents adaptive re-ranking over a BM25 corpus graph using monoELECTRA.

3.3 Experimental Setup

We use the following retrieval components, grouped into methods that perform retrieval or re-ranking and methods that perform query expansion in the form of query reformulation (QR), pseudo-relevance feedback (PRF), or adaptive re-ranking (GAR).

Retrieval:

- DPH [3] & BM25 [18]: Lexical retrieval from a Terrier inverted index over the msmarco-passage-v2 corpus.
- SPLADE[2]: A distilled SPLADE++ learned sparse model [10, 11].
- monoELECTRA[3]: A cross-encoder pre-trained with ELECTRA-style objectives [7] and fine-tuned with 31 localized negatives [16].

Query Expansion:

- Bo1 : Pseudo-relevance feedback using the DFR Bo1 model [2] over a Terrier index.
- Gen-QR (QR(LLM)): Using the generative encoder-decoder FLAN-T5[4] [6] to provide zero-shot query expansion terms with term weights controlled uniformly by the parameter $\beta = 0.5$. In all cases, we use tuned generation hyperparameters from Wang et al. [22].
- Gen-PRF (PRF_C(LLM)): Using the generative encoder-decoder FLAN-T5 to provide query expansion terms using a first-stage ranking as context C. Term weights are again controlled uniformly by the parameter $\beta = 0.5$.
- $\text{GAR}_G(S)$: Graph-based Adaptive Re-Ranking using corpus graph G and scoring function S [13]. We use a BM25-based corpus graph with a re-ranking budget of 5000 and 32 nearest neighbours.

All experiments were conducted in PyTerrier using the ir-datasets package for corpora and additional test sets (DL-2021 [8], DL-2022 [9]). PyTerrier is available from https://github.com/terrier-

[1] Prompt for Gen-PRF: Improve the search effectiveness by suggesting expansion terms for the query:{input_query}, based on the given context information: {context}

[2] naver/splade-cocondenser-ensembledistil

[3] crystina-z/monoELECTRA_LCE_nneg31

[4] google/flant5-xxl

org/pyterrier and the implementation of monoELECTRA is available from https://github.com/terrierteam/pyterrier_dr. The implementation of GAR is available from https://github.com/terrierteam/pyterrier_adaptive and the implementation of Generative QR and PRF are available from https://github.com/Parry-Parry/pyterrier_GenerativeQR. We perform all experiments on a de-duped msmarco-v2 corpus due to the added benefits for GAR corpus graph traversal preventing duplicate nearest neighbours. We then added duplicates before submission. All experiment code is available from https://github.com/Parry-Parry/terrier_trec_2023.

4 SUBMITTED RUNS

We submitted six group runs to the passage ranking task. We also submitted five baseline runs. We did not participate in the document ranking task.

4.1 Baseline Runs

As baseline runs submitted to the 2023 Deep Learning passage ranking track, we chose two sparse retrieval models, one learned and one non-parametric model, one sparse retrieval model with query expansion and two sparse retrieval models re-ranked by a monoELECTRA cross-encoder. All baselines are summarised as follows:

- uofg_tr_dph: Performs DPH retrieval on our passage sparse index.
- uofg_tr_s : Performs SPLADE retrieval retrieval on an msmarco-passage-v2 SPLADE learned sparse index.
- uofg_tr_dph_bo1: Performs Bo1 divergence from randomness query expansion over DPH retrieval on our passage sparse index.
- uofg_tr_be: Re-Ranks a first-stage BM25 retrieval using a monoELECTRA cross-encoder.
- uofg_tr_se: Re-Ranks a first-stage SPLADE retrieval using monoELECTRA.

4.2 Submitted Group Runs

For the 2023 Deep Learning passage ranking track, we submitted the following three group runs:

- uofg_tr_se_gb: Performs SPLADE retrieval, adaptively re-ranked by monoELECTRA and a BM25 graph.
- uofg_tr_qr_be_gb: Performs Generative Query Reformulation before BM25 retrieval, adaptively re-ranked by monoELECTRA and a BM25 graph.
- uofg_tr_b_grf_e_gb: Performs Generative Relevance Feedback over a first-stage BM25 retrieval re-ranked by monoELECTRA and a BM25 graph.

4.3 Additional Runs

For the 2023 Deep Learning passage ranking track, we submitted the following additional three runs:

- uofg_tr_qr_be: Performs Generative Query Reformulation before BM25 retrieval re-ranked by monoELECTRA.
- uofg_tr_b_grf_e: Performs Generative Relevance Feedback over a first-stage BM25 retrieval re-ranked by monoELECTRA.
- uofg_tr_be_gb: Performs BM25 retrieval, adaptively re-ranked by monoELECTRA and a BM25 graph.

5 RESULTS & ANALYSIS

In Table 1, we present the performance of each of our runs contrasting group runs with baseline runs[5] and the Best and Median results averaged across all judged topics. Across all metrics, our group runs improve over the median, with our additional runs also improving over the median in some cases, notably when adaptive re-ranking is applied with monoELECTRA over BM25 (uogtr_be_gb).

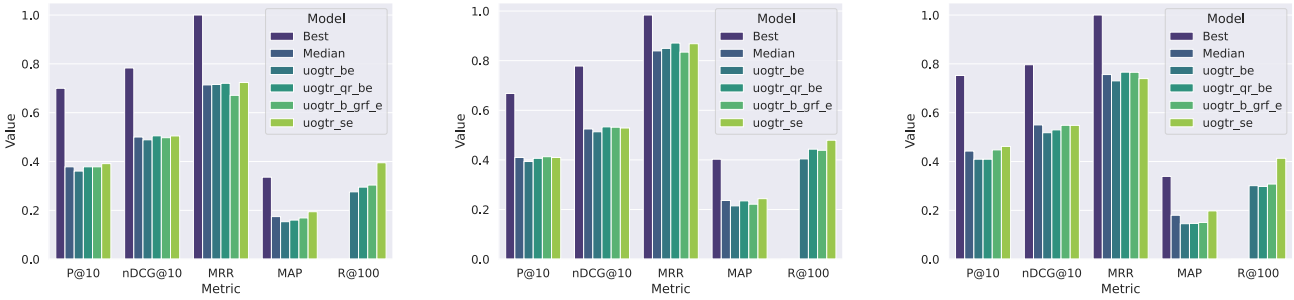
Generative methods versus SPLADE. Our SPLADE baseline re-ranked with monoELECTRA (uogtr_se) outperforms all runs with BM25 first stages in terms of Recall@100 and MAP even when adaptive re-ranking is applied and outperforms all Gen-QR runs in terms of nDCG@10. As generative relevance feedback is a completely zero-shot method, this is not unexpected and consistent performance improvements are observed across all metrics. Comparing uogtr_be and uogtr_qr_be, the diversification of query terms by Gen-QR is effective in improving first-stage lexical ranking models but fails to outperform the expansions of the learned SPLADE model. We observe improvements over SPLADE when applying Gen-PRF in nDCG@10, MAP, and R@100, suggesting that the context provided from the first-stage ranking is sufficient to ground expansion terms to suitable topics.

GAR is a stronger standalone method. Contrasting the use of generative relevance feedback (uogtr_qr_be and uogtr_b_grf_e) against the use of GAR (uogtr_be_gb) we observe that GAR alone provides stronger candidates to the re-ranking stage than Gen-QR or Gen-PRF improving both precision and recall based metrics. Furthermore, measured by 4 of 5 metrics, the use of GAR is more effective in improving performance than the use of SPLADE expansions. We propose that this is due to the rank bias of the GAR algorithm, as candidate texts are chosen from the neighbourhood of the highest-ranking documents aligning well with the cluster hypothesis in contrast to the zero-shot expansion terms provided by generative relevance feedback. However, the computational expense of GAR is determined by a compute budget, and further research is warranted to find the Pareto-optimal approach for smaller budgets.

Gen-PRF with GAR is most effective. Our most effective run by P@10 and nDCG@10 uses Gen-PRF and adaptive re-ranking. We observe improvements in nDCG@10 over SPLADE expansions (0.5489 versus 0.5394) when using GAR, suggesting that the diversification provided by generative expansion terms can be more effective in providing a strong initial pool for GAR corpus graph traversal. We note that SPLADE-based runs (uogtr_se and uogtr_se_gb) have greater recall and MAP than any generative relevance feedback pipelines. However, the zero-shot nature of these approaches coupled with user tendencies to only interact with the top 10 results of a search [12] reinforces generative relevance feedback as a compelling approach compared to learned first-stage models. Furthermore, addressing RQ (1), we find that Gen-QR and Gen-PRF generalise to a new test set and improve the retrieval performance of a monoELECTRA-based pipeline with and without adaptive re-ranking.

Table 1: Results on the TREC Deep Learning 2023 Passage Ranking task. Best and Median performance is an aggregate over the best-performing model for each topic. Runs that outperform the median are denoted with ⋄. The strongest performance in each metric is denoted bold, the second strongest is underlined.

Run ID	Pipeline	P@10	NDCG@10	MRR	MAP	R@100
Best (Per-topic aggregate)		0.7000	0.7892	0.9939	0.3839	-
Median (Per-topic aggregate)		0.4085	0.5329	0.7803	0.2159	-
Baseline Runs						
uogtr_dph	DPH	0.1805	0.2825	0.4320	0.0840	0.2310
uogtr_s	SPLADE	0.3549	0.4706	0.6725	0.1925	0.4144
uogtr_dph_bo1	DPH » Bo1 » DPH	0.1317	0.2377	0.3613	0.0600	0.1288
uogtr_be	BM25 » monoELECTRA	0.3939	0.5227	0.7881⋄	0.1940	0.3558
uogtr_se	SPLADE » monoELECTRA	0.4256⋄	0.5364⋄	0.7938⋄	0.2348⋄	0.4531
Additional Runs						
uogtr_qr_be	QR(FLAN-T5) » BM25 » monoELECTRA	0.3963	0.5316	0.8018⋄	0.1994	0.3668
uogtr_b_grf_e	BM25 » PRF$_{Top\text{-}P}$(FLAN-T5) » BM25 » monoELECTRA	0.4122⋄	0.5376⋄	0.7790	0.1996	0.3770
uogtr_be_gb	BM25 » GAR$_{BM25}$(monoELECTRA)	0.4159⋄	0.5451⋄	**0.8046**⋄	0.2285⋄	0.4240
Group Runs						
uogtr_se_gb	SPLADE » GAR$_{BM25}$(monoELECTRA)	0.4293⋄	0.5394⋄	0.7934⋄	**0.2401**⋄	**0.4706**
uogtr_qr_be_gb	QR(FLAN-T5) » BM25 » GAR$_{BM25}$(monoELECTRA)	0.4244⋄	0.5488⋄	0.7953⋄	0.2315⋄	0.4248
uogtr_b_grf_e_gb	BM25 » PRF$_{Top\text{-}P}$(FLAN-T5) » BM25 » GAR$_{BM25}$(monoELECTRA)	**0.4305**⋄	**0.5489**⋄	0.7885⋄	0.2314⋄	0.4328



(a) Open Directed Queries (b) Closed Directed Queries (c) Advice Queries

Figure 1: Results for each main query type comparing BM25 » monoELECTRA, QR(FLAN-T5) » BM25 » monoELECTRA, BM25 » PRF$_{Top\text{-}P}$(FLAN-T5) monoELECTRA and SPLADE » monoELECTRA.

5.1 Generative Relevance Feedback

To assess where generative relevance feedback is most effective, we compare generative expansions with learned expansions via SPLADE as well as a standard re-ranking pipeline across different query types. We refer users to the query type definitions proposed by Broder [4] and expanded by Rose and Levinson [19]. We focus on directed queries and advice, as the majority of test queries can be grouped into these classes. As defined by Rose and Levinson [19], directed queries aim to learn a particular piece of information about a topic. They can be either closed when there is "a single, unambiguous answer" to a query or open when the query is an "open-ended question or one with unconstrained depth". Advice queries are requests for "advice, ideas, suggestion or instructions".

We manually classified each Deep Learning 2023 test query into the groupings mentioned above. From 82 queries, we found that 31 fell in the directed closed class, 23 in the directed open class and 21 in the advice class. We investigate the performance of our runs within these query groupings.

As shown in Figure 1 (a), in open-directed queries, we find that Gen-QR is more effective than Gen-PRF or SPLADE in terms of nDCG@10. As the zero-shot expansions of Gen-QR are not grounded by either training in-domain or few-shot context from a first-stage ranker we hypothesize that in cases where the information need is broad, the relatively unconstrained diversification provided by such expansion terms leads to an improvement in both P@10 and nDCG@10 over other methods. A SPLADE first-stage retrieval (uogtr_se) shows the strongest performance in terms of MRR, MAP, and R@100, suggesting that the in-domain training of

[5]Due to an issue with indexing, expansion terms degrade DPH (uogtr_dph_bo1).

SPLADE allows for the retrieval of more partially relevant documents than a generative approach.

In Figure 1 (b) we present performance on closed-directed queries. We observe that any linear trend in metric performance across all official evaluation metrics observed in Table 1 does not hold for queries with an unambiguous intent. We consider that diversification of terms may be unnecessary in such cases with Gen-PRF (uogtr_b_grf_e) actually degrading performance over a standard monoELECTRA re-ranker (uogtr_be) in terms of MRR. Furthermore, we observe that in terms of R@100 the grounding of Gen-PRF using first-stage ranking context does not improve recall over Gen-QR, reinforcing that pseudo-relevance feedback is unnecessary in these cases.

In Figure 1 (c) we present performance on advice queries. We observe that Gen-QR generally performs worse than Gen-PRF and SPLADE in terms of P@10 and nDCG@10. Upon inspection of generated expansions by Gen-QR, we observe that without either fine-tuning or first-stage context to ground expansion terms, the LLM directly generated advice instead of expansion terms. As we removed stopwords post-generation, much of the semantic structure was not used. However, many terms formed a part of an answer as opposed to more general terms for expansion. Inherently this is an artefact of the instruction-tuning of FLAN-T5, it does not appear to affect search more generally, as we still observe strong performance across other query types. However, this finding may explain the margin between Gen-QR and Gen-PRF. In future applications, either fine-tuning or the use of a stronger LLM may alleviate this issue. In the case of question-style or conversational-style queries, the LLM misunderstands the full instruction and instead attempts to answer the question directly which is not always conducive to the generation of effective expansion terms.

Concerning RQ (2), we find that generative relevance feedback can improve directed queries with both clear and ambiguous intent. However, performance can be harmed when a query is posed as a conversational style question, as the underlying LLM can attempt to answer the question instead of generating suitable expansion terms.

5.2 Convergence of Adaptive Re-Ranking

Observing similar performance (identical top 10 texts for multiple test queries) when adaptive re-ranking is applied with a large corpus graph and budget regardless of first-stage ranker (uogtr_be_gb and uogtr_se_gb), we consider that by using GAR, the need for complex first-stage retrieval may be reduced. As relevance judgements are not available at the time of analysis, we instead use Deep Learning 2022 relevance judgements and test queries for this analysis. To assess the effect of compute budget on GAR, we apply adaptive ranking with a compute budget ranging from 100 to 5000 texts. Following TREC Deep Learning assessments, we truncate the final rankings to 100 texts. We contrast GAR with full re-ranking, allowing the first-stage retriever to rank up to 5000 texts.

We observe in Figure 2 (a) that rankings by Rank Biased Overlap (RBO) become increasingly correlated as budget increases, with the trend plateauing around a budget of 4000. Around a budget of 1000, variance begins to increase as rankings converge, whereas, below this point, BM25 and SPLADE rankings are almost completely diverging. As one would expect, the correlation between full re-rankings of BM25 and SPLADE is not as strong, however does follow a similar trend to adaptive re-ranking.

We also observe that from a budget of 2000, it can be seen in Figure 2 (b) that BM25 and SPLADE first stages are within a small margin of each other in terms of nDCG@10, we found that on both P@10 and R@100, a similar trend was present. We observe that SPLADE is only marginally improved by adaptive re-ranking, as is shown by contrast with full re-ranking and performance measured on the Deep Learning 2023 test queries (uogtr_se versus uogtr_se_gb) in Table 1. We observe that full re-ranking of BM25 fails to show continuous improvement with nDCG@10 performance plateauing around a budget of 2000, similar to adaptive re-ranking, however, at a significantly lower value (t-test, $p < 0.05$). This can be attributed to lexical mismatch as full re-ranking is limited by term overlap with a query, whereas GAR can make document-wise term overlap comparisons. We find that metric performance between SPLADE and BM25 using adaptive re-ranking is insignificant from a budget of 2000 (t-test, $p < 0.05$). Addressing RQ (3), in cases where labelled data is unavailable or financially infeasible to collect, this finding is compelling as the performance of a learned sparse model can be closely replicated by BM25 when using adaptive re-ranking with the caveat that one must have a suitable re-ranker.

6 CONCLUSIONS

In summary, our participation in the TREC 2023 Deep Leaning track has been insightful in validating recent approaches to the expansion of both query terms and first-stage rankings. In answering our research questions, we have found that generative relevance feedback can transfer to a monoELECTRA cross-encoder and is further bolstered by adaptive re-ranking. We find that though generative relevance feedback can be generally effective, the approach is sensitive to the form of the query being direct as opposed to conversational, in which case performance can degrade due to artefacts from instruction-tuning. We also find that with sufficient compute budget and corpus graph size, a first-stage lexical model can closely replicate the metric performance of a learned sparse retrieval model, with both models' rankings becoming increasingly correlated by RBO reaching a maximum correlation of 0.80.

REFERENCES

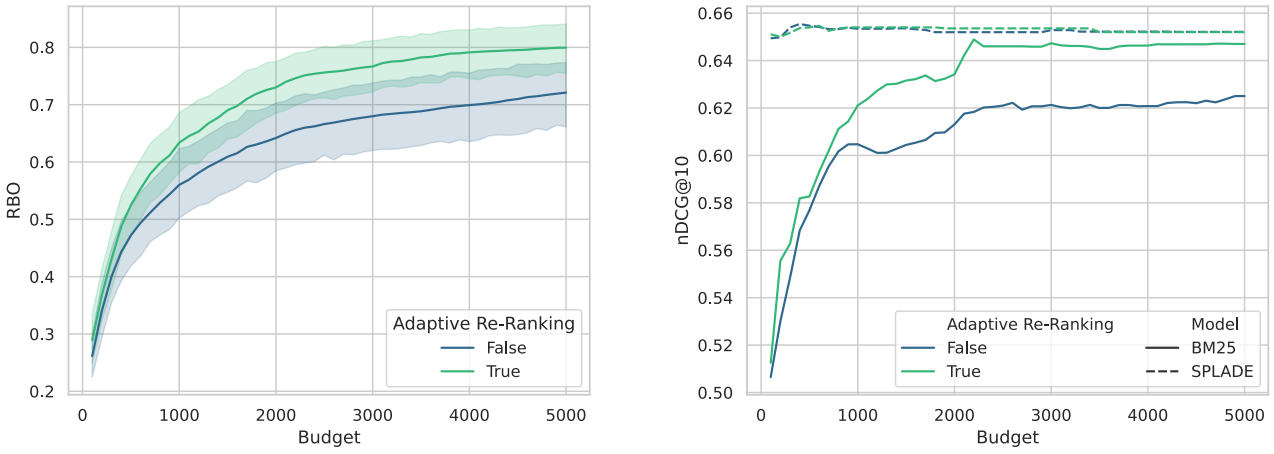
[1] Sophia Althammer, Guido Zuccon, Sebastian Hofstätter, Suzan Verberne, and Allan Hanbury. 2023. Annotating Data for Fine-Tuning a Neural Ranker? Current Active Learning Strategies are not Better than Random Selection. *CoRR* abs/2309.06131 (2023). https://doi.org/10.48550/ARXIV.2309.06131 arXiv:2309.06131

[2] Giambattista Amati. 2003. *Probability models for information retrieval based on divergence from randomness.* PhD. University of Glasgow. https://eleanor.lib.gla.ac.uk/record=b2151999

[3] Giambattista Amati. 2006. Frequentist and Bayesian Approach to Information Retrieval. In *Advances in Information Retrieval, 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 3936)*, Mounia Lalmas, Andy MacFarlane, Stefan M. Rüger, Anastasios Tombros, Theodora Tsikrika, and Alexei Yavlinsky (Eds.). Springer, 13–24. https://doi.org/10.1007/11735106_3

[4] Andrei Z. Broder. 2002. A taxonomy of web search. *SIGIR Forum* 36, 2 (2002), 3–10. https://doi.org/10.1145/792550.792552

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter,



(a) RBO Correlation **(b) nDCG@10 Performance**

Figure 2: Contrasting adaptive re-ranking over BM25 and SPLADE first stages with full re-ranking at different budgets on TREC Deep Learning 2022 queries. Rankings are truncated to the top 100 results. Post-ranking, duplicates were added to the rankings to follow the current TREC procedure on MS MARCO-v2. Error bands are omitted from Figure (b) for clarity.

Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *CoRR* abs/2005.14165 (2020). arXiv:2005.14165 https://arxiv.org/abs/2005.14165

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. *CoRR* abs/2210.11416 (2022). https://doi.org/10.48550/ARXIV.2210.11416 arXiv:2210.11416

[7] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *CoRR* abs/2003.10555 (2020). arXiv:2003.10555 https://arxiv.org/abs/2003.10555

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Overview of the TREC 2021 Deep Learning Track. In *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021, online, November 15-19, 2021 (NIST Special Publication, Vol. 500-335)*, Ian Soboroff and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec30/papers/Overview-DL.pdf

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2022. Overview of the TREC 2022 Deep Learning Track. In *Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, online, November 15-19, 2022 (NIST Special Publication, Vol. 500-338)*, Ian Soboroff and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec31/papers/Overview_deep.pdf

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

[11] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2288–2292. https://doi.org/10.1145/3404835.3463098

[12] Diane Kelly and Leif Azzopardi. 2015. How many results per page?: A Study of SERP Size, Search Behavior and User Experience. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto (Eds.). ACM, 183–192. https://doi.org/10.1145/2766462.2767732

[13] Sean MacAvaney, Nicola Tonellotto, and Craig Macdonald. 2022. Adaptive Re-Ranking with a Corpus Graph. *CoRR* abs/2208.08942 (2022). https://doi.org/10.48550/ARXIV.2208.08942 arXiv:2208.08942

[14] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*, Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[15] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. 2005. Terrier Information Retrieval Platform. In *Advances in Information Retrieval, 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005, Proceedings (Lecture Notes in Computer Science, Vol. 3408)*, David E. Losada and Juan M. Fernández-Luna (Eds.). Springer, 517–519. https://doi.org/10.1007/978-3-540-31865-1_37

[16] Ronak Pradeep, Yuqi Liu, Xinyu Zhang, Yilin Li, Andrew Yates, and Jimmy Lin. 2022. Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13185)*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer, 655–670. https://doi.org/10.1007/978-3-030-99736-6_44

[17] Radford, Narasimhan, Saliman, and Sutskever. 2017. Improving language understanding with unsupervised learning. https://openai.com/research/language-unsupervised

[18] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994 (NIST Special Publication, Vol. 500-225)*, Donna K. Harman (Ed.). National Institute of Standards and Technology (NIST), 109–126. http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

[19] Daniel E. Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills (Eds.). ACM, 13–19. https://doi.org/10.1145/988672.988675

[20] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca Publication Title: GitHub repository.

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). https://doi.org/10.48550/ARXIV.2302.13971 arXiv:2302.13971

[22] Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2023. Generative Query Reformulation for Effective Adhoc Search. *CoRR* abs/2308.00415 (2023). https://doi.org/10.48550/ARXIV.2308.00415 arXiv:2308.00415