Re-Ranking Dense Retrieval

Jake Norton
University of Otago
Dunedin, Otago, New Zealand
norja159@student.otago.ac.nz

ABSTRACT

This paper explores advanced retrieval techniques aimed at enhancing the efficiency and effectiveness of information retrieval systems through use of graph-based and latent feature methodologies. I analyze three distinct approaches: Latent Approximate Document Retrieval (LADR)[5], which utilizes latent features to optimize document retrieval; adaptive re-ranking with a corpus graph (GAR)[6], which dynamically adjusts document rankings within a corpus graph to improve relevance; and adhoc retrieval through traversal of a query-document graph(QDGT)[2], which employs direct traversal methods to optimize query-specific document retrieval. I propose to build ontop of these by trying the query-document mapping but using the LADR methodologies. Try other LADR with smarter exploration ideas.

CCS CONCEPTS

• **Information systems → Novelty in information retrieval**.

KEYWORDS

dense retrieval, approximate k nearest neighbour, adaptive re-ranking, neural re-ranking, clustering hypothesis

1 INTRODUCTION

Dense retrieval is gaining popularity, driven by the widespread adoption of pre-trained language models such as BERT and GPT, which excel at learning semantic representations of words. This approach involves several trade-offs, particularly noticeable in re-ranking pipelines where choices must be made about the size of the initial ranked set and the computational budget allocated for further re-ranking.

One of the key advantages of dense retrieval is its ability to improve recall by enabling queries based on semantic meaning rather than exact lexical matches. However, this benefit comes at the cost of increased computational demand during both indexing and query-time. Additionally, techniques that use approximation approaches in the second re-ranking phase often suffer from diminished recall, which is counterproductive to their primary objective.

Ideally, to surpass the performance of traditional models, an exhaustive search over all documents would be performed. However, such a strategy scales linearly with the number of documents and allows no pruning, in contrast to lexical approaches.

Enhancing recall remains a significant challenge in information retrieval. Alternative methods, such as those employed by models like SPADE[1], leverage query expansion to identify semantically similar documents in the initial stage, potentially improving recall with reduced computational overhead at query-time, though still requiring additional effort during indexing.

In dense retrieval, models are broadly classified into two categories, interactive and non-interactive. Non-interactive models employ static vector embeddings which remain unchanged regardless of the query context. In contrast, interactive models dynamically adjust the interaction between the query and document representations in the index, enabling more responsive and context-sensitive retrieval.

A prevalent technique in this field is the use of the top-ranked results from an initial query as a "seed" to guide the re-ranking and exploration processes. This approach leverages the initial results to refine and enhance subsequent retrieval efforts.

Supporting these techniques is the clustering hypothesis[4], which posits that documents similar in content are likely to be relevant to the same queries. This hypothesis is fundamental to many advanced exploration techniques in dense retrieval, as it allows for more efficient and effective identification of relevant documents through clustering-based strategies.

2 LEXICALLY-ACCELERATED DENSE RETRIEVAL

Lexically-Accelerated Dense Retrieval (LADR)[5] uses lexical retrieval to seed a dense retrieval exploration, leveraging a document proximity graph. This method establishes a new effectiveness-efficiency Pareto frontier, requiring only a CPU, which offers significant advantages in terms of cost and ease of deployment.

Instead of developing an entirely new paradigm, LADR builds on traditional lexical techniques. Specifically, it uses an initial ranking with BM25 to efficiently create a top-K document pool, followed by re-ranking with semantically aware vector embeddings. This approach does not solve the recall problem entirely, as the pool of documents is limited by the initial ranking, which primarily improves the ordering of the already gathered pool, enhancing precision.

Broadly speaking, LADR uses the initial ranking to seed an exploration of semantically similar documents in subsequent re-ranking rounds. This approach is similar to an idea proposed by GAR[6] for finding additional documents to score when re-ranking. However, LADR differs by exploring similar documents as a method of pruning the dense retrieval search space while maintaining comparable performance to an exhaustive search.

Documents are indexed using a standard lexical retrieval model. LADR employs two strategies:

(1) Obtain n documents using the query input into a BM25 model as the seed.
(2) Expand the document pool by unioning the seed documents with the K nearest neighbours of those documents.
(3) Store the final results of the expansion in R1, providing both the ranking and the expanded pool.

Algorithm 1 Proactive Algorithm

```
1: Input: $q$ query, $n$ seed set size, $k$ number of neighbours
2: Output: $R1$ dense retrieval results
3: let $R0 \leftarrow$ obtain $n$ documents using the query input into an initial model
4: let $P \leftarrow R0$
5: let $R1 \leftarrow 0$
6: $R1 \leftarrow R1 \cup$ top-$k$ neighbours
7: $R \leftarrow$ top ranked docs from $R1$ with respect to $q$
```

Algorithm 2 Adaptive Algorithm

```
1: Input: $q$ query, $n$ seed set size,$c$ exploration depth, $k$ number of neighbours
2: Output: $R1$ dense retrieval results
3: let $R0 \leftarrow$ obtain $n$ documents using the query input into an initial model
4: let $R \leftarrow$ second stage scoring of seed documents $R0$
5: let $N \leftarrow$ top ranked $c$ from $R$, $c$ neighbours
6: let $N \leftarrow$ skip documents in R that have been seen
7: while $N \neq 0$ do
8:     let $R \leftarrow$ Score and add neighbours
9:     let $N \leftarrow$ top-$c$ from $R$, $c$ neighbours
10: end while
```

(1) Obtain n documents using the query input into a BM25 model as the seed.
(2) Score the seeds.
(3) Expand the document pool by unioning the seed documents with the K nearest neighbours of those documents, skipping previously seen documents.
(4) Store the final results of the expansion in R1, providing both the ranking and the expanded pool.

2.1 Research Questions

- **R1:** How does LADR compare with other approximate nearest neighbour techniques in dense retrieval w.r.t performance and computational load?
- **R2:** What should be considered when choosing between proactive and adaptive LADR?
- **R3:** Does the nearest neighbour graph need to be constructed using an exhaustive search? Or can it be optimized by using an approximation approach like Hierachical Navicable Small World graphs?[7]

2.2 Main Contributions

LADR is a new technique to reduction computational load in dense retrieval. It is competitive with other approximate nearest neighbour techniques, its parameters allow for a system which is flexible and able to be optimized for specific use cases. State of the art for low latency approximate dense retrieval.

3 ADAPTIVE RE-RANKING WITH A CORPUS GRAPH

The most effective retrieval strategies tend to be those that first retrieve a pool of candidate documents. Cross-encoders model both the query and the document to compare the two, enhancing the quality of re-ranking. However, re-ranking requires a budget, as it typically involves using expensive ranking functions in the later stages.

GAR strategically allocates its ranking budget by focusing on neighbouring documents with the highest current scores. This process is underpinned by a pre-computed corpus graph, which serves as a directed map linking documents through a similarity function. To manage the graph's size and complexity, each document is connected by a set number of edges, denoted as K. In practice, the selection process prioritizes the most similar neighbours based on this graph structure.

GAR is versatile, easily integrating into and enhancing a variety of re-ranking pipelines. Its graph traversal technique is effective even with simple models, leveraging only lexical signals (BM25) for initial retrieval and similarity assessments, this system maintains strong performance compared to other more expensive models.

Algorithm 3 Re-ranking Algorithm

```
1: input: query, budget, corpus graph $g$
2: output: ranked list $r1$
3: let $r0 \leftarrow$ obtain $n$ documents using the query input into an initial model
4: let $b \leftarrow$ batch size
5: let $P \leftarrow R2$
6: let $R1 \leftarrow 0$
7: let $F \leftarrow 0$
8: while $R1 <$ budget do
9:     $B \leftarrow$ re-rank top $b$ documents from $P$ using second stage scoring function
10:    Add batch to $R1$ by taking the union of $R1$ and top $B$
11:    $R0 \leftarrow$ discard the batch from the initial ranking
12:    $F \leftarrow$ Discard batch from frontier
13:    $F \leftarrow$ Update frontier with neighbours of $B$ using the precalculated corpus graph $G$
14:    $P \leftarrow$ Alternate between initial ranking and the frontier
15: end while
16: $R1 \leftarrow$ backfill $R1$ with remaining items if the budget did not allow the algorithm to finish ranking
```

Alternating the previous ranked set P between the frontier allows for a balanced approach between ranking the documents from the initial ranking and their neighbours. This strategy allows for documents to be found which can be multiple graph nodes away from the origin document.

3.1 Research Questions

- How does GAR compare with other re-ranking techniques in dense retrieval w.r.t performance and computational load?
- What is the computational overhead?

- Does GAR improve other neural information retrieval systems?

3.2 Main Contributions

Provides a novel technique which uses a feedback loop to efficiently re-rank documents. This technique can improve precision and recall when added to many other re-ranking pipelines. It uses a corpus graph structure to guide the exploration process. This can allow documents to be ranked that would previously have been missed due to budget constraints. It is still an effective technique when used with a low cost similarity scoring model like BM25 as well as a range of other pipelines/ models.

4 EFFECTIVE ADHOC RETRIEVAL THROUGH TRAVERSAL OF A QUERY-DOCUMENT GRAPH

GAR[6] does not take into account the users query itself, relying on the document similarity mapping to higher match towards a query. Utilising a query document graph allows for the ranking to focus on documents that are specific to the query itself, instead of to the parent document.

This technique uses a bipartite graph of documents to queries, Query-Document Graph(QDG)[2]. This is made such that there are no document-document connections, similar documents are only connected through queries. This allows for the end-users query to be mapped more directly to the precalculated graph. The issue with this approach in general is that it requires there to be queries for each document, something which does not exist for any new corpus. In this paper, Doc2Query[3] and are used as generative models to create queries from a given document. These are then filtered on relevance to only keep the highest quality queries.

Another challenge is how it is possible to calculate the nearest documents to a document in question, when it has to be accessed indirectly through a query. We can estimate the similarity of two documents by getting the product of the relevance scores with respect to the common query. In addition, as we no longer need the original document, we can find the similarity between the hypothesised query q′ and the uses original query with the potential documents. So :

$$\text{Similarity}(d_1, d_2 \mid q_{\text{user}}) = \text{Similarity}(q_{\text{user}}, q') \times \text{Relevance}(q', d_2)$$

There are a few different approaches posited in QDGT[2].First approach is to use the same algorithm as GAR[6] , but substituting the corpus graph with the QDG, adding similar documents found with the above method and expanding the frontier in the same way.

The second approach is to use the candidate queries from a given candidate query as a neighbourhood. Then to see which candidate query has the highest relevance, rank a smaller subset of each query neighbourhood to decide which query neighbourhood should be prioritized. This way resources are spent on documents with the highest likelihood of relevance. This technique is called Reverted Adaptive Retrieval(RAR). This approach also adds potential for interpretability, by mapping the candidate queries which are picked in the frontier expansion phase.

Resource selection, takes the previous approach further, the subset of the documents which have been ranked around a candidate query are averaged. Then these neighbourhoods are added to the new frontier as prioritized neighbourhoods. This way neighbourhoods with the highest likelihood of relevance are picked to be fully ranked in the next batch of ranking, as well as reducing the chance of wasting resources on documents that aren't relevant.

4.1 Research Questions

- Do the new techniques stated allow for more relevant documents to be found?
- Do these techniques improve performance with a lower re-ranking budget?

4.2 Main Contributions

Present a new method to rank relevancy of documents based on the inverse of the clustering hypothesis, that "documents relevant to the same query are similar"[2]. The QDG can be used by multiple re-ranking pipelines and show an increase in the nDCG@10 metric compared to others in the same class. All three approaches stated, are competitive against both re-ranking pipeline baselines as well as against full dense retrieval baselines.

5 RELATIONS BETWEEN THE PAPERS

All three papers aim to improve retrieval performance without sacrificing computational efficiency. LADR and GAR share a strategy of using initial rankings to inform more complex re-ranking strategies, while QDG introduces a similar concept but incorporates the query more directly into the re-ranking process.

Both GAR and QDG utilize graph-based approaches, albeit in different forms (corpus graph vs. query-document graph), highlighting a trend towards exploiting relational data for information retrieval. These approaches suggest that graph-based methods are becoming crucial in navigating the complexities of modern datasets.

There is potential for integrating the methodologies from these papers. For example, the graph-based approaches of GAR and QDG could be combined with LADR's lexically-seeded retrieval to create a hybrid system that leverages both document and query relations effectively.

Each paper balances exploration (finding new, relevant documents) and exploitation (utilizing known good documents) differently. LADR begins with a traditional method and expands, GAR refines and focuses using existing scores within a graph, and QDG explores relations through queries, suggesting different strategies for managing the trade-offs between breadth and depth in search strategies.

6 RESEARCH Q1

Use the resource gathering technique as a way to chose potentially more likely neighbourhoods Can the use of the resource gathering seen in QDGT[2] be used to improve the recall of both Proactive and Adaptive LADR?

7 RESEARCH Q2

What is the optimal neighbourhood sample size for achieving the highest retrieval performance in the modified LADR model that incorporates neighbourhood-based ranking?

7.1 Experiment

It makes sense to answer both research questions in one experiment as the neighbourhood size will also be relative to all the other parameters, and so needs to be tested at the same time.

7.1.1 Objective. To test the effectiveness of a modified LADR method where a subset of ranked neighbours from each seed document is added to the document pool for ranking, aiming to enhance the efficiency and potentially the recall of the retrieval process.

To determine the most effective neighbourhood sample size that balances retrieval accuracy and computational efficiency in the modified LADR model.

7.1.2 Hypothesis. Ranking and selecting a subset of neighbours before adding them to the pool of new documents to be ranked will lead to more relevant document retrieval and therefore more budget efficient ranking thereby improving the recall of the lexically-accelerated dense retrieval process. I expect that smaller budgets will benefit more from smaller neighbourhood sizes.

7.1.3 Datasets. As described in LADR[5] I will also be using the same datasets to benchmark against.

- TREC DL 2019
- TREC DL 2020
- MS MARCO small

7.2 Procedure

Baseline Setup

Collect data on average sizes of clusters surrounding documents to ascertain reasonable values, for use as sample sizes when sampling the seed document neighbourhoods.

- Use the standard LADR system as per the original paper, as implemented on GitHub[1].
- Perform the suite of experiments as seen in the LADR paper to establish baseline metrics for recall, precision, and computational efficiency on our system.

7.3 Experiment Execution

Modified Setup

- Modify the LADR algorithm to incorporate the ranking of neighbours. Specifically, for each seed document identified in the initial retrieval:
  - For each parameter from the LADR implementation, i.e budget, cluster size, neighbourhood size.
  - Retrieve a larger set of neighbouring documents
  - Rank these documents based on a predefined criterion (e.g., relevance scores, semantic similarity).
  - Select the top-ranked subset of these neighbours and add only this subset to $R1$.
- Execute retrieval tasks using both the baseline and modified LADR systems.
- Record and compare the performance metrics (recall, precision, computational time).

[1]https://github.com/Georgetown-IR-Lab/ladr?tab=readme-ov-file

7.4 Expected Results

It is expected that this selective ranking and addition process will prioritize the neighbourhoods which are the most relevant. I would expect it to have more impact on the Proactive LADR, as I am unsure how much this would effect the adaptive LADR given that it keeps going until convergence, however given it still work off a budget, for budgets where it does not reach convergence it would still be highly relevant.

8 CONCLUSION

In conclusion, this paper critically examines several advanced retrieval strategies that leverage both graph-based and latent feature methodologies to enhance the efficiency and effectiveness of information retrieval systems. By assessing three distinct methods—Latent Approximate Document Retrieval (LADR), adaptive re-ranking with a corpus graph (GAR), and adhoc retrieval through a query-document graph (QDGT)—this study underscores the potential of integrating these techniques to optimize retrieval processes further. These methods collectively demonstrate how dynamic adjustment of document relationships and direct traversal of query-document mappings can significantly refine the relevance and precision of search results in large datasets. Future research could explore the synergistic integration of these approaches, possibly leading to more sophisticated systems capable of handling the increasing complexity of information retrieval tasks with higher computational efficiency and improved accuracy.

REFERENCES

[1] Eunseong Choi, Sunkyung Lee, Minjin Choi, Hyeseon Ko, Young-In Song, and Jongwuk Lee. 2022. SpaDE: Improving Sparse Representations using a Dual Document Encoder for First-stage Retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 272–282. https://doi.org/10.1145/3511808.3557456

[2] Sean Macdonald Craig Ounis Iadh Frayling, Erlend MacAvaney. 2024. Effective Adhoc Retrieval Through Traversal of a Query-Document Graph. In *Advances in Information Retrieval*, Nicola He Yulan Lipani Aldo McDonald Graham Macdonald Craig Ounis Iadh Goharian, Nazli Tonellotto (Ed.). Springer Nature Switzerland, Cham, 89–104.

[3] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. 2023. Doc2Query−: When Less is More. In *Advances in Information Retrieval*, Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo (Eds.). Springer Nature Switzerland, Cham, 414–422.

[4] N. Jardine and C.J. van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval* 7, 5 (1971), 217–240. https://doi.org/10.1016/0020-0271(71)90051-9

[5] Hrishikesh Kulkarni, Sean MacAvaney, Nazli Goharian, and Ophir Frieder. 2023. Lexically-Accelerated Dense Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (, Taipei, Taiwan,) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 152–162. https://doi.org/10.1145/3539618.3591715

[6] Sean MacAvaney, Nicola Tonellotto, and Craig Macdonald. 2022. Adaptive Re-Ranking with a Corpus Graph. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 1491–1500. https://doi.org/10.1145/3511808.3557231

[7] Yu A. Malkov and D. A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2020), 824–836. https://doi.org/10.1109/TPAMI.2018.2889473