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. We analyze three distinct approaches: Latent Approximate Document Retrieval (LADR), which utilizes latent features to optimize document retrieval; adaptive re-ranking with a corpus graph (GAR), which dynamically adjusts document rankings within a corpus graph to improve relevance; and adhoc retrieval through traversal of a query-document graph, 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

2 LEXICALLY-ACCELERATED DENSE RETRIEVAL RESEARCH

Lexically-Accelerated Dense Retrieval (LADR) 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] 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: proactive and adaptive.

2.1 Proactive Algorithm

(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 neighbors of those documents.
(3) Store the final results of the expansion in R1, providing both the ranking and the expanded pool.

The complexity and time constraints are bounded by n (seed set size) and k. The time complexity is O(kn), and the space complexity is O(Dn).

2.2 Adaptive Algorithm

(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 neighbors 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.

The complexity and time constraints are bounded by n (seed set size) and k. The time complexity is O(D), and the space complexity is O(Dn).

2.3 Research Questions

- **R1:**How does LADR compare with other ANN techniques in dense retrieval w.r.t performance and computational load?
- **R2:**What should be considered when choosing better 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 HNSW?

2.4 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 apporximate 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 focuses its ranking budget on the neighboring documents that have the highest current scores. The expansion for GAR is based on a pre-computed corpus graph. The corpus graph is a directed mapping between documents and the similarity function. Each document has a set K number of edges to decrease the size and complexity of the graph. When used, the neighbours are selected taking the most similar. This method is versatile and can be integrated into many re-ranking pipelines to improve their performance. Even simple models, i.e using BM25 for both the first, second ranking function as well as creation of the graph, benefit from the graph traversal itself.

GAR is versitile in that it can be added to many re-ranking pipelines, and improve their performance. The graph traversal by itself is useful even when using simple models. The whole system running on lexigraphical signals, BM25 for first retrieval and similarity still performs well against other vastly costlier models.

Algorithm

Algorithm 1 Re-ranking Algorithm

Input: Query, budget, corpus graph G
Output: Ranked list $R1$
let $b \leftarrow$ batch size
let $R0 \leftarrow$ Obtain n documents using the query input into an initial model
let $P \leftarrow R0$
let $R1 \leftarrow 0$
let $F \leftarrow 0$
while $R1 <$ budget **do**
    $B \leftarrow$ re-rank top b documents from P using second stage scoring function
    Add batch to $R1$ by taking the union of $R1$ and top B
    $R0 \leftarrow$ discard the batch from the initial ranking
    $F \leftarrow$ Discard batch from frontier
    $F \leftarrow$ Update frontier with neighbours of B using the precalculated corpus graph G
    $P \leftarrow$ Alternate between initial ranking and the frontier
end while
$R1 \leftarrow$ 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

Provide a novel technique which provides 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 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). This is made such that there are no document-document connections, similar documents are only connected through queries. This allows for the end-users to 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 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 by 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.

First approach is to use the same algorithm as GAR, 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 Question

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"[query-document]. The QDG can be used by multiple re-ranking pipelines and show and 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

Graph-Based and Advanced Retrieval Techniques Each paper explores different facets of improving retrieval effectiveness and efficiency by introducing novel structures or frameworks:

LADR explores the use of latent features and approximation techniques to improve the retrieval process.

Adaptive Re-ranking with a Corpus Graph (Gar) introduces a graph-based method to dynamically adjust rankings based on a corpus graph, enhancing the relevance of retrieved documents.

Adhoc Retrieval through Traversal of a Query-Document Graph investigates the use of a graph that includes both queries and documents to potentially improve retrieval outcomes through strategic traversal methods. Methodological Innovations

All three papers push the boundaries of conventional retrieval methods:

LADR challenges traditional approximation techniques with a potentially more effective and efficient method.

Gar employs a corpus graph, which adapatively re-ranks documents based on additional contextual relationships within the corpus.

Traversal of a Query-Document Graph suggests that direct interaction between queries and documents via graph traversal could yield better retrieval results than isolated query-document evaluations.

Enhanced Retrieval Metrics Each study focuses on evaluating and improving key retrieval metrics like precision, recall, and nDCG:

LADR is assessed on its effectiveness and efficiency, likely impacting metrics such as nDCG and precision.

Gar is evaluated through its impact on retrieval effectiveness, comparing its performance with traditional re-ranking and advanced neural IR systems.

Traversal of a Query-Document Graph addresses the impact on retrieval metrics through unique graph traversal techniques, offering potential improvements in precision and recall.

Potential for Future Research Integration The insights and methods from each paper could be integrated into a comprehensive retrieval system:

Techniques from LADR could be combined with the graph-based approaches of Gar and Traversal of a Query-Document Graph to develop a highly sophisticated IR system that uses both latent features and graph structures. The adaptive re-ranking methods of Gar could benefit from the dynamic traversal methods explored in the Traversal of a Query-Document Graph, potentially enhancing the adaptiveness and contextual awareness of the re-ranking process.

6 RESEARCH Q1

New training direction for dense models using intra-document similarity?

Try Gar process with LADR idea?

Try different strategy for alternating

6.1 Experiment

6.2 Hypothesis

7 RESEARCH Q2

7.1 Experiment

7.2 Hypothesis

8 CONCLUSION

These papers contribute to the evolving landscape of information retrieval by demonstrating how advanced data structures like graphs and innovative algorithms can be leveraged to solve complex retrieval challenges. By exploring different aspects of IR enhancements—whether through latent features, adaptive re-ranking, or traversal mechanisms—they collectively push the field toward more nuanced and effective retrieval solutions, offering a foundation for future research that could integrate these varied approaches into a unified framework.

REFERENCES