Optimization in Information Retrieval: A Quick View of Techniques for Performance and Scalability

André Tomitan Bocces[1]
Alexandro Baldassin[1]
Allberson Dantas

[1]"Instituto de Geociências e Ciências Exatas" - IGCE,
Universidade Estadual Paulista "Júlio de Mesquita Filho" - UNESP,
Av. 24, s/n, Bela Vista, Rio Claro, SP, 13506-900, Brazil.

at.bocces@unesp.br

alexandro.baldassin@unesp.br

allberson@unilab.edu.br

Abstract. *Information Retrieval (IR) systems often must manage large-scale datasets while requiring efficient response times. This paper presents a summary of optimization techniques aimed at improving performance and scalability in IR algorithms. We explore methods such as parallel processing, memory hierarchy optimizations, and GPU acceleration. Our analysis identifies key factors influencing the effectiveness of these optimizations, including workload adaptability and hardware constraints. We also discuss the challenges of implementing these techniques and their impact on modern IR pipelines.*

1. Introduction

Information Retrieval has emerged as a powerful approach in modern search [Buttcher et al. 2016], capturing complex semantic relationships between queries and documents by representing data as high-dimensional vectors, referred to as embeddings. Considerable computational challenges arise as data volumes grow [Shaikh 2017], particularly in terms of scalability and efficiency. The high-dimensional nature of embeddings, coupled with the necessity for fast nearest-neighbor searches [Wei et al. 2023], leads to significant demands on memory, processing power, and storage bandwidth. Therefore, optimizing these systems becomes crucial to maintaining real-time performance and ensuring cost-effective deployment.

This summarized study is based on a systematic review that followed the protocol outlined by Carrera-Rivera et al. [Carrera-Rivera et al. 2022], covering 20 publications from 2018 to 2024. The selection process follows established methodologies for systematic reviews, ensuring the inclusion of relevant and high-impact research. The primary objective of this review is to identify and discuss prominent optimization strategies that enhance the efficiency, scalability, and accuracy of IR systems.

The reviewed studies focus on a diverse range of approaches, including parallel computing, GPU acceleration, indexing optimizations, and memory-efficient embedding representations. By consolidating findings from multiple sources, this work aims to provide a view of state-of-the-art techniques scoped to various IR scenarios, including

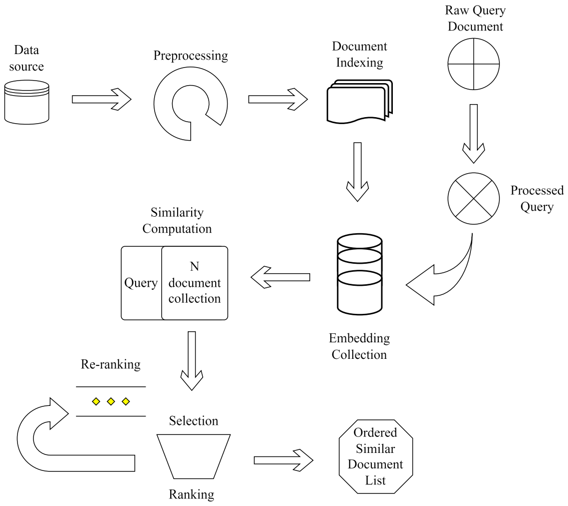

Figure 1. A generic Information Retrieval Task Pipeline.

textual, image, and multimodal retrieval systems. Understanding these optimization techniques is critical for both academics and industry professionals working on large-scale information retrieval applications.

2. Background

Figure 1 shows a generic Information Retrieval pipeline, comprising of several essential stages, each tailored to process diverse data types efficiently. The process begins with data ingestion, which accommodates structured databases, unstructured text, multimedia, and sensor-generated inputs. Preprocessing standardizes raw data into suitable formats through techniques such as tokenization for text, feature extraction for images, and spectral transformations for audio and video. The indexing stage structures this processed data to enable fast retrieval, employing inverted indexes for text, feature-based structures for multimedia, and hash maps or B-Trees for structured data. Next, embedding collection transforms data into vector representations that encapsulate semantic or structural attributes, facilitating similarity-based retrieval. Queries undergo vectorized transformation to ensure compatibility with indexed data—e.g., converting images into feature vectors via deep learning models. The system then computes similarity scores between query and document embeddings using metrics like cosine similarity or Euclidean distance, determining the most relevant matches. These results are ranked, with re-ranking techniques refining the final order to improve retrieval accuracy. The system then presents an ordered list of relevant documents, delivering efficient retrieval aligned with user queries.

3. Optimization Techniques in Information Retrieval

This section will go over techniques in parallel computing, GPU acceleration, and memory hierarchy management that have enabled IR systems to process vast datasets with improved efficiency, scalability, and reduced latency.

3.1. Parallel Processing and GPU Acceleration

Parallel processing, particularly on GPUs, has become a cornerstone in optimizing IR algorithms, offering substantial improvements in both speed and scalability. Techniques like parallel Top-K algorithms and intra-query parallelism [Zhang et al. 2023] have demonstrated remarkable reductions in query processing times by distributing computational workloads across many GPU cores. This level of parallelism is particularly effective for compute-heavy tasks such as similarity search, ranking, and clustering large datasets.

As for extracting the best of both CPU and GPU utilization, Griffin's dynamic query partitioning strategy [Liu et al. 2018] effectively mitigates processing bottlenecks. Building upon this, intelligent load-balancing mechanisms can partition workloads while predicting computational demand fluctuations, ensuring optimal distribution between CPU (efficient in branch-intensive tasks) and GPU (SIMD-efficient) for sustained gains. Furthermore, hybrid parallelism frameworks, such as those explored by [Fazlali et al. 2024], successfully integrate OpenMP and CUDA to enhance workload distribution. An adaptive parallel execution model further improves upon this by dynamically shifting workloads between CPU and GPU cores based on real-time performance profiling, optimizing resource utilization for computationally demanding retrieval tasks.

3.2. Memory Hierarchy Optimizations

Efficient memory management is paramount in embedding-based retrieval tasks, where large-scale embedding spaces demand optimized memory hierarchies to balance performance and scalability. Recent advancements, such as the ESPN model ([Shrestha et al. 2024]), introduce structured multi-vector storage strategies that reduce redundancy and enhance retrieval efficiency, significantly lowering query latency in large-scale search applications.

Memory and cache optimization techniques have also proven effective in reducing computational overhead. Cache-efficient retrieval algorithms ([Ni 2023]) dynamically adjust caching policies based on workload patterns, minimizing memory latency and improving query response times. Additionally, dimensionality reduction approaches, such as PCA and autoencoders ([Hofstätter et al. 2019]), reduce storage requirements while preserving retrieval accuracy, making them highly valuable in scenarios with constrained memory resources. Employing a layered memory architecture that combines cache-aware retrieval methods and optimized storage models ([Wang et al. 2023]), IR systems can efficiently scale to accommodate larger datasets without compromising speed. This results in improved scalability and computational efficiency, essential for high-performance information retrieval in large-scale applications.

4. Summary of Optimization Techniques

Table 1. IR techniques and best-use scenarios

Technique	**Best Use Case**
Parallelism	Large datasets with high compute (e.g., Top-K, similarity)
Hybrid CPU-GPU	Mixed workloads in heterogeneous systems
Structured Storage	Embedding-based retrieval with low redundancy
Cache-Awareness	Frequent or bursty queries with latency sensitivity
Dimensionality Reduction	Memory-constrained retrieval with accuracy retention
Memory Hierarchy	Scaling to massive datasets efficiently

Table 1 summarizes the optimization strategies explored in this work, highlighting their most suitable application contexts. This concise view facilitates quick identification of methods best aligned with system constraints and performance goals.

5. Conclusion

The landscape of IR optimizations is evolving rapidly, and IR has benefited significantly from optimizations in parallel processing, GPU acceleration, and memory hierarchy management. Future work should explore more adaptive and intelligent optimization strategies involving hardware-specific enhancements, for instance, expanding the memory hierarchy by including additional layers like Persistent Memory (PM). By continuously refining optimization techniques, IR systems can achieve higher efficiency and scalability, ensuring they meet the demands of modern large-scale search applications.

References

Buttcher, S., Clarke, C. L., and Cormack, G. V. (2016). *Information retrieval: Implementing and evaluating search engines*. Mit Press.

Carrera-Rivera, A., Ochoa, W., Larrinaga, F., and Lasa, G. (2022). How-to conduct a systematic literature review: A quick guide for computer science research. *MethodsX*, 9:101895.

Fazlali, M., Mirhosseini, M., Shahsavari, M., Shafarenko, A., and Mashinchi, M. (2024). GPU-based Parallel Technique for Solving the N-Similarity Problem in Textual Data Mining. In *DCHPC 2024*, pages 1–6.

Hofstätter, S., Rekabsaz, N., Eickhoff, C., and Hanbury, A. (2019). On the effect of low-frequency terms on neural-IR models. In *(SIGIR)*, pages 1137 – 1140. Association for Computing Machinery, Inc.

Liu, Y., Wang, J., and Swanson, S. (2018). Griffin: Uniting CPU and GPU in Information Retrieval Systems for Intra-Query Parallelism. *ACM SIGPLAN Notices*, 53(1.0):327 – 337.

Ni, C. (2023). Top-k query optimization on the hierarchical memory structure. In *(AUTEEE)*, pages 1075–1080.

Shaikh, T. (2017). 5. comparing performance of various optimization algorithms for effective information retrieval – a review. *International Journal for Research in Applied Science and Engineering Technology*.

Shrestha, S., Reddy, N., and Li, Z. (2024). ESPN: Memory-Efficient Multi-vector Information Retrieval. In *(ISMM)*, pages 95 – 107. Association for Computing Machinery.

Wang, D., Liu, L., and Liu, Y. (2023). Normalized Storage Model Construction and Query Optimization of Book Multi-Source Heterogeneous Massive Data. *IEEE Access*, 11:96543–96553.

Wei, C., Qingbo, L., Na, D. E., and Congli, C. (2023). A Novel Redundant Data Retrieval Model based on Parallel Batch Algorithm. In *(ICICT)*, pages 518–522.

Zhang, J., Naruse, A., Li, X., and Wang, Y. (2023). Parallel Top-K Algorithms on GPU: A Comprehensive Study and New Methods. In *(SC23)*, pages 1–13.