Practical, Efficient, In-Memory Inverted Indexes

Joel Mackenzie[1][0000−0001−7992−4633], Sean MacAvaney[2][0000−0002−8914−2659], Antonio Mallia[3][0000−0002−7817−6140], and Michał Siedlaczek[4][0000−0002−9168−0851]

[1] The University of Queensland, St Lucia, Australia
[2] University of Glasgow, Glasgow, UK
[3] Seltz, San Francisco, US
[4] MongoDB, New York, US

Abstract. Inverted indexes are the backbone of most large-scale information retrieval systems. Although conceptually simple, high-performance inverted indexes require a deep understanding of low-level system optimizations, memory layouts, data compression techniques, and index traversal strategies. With the widespread adoption of in-memory search engines, the rise of learned sparse retrieval (LSR), and the increasing complexity of ranking pipelines, the design space for efficient indexing and retrieval systems has expanded significantly. This tutorial addresses a critical knowledge gap between textbook-style explanations and advanced techniques required for efficient and optimized retrieval. It aims to equip researchers and practitioners with a comprehensive understanding of how modern in-memory search systems are designed, built, and optimized for high-performance retrieval across large-scale document collections.

Keywords: Efficiency, Inverted Index, Query Processing, Compression

1 Motivation

Efficient indexing and retrieval remain foundational to information retrieval (IR) systems at all scales, from academic testbeds to large-scale commercial search engines. At the heart of these systems lies the inverted index, a decades-old data structure that continues to evolve in response to new retrieval paradigms and performance demands. While many students and early-career researchers are introduced to inverted indexes through textbooks or simplified implementations, the gap between these resources and the requirements of high-performance, in-memory search engines remains substantial. Furthermore, the complexity of these highly optimized solutions presents a significant barrier to entry for those interested in understanding and working in the efficiency space.

Recent advances in retrieval – particularly the rise of learned sparse retrieval (LSR) models [1, 4, 9, 14, 19, 26, 36], hybrid search pipelines [5, 7, 11, 15], and retrieval-augmented generation (RAG) applications [10] – have renewed interest in efficient sparse retrieval techniques that can be tightly integrated with modern machine learning and cascade ranking workflows. In particular, the

use of traditional sparse indexes to serve model-generated representations has led to new challenges in indexing, scoring, and top-k retrieval, motivating the development of specialized pruning strategies such as BM25-guided traversal [8, 20, 28], Block-max Pruning [25], and list decomposition [16] among many others.

Despite these developments, many researchers and practitioners face a steep learning curve when experimenting with efficient retrieval infrastructures, making it difficult to reproduce results, run scalable experiments, or explore new optimization techniques. To address this gap, this tutorial provides both theoretical foundations and practical guidance for building high-performance sparse retrieval systems. Using the open-source *Performant Indexes and Search for Academia* (PISA) engine [22] and its Python bindings via PyTerrier [13], we demonstrate how classical indexing techniques are implemented in practice and how they can be extended to emerging applications such as LSR and RAG.

The tutorial is particularly relevant to the ECIR community, where interest in efficient retrieval, reproducible experimentation, and integration with neural models has grown rapidly in recent years. By equipping attendees with both the conceptual background and the practical skills required to build and experiment with state-of-the-art sparse retrieval systems, this tutorial aims to lower the barrier to entry and foster new research directions at the intersection of classical IR and modern machine learning. The tutorial will also improve the visibility of ongoing work and open directions for inverted index-based search systems.

2 Objectives

We organize our tutorial around a set of Intended Learning Outcomes (ILOs) that attendees will be able to achieve by the end of the tutorial. These outcomes are designed to accommodate a broad audience – from those new to information retrieval, to more experienced researchers – by combining foundational theory with hands-on practical skills via the PISA engine. Furthermore, we aim to provide valuable content to both *end-users* (those who want to effectively apply inverted indexes in their research) and *tinkerers* (those who want to dig into the implementations and contribute to new research in this domain). The ILOs for the tutorial are described below.

ILO1: Theoretical Understanding of Inverted Indexes. Attendees will develop a solid understanding of how inverted indexes are structured and used in IR systems. They will learn how core components – such as posting lists, lexicons, and skip lists – are implemented and organized in memory, how they are compressed, and how these design choices impact retrieval scalability.

ILO2: Fast Top-k Retrieval with Dynamic Pruning. Attendees will gain insight into *dynamic pruning-based* retrieval techniques (such as MaxScore [33] and BMW [3, 6, 21]) that accelerate top-k search. They will understand the principles behind these algorithms, including any assumptions, sensitivities to different indexing or parameter choices, and the underlying trade-offs involved with different pruning strategies. Attendees will also gain insights

into the current state-of-the-art methods for accelerating inverted index-based algorithms, including threshold estimation [17, 24, 27], advanced skipping or pruning methods [23, 29, 30, 34], and anytime retrieval strategies [18].

ILO3: Current Trends and Research Directions. Attendees will understand how traditional inverted index-based retrieval fits into modern IR systems. They will learn how emerging methods – such as learned sparse retrieval – influence traversal strategies, including problems caused by learned sparse data distributions. Then, we will introduce ongoing research to remedy these issues including BM25-guided traversal, list decomposition, and *anytime* retrieval, as well as emerging methods tailored to sparse neural representations such as Block-Max Pruning (BMP) [25] and Seismic [2].

ILO4: Building and Running Experiments with PISA. Attendees will gain practical experience using the PISA engine to build indexes, run retrieval experiments, and evaluate performance. They'll learn how to apply different scoring functions and retrieval algorithms to analyze efficiency and effectiveness trade-offs. We plan to offer two levels of access; (1) the PyTerrier-PISA python bindings for users who wish to interact with PISA via the PyTerrier interface; and (2) directly via PISA's command line tool interface.

ILO5: Integration of PISA in Modern Applications. Attendees will be able to use PISA as a backend in applied research contexts, such as incorporating keyword search into retrieval-augmented generation (RAG) pipelines.

3 Relevance

This tutorial is designed for both newcomers to information retrieval and experienced researchers who want to better understand efficient retrieval methods. Although there has been a significant focus on semantic search methods over the last decade, lexical retrieval is still a core component of many search systems at various scales, and the efficient deployment of lexical search systems is a key requirement for the community. Our tutorial complements recent tutorials [31, 35] and surveys [32] by focusing specifically on the efficient design and implementation of such retrieval systems, from traditional models and simple indexes to state-of-the-art techniques. To ensure relevance with modern IR trends and ongoing work, we specifically focus on how traditional inverted index methods can be integrated into modern applications, and dedicate a section of the tutorial to ongoing work and emerging trends.

4 Format and Schedule

The tutorial will be presented as a series of modules, interleaving the theoretical aspects with practical, hands-on activities. We also intend on providing a more advanced "post tutorial" extension task for those who wish to continue to develop their knowledge, but our main goal is to reduce the barrier to entry for attendees.

To this end, no "pre-work" is required, and we expect to have both runnable notebooks available to support the python-oriented sections, as well as a Docker image with PISA pre-installed for the command line work. Thus, participants are only expected to have Docker installed and a functioning Python environment. This will facilitate rapid set-up, and these resources will be made available before the workshop with support available from the team prior to, during, and after the tutorial. Note that all timelines listed below are indicative, and are intended to have some level of flexibility to adapt to the audience needs or interests.

Session 1: Indexing and Retrieval [75 min.]. The first session will be focused on exploring the fundamentals of the inverted index. We will begin with a discussion on where the PISA family (including Python bindings) fits into the wider tool landscape, and compare it against other common tools that serve a similar purpose. We will then motivate the need for inverted indexes, followed by a series of brief, visual tutorials on in-memory indexing. These will include details on the representation of inverted indexes, and practical examples of indexing common IR collections. Next, we will move from indexing to retrieval, again using visual tutorials to outline how exhaustive retrieval operates over inverted indexes. Following this, we will move to efficient dynamic pruning algorithms (using the more simple WAND or MaxScore algorithms as exemplars). We will complete this session with a small empirical comparison of top-k retrieval algorithms using PISA on the indexed collection. An indicative timeline is as follows:

- Introduction and motivation [5-10 minutes]
- Inverted indexing and retrieval fundamentals [20 minutes]
- Fast and compact in-memory index representations [20 minutes]
- From exhaustive to fast top-k retrieval [20 minutes]

Practical Aspects. The first session will aim at familiarizing attendees with the notebook/Docker image, and to run some basic indexing and querying. We expect this part to rely on a small and freely available collection such as a subset of an open-source Wikipedia dump. We will start with building a basic index, and running simple conjunctive and disjunctive queries. Then, we will build additional structures required to support (fast) ranking, and explore exhaustive document-at-a-time top-k retrieval. Finally, we will move to deploying dynamic pruning algorithms like WAND and MaxScore.

Break [15 minutes]. Aligned with the conference coffee break.

Session II: Learned Sparse Retrieval [60 min.]. The second session will focus on the challenges and opportunities of combining traditional inverted index structures with modern learned sparse retrieval (LSR) techniques. We begin with a high-level overview of LSR, covering their motivation, model architectures, and how they differ from statistical rankers like BM25. We then discuss how LSR changes the structure and usage patterns of inverted indexes – including new characteristics of posting lists, term distributions, and score distributions that arise from model-generated representations. Next, we explore how traditional indexing and retrieval algorithms can be adapted to efficiently support LSR,

with a particular focus on pruning strategies that help mitigate long and noisy LSR postings. An indicative timeline is as follows:

- From Statistical to Learned Models [10 minutes]
- LSR: A trouble-maker for Inverted Indexes [20 minutes]
- Accelerating Inverted Indexes in the context of LSR [30 minutes]

Practical Aspects. This session will include a practical walk-through, showing how PISA can be used to index LSR-generated output (e.g., in JSON or CIFF [12]) format and how these indexes can be queried using the same high-performance retrieval infrastructure. These experiments will also demonstrate the slowdowns caused by LSR in practice. The session concludes with a demonstration of how PISA can be integrated into Python-based research pipelines via PyTerrier, enabling its use as a first-stage retriever in modern workflows, including hybrid retrieval and, if time permits, retrieval-augmented generation (RAG) systems.

Session III: Alternatives & New Directions [30 min.]. Our final session will broaden the focus of the tutorial beyond document-ordered inverted indexes with the intent of providing useful pointers to alternative approaches being investigated by the efficiency community. We will also outline some open problems and ongoing work in the efficiency space.

5 Materials

Attendees will be provided with: (1) Slides covering theoretical content and code walk-throughs, including diagrams or animations to improve understanding; (2) Access to a Docker image for PISA, and notebooks (Jupyter or Colab) for the PyTerrier interface; (3) Sample datasets and prebuilt indexes for experimentation; and (4) Links to relevant publications, code repositories, and further readings.

We intend to provide some post-tutorial resources and guides for those attendees interested in learning or engaging further. For example, we plan to provide links to the PISA repository (and the PyTerrier counterpart), detailed documentation, the PISA Slack channel, and a guide on how to contribute further to the project. We will also provide a set of references to recent and ongoing work, and any further material we develop that was not covered during the tutorial.

6 Presenters

The presenters have rich experience with the long history and practical aspects of inverted index-based retrieval, including the internal data structures, algorithms, and emerging topics such as learned sparse ranking models. This is the *second offering* of the tutorial – it was also offered at SIGIR 2025 as a half day tutorial.

Joel Mackenzie is a senior lecturer (Assistant Professor) at the University of Queensland in Brisbane, Australia. His research is focused on efficient

and effective representations for large-scale search engines, including indexing, compression, and retrieval. He is a maintainer of the state-of-art PISA search system, and teaches undergraduate- and graduate-level *Algorithms & Data Structures* courses at UQ. He was also a co-organizer of the SIGIR ReNeuIR workshop in 2023 and 2024.

Sean MacAvaney is a lecturer (Assistant Professor) at the University of Glasgow. He wrote the Python bindings for the PISA engine and maintains shared software packages such as PyTerrier and IR-Datasets. His research focuses on efficiently leveraging neural language models for information retrieval. He teaches the graduate-level *Text as Data* course at Glasgow and has presented at the European Summer School on Information Retrieval (ESSIR).

Antonio Mallia is an Independent reasearcher. Previously, he was a Staff Research Scientist at Pinecone, leading research on scalable retrieval systems and models; before that, he was an Applied Scientist on Amazon's Artificial General Intelligence (AGI) team, working on large-scale web search. He holds a Ph.D. in Computer Science from New York University, where he studied under Professor Torsten Suel, focusing on enhancing the effectiveness and efficiency of Information Retrieval systems. He is a founder of the PISA search engine.

Michał Siedlaczek is a Senior Software Engineer at MongoDB, working on the scalability, availability, and reliability of a search engine implementation. Previously, he worked at IBM in a team responsible for delivering a search experience for internal use. He earned his Ph.D. in Computer Science at New York University, researching search engine performance. He is one of the maintainers of PISA.

7 Conclusion

Efficient indexing and retrieval remain central challenges in the design of high-performance information retrieval systems. This tutorial bridges the gap between foundational concepts and cutting-edge developments by offering both theoretical insights and hands-on experience with state-of-the-art tools. It focuses on in-memory inverted indexes and their role in both classical and modern retrieval pipelines — including learned sparse retrieval and retrieval-augmented generation. This tutorial is designed to support both newcomers to information retrieval and experienced researchers interested in efficiency-focused methods. By the end of the session, attendees will have a strong understanding of core indexing and retrieval techniques, along with practical experience that equips them to apply these approaches in their own research and experimental workflows.

Disclosure of Interests. The authors have no competing interests of any sort.

References

[1] Basnet, S., Gou, J., Mallia, A., Suel, T.: Deeperimpact: Optimizing sparse learned index structures. In: Proc. ReNeuIR at SIGIR (2024)

[2] Bruch, S., Nardini, F.M., Rulli, C., Venturini, R.: Efficient inverted indexes for approximate retrieval over learned sparse representations. In: Proc. SIGIR, pp. 152–162 (2024)

[3] Ding, S., Suel, T.: Faster top-k document retrieval using block-max indexes. In: Proc. SIGIR, pp. 993–1002 (2011)

[4] Formal, T., Piwowarski, B., Clinchant, S.: SPLADE: sparse lexical and expansion model for first stage ranking. In: Proc. SIGIR, pp. 2288–2292, ACM (2021)

[5] Gao, L., Dai, Z., Callan, J.: COIL: revisit exact lexical match in information retrieval with contextualized inverted list. In: Proc. NAACL-HLT, pp. 3030–3042 (2021)

[6] Grand, A., Muir, R., Ferenczi, J., Lin, J.: From MaxScore to Block-Max Wand: The story of how Lucene significantly improved query evaluation performance. In: Proc. ECIR, pp. 20–27 (2020)

[7] Huang, K., Venkatesh, T., Dingankar, U., Mallia, A., Campos, D., Jiao, J., Potts, C., Zaharia, M., Boahen, K., Khattab, O., Sarup, S., Santhanam, K.: ColBERT-Serve: efficient multi-stage memory-mapped scoring. In: Proc. ECIR, pp. 21–30 (2025)

[8] Kulkarni, H., MacAvaney, S., Goharian, N., Frieder, O.: Lexically-accelerated dense retrieval. In: Proc. SIGIR, pp. 152–162 (2023)

[9] Lassance, C., Clinchant, S.: An efficiency study for splade models. In: Proc. SIGIR, pp. 2220–2226 (2022)

[10] Lewis, P.S.H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proc. NeurIPS (2020)

[11] Lin, J., Ma, X., Lin, S., Yang, J., Pradeep, R., Nogueira, R.: Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proc. SIGIR, pp. 2356–2362 (2021)

[12] Lin, J., Mackenzie, J., Kamphuis, C., Macdonald, C., Mallia, A., Siedlaczek, M., Trotman, A., de Vries, A.: Supporting interoperability between open-source search engines with the common index file format. In: Proc. SIGIR, pp. 2149–2152 (2020)

[13] MacAvaney, S., Macdonald, C.: A python interface to pisa! In: Proc. SIGIR, pp. 3339–3344 (2022)

[14] MacAvaney, S., Nardini, F.M., Perego, R., Tonellotto, N., Goharian, N., Frieder, O.: Expansion via prediction of importance with contextualization. In: Proc. SIGIR, pp. 1573–1576 (2020)

[15] Macdonald, C., Tonellotto, N., MacAvaney, S., Ounis, I.: Pyterrier: Declarative experimentation in python from BM25 to dense retrieval. In: Proc. CIKM, pp. 4526–4533 (2021)

[16] Mackenzie, J., Mallia, A., Moffat, A., Petri, M.: Accelerating learned sparse indexes via term impact decomposition. In: Findings of the ACL: EMNLP 2022, pp. 2830–2842 (2022)
[17] Mackenzie, J., Moffat, A.: Examining the additivity of top-k query processing innovations. In: Proc. CIKM, pp. 1085–1094 (2020)
[18] Mackenzie, J., Petri, M., Moffat, A.: Anytime ranking on document-ordered indexes. ACM Trans. Inf. Syst. **40**(1), 1–32 (2021)
[19] Mallia, A., Khattab, O., Suel, T., Tonellotto, N.: Learning passage impacts for inverted indexes. In: Proc. SIGIR, pp. 1723–1727 (2021)
[20] Mallia, A., Mackenzie, J., Suel, T., Tonellotto, N.: Faster learned sparse retrieval with guided traversal. In: Proc. SIGIR, pp. 1901–1905 (2022)
[21] Mallia, A., Ottaviano, G., Porciani, E., Tonellotto, N., Venturini, R.: Faster blockmax WAND with variable-sized blocks. In: Proc. SIGIR, pp. 625–634 (2017)
[22] Mallia, A., Siedlaczek, M., Mackenzie, J.M., Suel, T.: PISA: Performant indexes and search for academia. In: Proc. OSIRRC at SIGIR, vol. 2409, pp. 50–56 (2019)
[23] Mallia, A., Siedlaczek, M., Suel, T.: Fast disjunctive candidate generation using live block filtering. In: Proc. WSDM, pp. 671–679 (2021)
[24] Mallia, A., Siedlaczek, M., Sun, M., Suel, T.: A comparison of top-k threshold estimation techniques for disjunctive query processing. In: Proc. CIKM, pp. 2141–2144 (2020)
[25] Mallia, A., Suel, T., Tonellotto, N.: Faster learned sparse retrieval with block-max pruning. In: Proc. SIGIR, pp. 2411–2415 (2024)
[26] Nguyen, T., MacAvaney, S., Yates, A.: A unified framework for learned sparse retrieval. In: Proc. ECIR, vol. 13982, pp. 101–116 (2023)
[27] Qiao, Y., Carlson, P., He, S., Yang, Y., Yang, T.: Threshold-driven pruning with segmented maximum term weights for approximate cluster-based sparse retrieval. In: Proc. EMNLP, pp. 19742–19757 (2024)
[28] Qiao, Y., Yang, Y., Lin, H., Yang, T.: Optimizing guided traversal for fast learned sparse retrieval. In: Proc. WWW, pp. 3375–3385 (2023)
[29] Siedlaczek, M., Mallia, A., Suel, T.: Using conjunctions for faster disjunctive top-k queries. In: Proc. WSDM, pp. 917–927 (2022)
[30] Tolosa, G., Mallia, A.: Many are better than one: Algorithm selection for faster top-k retrieval. Inf. Proc. & Man. **60**(4) (2023)
[31] Tonellotto, N., Macdonald, C.: Efficient query processing infrastructures: A half-day tutorial at sigir 2018. In: Proc. SIGIR, p. 1403–1406 (2018)
[32] Tonellotto, N., Macdonald, C., Ounis, I.: Efficient query processing for scalable web search. Found. Trends Inf. Retr. **12**(4-5), 319–500 (2018)
[33] Turtle, H.R., Flood, J.: Query evaluation: Strategies and optimizations. Inf. Proc. & Man. **31**(6), 831–850 (1995)
[34] Yafay, E., Altingovde, I.S.: Faster dynamic pruning via reordering of documents in inverted indexes. In: Proc. SIGIR, pp. 2001–2005 (2023)
[35] Yates, A., Lassance, C., MacAvaney, S., Nguyen, T., Lei, Y.: Neural lexical search with learned sparse retrieval. In: Proc. SIGIR-AP, pp. 303–306 (2024)
[36] Yu, P., Mallia, A., Petri, M.: Improved learned sparse retrieval with corpus-specific vocabularies. In: Proc. ECIR, pp. 181–194 (2024)