

Efficient RAG for MediaWiki and Wikibase: Hyperparameter Tuning for Semantic Retrieval under Resource Constraints

Kolja Bailly^{1,*}, Lozana Rossenova¹ and Ina Blümel¹

¹TIB Leibniz Information Centre for Science and Technology - Open Science Lab, Welfengarten 1B, Hannover, 30167, Germany

Abstract

Retrieval-Augmented Generation (RAG) improves the factuality and domain adaptation of large language models, but optimizing their configuration requires computationally expensive evaluations, manually curated datasets, and machine learning expertise. These requirements are challenging for scholarly knowledge infrastructures and resource-constrained research environments.

We present an automated, resource-efficient workflow for optimizing semantic retrieval in MediaWiki and Wikibase-based RAG systems. The approach uses hyperparameter optimization guided by Mean Reciprocal Rank (MRR) and automatically generates question-context-answer (QCA) triplets using the lightweight IBM Granite 4 350M Nano language model. Persona-aware question generation enables the creation of retrieval benchmarks tailored to different user groups and information needs.

The workflow runs entirely on commodity hardware without manually curated evaluation datasets. Across three MediaWiki-based datasets, optimized configurations achieved MRR improvements ranging from 9% to 52%, with an average improvement of 30%. Results indicate that small on-device language models can serve as effective proxies for automated retrieval optimization, reducing computational costs while maintaining retrieval quality.

The implementation is integrated into the open-source Wikibase4Research ecosystem to support accessible, reproducible, and sustainable RAG deployment in scholarly knowledge infrastructures.

Keywords

RAG, Hyperparameter Optimization, Sustainable AI, MediaWiki, Retrieval, Small Language Models

1. Introduction

While Retrieval-Augmented Generation (RAG) systems provide an effective mechanism for combining semantic retrieval with large language models, the quality of generated responses strongly depends on the configuration of the retrieval pipeline. Important factors include chunking strategies, embedding models, retrieval methods, ranking algorithms, and context selection parameters. Since the optimal configuration depends heavily on the characteristics of the underlying knowledge base and the intended usage scenario, there is no universally applicable default configuration [1].

Traditionally, optimizing these parameters requires extensive experimentation and domain expertise and is therefore commonly performed by data scientists or machine learning engineers. This process, commonly referred to as hyperparameter tuning or hyperparameter optimization, is often computationally expensive and technically demanding due to the large number of possible parameter combinations and evaluation iterations [2].

To address these challenges, we build an automated retrieval optimization workflow into the RAG Pipeline of Wikibase4Research, a docker-based framework for easy setup of custom MediaWiki/Wikibase environments [3]. MediaWiki is the multilingual and collaborative open source framework behind Wikipedia. Wikibase is the MediaWiki extension for the management of structured, linked open data (e.g. in the public Wikidata instance). Self-hosted instances, such as the ones Wikibase4Research facilitates, are widely used by institutions, organizations and communities for a variety of data management use

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

*Corresponding author.

✉ kolja.bailly@tib.eu (K. Bailly); lozana.rossenova@tib.eu (L. Rossenova); ina.bluemel@tib.eu (I. Blümel)

🆔 0009-0001-3533-0275 (K. Bailly); 0000-0002-5190-1867 (L. Rossenova); 0000-0002-3075-7640 (I. Blümel)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

cases. [4]. The usecase is, to provide a RAG tool within the MediaWiki Framework, that helps users to find relevant information within the wiki pages, based on natural language queries.

2. Methodology

2.1. Requirements and Constraints

The requirements for retrieval optimization are derived from the target users of Wikibase4Research and the constraints of publicly funded research infrastructures. Researchers typically work with domain-specific data but often lack expertise in machine learning engineering. Therefore, optimization workflows should require minimal configuration and no advanced technical knowledge.

Because manually curated evaluation datasets are rarely available, the system must automatically generate question-context-answer (QCA) triplets for retrieval benchmarking and hyperparameter optimization. In addition, retrieval quality depends on the intended user group and query characteristics. [5, 6] To account for different information needs, the workflow supports persona-aware evaluation through configurable user personas.

Finally, the optimization process should run efficiently on commodity hardware to ensure accessibility, reproducibility, and resource-efficient AI usage.

Based on the conditions above, the following requirements were identified for our approach:

- Adaptation to the characteristics of the underlying knowledge base data
- Fully automated retrieval optimization
- No requirement for advanced technical expertise
- Automatic generation of evaluation datasets
- Support for persona-aware retrieval strategies
- Efficient execution with minimal computational resources

2.2. Datasets

The evaluation was conducted using subset sampling evaluation[7] from three MediaWiki-based knowledge collections: 'Wikipedia', 'Manor Houses of the Baltic Sea' and 'Kung Fu Wiki'.

For each dataset, 50 random articles were selected as a pseudo ground truth [7] for retrieval optimization and evaluation. To assess retrieval quality, question-context-answer triplets were generated automatically using the multilingual language model IBM Granite 4 350M Nano. The model was specifically designed for low-resource and on-device environments, making it suitable for sustainable and computationally efficient evaluation workflows. To ensure dataset quality even when using a small language model, we used a two-step process, first generating Question-Context pairs and afterwards extending it to Question-Context-Answer triples motivated by Gao et al. [8]. However, for comparing retrieval performance of different parameter configurations, the quality of the generated answers is not relevant, as only the retrieval quality is evaluated. For the datasources 'Manor Houses' and 'Kung Fu Wiki', the retrieved wiki pages contained knowledge of a specific domain, while the Wikipedia pages contained general knowledge of different domains.

2.3. Retrieval Pipeline Parameters

In a RAG pipeline, the underlying knowledge base is indexed in a vector database to enable efficient semantic retrieval of relevant content. Most parameters optimized in our approach affect the information extraction and indexing process, determining how textual information is segmented, enriched, and represented before retrieval. For vectorization, we used the embedding model Qwen3-embedding:0.6B. The following parameters were optimized within the RAG Pipeline:

Chunk Size Defines the size of text segments that can later be retrieved individually.

Chunk Overlap Defines the degree of overlap between neighboring text segments.

Extractors Enrichment strategies applied during indexing, such as summarization/question generation.

Top K Defines the number of retrieved chunks considered as relevant search results.

2.4. Retrieval Optimization

Once the data has been indexed, the retrieval system searches for relevant information chunks in response to user queries. In our optimization workflow, the parameter `Top_K` is particularly important, as it controls the number of retrieved chunks included in further processing. Small values may exclude relevant sources, whereas excessively large values may introduce large amounts of irrelevant context and reduce retrieval precision.

2.5. Persona-Aware Question Generation

To generate realistic evaluation queries that reflect anticipated usage scenarios, the prompts provided to the language model were extended with customizable persona descriptions. This persona-aware prompt engineering approach enables the automatic generation of domain-specific and user-oriented retrieval benchmarks. The personas can be configured through external configuration files and represent different expected user groups and information needs. The persona definition was used to increase the quality of the generated questions, regarding to the anticipated end user group. As different personas lead to different evaluation datasets, no comparison of different persona prompts during the optimization process was performed.

Example Prompt: "You are a scientist who wants to learn about historic manor houses in Europe."

2.6. Optimization Strategy

We employed a greedy optimization strategy as suggested in Orbach et al. [9]. This iterative approach optimizes one parameter at a time. While this strategy does not guarantee globally optimal solutions, it enables sufficiently good retrieval configurations with significantly lower computational costs. As the primary evaluation metric, we used the Mean Reciprocal Rank (MRR), which measures the ranking position of relevant retrieval results within the returned result lists. A major advantage of this methodology is that the process is performed entirely without answer generation by LLMs. We evaluate only the retrieval quality, not the quality of generated answers. This substantially reduces computational costs.

3. Technical Implementation

The optimization workflow was implemented within the Wikibase4Research ecosystem using LlamaIndex, Ollama, and a Docker-based Python API. The system extends the existing RAG module of Wikibase4Research, which supports MediaWiki, Wikibase, and hybrid semantic knowledge infrastructures. All experiments were executed on a standard consumer laptop Intel Core i7-11850H CPU (8 cores, 16 GB RAM) without GPU acceleration.

4. Results

Despite the intentionally lightweight hardware setup, the optimization process could typically be completed within approximately one hour. Across all evaluated datasets, the optimization workflow produced substantial improvements in retrieval quality. The process requires only a one-click execution of the optimization workflow, without any manual intervention or configuration, thus enabling researchers to optimize retrieval performance without requiring machine learning expertise or access to specialized computing infrastructure. The increased retrieval quality is reflected in higher MRR values,

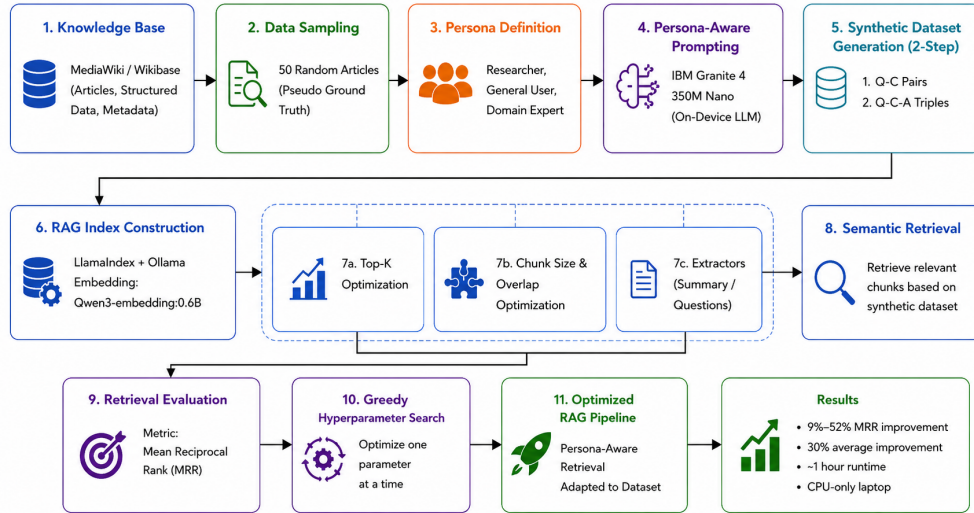


Figure 1: Workflow of the proposed AutoRag system

indicating, the parameter optimization adapts to the characteristics of the underlying knowledge base and the intended usage scenario.

Table 1

Retrieval optimization results across the evaluated datasets.

Dataset	Min MRR	Max MRR	Relative Improvement
Kung Fu Wiki	0.5663	0.7345	+29.7%
Manor Houses	0.3302	0.5010	+51.7%
Wikipedia	0.7961	0.8693	+9.2%
Average	–	–	+30.2%

Using Mean Reciprocal Rank (MRR) as the primary evaluation metric, retrieval performance improved by approximately 30% compared to the lowest performing parameter configuration. Thus, relevant sources appeared more frequently and at higher positions within the retrieval result lists after optimization. For individual datasets, improvements of up to 52% were observed.

The magnitude of improvement varied between datasets, indicating that retrieval performance remains dependent on knowledge-base characteristics and parameter interactions. For Kungfu Wiki and Wikipedia, the parameter with most impact to retrieval performance was `top_k`, while for Manorhouses dataset, the most important parameter during optimization was `chunk_size`. This observation aligns with the best found parameter set: Kungfu Wiki and Wikipedia had best MRR using smaller chunk size of 128 tokens whereas Manorhouses performed best at larger chunks of 4096 tokens. Using larger chunks with more context information, the `top_k` parameter seems to become less important. Vice versa, for the datasets using small chunk sizes, the parameter `chunk_overlap` had the fewest impact of all parameters.

Limitations and Observations During development, several limitations were observed. First, question generation occasionally produced hallucinations when retrieved pages contained little or no useful content, even if the document itself was not empty. Second, increasing the retrieval parameter Top-K often improved retrieval metrics such as MRR by increasing recall, but did not necessarily lead to better overall RAG performance. Larger Top-K values may introduce irrelevant context, reducing answer precision and increasing the risk of hallucinations during generation.

5. Conclusion

For scientific infrastructures such as digital libraries, domain repositories, and research data platforms, the efficient and resource-conscious operation of AI systems is becoming increasingly important. Our results demonstrate that meaningful retrieval optimization for RAG systems can be achieved without access to specialized computing infrastructure or data centers. This substantially lowers technical barriers, reduces operational costs, and enables the practical adoption of modern AI technologies in smaller research projects and publicly funded infrastructures.

Ethics and Environmental Footprint The proposed workflow lowers energy consumption for RAG systems. An optimization run required an estimated 0.05–0.07 kWh of energy, corresponding to roughly 15–25 g CO₂ emissions depending on the electricity mix. Potential biases introduced by synthetic dataset generation remain a limitation and should be considered when interpreting optimization results.

6. Future Work

The embedding models used for semantic retrieval have a substantial impact on both retrieval quality and computational performance, as also observed by Orbach et al.[9]. Our experiments indicate that there are limitations regarding the transferability of optimization results obtained with extremely small or highly simplified embedding models. If the embedding models are insufficiently adapted to the target knowledge domain, the resulting retrieval optimization remains suboptimal. The system could be further improved by including embedding model choosing in retrieval optimization as suggested in Orbach et al. [9] and including generation optimization as shown in Chen et al. [10]. In Future work, we will investigate the performance of the shown approach for Graph based RAG systems, based in wikibase data instead of mediawiki pages. However, the current approach is already applicable to Wikibase-based RAG systems, as the underlying RAG pipeline may process Wikibase pages similar to MediaWiki pages.

Acknowledgments

Wikibase4Research developments and RAG extension are partly funded by NFDI4Culture, funded by the Deutsche Forschungsgemeinschaft (DFG) under grant no. 441958017, and the EU HORIZON project ECHOLOT funded by the European Union under Grant Agreement n.101233096.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-5.5 in order to: Text Translation, Improve writing style, Generate images, Peer review simulation. After using these tool(s)/service(s), the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] J. Gao, Q. H. Pham, S. Varghese, S. Saurav, V. Hoskore, Ragalyst: Automated human-aligned agentic evaluation for domain-specific rag, arxiv.org:2511.04502 (2025). URL: <https://arxiv.org/abs/2511.04502>.
- [2] Z. Chen, Y. Liu, W. Xie, Y. Liang, P. Chen, S. Wang, Raise: Rag design as an architecture search problem, arxiv.org:2605.30029 (2026). URL: <https://arxiv.org/abs/2605.30029>.
- [3] L. Günther, Wikibase4research, MediaWiki Users and Developer Conference Fall 2025 (2025). URL: https://www.semantic-mediawiki.org/wiki/MediaWiki_Users_and_Developers_Conference_Fall_2025/Wikibase4Research.

- [4] L. Rossenova, P. Duchesne, I. Blümel, Wikidata and wikibase as complementary research data management services for cultural heritage data. proceedings of the 3rd wikidata workshop 2022 co-located with the 21st international semantic web conference (iswc2022), ISWC2022 (2022). URL: <https://doi.org/10.25968/opus-2573>.
- [5] D. Abián, A. Meroño-Peñuela, E. Simperl, An Analysis of Content Gaps Versus User Needs in the Wikidata Knowledge Graph, 2022, pp. 354–374. doi:10.1007/978-3-031-19433-7_21.
- [6] R. Ye, Z. Wang, Z. Ling, Y. Xiao, M. Li, X. Ma, B. Hui, Your language model secretly contains personality subnetworks, arXiv:2602.07164 (2026). URL: <https://arxiv.org/abs/2602.07164>.
- [7] Z. Yuheng, W. Yiran, T. Yuzhu, Z. Min, H. Yanhua, Seara: An automated approach for obtaining optimal retrievers, arxiv.org:2507.06554v2 (2025). URL: <https://arxiv.org/abs/2507.06554>.
- [8] X. Gao, Q. Pei, Z. Tang, Y. Li, H. Lin, J. Wu, C. H. and Lijun Wu, A strategic coordination framework of small llms matches large llms in data synthesis, huggingface (2025). URL: <https://huggingface.co/papers/2504.12322>.
- [9] M. Orbach, O. Eytan, B. Sznajder, A. Gera, O. Boni, Y. Kantor, G. Bloch, O. Levy, H. Abraham, N. Barzilay, E. Shnarch, M. E. Factor, S. Ofek-Koifman, P. Ta-Shma, A. Toledo, An analysis of hyperparameter optimization methods for retrieval augmented generation, arXiv arXiv:2505.03452v3 (2025). URL: <https://arxiv.org/abs/2505.03452>.
- [10] P. Chen, T. Chen, Cds4rag: Cyclic dual-sequential hyperparameter optimization for rag, arXiv:2605.08333v1 (2026). URL: <https://arxiv.org/abs/2605.08333>.

7. Online Resources

The software implementations described in this paper are available as open-source code repositories under the following links:

- <https://gitlab.com/nfdi4culture/wikibase4research/wikibase4research>
- <https://gitlab.com/nfdi4culture/wikibase4research/wikibase-RAG>,