PD.TVZ.HR

POLYTECHNIC & DESIGN Vol. 12, No. 4, 2024.

Mrežna inačica: ISSN 2459-6302
Tiskana inačica: ISSN 1849-1995

DOI: 10.19279/TVZ.PD.2024-12-4-19

USKLAĐIVANJE TROŠKA, BRZINE I RELEVANTNOSTI U SUSTAVIMA UMJETNE INTELIGENCIJE ZA TURIZAM.

BALANCING COST, SPEED AND RELEVANCE IN ARTIFICIAL INTELLIGENCE SYSTEMS FOR TOURISM

Tin Popović

Bulb Technologies, Ulica Grada Vukovara 23, Zagreb, Hrvatska

SAŽETAK

Turistički sektor suočava se s rastućim zahtjevima za personalizacijom u stvarnom vremenu s čime se tradicionalni sustavi teško nose. U ovome radu predstavljamo skalabilni AI sustav koji kombinira velike jezične modele (LLMs , engl. Large Language Models) s Retrieval-Augmented Generation (RAG) arhitekturama radi poboljšanja kvalitete preporuka u turističkim aplikacijama. Istražene su tri konfiguracije RAG-a za generiranje prilagođenih prijedloga smještaja, atrakcija i upita. Sustav se temelji na modularnim komponentama koje omogućuju fleksibilno prilagođavanje korisničkom kontekstu. Učinkovitost je ocijenjena kompozitnim indeksom RCT (engl. Relevance–Cost–Time) koji kvantificira kompromis između kvalitete odgovora, brzine i operativnog troška. Eksperimentalni rezultati pokazuju da naprednije RAG strategije značajno povećavaju relevantnost odgovora, ali uz veće troškove i latenciju, dok osnovni RAG postiže najbolju ekonomičnost. Dobiveni uvidi pružaju praktične smjernice za dizajn AI asistenata koji uravnotežuju personalizaciju s računalnom učinkovitošću.

Ključne riječi: *AI u turizmu, RAG arhitektura, veliki jezični modeli, RCT index,*

ABSTRACT

The tourism sector faces growing demands for real-time personalization and intelligent decision-making, which traditional systems often struggle to meet. This study presents the development of a scalable AI concierge system that combines large language models (LLMs) with Retrieval-Augmented Generation (RAG) architectures to enhance recommendation quality in tourism applications. We evaluate multiple RAG configurations to deliver personalized suggestions for accommodations, attractions, and travel-related queries. The system is designed with modular retrieval components that enable flexible adaptation to user inputs and contextual relevance. Performance is assessed using a composite RCT (Relevance–Cost–Time) index, which captures trade-offs between answer quality, speed, and operational cost. Experimental results highlight the strengths and limitations of each approach, providing practical guidance for designing AI-driven tourism assistants that balance personalization with computational efficiency.

Keywords: *AI in tourism, RAG architecture, large language models, personalization, RCT index*

1. UVOD
1. INTRODUCTION

Tourism is an especially dynamic activity in which travellers more and more expect personalised, up-to-date information and intelligent assistance. The classic customer support systems in hotel business often lack adaptability to meet various, contextually sensitive needs of different travellers' profiles. The most recent progress in artificial intelligence (especially large language models) have opened a possibility of building interactive systems of cognitive contexts that can simulate human understanding and conversation [1]. However, large language models themselves struggle with the facts' accuracy and the data timeliness which bumped broader application of RAG architectures (Retrieval-Augmented Generation) [2].

RAG enables dynamic connection of large language models with external structured or unstructured knowledge bases where the system retrieves relevant information (e.g., information on hotels, travelling rules, attractions in the proximity). Based on that, RAG systems generate more accurate and contextually relevant responses [3].

In this research we examine three basic RAG-configurations:

- Naïve RAG – The highest *k* of the retrieved documents gets forwarded to the model as the additional context. [4]
- RAG Fusion – A query gets reshaped multiple times, and the results get connected through the reciprocal rang-fusing method for improved coverage [5].
- HYDE (Hypothetical Document Embeddings) – The system generates a hypothetical answer first, and then transforms it to a vector and uses it to retrieve relevant contents [6]

Each method represents a different balance between simplicity, retrieval depth and computing load. We have implemented them by using retrieval methods based on OpenAI models for vector transformations, linked to a vector data base for an efficient search by resemblance. The retrieved documents get dynamically inserted into command templates in order to generate responses by large language model.

In order to assess the compromises among the configurations, we introduce the complex RCT index (Relevancy – Cost – Time) which balances out the responses' accuracy, response and cost efficiency. Such comprehensive valuation framework enables parallel analysis and directs design decisions when creating scalable, intelligent recommendation systems in tourism.

2. SLIČNI RADOVI
2. RELATED WORK

Retrieval-Augmented Generation (RAG) systems have demonstrated considerable efficacy as a methodological framework for grounding the outputs of large language models in external knowledge sources, thereby enhancing factual accuracy and mitigating the occurrence of hallucinations. This approach has gained notable prominence in both academic research and industrial applications, establishing itself as a foundational paradigm for the development of chatbots and question–answering systems that necessitate domain-specific expertise. [7] Even though the advanced RAG strategies can increase the response relevancy, they introduce additional load. Recent evaluation shows that more sophisticated retrieving methods such as repeated ranking or HYDE significantly increase precision, but with higher latency [8]. The delay of the first token roughly doubles when retrieval is included into the LLM process compared to the LLM alone [9], while needless retrieving steps increase both latency and cost with no benefit for simple queries. Our paper builds on those references by explicitly comparing latency and cost benefit of different RAG systems in a unified framework.

Along with that, the conversational solutions of the artificial intelligence experience quick progress thanks to LLMs. Personalized systems are also increasingly capitalizing on the potential of large language models. AI tourism assistants and chatbots are already being used in hotels and travel services: from simple hotel booking bots to complex travel planners with personalized suggestions [10]. Even though numerous papers show the potential of the domain-specific RAG

applications, the majority rates the individual techniques separately. Contrary to that, this paper presents a new comparative analysis of several RAG methods (Naïve vs. Fusion vs. HYDE) within a unified evaluation framework. A complex RCT (Relevancy-Cost-Time) index is being introduced for the joint evaluation of the answer quality, system responsiveness and efficiency, inspired by multi-criteria evaluation in research. By comparing the three strategies of the RAG system for tourist queries and by quantifying their trade-offs by a single metric, we provide practical insights for the design of economic, low-latency and high-relevancy AI assistants. Our findings extend prior research on RAG and conversational recommendations showing that design modularity and metrics-driven evaluation can direct the development of personalized assistants which optimize the balance of technical performance and the user experience.

3. MATERIJALI I METODE
3. MATERIALS AND METHODS

To investigate the application of RAG architectures in personalized recommendation systems for tourism, we developed a custom framework called AI Studio. The framework is designed for rapid prototyping, designing, and testing AI processes that integrate retrieval modules, internal data, and generative language models. It supports modular deployment of various RAG configurations: from simple to complex chained mechanisms, enabling real-time operation with external data sources.

AI Studio internally uses a vector database for document indexing, GPT models from the OpenAI suite and other large language models for text generation, as well as semantic embedding models for similarity search. Personalized recommendations arise from dynamically retrieving contextually relevant data based on the user's query and forwarding it to large language models. AI Studio also offers integrations with external services (e.g., Google Cloud) via APIs, enriching context in real time.

AI Studio being an internal and experimental system, the source code is not publicly available. All architectural components, design patterns, and applied methodologies are thoroughly described in the paper. The framework served as the main environment for conducting tests, evaluating RAG variants, and simulating recommendations in realistic tourism scenarios.

3.1. PRIPREMA PODATAKA I VEKTORSKO INDEKSIRANJE
3.1. DATA PREPARATION AND VECTOR INDEXING

The foundation of quality retrieval lies in the rich relevance of the indexed data. In the initial phase, we collected representative content that reflects the information travellers typically seek when planning a stay: digital brochures, websites, online booking lists, and FAQ sections from various accommodation providers. From this analysis, a comprehensive dataset was synthesized to simulate the experience of a mid-sized hotel, including details about check-in/check-out times, types and features of rooms, amenities (spa, gym, Wi-Fi), booking and cancellation policies, nearby attractions, dining options, accessibility, and contact information.

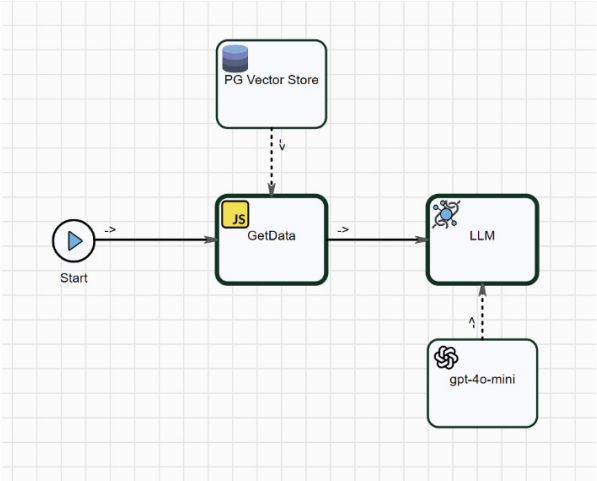
The documents are segmented into semantically coherent sections optimized for retrieval. Such context-oriented segmentation preserves the structure of the text and prevents the loss of meaning that occurs with naïve, fixed partitioning [11]. Each section is then transformed into a vector using OpenAI's text-embedding-3-large model, resulting in 3072-dimensional vector representations that retain semantic meaning. The transformed data are stored in a vector database for fast similarity search. The similarity between user queries and document sections is measured by cosine similarity, a metric useful for measuring semantic similarity between documents [12].

Slika 1 *Prikupljanje i obrada podataka*

Figure 1 *Data collection and processing*

3.2. NAIVNI RAG
3.2. NAÏVE RAG

The first implemented and tested architecture was the Naïve RAG, which serves as a basic reference point for more advanced RAG systems (Figure 2). After the system receives a user query, the custom JavaScript module GetData initiates a semantic search over the vector database and retrieves the k most relevant document sections. The returned results include not only the raw text but also metadata such as *chunkID* and *documentID*.



Slika 2 *Naivni RAG*

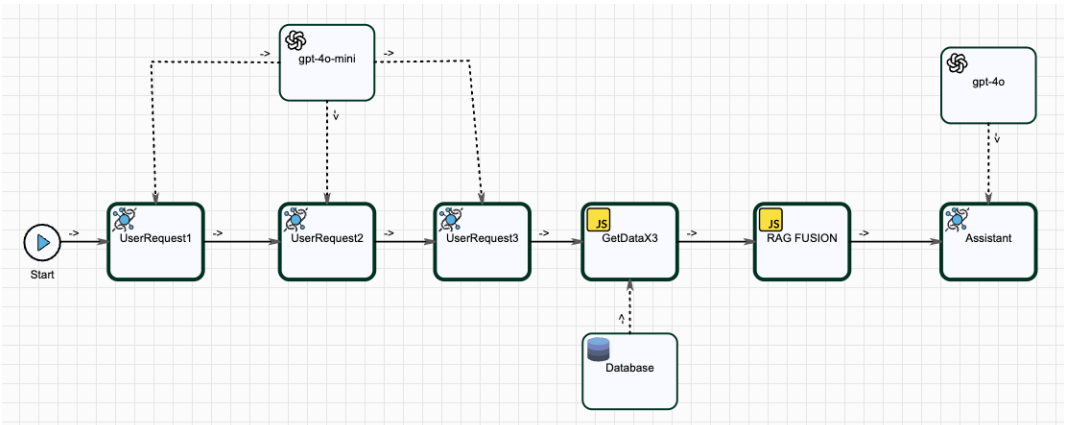
Figure 2 *Naïve RAG*

The collected data is then forwarded directly to the large language model block configured with OpenAI's gpt-4o-mini, without re-ranking or semantic filtering. The choice of the gpt-4o-mini model was deliberate: since the task involves answering grounded questions over structured hotel documents, it was assumed that the smaller model size maintains response quality while significantly reducing inference costs and latency. Because the architecture does not involve complex processes, a lighter model like gpt-4o-mini offers an effective balance between performance and resource consumption, making it ideal for baseline experiments in constrained environments. The model then processes both the query and the retrieved sections to generate a natural language response based on the combined input.

3.3. RAG FUSION ARHITEKTURA
3.3. RAG FUSION ARCHITECTURE

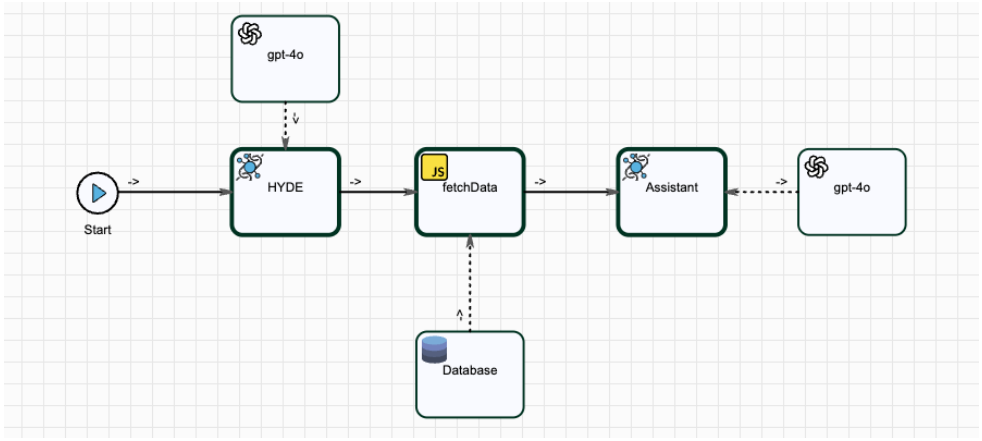
The most advanced architecture we implemented is based on the RAG Fusion paradigm (Figure 3) and introduces generation of multiple queries and result ranking for greater retrieval robustness. Its goal is to mitigate the limitations of a single query by creating several semantically related variations of the original query, which has proven effective by achieving good performance through multiple reformulations without significant loss of efficiency [13].

In the preprocessing phase, the input query first passes through a series of LLM command processing blocks (UserRequest1–3) which generate three new versions of the original user query. For example, the initial query "When can I check into the hotel?" may lead to variants such as "What is the earliest check-in time?", "At what time do guests usually arrive?", and "What are the hotel's check-in policies?". Each form is separately transformed into a vector and used to retrieve relevant document sections from the vector database.



Slika 3 *RAG Fusion*

Figure 3 *RAG Fusion*



Slika 4 *HYDE RAG*
Figure 4 *HYDE RAG*

Results connecting and re-ranking is performed by the RAG FUSION block, implemented by the JavaScript module. The *Reciprocal Rank Fusion* (RRF) algorithm with the parameter k = 60 connects the results of the three retrievals in a single ranked list. RRF justly ranks the best results of all the query variations and prevents excessive dependency on any of the formulations [14].

$$RRF\ (d\ \in D) = \sum_{r \in R} \frac{1}{k + r(d)} \quad (1)$$

Jednadžba 1 *Recipročni Rank Fuzije*
Equation 1 *Reciprocal Rank Fusion*

In the end, the highest-ranking section obtained by RRF is being combined with the original user's question and forwarded to the final LLM block, the Assistant (gpt-4o), which synthesizes the final answer.

3.4. HYDE RAG ARHITEKTURA
3.4. HYDE RAG ARCHITECTURE

The final architecture developed in this research is based on the HYDE (Hypothetical Document Embeddings) approach, which applies a generative retrieval method (Figure 4). Unlike classical RAG methods that directly vectorize the user query, HYDE leverages the capabilities of a large language model to first generate a probable answer to the query and only then performs retrieval, thereby improving alignment with relevant content.

When a user submits a query, it is first processed in the HYDE LLM block, where the gpt-4o model generates a hypothetical answer—a concise text reflecting what the assistant would say if it already had the necessary information. For example, the query "What amenities does the premium room have?" might result in the answer: "The premium room includes a king-size bed, a private balcony, a whirlpool tub, and free room service."

This synthetic answer is then vectorized and used as the input vector for similarity search in the vector database. The retrieved results are now aligned with the semantics of the generated hypothetical answer rather than the original query, which reduces ambiguity and improves semantic matching [15].

The selected sections are forwarded to the Assistant block, where they are combined with the original query and processed by the gpt-4o model to produce the final answer. This approach is particularly effective when the user query is vague or insufficiently specific, as the generated hypothetical answer "fills in" gaps in the query intent and improves retrieval alignment.

4. REZULTATI – RCT INDEKS
4. RESULTS – RCT INDEX

To evaluate the efficiency and effectiveness of the implemented RAG-based recommendation systems, we focused on three key dimensions of system performance: relevance of the generated response, response time, and computational cost—measured through generation and processing of text using large language models. These metrics were chosen to cover the critical components of the trade-off between answer

quality, speed, and resource consumption, which is especially important in the context of personalized AI applications in real-time tourism.

For the evaluation, we compiled a set of 100 standalone questions that tourists frequently ask. They covered a wide range of topics such as hotel services, local attractions, transportation, and booking policies. Several examples of these questions are shown in Figure 5. The questions were deliberately formulated as one-way queries to simplify testing and ensure consistency across all RAG configurations. Although real interactions often involve multi-turn conversations, this approach allowed us to observe system performance in isolation for each individual query. Moreover, since each standalone query can be considered a potential starting point for a broader dialogue, the results are still indicative of how the system might perform in more complex conversational environments.

Do I need to book the sauna in advance?
Are there any spa treatments for couples?
How much is a 1-hour massage?
Can I access the spa without booking a treatment?
What are the spa opening hours?
Is your gym open 24/7?
Do you offer personal training sessions?

Slika 5 *Primjer pitanja za evaluaciju*

Figure 5 *Example of questions for evaulation*

After testing on the collected dataset, the generation time, language model algorithm cost, and degree of relevance were measured for each response. Relevance was assessed with human intervention, where system responses were carefully analyzed and classified into categories: relevant, partially relevant, and irrelevant response.

In the context of AI-based tourism assistants, achieving high recommendation relevance often comes at the cost of increased latency and computational cost. More advanced architectures like RAG-Fusion and HYDE offer significantly better alignment with user intent but require multiple calls to the LLM or generative data processing steps, which increases the cost per query and extends response time. This reaffirms the classic trade-off in personalized AI systems: balancing output quality, responsiveness, and resource efficiency. In tourism applications, where real-time accuracy and user experience are equally critical, choosing the optimal architecture must therefore be based on multidimensional evaluation criteria.

To address this, we introduce a complex metric called the RCT index (Relevance–Cost–Time). A single indicator (e.g., accuracy alone) is insufficient for complex AI systems [16], so composite metrics combining multiple criteria are recommended to provide a more balanced insight into performance [17]. This approach aligns with evaluations in tourism and recommendation systems that merge different goals and constraints into a single function [18].

Our RCT measure integrates three normalized components: relevance, cost, and execution time.

(i) Relevance score (reflects the rate of accurate/ relevant responses)

$$Relevance\ score\ =\ \frac{number\ of\ relevant\ responses}{total\ nmber\ of\ resonses} \quad (2)$$

Jednadžba 2 *Ocjena relevantnosti*

Equation 2 *Relevance score*

(ii) Cost score is defined by the reference cost which represents the acceptable high threshold per query

$$Cost\ score\ =\ 1\ -\ \frac{Average\ Cost}{Reference\ Cost} \quad (3)$$

Jednadžba 3 *Ocjena isplativosti*

Equation 3 *Cost score*

(iii) Time efficiency score:

$$Time\ score\ =\ 1\ -\ \frac{Average\ time\ -\ Min\ time}{Max\ time\ -\ Min\ time} \quad (4)$$

Jednadžba 4 *Ocjena vremenske učinkovitosti*

Equation 4 *Time efficiency score*

We designed the RCT so that higher values indicate better overall performance—an ideal

method (i.e., high relevance, low cost, and fast response time) would achieve a composite score close to 1. The RCT is calculated as the weighted average of three normalized components: relevance score, cost-efficiency score, and time-efficiency score. To maintain interpretability and a range between 0 and 1, the sum of the weights must be 1. It is important to note that the weights can be adjusted: they can be changed to emphasize priorities specific to a particular domain or according to individual preferences. For example, in environments where latency is critical, the importance of the time component can be increased, while in cost-sensitive contexts, the cost factor might have greater significance. Overall, the RCT metric provides a unified measure of performance encompassing quality, efficiency, and speed, offering a practical basis for comparing methods across multiple criteria.

$$RCT_{Index} = w_1 relevance_{score} + w_2 Cost_{score} + w_3 Time_{score} \quad (5)$$

Jednadžba 5 *RCT indeks*

Equation 5 *RCT index*

To demonstrate how the RCT metric adapts to different priorities, we evaluated each RAG configuration with three sets of parameters. In the first scenario, we assigned equal importance to each component — relevance, cost, and time (weights: 0.33 each) — and used a reference cost of $0.02 per query. The resulting RCT scores are shown below.

Tablica 1 *RCT indeks za jednake težine*

Table 1 *RCT Index for equal weights*

RAG technique	RCT index
NAIVE RAG	0.76
RAG FUSION	0.91
RAG HYDE	0.85

In the second scenario, greater emphasis was placed on relevance, assigned a weight of 0.9, while the time component was neglected, and the remaining 0.1 was assigned to cost. This weight distribution reflects use cases where response quality is the highest priority—such as in premium tourism services—while cost is a secondary concern and latency is less important. The updated RCT results, shown in Table 2, demonstrate how changing the importance of individual factors affects overall performance scores, favoring methods that consistently provide accurate and contextually appropriate answers.

Tablica 2 *RCT indeks s prioritetom na relevantnosti*

Table 2 *RCT Index with Relevancy Priority*

RAG technique	RCT index
NAIVE RAG	0.70
RAG FUSION	0.92
RAG HYDE	0.87

In the third scenario, emphasis was placed on execution time, assigned a weight of 0.9, while the remaining 0.1 was assigned to cost, and relevance was neglected. This weight distribution is suitable for real-time applications, such as chatbots that must respond quickly, where speed is more important than peak accuracy. The RCT evaluation results, shown in Table 3, clearly demonstrate the advantage of simpler systems like Naïve RAG, which responds the fastest with minimal cost, while more complex methods like RAG FUSION lag due to higher latency and execution cost.

Tablica 3 *RCT indeks s prioritetom vremenske složenosti*

Table 3 *RCT Index with Time Complexity Priority*

RAG technique	RCT index
NAIVE RAG	0.93
RAG FUSION	0.78
RAG HYDE	0.86

5. ZAKLJUČAK.
5. CONCLUSION

In the final evaluation of the performance of our digital tourism assistant, the RCT (Relevance–Cost–Time) metric proved to be an extremely useful tool for objectively comparing different RAG approaches across three key dimensions: response relevance, execution time, and computational cost. By introducing this unified metric, a more precise understanding of the trade-offs each system makes has been enabled, depending on what is deemed most important

in a given context—response quality, speed, or cost-effectiveness. The evaluation conducted across three different scenarios with varying importance settings clearly demonstrated the flexibility and adaptability of the RCT index to the real needs of users.

In the first scenario, with an equal distribution of importance among all three components (0.33 each), RAG-Fusion achieved the highest RCT score (0.91), followed by RAG-HYDE (0.85), while Naïve RAG had the lowest score (0.76). This indicates that RAG-Fusion is a balanced solution that effectively balances all system aspects—relevance, speed, and cost.

In the second scenario, where almost exclusive priority was given to relevance (weight 0.9), RAG-Fusion again dominated (0.92), followed by RAG-HYDE (0.87) and Naïve RAG (0.70), which is expected given its robust architecture that generates contextually most appropriate answers.

In the third scenario, where the highest importance was assigned to execution time (weight 0.9), the ranking changed: Naïve RAG came first (0.93), ahead of RAG-HYDE (0.86) and RAG-Fusion (0.78). These results confirm that different configurations become significant depending on specific requirements—Naïve RAG is ideal for real-time applications requiring fast and lightweight processing, while more complex methods are better suited to scenarios demanding a high degree of accuracy.

The applicability of the RCT metric goes beyond the tourism domain. This evaluation approach for RAG systems can be easily applied in numerous other fields using AI for response generation, such as medical assistants, educational tutoring systems, customer support, and financial advisors. In each of these domains, priorities differ—for example, in healthcare relevance may have absolute priority, whereas in customer support speed and cost might play a more important role. This is precisely why the RCT metric, with its ability to adjust weights according to system priorities, enables informed decision-making during AI solution design and implementation.

Our findings confirm insights from the literature. Qi et al. [19] showed that increasing relevance—for example, by switching to an HNSW-based index—brings significantly higher time and computational demands, which aligns with our observations in the RAG-Fusion method. Additional research also indicates that methods like HYDE achieve the highest accuracy but with the greatest costs and latency [20]. Moreover, several authors have highlighted the lack of comprehensive metrics that consolidate quality and efficiency [21], which the RCT directly addresses by combining all three dimensions into one scalable and adaptable measure.

Despite the usefulness of the RCT approach, several limitations should be considered. The experiments were conducted in a controlled environment with a predefined question set and static data, which does not fully reflect the dynamic nature of real user interactions. Additionally, the current system does not include mechanisms for dynamic adjustment of the retrieval strategy based on query complexity. Future research should focus on developing hybrid architectures that include agent orchestration and adaptive retrieval mechanisms that could balance quality, speed, and cost in real time. This way, AI assistants—not only in tourism but also in other sectors—could better adapt to the specific needs of users and the demands of real-world environments.

6. REFERENCE

6. REFERENCES

[1.] Zhehao Zhang, et al. , Personalization of Large Language Models: A Survey, arXiv preprint , DOI: 10.48550/arXiv.2411.00027 , 2024.

[2.] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 9459 – 9474. DOI 10.48550/arXiv.2005.11401

[3.] Elmitwalli, S., Mehegan, J., Braznell, S. et al. Scalable evaluation framework for retrieval augmented generation in tobacco research using large Language models. Sci Rep 15, 22760 (2025). https://doi.org/10.1038/s41598-025-05726-2.

[4.] I. P. A. E. Pratama, I. M. O. Widyantara, Linawati, N. Gunantara, S. Suakanto and

E. T. Irawan, "Technology-Enhanced Learning for User Security Awareness Using AI-based Naïve RAG: A Design and Prototype," 2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS), Bandung, Indonesia, 2025, pp. 1-6, doi: 10.1109/ICADEIS65852.2025.10933283.

[5.] Izacard Gautier; GRAVE, Edouard. Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint, DOI: 10.48550/arXiv.2007.01282 , 2020.

[6.] Gao, L., et al. (2023). Precise Zero-Shot Dense Retrieval without Relevance Labels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023), Volume 1: Long Papers, Toronto, Canada, pp. 1762 – 1777. DOI 10.18653/v1/2023.acl-long.99

[7.] Klesel, M., Wittmann, H.F. Retrieval-Augmented Generation (RAG). Bus Inf Syst Eng (2025). DOI: 10.1007/s12599-025-00945-3.

[8.] Wang, X., et al. (2024). Searching for Best Practices in Retrieval-Augmented Generation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024), Miami, Florida, USA, pp. 17716 – 17736. Association for Computational Linguistics. DOI 10.18653/v1/2024.emnlp-main.981.

[9.] Huang, Y.; Han, X.; Sun, M. (2024). FastFiD: Improve Inference Efficiency of Open Domain Question Answering via Sentence Selection. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Vol. 1: Long Papers, Bangkok, Thailand, pp. 6262 – 6276. Association for Computational Linguistics. DOI 10.18653/v1/2024.acl-long.340

[10.] Wüst, K., & Bremser, K. (2025). Artificial Intelligence in Tourism Through Chatbot Support in the Booking Process—An Experimental Investigation. Tourism and Hospitality, 6(1), 36. https://doi.org/10.3390/tourhosp6010036

[11.] Dong, K., et al. (2024). MC-indexing: Effective Long Document Retrieval via Multi-view Content-aware Indexing. In Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, pp. 2673 – 2691. Association for Computational Linguistics. DOI 10.18653/v1/2024.findings-emnlp.150

[12.] Gunawan, D., Sembiring, C. A., & Budiman, M. A. (2018). The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. Journal of Physics: Conference Series, IOP Publishing, 2018. p. 012120. DOI 10.1088/1742-6596/978/1/012120

[13.] Ivica Kostric and Krisztian Balog. 2024. A Surprisingly Simple yet Effective Multi-Query Rewriting Method for Conversational Passage Retrieval. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24). Association for Computing Machinery, New York, NY, USA, 2271–2275. DOI 10.1145/3626772.3657933 , 2024

[14.] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09). Association for Computing Machinery, New York, NY, USA, 758–759, DOI 10.1145/1571941.1572114

[15.] Vake, Domen and Vičič, Jernej and Tošić, Aleksandar, Bridging the Question-Answer Gap in Retrieval-Augmented Generation: Hypothetical Prompt Embeddings. DOI: 10.2139/ssrn.5139335

[16.] Jannach D and Abdollahpouri H (2023) A survey on multi-objective recommender systems. Front. Big Data 6:1157899. DOI: 10.3389/fdata.2023.1157899

[17.] Alabbas, A., & Alomar, K. (2025). A Weighted Composite Metric for Evaluating User Experience in Educational Chatbots: Balancing Usability, Engagement, and Effectiveness. Future Internet, 17(2), 64. DOI: 10.3390/fi17020064.

[18.] Tsai, C.-Y., Chuang, K.-W., Jen, H.-Y., & Huang, H. (2024). A Tour Recommendation System Considering Implicit and Dynamic Information. Applied Sciences, 14(20), 9271. DOI: 10.3390/app14209271

[19.] Jinhu Qi, Shuai Yan, Yibo Zhang, Wentao Zhang, Rong Jin, Yuwei Hu, and Ke Wang. 2025. RAG-Optimized Tibetan Tourism LLMs: Enhancing Accuracy and Personalization. In Proceedings of the 2024 7th International Conference on Artificial Intelligence and Pattern Recognition (AIPR '24). Association for Computing Machinery, New York, NY, USA, 1185–1192. DOI: 10.1145/3703935.3704112

[20.] Chen, J., Lin, H., Han, X., & Sun, L. (2024). Benchmarking Large Language Models in Retrieval-Augmented Generation. Proceedings of the AAAI Conference on Artificial Intelligence, 38(16), 17754-17762. DOI : 10.1609/aaai.v38i16.29728.

[21.] Gan, A., et al. (2025). Retrieval Augmented Generation Evaluation in the Era of Large Language Models: A Comprehensive Survey. arXiv preprint, arXiv:2504.14891. DOI 10.48550/arXiv.2504.14891.

AUTORI · *AUTHORS*

• Tin Popović - rwas born on August 28, 2000, in Mostar, and graduated in 2025 from the Faculty of Electrical Engineering and Computing (FER), University of Zagreb, with a specialization in Data Science. He is currently employed at Bulb Technologies as a Specialist for R&D Software Projects Delivery. Previously, he gained work experience as a software developer at Ericsson and as a machine learning engineer at CROZ. His professional focus is on the development of advanced AI solutions, particularly agent-based systems and Retrieval-Augmented Generation (RAG) architectures for business process automation. He has participated in numerous research and commercial projects involving personalized assistants, data processing, and AI system evaluation.

Korespondencija · *Correspondence*
tin.popovic@bulbtech.com