Modelling and Constructing Avatars for Museum Exhibitions

Dušan Tatić[1][0000-0002-9228-6020], Maxim Goynov[2][0009-0007-2244-4314], Radomir Stanković[1]

[1] Mathematical Institute of the Serbian Academy of Sciences and Arts,
36 Kneza Mihaila Street, 11000 Belgrade, Serbia
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
8 Acad. Georgi Bonchev Street, 1113 Sofia, Bulgaria
dule_tatic@yahoo.com, goynov@gmail.com,
radomir.stankovic@gmail.com

Abstract. This paper presents a development in constructing the Avatar system based on artificial intelligence (AI) technology to answer the textual questions in its initial phase. The proposed solution is developed using the Ollama tool and implements a Retrieval-Augmented Generation (RAG) architecture. The advantage of this system is that it works locally and ensures responses are strictly based on a given document, which prevents imaginary answers. Use case is given based on the exhibition catalogue dedicated to Serbian mathematician Mihailo Petrović Alas as the knowledge source. The four systematic experiments are conducted with varying chunking parameters, retrieval methods, and system configurations. These experiments are evaluated using 25 test questions across different system setups. Based on the results, research establishes a foundation for developing reliable, AI text answering context-aware systems as the initial phase of creating AI avatars that can enhance visitor engagement.

Keywords: Kiosk System, RAG, AI, Ollama.

1 Introduction

Memorial rooms and houses where renowned people once lived are specific museum objects, often posing many challenges when creating permanent exhibitions there. In such spaces, visitors typically express a desire to learn more about the personal lives and work of the people who used to live and work there. In such situations, museum guides and curators answer many and various questions. The problem is that, among the visitors, many have interests in various topics, and the guide or curator answers only one question at a time. Since the visiting time is naturally limited, many visitors will leave the exhibition without a proper answer to their possible questions, since they simply did not get the opportunity to ask.

A possible answer to this problem, offered by contemporary information technologies, could be to develop museum avatars that will answer visitors' questions by using

Digital Presentation and Preservation of Cultural and Scientific Heritage. Volume 16, pp. 307-316, 2026.
DOI: 10.55630/dipp.2026.16.25, ISSN 1314-4006, eISSN 2535-0366.

artificial intelligence (AI) tools (Sylaiou & Fidas, 2022). The problem is that when allowing an open AI search over the Internet, it may often happen that the provided answer is inaccurate or completely freely imagined by the AI tools (Huang et al., 2025). To avoid such possibilities, it is much better to construct a closed system allowing search for answers over a limited, strictly controlled set of relevant documents. Obviously, the documents should be carefully selected, aiming to cover as wide a range of expected reasonable questions as possible. When the answer cannot be detected in the provided documents, the system responds that it does not know it.

The design of such an avatar system can be based on various programming methods and related tools. In this paper, we present some experiences gathered in developing an AI text-answering system using the Ollama tool and the concept for the Retrieval-Augmented Generation architecture (Murthy & Iliev, 2025). The intended application is for an exhibition presenting the life and work of the Serbian mathematician from the last century, Mihailo Petrovic Alas. We provide the experiments and test results based on a catalogue prepared on the occasion of an exhibition in 2018 dedicated to the memory of Mihailo Petrović Alas.

2 Documents Used for the System

The Serbian mathematician Mihailo Petrović Alas was one of the greatest Serbian mathematicians. He was an academician who developed the Serbian school of mathematics to the European level, as he was influenced by his French professors and mathematicians Henri Poincaré, Charles Hermite, and Charles Émile Picard. Besides his achievements and influences as a university professor of mathematics, he was a versatile personality. He was a fisherman, from whom he got his nickname – Alas, a Serbian word for fisherman, and had scientific work in ichthyology. Also, he was in that period a traveler and one of the few world explorers who sailed to the North and South Poles, and in the Atlantic and Indian oceans. Based on these traveling opportunities, he was inspired to write travel books. Among the mentioned things, he enjoyed playing the violin actively with his band. In addition to being a mathematician, fisherman, writer, philosopher, musician, world traveler, and travel writer, he has left his mark in archeology and cultural heritage.

The Serbian Academy of Sciences and Arts published a catalogue on the life and achievements dedicated to the celebration of 150 years since Mihailo Petrović Alas's birth (Mijajlović, 2018). The text of the catalogue is used to build a database for the AI text-answering system.

3 System Design

From the technical installation point of view, the system is imagined as a multimedia system with a touch screen, microphone, speakers, and projection plane. Using the touch screen, visitors can choose their preferred language and ask a question by typing a text message. Additionally, a microphone is installed for visitors who prefer to ask questions by voice.

The system's answers are provided similarly. The textual answer will be displayed on the screen while the audio narration is played on the system speakers so that visitors with either visual or auditory disabilities can use it equally. Also, we want to achieve an additional effect by projecting the video of the historical person onto the projection plane. In this way, visitors have the impression that a historical person is present at the exhibition space. At the exhibition space, a few systems that can work independently can be installed, depending on the expected average number of visitors.

The primary goal of this paper is to test an AI text-answering system using the Ollama tool as a part of the initial phase of developing an Avatar system. This involves testing multiple components and parameters to find the one that yields the most precise response for a given test question.

4 AI Text-answering System

The Ollama tool is used for developing an AI text-answering system. The feature of Ollama is that it enables downloading various large language models (LLMs) and simplifying their usage on a local computer. Additionally, LLMs can be adapted to specific needs without usage limitations, unlike those in cloud-based services. Therefore, we used this tool to implement the Retrieval-Augmented Generation (RAG) architecture to create an Avatar system (Lewis et al., 2020).

The advantage of RAG integration is using the given text to create a local database that enables answering questions strictly based on information stored in the local database. If a question is not relevant to the text stored in the database, the RAG should not give an imaginary answer. Accordingly, data are not shared with third parties and can be easily updated. Also, the system is not limited and can respond in real-time without waiting for the latency that is present in cloud services.

The main architecture of the RAG system is given as:

Indexing is the initial process of organizing and storing documents into a vector database. This considers the process of parsing and splitting documents into smaller textual segments called chunks. Then, an embedding model is used to vectorize and store the chunks in a vector database.

Retrieval is the first phase of the RAG system that identifies the most relevant chunks. This phase starts when a visitor queries the system. Then the embedding model vectorizes the query to search the vector database. Comparing the vectorized question with vectors in the database, the system returns the semantically most relevant documents (chunks).

The augmentation phase starts when the system retrieves relevant documents. In this phase, those documents are grouped with a prompt and the visitor's question. Gathered information in this way is prepared to be forwarded to the LLM to generate the answer.

In the generation phase, LLM analyses the defined prompt, retrieved documents, and visitors' questions. Accordingly, the LLM generates a coherent response to the visitor based only on the information stored in the database.

Our RAG system is realized with the Ollama tool and the LangChain framework. The indexing process uses the Qwen3 embeddings model and ChromaDB as a vector database. For generating the response, Qwen 3 LLM is used.

5 Experiments Setup

The document we use for analysis is an exhibition catalogue dedicated to Mihailo Petrovic Alas, 150 years of birth, which has a Serbian and English version. Only the English version is used for the experiment in this paper. Questions can be asked to the system in English, but it is also possible to query in Serbian. The system response is designed to provide answers in English.

The usage considered the conversion of the PDF catalogue into a textual (.txt) format. This conversion removed the images and figure captions. We had about 45 pages of text and 21056 words. The documents have been split into chapters before the chunking process.

During the creation of the vector database, each chapter is processed separately during the chunking process. For the experiments, we try different settings that concern the chunk parameters considered chunk size and chunk overlap, given in Table 1.

Table 1. The parameters of the chunking process for the experiments.

	Option 1	**Option 2**	**Option 3**	**Option 4**
Chunk size	500	800	1000	1300
Chunk overlap	200	250	250	400

Accordingly, for testing purposes, we made 25 questions in the English language for querying the system for those four different combinations of chunking parameters during the embedding process. Assessment of the questions has been done, giving a score of 1 for the correct answer, 0.5 for a partially answered question, and 0 for not valid answer.

The correct answer means that the answer is appropriate for a given question and contained in the chunk context, not imagined. Also, some of the provided questions are created in a way that the system should not provide an answer. If the system responds that the answer was not found, that is the correct answer and provides a score of 1.

A partially correct answer is generated from the given context, not imagined, but it should be more precise. For example, if the system provides the answer that Mihailo Petrovic was a mathematician and fisherman and doesn't point out that he was an academician at the university, then we give a partial score of 0.5.

Not a valid answer is the one that was provided from context that is not appropriate. The imagined answer provided by the LLM, out of the scope of the provided context, is rated as 0.

We conducted four experiments with different optimizations to improve the system:

- Experiment 1 – has been created as an RAG system with default parameters,
- Experiment 2 – some parameters are customized,

- Experiment 3 – added balance between relevance and diversity,
- Experiment 4 – included a reranker component to improve precision.

Those experiments were realized with the following configuration of PC: RAM - 16 GB, graphics card - NVIDIA RTX 4060 Ti 8 GB, processor - Intel i7-12700KF.

5.1 Experiment 1

We used the Qwen3 Embedding to create a vector database and query the system. For our RAG retrieval, we used default parameters. The default parameters use the Similarity search type, which returns the top 4 (k=4) chunks based on embedding similarity. Also, we created a basic prompt for the LLM that behaves as Mihailo Petrovic Alas, a famous Serbian mathematician, academician, and university professor. Based on the question, retrieved chunks, and provided prompt, the LLM generates answers. The results of answering the test questions in Experiment 1 are given in Table 2.

Table 2. The results of experiment 1.

Chunk:	500/200	800/250	1000/250	1300/400
Retriever	k=4	k=4	k=4	k=4
Correct:	13	18	15	17
Semi correct:	7	3	6	4
Not correct:	5	4	4	4
Score:	16.5	19.5	18	19
Percents:	66%	78%	72%	76%

The first row in the table presents the chunk size and chunk overlap parameters, which define the chunks in the vector database. The number of chunks retrieved from the database after querying the system is given in the second row. The type and number of system answers for the given database are given in the third, fourth, and fifth rows. Assessment based on the answers is given as a score and percentages in the sixth and seventh rows.

5.2 Experiment 2

Experiment 2 examined improvements of the previous experiment, considering the text document, chunks, and prompt.

It was noticed that some chunks are not well-formed. For example, the titles of chapters and some sentences were created as standalone chunks. Therefore, we rearranged the text document to make better chunk quality. We removed titles from the text document and added them as metadata. That metadata was later used during the augmentation phase and later for generating the answer.

Also, we increased the number of chunks the system retrieves. Instead of using the default 4 chunks (k=4) during retrieval for similarity search, we used a different number of retrieved chunks for various chunk sizes of the chunks.

During the generation of the answers, it was noticed that LLM put at the same level of importance, the area of mathematics and other secondary works of Mihailo Petrović. Therefore, we improved the prompt where we provide priority order in the response. We set that primary identity as a mathematician, academician, and university professor, and all other roles (such as fisherman, traveler, writer, cryptographer) are secondary. The results of experiment 2 are given in Table 3.

Table 3. The results of experiment 2.

Chunk:	500/200	800/250	1000/250	1300/400
Retriever	K=7	K=6	K=6	K=5
Correct:	18	18	19	18
Semi correct:	4	3	2	1
Not correct:	3	4	4	6
Score:	20	19.5	20	18.5
Percents:	80%	78%	80%	74%

5.3 Experiment 3

Similarity search gives the most relevant number of chunks (k), comparing the vector of the question with the vectors stored in the vector database. While it is looking for the precise results in the semantic sense, it can result in retrieving similar chunks, which can cause redundancy. Therefore, another approach using Maximal Marginal Relevance (MMR), provides a more complex selection of chunks. MMR uses similarity but also incorporates diversity between chosen chunks.

This approach retrieves a larger number of chunks (fetch_k). Then, MMR iteratively chose the final number of chunks (k), which are relevant for the question, but also sufficiently different from each other. This balance between relevance and diversity is determined by a parameter (lambda_mult). If the value of this parameter is closer to 1, priority is given to relevance, and the system behaves as a similarity search. If the value of this parameter is closer to 0, greater importance is given to diversity, but relevance is reduced. The results of the given experiments are given in Table 4.

Table 4. The results of experiment 3.

Chunk:	500/200	800/250	1000/250	1300/400
Retriever	K= 7 fetch_k=30 lambda_mult=0 .7	K= 6 fetch_k=25 lambda_mult= 0.7	K= 6 fetch_k=25 lambda_mult= 0.7	K= 5 fetch_k=20 lambda_mult= 0.7

Correct:	15	19	20	19
Semi correct:	6	3	2	2
Not correct:	4	3	3	4
Score:	18	20.5	21	20
Percents:	72%	82%	84%	80%

5.4 Experiment 4

In this experiment, one more layer of selection is added to improve the RAG system based on Experiment 3. This is done by embedding the reranker component into the RAG system. In our system, it is used to choose the best chunks after the MMR retrieval. This component is embedded to precisely rank chunks semantically. It compares the query with the chunks and gives the scores. After the scoring process, a defined number of the best chunks (top_k) are sent to the generation of an answer. The results of Experiment 4 are given in Table 5.

Table 5. The results of experiment 4.

Chunk:	500/200	800/250	1000/250	1300/400
Retriever	top_k= 7 k=40 fetch_k=120 lambda_mul t=0.7	top_k= 6 k=35 fetch_k=105 lambda_mul t=0.7	top_k= 6 k=30 fetch_k=90 lambda_mul t=0.7	top_k= 6 k=25 fetch_k=75 lambda_mul t=0.7
Correct:	16	20	23	22
Semi correct:	7	3	1	1
Not correct:	2	2	1	2
Score:	19.5	21.5	23.5	22.5
Percents:	78%	86%	94%	90%

6 Discussion

The provided experimental results have been summarized in Figure 1. These experiments show the steps of the RAG system improvement. The first experiment was the starting point, by creating an RAG answering system with chunking, default retrieval parameters, and a simple prompt. The best results for this experiment are retrieved for the databases made of chunk size 800 tokens and overlap 250 tokens, with a retriever configured to return top 4 chunks (k=4), with a score of 19.5 out of a maximum 25 (78%).

Improving the chunking process by optimizing the text, setting different retrieval parameters, and providing an improved and stricter prompt in experiment 2 yields better

system responses to the given text questions. The best results for this experiment were obtained using two different configurations. The first configuration used databases made of chunk size 500 tokens and chunk overlap 200 tokens, with a retriever configured to return the top 7 chunks (k=7). The second configuration used a database with a chunk size of 1000 tokens and chunk overlap of 250 tokens, with a retriever configured to return the top 6 chunks (k=6). Both configurations achieved the score of 20 out of a maximum of 25 (80%).

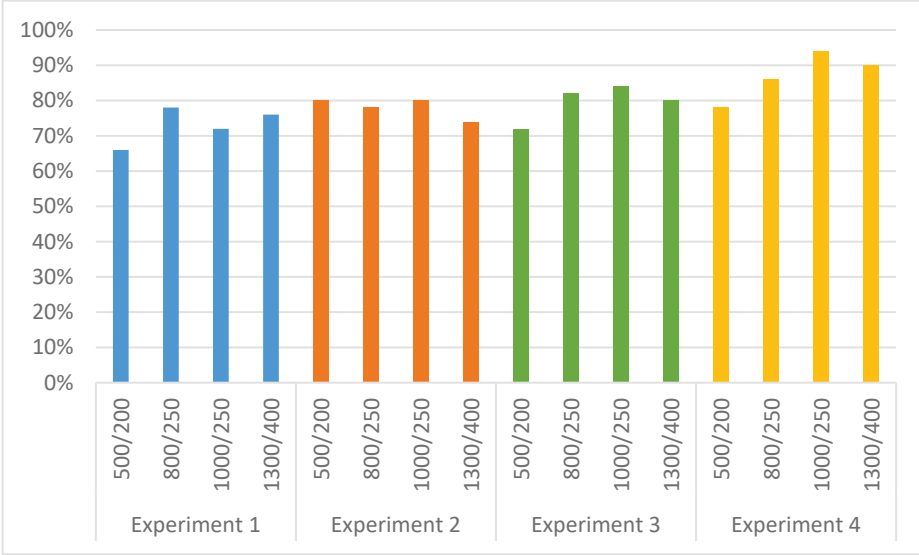

Fig. 1. Results of the experiments for the AI answering system.

In experiment 3, we switch from similarity search to MMR in chunk retrieval. This was done not only to use the most relevant chunks for retrieval but also to obtain more diverse answers. This diversification slightly improved the system. The best results for this experiment are retrieved for the databases made of chunk size 1000 tokens and chunk overlap 250 tokens, with a retriever configured to return the top 6 chunks (k=6) from selected 25 (fetch_k=25) chunks, with diversity parameter lambda_mult=0.7, with a score of 21 out of a maximum 25 (84%).

The greatest improvement we achieved was by adding a reranker to the diversified version in retrieval. The best results for this experiment are retrieved for the databases made of chunk size 1000 tokens and chunk overlap 250 tokens. The best performing setup used a reranker top 6 chunks (top_k= 6) selected from 30 candidates (k=30) retrieved from a larger set of chunks obtained through MMR search (fetch_k = 90) with a diversity parameter of lambda_mult = 0.7. This configuration achieved a score of 23.5 out of a maximum 25 (94%).

7 Limitation

This work is considered the first phase of development of the avatar system. Therefore, we tested different options based on the RAG architecture, and the assessment was done manually on the provided 25 questions. The next phase of the research will be to test different RAG system architectures. Also, we will add more text to the system and perform automatic testing of answers using larger online LLMs. Also, when the system is in the final stage, we plan to conduct the survey with the users.

8 Conclusions

Visitors of memorial rooms and houses, where renowned people once lived, desire to learn more about their personal lives and work. Museum guides and curators sometimes don't have the capacity and time to answer their questions. The research in this paper establishes a solution as an AI text answering system as the initial phase of constructing the Museum Avatar system. This solution is developed using the Ollama tool and implementing the RAG architecture.

The RAG architecture enables the system to work offline, which enables usage without limitations in terms of querying that exist in cloud-based systems. Also, this system's answers are based on the documents stored in the local database. If the question is out of the scope of the database documents, the system should answer that the information is not available. In this way, the system prevents imaginary answers.

Use case is given based on the exhibition catalogue of the greatest Serbian mathematician, Mihailo Petrović Alas, dedicated to the celebration of 150 years since his birth. Four experiments have been conducted based on various parameters to improve the system. The experiment evaluation is realized on 25 test questions across different system setups. The combination of chunking parameter, chunk size 1000 tokens, and chunk overlap 250 tokens, MMR retrieval, and the addition of a reranker component significantly improves response accuracy, achieving 94% correct answers as the best configuration.

While this initial phase shows promising results, future work should expand the knowledge base, implement different approaches, or use a combination of these as hybrid RAG approaches. Using these approaches in the plan is to make more complex assessments and conduct user studies to evaluate the system's effectiveness.

Acknowledgements

This research work was carried out and is supported by the joint research project "Advanced Technologies for the Digital Preservation and Presentation of Cultural Heritage" between the Institute of Mathematics and Informatics, Bulgarian Academy of Sciences and the Mathematical Institute of the Serbian Academy of Sciences and Arts

(2026-2028). This work was also supported by the Serbian Ministry of Science, Technological Development, and Innovation through the Mathematical Institute of the Serbian Academy of Sciences and Arts.

References

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, *43*(2), Article 42, 1-55. https://doi.org/10.1145/3703155

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *33*, 9459-9474.

Mijajlović, Ž. (Ed.). (2018). *Mihailo Petrović Alas: The founding father of the Serbian school of mathematics* [Exhibition catalog]. Serbian Academy of Sciences and Arts.

Murthy, H. K., & Iliev, A. I. (2025). Secure Curation and Reuse of Digital Scientific Data Using Retrieval-Augmented Generation Systems. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, *15*, 187–196. https://doi.org/10.55630/dipp.2025.15.17

Sylaiou, S., & Fidas, C. (2022). Virtual humans in museums and cultural heritage sites. *Applied Sciences*, *12*(19), 9913. https://doi.org/10.3390/app12199913

Received: March 15, 2026
Reviewed: May 10, 2026
Finally Accepted: June 02, 2026