

Multi-Agent System for Policy-Making Support: Initial Implementation of the Data Exploration LLM-RAG Agent

Ilija Kamenko^{1,*†}, Dragan Kukolj^{1,†} and Dubravko Culibrk^{2,†}

¹The Institute for Artificial Intelligence Research and Development of Serbia, Fruskogorska 1, Novi Sad, Serbia

²Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovica 6, Novi Sad, Serbia

Abstract

We present research in progress on the development of a multi-agent framework designed to enhance policy making through advanced data analysis techniques. The framework relies on specialized agents collaborating to transform complex datasets into actionable insights for decision-makers. Substantial progress has been achieved with the implementation of the first core component, the Data Exploration LLM-RAG Agent, which facilitates structured and intuitive interaction with complex tabular data by leveraging large language model retrieval-augmented generation techniques (LLM-RAG). A case study focusing on researcher productivity in Serbia illustrates the initial functionality and practical relevance of the approach. Further development is actively underway, with ongoing efforts directed toward the integration of domain knowledge and policy recommendation agents, ultimately aiming to establish a comprehensive, intelligent decision-support ecosystem.

Keywords

Multi-Agent Systems, Data Exploration, RAG, LLM, Policy-Making Support

1. Introduction

Increasing availability of large-scale datasets in science, education, and governance presents new opportunities for the development of new decision-making systems in public policy domain. Rapid development of large language models (LLM), capable of processing and generating huge amounts of text as well as simulating human behavior and reasoning, enables new approaches in the development of decision-making systems. Recent studies show that the introduction of the concept of multi-agent LLM-based systems [1] contributes to enhanced performance, i.e. reduced bottlenecks, enhanced fault tolerance, improved accuracy, or more refined decisions. For instance, in the domain of societal simulation a configurable multi-agent interaction framework is utilized to simulate classroom interactions between teachers and students [2]. Simulation of human-like behavior using LLM-based agents, e.g. changing attitudes and emotions in response to social events is presented in [3]. Another multi-agent framework named Cognitive Agents and Social Evolution Simulator, represents the simulation of social interaction and communication based on complex networks [4]. Its capabilities are illustrated by an election process simulation in which agents represent voters that reproduce voter behavior.

A Multi-agent LLM-based system is a framework where multiple agents powered by LLMs work together, communicate, collaborate and adapt, in order to solve a given problem. Multi-agent LLM-based systems are a powerful approach to complex problem-solving. Their power lies in the following features: parallel processing by distribution of tasks among multiple agents, specialization of agents in various domains with different tools attached, and advanced reasoning by powerful LLM models with adapted system prompts.

This work presents an evolving multi-agent framework, where specialized agents collaboratively analyze data, validate findings, enrich outputs with domain-specific knowledge, and ultimately generate

HHAI-WS 2025: Workshops at the Fourth International Conference on Hybrid Human-Artificial Intelligence (HHAI), June 9-13, 2025, Pisa, Italy

*Corresponding author.

†These authors contributed equally.

✉ ilija.kamenko@ivi.ac.rs (I. Kamenko); dragan.kukolj@ivi.ac.rs (D. Kukolj); dculibrk@uns.ac.rs (D. Culibrk)

ORCID 0000-0003-3352-7637 (I. Kamenko); 0000-0003-0711-0168 (D. Kukolj); 0000-0003-3417-1687 (D. Culibrk)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

policy-relevant recommendations. At the current stage, substantial progress has been made with the implementation of the Data Exploration LLM-RAG Agent, which enables flexible and dynamic interaction with structured datasets to support preliminary data understanding, exploratory analysis, and trend identification. Development of higher-level agents including the Domain Knowledge Agent and the Policy Recommendation Agent is actively underway, with future research phases aimed at achieving a fully integrated, decision-support ecosystem.

2. Multi-Agent Framework Concept

The full system is envisioned as a multi-agent framework designed to coordinate specialized agents, each focused on a distinct set of responsibilities within the data-driven research analysis and policy support workflow. This modular architecture ensures scalability, flexibility, and efficiency by distributing tasks across agents with complementary capabilities.

The overall structure of the framework is illustrated in Figure 1, showing the flow from the user's input through orchestration, specialized agent collaboration, and final policy-oriented outputs.

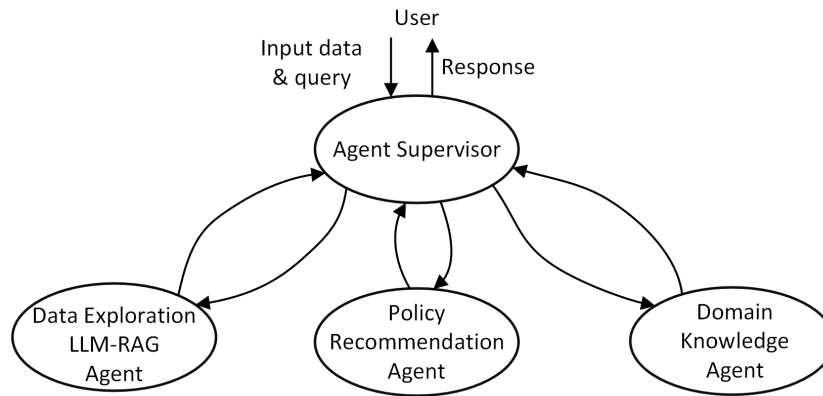


Figure 1: Block diagram of Multi-agent LLM-based system

In this framework, the Agent Supervisor receives input statements from the user and devises a plan to break down the problem into smaller tasks and then forwards the tasks to agents who are best suited for the specific tasks. The supervisor checks the validity of the task executed by the assigned agent. The reasoning process is inspired in part by [5]. The Data Exploration LLM-RAG Agent executes structured data querying, statistical analysis and visualization based on user prompts and retrieved data. The Domain Knowledge Agent enhances the analytical process by integrating domain-specific expertise, models and policy frameworks to ensure contextually relevant outputs. The Policy Recommendation Agent synthesizes analytical results and domain knowledge into coherent, evidence-based policy options and strategic insights.

Through seamless interaction between these agents, the system is capable of addressing complex domain questions and translating analytical findings into actionable recommendations. This multi-agent setup fosters distributed intelligence, allowing specialized components to collaborate dynamically and deliver rich, multi-dimensional insights to users.

3. Realized Component: Data Exploration LLM-RAG Agent

The Data Exploration LLM-RAG Agent is designed to transform natural language queries into structured, executable analyses. It implements a full Retrieval-Augmented Generation (RAG) architecture, consisting of three integrated modules: retrieval, augmentation, and generation as is illustrated in Figure 2. These modules enable the system to guide large language models (LLMs) with contextual data, relevant analytical methods, and execution logic, delivering both statistical summaries and visual outputs.

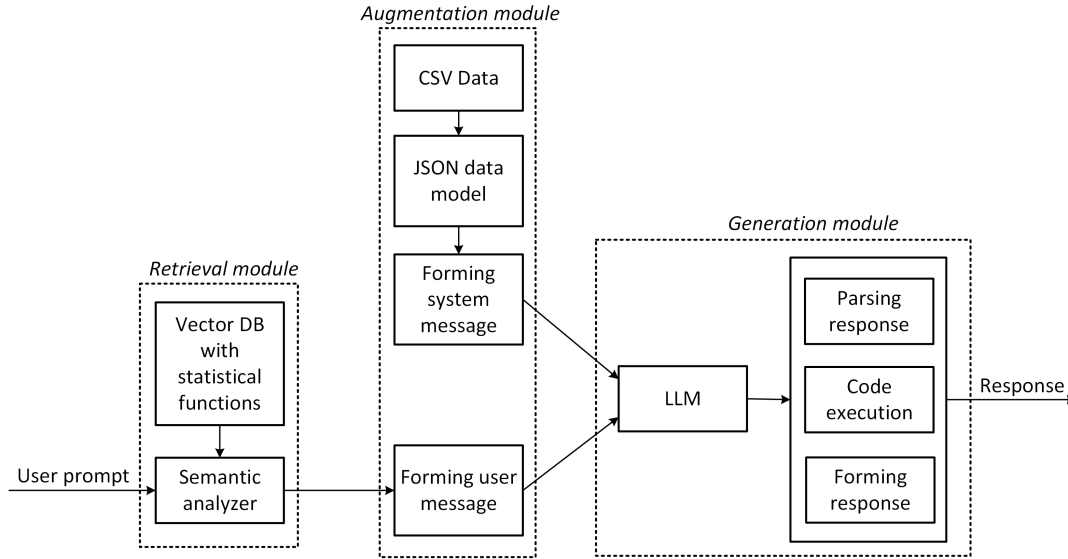


Figure 2: Block diagram of the Data exploration LLM-RAG agent

3.1. Retrieval

The retrieval module is responsible for identifying statistically relevant methods to guide the analysis. When a user submits a query, the system semantically analyzes the text to find appropriate statistical techniques. It uses a vector database (ChromaDB) to store embeddings of the statistical method descriptions such as t-tests, ANOVA, Pearson correlation, regression models, and clustering algorithms. The user's query as well as embeddings in the vector database are encoded using an embedding model (all-MiniLM-L6-v2) and then compared to the stored vectors.

If a statistically relevant method is found with a similarity score below a defined threshold, the corresponding method name and description are retrieved. This result is used to enhance the prompt sent to the language model, ensuring the generated analysis aligns with appropriate statistical practices. If no relevant method is confidently retrieved, the system defaults to generating general-purpose code, preserving robustness and flexibility.

This retrieval component forms the first stage of the RAG pipeline, enabling dynamic incorporation of external, domain-relevant knowledge into the analysis process.

3.2. Augmentation

The augmentation module prepares the language model context by constructing a complete, data-aware prompt. First, the structured dataset is automatically parsed and modeled into a JSON schema, describing each field's name, type, and semantic role. This schema acts as a formalized summary of the dataset's structure and is critical for guiding how the language model interacts with the data.

The system then assembles a system message, which defines constraints and operational guidelines such as instructing the language model to work only with the preloaded Pandas data (a Python library for data manipulation and analysis) and prohibiting external imports or data redefinition. Simultaneously, the user message contains the natural language query, optionally augmented with the statistical method retrieved during the previous phase.

These two messages form the composite prompt sent to the LLM model. Together, they give the model both the semantic intent of the query and the structural context of the dataset, enabling it to generate appropriate and executable analytical code.

3.3. Generation

In the generation module, the LLM model returns markdown-formatted output containing one or more python code blocks. These blocks may include data filtering, aggregation, statistical testing, or visualization instructions depending on the complexity of the query and any injected statistical method.

The system parses these code blocks, checks for syntax validity, and then executes them in a secure, isolated environment. Textual results, such as statistical test outputs or numerical summaries, are captured, while any plots generated using Matplotlib (a python library for data visualization) are rendered and encoded as Base64 images for downstream presentation.

To support iterative exploration, the agent also maintains conversational history, allowing users to refine their questions or ask follow-up queries based on previous results. This module completes the RAG loop by producing tangible outputs from semantically guided code generation.

4. Case Study: Researcher Productivity in Serbia

The effectiveness of the Data Exploration LLM-RAG Agent was evaluated through a real-world case study focused on researcher productivity in Serbia. This domain was selected because it offers rich, structured data across multiple dimensions (demographics, academic progression, and bibliometric indicators) making it an ideal testbed for demonstrating the agent's capabilities in extracting actionable insights to support policy development.

4.1. Dataset Overview

The system was evaluated using a national dataset of Serbian researchers spanning 15 years. Key fields include:

- Researcher identifiers (national ID, ORCID)
- Academic fields and titles
- Gender, birth year, and institutional affiliation
- Academic promotion histories
- Citation counts from multiple bibliometric databases
- Research outputs classified by national standards

The dataset was ingested in CSV format and dynamically modeled into JSON schemas for internal use.

4.2. Example Analytical Tasks

To demonstrate the flexibility and capabilities of the Data Exploration LLM-RAG Agent, different types of analytical tasks are showcased in the following examples. The ability of the Agent to return purely textual insights, generate visual outputs, and combine statistical analysis with textual and graphical responses is highlighted. Through these varied formats, the versatility of the Data Exploration LLM-RAG Agent in supporting a wide range of user needs, from simple descriptive queries to advanced, statistically driven explorations, as illustrated.

4.2.1. Example 1: Textual Response Only

In this example task, the question "How many researchers are between 30 and 45 years old?" is posed. The request is processed by the agent, and a textual response is generated: "Number of researchers between 30 and 45 years old: 8,157." In this case, no statistical function was applied, as the agent did not identify sufficient semantic similarity between the user prompt and the available vector database records, resulting in a direct retrieval and simple counting operation.

4.2.2. Example 2: Graphical Response Only

In this example task, the question "Compare graphically the trend of total citations over the last 10 years between male and female researchers holding the rank of associate professor" is posed. The request is processed by the agent, and only a graphical response is generated, illustrating the comparison of citation trends between the two groups (Figure 3). In this case, no statistical function was applied, as the agent did not identify sufficient semantic similarity between the user prompt and the statistical operation templates stored in the vector database.

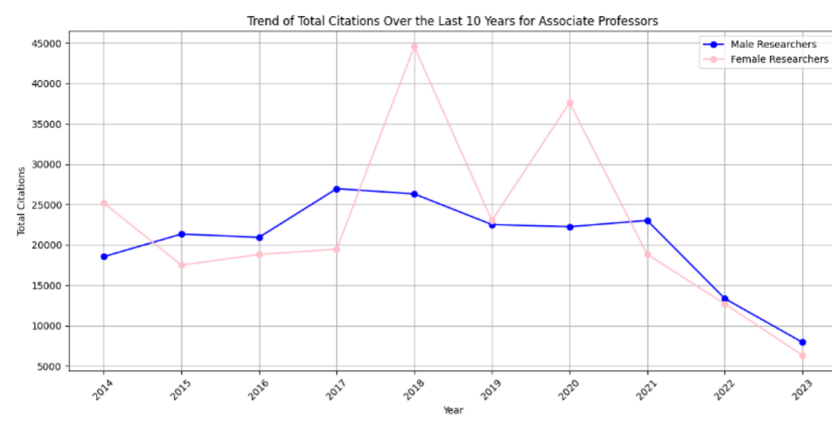


Figure 3: Example 2 Graphical response

4.2.3. Example 3: Textual and Graphical Response

In this example task, the question "Create an overview of the number of researchers by gender by year of birth in 5-year intervals. For each value, calculate the statistical significance in relation to the others and mark it with a shade from the color palette" is posed. The request is processed by the agent, and a statistical function is applied. The Chi-squared test is performed, and the resulting p-value is returned: "Chi-squared test p-value: 2.772350560555721e-51." In addition to the textual output, a corresponding graphical response is generated, illustrating the number of researchers by gender across 5-year birth intervals with color shading used to indicate statistical significance (Figure 4).

5. Conclusion

This work represents a foundational phase toward the development of a modular, multi-agent system for supporting evidence-based policymaking from complex datasets. The realization of the Data Exploration Agent demonstrates that structured, natural-language-driven data analysis is not only feasible but also capable of producing outputs relevant to informing and guiding policy discussions.

Active research efforts are now directed toward the implementation of the Supervisor Agent, the Domain Knowledge Agent, and the Policy Recommendation Agent, with the goal of building a comprehensive decision-support framework capable of delivering collaborative, verified, and policy-aligned insights.

6. Acknowledgements

This work is part of the TANGO project, which has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101070052.

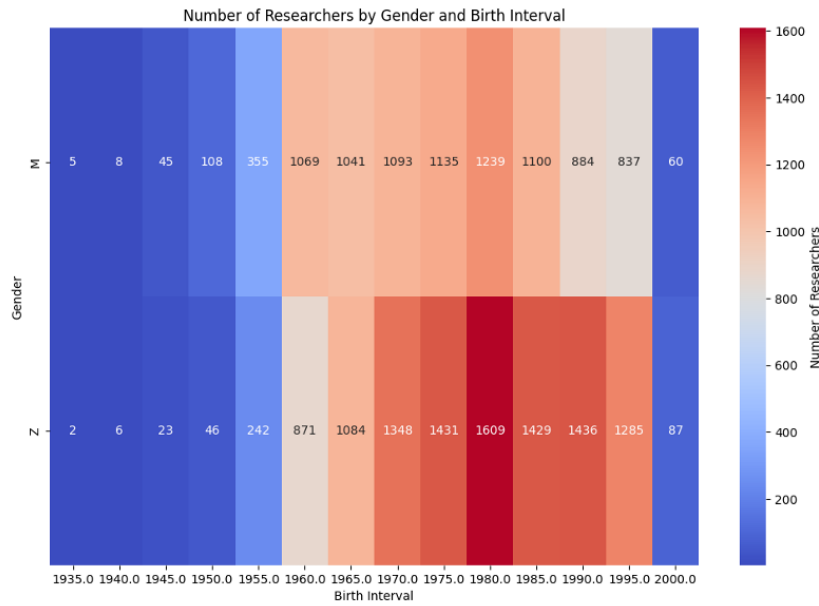


Figure 4: Example 3: Graphical response

Declaration on Generative AI

In preparing the work, no generative AI tools were used.

References

- [1] X. Li, S. Wang, S. Zeng, Y. Wu, Y. Yang, A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges, *Vicinagearth* 1 (2024). doi:10.1007/s44336-024-00009-2.
- [2] J. Sun, J. Zhang, Y. Wang, X. Wu, J. Liu, L. He, Cgmi: Configurable general multi-agent interaction framework, *arXiv preprint* (2023). [arXiv:2308.1250](https://arxiv.org/abs/2308.1250).
- [3] N. Ghaffarzadegan, A. Majumdar, R. Williams, N. Hosseinichimeh, Generative agent-based modeling: Unveiling social system dynamics through coupling mechanistic models with generative artificial intelligence, *arXiv preprint* (2023). [arXiv:2309.11456](https://arxiv.org/abs/2309.11456).
- [4] Z. Jiang, Y. Shi, M. Li, H. Xiao, Y. Qin, Q. Wei, Y. Wang, Y. Zhang, Casevo: A cognitive agents and social evolution simulator, *arXiv preprint* (2024). [arXiv:2412.19498](https://arxiv.org/abs/2412.19498).
- [5] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. Lee, E. Lim, Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, volume 1, Toronto, Canada, 2023, pp. 2609–2634. doi:10.18653/v1/2023.acl-long.147.