Multi-agent systems for improved information retrieval – leveraging autonomous agents and LLM models

1st Aneta Poniszewska-Marańda
Institute of Information Technology
Lodz University of Technology
Lodz, Poland
aneta.poniszewska-maranda@p.lodz.pl

2nd Maciej Kopa
Institute of Information Technology
Lodz University of Technology
Lodz, Poland
maciej.kopaa@gmail.com

3rd Bożena Borowska
Institute of Information Technology
Lodz University of Technology
Lodz, Poland
bozena.borowska@p.lodz.pl

***Abstract*—In the era of dynamic technological development and growing needs for data processing and analysis, the architecture of multi-agent systems is gaining importance. These systems, combined with Large Language Models (LLMs), offer an innovative approach to the information retrieval process that can enhance the efficiency, speed, and reliability of fact-finding and question-answering. This paper proposes the use of a designed multi-agent system architecture that uses autonomous agents and LLM models to efficiently acquire and process large amounts of data. It is shown how the integration of these technologies allows for more effective and precise information acquisition, which can lead to innovative solutions in both business and science. The effectiveness of solution were evaluated using specific metrics and testing.**

***Index Terms*—Multi-agent system, autonomous agents, information retrieval, LLM models, natural language processing, question answering.**

I. Introduction

Large Language Models (LLMs) and autonomous agents are rapidly evolving technologies with the potential to revolutionize a variety of industries. The advanced capabilities of LLMs allow them to understand and generate human-like text, making them invaluable for tasks such as content creation, decision-making, and data analysis. The ability of LLMs and autonomous agents to perform complex tasks, facilitate communication, and automate decision-making without human intervention is one of their key strengths. This ability to function without constant human supervision can lead to increased efficiency and innovation in a variety of sectors. However, while LLMs and autonomous agents offer significant advances in automation and intelligence, the complexity of developing and integrating these systems poses a challenge for developers seeking to fully leverage their capabilities.

Large Language Models and autonomous agents pose a number of challenges, particularly in the areas of knowledge management and the reliability of the information they provide. Developing and implementing these technologies requires developers to have a deep understanding of natural language processing, machine learning algorithms, and data management. LLMs often struggle to provide consistently accurate and reliable information, as they can generate plausible-sounding but incorrect or misleading answers. This problem is compounded by the fact that LLMs may not have access to up-to-date information in real time, leading to outdated or contextually inappropriate results. Furthermore, managing and preprocessing the massive amounts of data required to train LLMs is a complex task that requires significant expertise and resources. Consequently, overcoming these challenges is crucial to ensuring that large language models and autonomous agents can be effectively used in applications where accuracy and reliability are paramount.

The paper presents the analysis of possibilities of using the selected large language models in autonomous agent systems, focused on the implementation of information retrieval tasks. Our study focused on a detailed evaluation of selected LLM models and a critical review of existing solutions in this field.

Based on this analysis, the optimal methods used for system design and implementation were selected. The key element was the design and implementation of autonomous agents system capable of effective information retrieval in an environment with many available tools. Moreover, the comprehensive analysis of possibilities of optimizing the proposed solution, aimed at increasing its efficiency and effectiveness in the implementation of the set tasks.

II. Multi-agent systems and LLMs

An autonomous agent is a computational system or computer program located in a specific environment, capable of making decisions and performing actions independently to achieve its goals, without direct human intervention or control [1]. Autonomous agents have the ability to act autonomously, purposefully, and adaptively in complex and dynamic environments, using techniques such as machine learning, decision-making algorithms, and knowledge representation to process, reason, and respond to situations autonomously. They can range from simple rule-based systems to highly advanced AI models capable of learning, adapting, and exhibiting intelligent behavior similar to humans or animals [1]. Agents are often

2151-0849/25/$31.00 ©2025 IEEE
DOI 10.1109/ASEW67777.2025.00062

deployed in environments where they must interact with other software to effectively perform their tasks.

Autonomous agents are characterized by several key features that enable their independent action, intelligent behavior, and interaction in their environment: autonomy, reactivity, proactivity, social capabilities, learning and adaptation, mobility [2]–[4]. These characteristics allow autonomous agents to act autonomously, respond to change, pursue goals proactively, interact with other entities, learn and adapt, and potentially navigate their environment, making them versatile and able to cope with complex tasks and scenarios.

Autonomous agents can be classified into different types depending on their architecture, decision-making processes, and capabilities – several common types of autonomous agents [2], [5], [6]: reactive agents, deliberative agents, hybrid agents, learning agents, mobile agents.

Multi-agent systems (MAS): Composed of multiple autonomous agents, these systems involve agents in interactions, collaboration, and coordination to achieve common goals or solve complex problems [2], [3]. Autonomous agent systems have revolutionized the landscape of modern technology, presenting extraordinary opportunities in many applications. These intelligent systems are achieving significant advances in various fields, from improving information acquisition and data management, through improving healthcare and telemedicine, optimizing e-commerce and decision support systems, to the development of robotics and autonomous systems.

On the other hand, the development and implementation of autonomous agents poses a number of technical challenges that need to be addressed. These challenges cover various aspects, including agent architecture, decision-making processes, interactions, and consideration of real-world deployment conditions.

Large language models are advanced AI systems that use deep learning techniques to process and generate human-like text. They are trained on massive amounts of text data, enabling them to understand and produce natural language with remarkable fluency and consistency [8]. LLMs are neural networks specifically designed to handle sequential data such as text, but there are also architectures capable of transforming text to image or text to audio and vice versa. They use a transformer architecture that uses self-attention mechanisms to capture long-range dependencies within input sequences. By learning from massive text corpora, LLMs develop a deep understanding of language patterns, semantics, and context [9]. These models can perform a wide range of natural language processing tasks, including text generation, translation, summarization, question answering, and many others. Their ability to understand and generate human-like text has revolutionized fields ranging from writing to customer service and content creation [8]–[10].

Large language models have revolutionized the field of artificial intelligence, demonstrating extraordinary capabilities in thousands of applications. From powering advanced chatbots and virtual assistants to streamlining content creation and automating customer service, LLMs are proving invaluable tools across industries. Their applications span healthcare, finance, education, and advanced research, where they are used to analyze complex data sets, generate insightful reports, and deliver personalized user experiences.

III. Related works

Large language models are increasingly being used in the fields of scientific research and data analysis, offering promising applications and tools to assist researchers and analysts. LLMs can be used to efficiently search and summarize relevant scientific literature, helping researchers conduct comprehensive literature reviews and identify key findings and insights much faster [11]. Having an assistant with extensive knowledge can greatly increase the efficiency of researchers' work and increase their productivity. LLMs can also support the generation of research questions, hypotheses, and experiments from existing literature or data, facilitating the creation and exploration of new research directions [11]. In addition, LLMs can be tuned to interpret and analyze complex data sets, identifying patterns, trends, and insights that may be difficult for humans to spot [12].

In the field of information retrieval systems, a number of innovative approaches have been proposed, combining both conventional methods and recent developments in large language models. LLM models have relatively recently reached a level of effectiveness that allows their effective use in a variety of tasks, including advanced information retrieval.

Paper [7] presents an innovative concept of Professional Agents (PAgents). The idea is to use the potential of large language models to create autonomous agents equipped with specialized, interactive and professional competences. Work [13] presents a comprehensive review of various methods of using large language models in the field of information retrieval. The authors present a number of innovative concepts that significantly extend the capabilities of traditional search systems.

GPT Researcher appears as a breakthrough tool in the field of artificial intelligence, offering an advanced solution for autonomous research and information gathering [14]. The concept of an AI-driven research assistant is at the core of GPT Researcher, whose main goal is to optimize the processes of gathering, analyzing and synthesizing information from various sources. Authors of [15] present the development of BabyAGI, an autonomous task-driven agent. A prototype of the system was created to autonomously perform tasks for him, generate subsequent ones and set their priorities, using a simple but effective architecture. The work [16] presents a novel approach to improving the performance of linguistic agents by integrating with verbal reinforcement learning. The proposed solution enables the agent to analyze its previous decisions, learning from successes and failures, which leads to improved problem-solving abilities.

Using LLM models as a core component of agents for information retrieval tasks represents a significant advance in the field of data retrieval. The main challenges include:

zero-shot transfer to new tasks, limitations in open-ended reasoning, modeling social and interactive behavior, autonomous problem-solving abilities, and ensuring factual accuracy, especially in the face of increasing task complexity.

IV. Information retrieval using multi-agent systems and LLMs

This section presents a case study on the use of autonomous agents powered by large language models to optimize complex research processes.

The development of the proposed system was primarily motivated by the lack of a reliable, open and cost-effective solution for comprehensive information retrieval. In response to this gap, an attempt was made to create an innovative platform that combines innovative features, distinguishing it from other currently existing multi-agent systems.

The proposed method was designed to excel in fast information retrieval and synthesis in a wide range of topics, from daily news to in-depth academic research. Its basic function is based on an advanced process that includes question understanding, planning, finding helpful facts and detailed information analysis, report generation and rigorous quality control at each stage of the process aimed at finding an answer to the user's question. The overarching goal of this system is to provide highly reliable and trustworthy information, establishing a new approach in automated research processes.

The architecture of the proposed solution is conceptually organized into four main modules, each representing a separate phase in the information retrieval and synthesis process. The modules can be presented as specialized teams of agents within a larger, coherent system (Fig. 1):

1) *User interface and data preparation*: responsible for communication with the user and understanding the problem, effectively translating the user's queries into feasible research directives.
2) *Research module*: component dedicated to comprehensive information collection and preliminary analysis of relevant data sources.
3) *Writing module*: component focused on synthesizing the collected information into coherent and well-structured reports.
4) *Quality assurance module*: component responsible for ensuring the accuracy, relevance, and overall quality of the generated responses through multiple validation processes.

The system is designed to provide comprehensive and authoritative answers to user queries, while rigorously adhering to citation guidelines. The entire response process is presented in figure 2. The process begins with a thorough analysis of the user's query to ensure a complete understanding of the user's intent. This initial step is particularly important for overly general questions that require refinement to achieve optimal question precision. For example, a general question such as "What are the effects of climate change?" can be refined to "What are the effects of climate change on coastal ecosystems in the Mediterranean over the past decade?". After

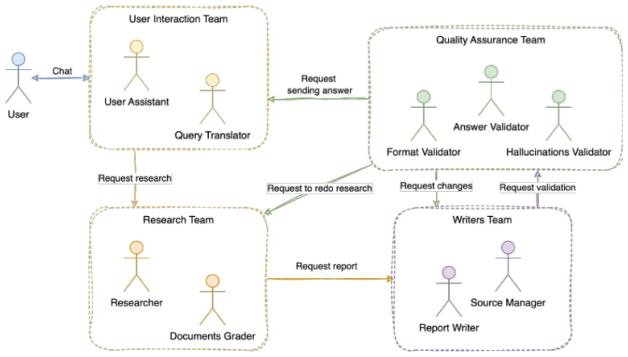

Fig. 1. Agent modules cooperating within a multi-agent system and their inter-team communication.

an initial task understanding phase, the system generates a series of targeted search queries. These queries are designed to explore a variety of online resources, including academic databases, recent press releases, and trustworthy websites. This comprehensive search strategy ensures that a broad spectrum of information is collected, covering both academic research and breaking news.

Then, another agent analyzes and evaluates the relevance and credibility of the collected information. If the collected data does not meet predefined quality standards, the system initiates a recursive research process to obtain more relevant information. On the other hand, if the collected data is deemed good enough and reliable, it is forwarded to specialized agents responsible for writing reports and managing sources. These agents synthesize the information into a coherent and structured response that directly addresses the user's query.

The generated response then undergoes a rigorous verification process. This complex verification includes checking the accuracy of sources, detecting potential hallucinations or unsupported claims, and assessing the overall relevance and usefulness of the response with respect to the original question. Only after these verification procedures have been successfully completed is the final response presented to the user via an intuitive and user-friendly interface. This designed process ensures the delivery of high-quality, well-documented and directly related to the user's query responses, maintaining the system's commitment to the accuracy and credibility of the information disseminated.

Figure 2 presents the complete process of the proposed multi-agent solution with a detailed description of each agent, its responsibilities, and the communication process between agents. Each of the listed agents plays only a specific role within the system, contributing to its overall functioning and efficiency.

The *User Assistant* serves as a key interface between the user and the system, performing a number of important functions. Its primary task is to evaluate the specificity and clarity of the user's initial query. In situations where the query does not demonstrate sufficient precision to generate a com-

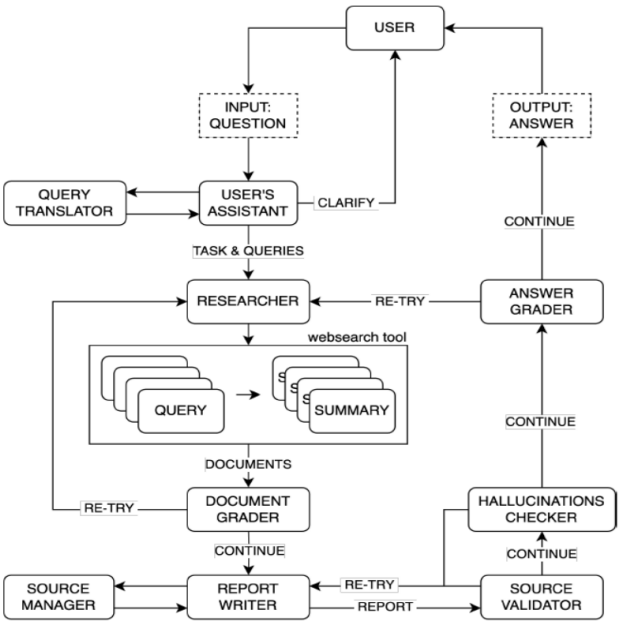

Fig. 2. Workflow schema of LLM-based agent system.

prehensive answer, the agent initiates and maintains a dialogue with the user. This process includes requests for clarification and, where appropriate, proposals for potential improvements to the original query. This recursive interaction continues until the agent determines that the query has been fully clarified and is adequate for further processing. It should be noted that this procedure is usually not necessary for simple queries. However, for complex or imprecise tasks, this preliminary step significantly increases the reliability and efficiency of the system. By providing a comprehensive understanding of the user's intentions, the agent minimizes the risk of inefficient use of computational resources on potentially irrelevant or unhelpful answers. Once a satisfactory level of query clarity is achieved, the assistant initiates cooperation with the Translator agent and hands over the task to it (Fig. 3).

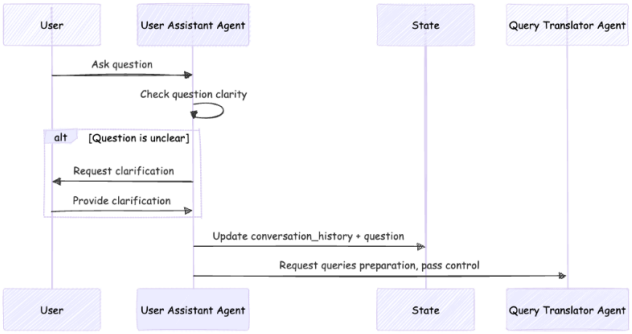

Fig. 3. Context diagram showing the User Assistant agent working.

The *Context Translator* plays a key role in the information retrieval process, serving as a bridge between the user's query and the query-structured topics related to the user's intentions. This specialized agent has a single, but extremely important task: to analyze the history of interactions between the user and the User Assistant, and then generate a set of precisely defined research topics, or search queries. The query translation process is fundamental to the efficiency of the system, as it transforms potentially ambiguous or colloquial user queries into precise, machine-readable queries. The Context Translator ensures that the next phase of the process is both precisely focused on the topic imposed by the user and comprehensive in this respect. The generated topics are encoded in the standard JSON format, which optimizes their use by the next agent — the Researcher. Using JSON as the output format is a strategic choice, facilitating smooth transfer and interpretation of data between agents. This standardization not only increases interoperability between different components of the system, but also enables efficient parsing and prioritization of research topics by the Researcher agent to which it transfers control.

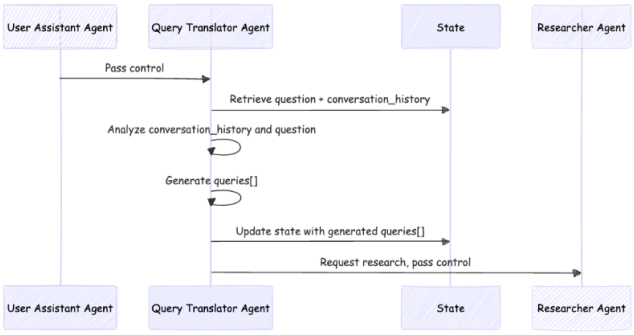

Fig. 4. Context diagram showing the Context Translator agent working.

The *Researcher* occupies a central and key place in the system's process architecture, responsible for conducting comprehensive and detailed information retrieval. It operates on the basis of a structured input consisting of a list of queries, implementing a multi-faceted research strategy. Before starting work, the Researcher retrieves topics for research from the system state. The research method used by the Researcher is based on the use of advanced search tools, such as the Tavily and Exa search engines. These services are used together to maximize the diversity and comprehensiveness of results, because each of the search engines uses different search algorithms and, consequently, different data sources. In cases requiring more detailed analysis of web content, the FireCrawl tool is also used to obtain more comprehensive and in-depth information. The information acquisition process is divided into 2 phases and optional 3rd phase:

1) At first, the Researcher uses input queries for both tools, Tavily and Exa, downloading the best k results from each. The use of multiple search services is intended to obtain a wider spectrum of information.
2) After the initial search phase, the Researcher synthesizes and summarizes the collected results, preparing them for

delivery to the next agent in the workflow—the Content Reviewer.

3) If the summarized results are deemed insufficient, the Researcher initiates a more detailed analysis. This deeper analysis phase involves using the URLs identified in the previous steps, in conjunction with a web parser, to extract and analyze entire web pages or scholarly articles.

Escalation to more in-depth content analysis is typically triggered by queries of significant complexity or those requiring more nuanced understanding. This phase represents a strategic deepening of the research process, allowing for more in-depth exploration of complex topics and providing a more robust and comprehensive response to challenging queries. Through this multi-level, adaptive approach to information collection and analysis, the Researcher plays a key role in ensuring the accuracy and diversity of the knowledge base, significantly improving the quality and reliability of the final output delivered to the user.

The *Content Verifier* plays a key decision-making role in the system's workflow, acting as an evaluation filter for the retrieved information. The primary responsibility of this agent is to evaluate the relevance and usefulness of documents collected by the Researcher, comparing them to the original user query to determine their usefulness and value. This ensures that only the most relevant information is passed on through the system. A key aspect of the Content Verifier's functionality is its role in optimizing the flow of information. By systematically identifying and excluding documents that do not meet established relevance standards, the Content Verifier efficiently optimizes the information corpus. This process is essential for streamlining subsequent stages of the workflow, ensuring that subsequent agents operate on a filtered and highly relevant dataset. Furthermore, the role of the Content Verifier goes beyond just filtering—it actively shapes the knowledge base for further processing. By removing irrelevant or outlier documents from the system's process state and sequencing these documents, the agent not only increases the efficiency of LLM models, but also minimizes the risk of introducing noise into the final output. Reducing the document set is essential to maintaining the LLMs focus on the most important information and the integrity of the entire process.

Reranking documents is essential to minimizing the impact of the lost in the middle phenomenon. In essence, the Content Checker acts as a gatekeeper, ensuring that only the most relevant and valuable information is retained for further processing. This critical evaluation step significantly contributes to the overall quality and accuracy of the system output, closely aligning with the user's information needs and expectations.

During the information generation process, two closely cooperating agents play a key role: the Report Author and the Source Manager.

The *Report Author* serves as the main information synthesizer, responsible for creating an accurate and comprehensive answer to the user's query. The *Source Manager*, working in synergy with the Report Author, plays a specialist role in the process of verifying and formatting citations. Their functions, although distinct, are complementary and essential in the process of generating the final answer for the user.

The *Report Author* acts as the main information synthesizer, responsible for creating an accurate and comprehensive answer to the user's query. He operates on the basis of selected summaries and analysis results provided by other components of the system. The primary task of the Report Author is to construct a coherent, substantive and precise answer, based solely on the context provided, while rigorously adhering to the principle of information fidelity. This approach ensures that the generated answer is solidly grounded in the source materials, minimizing the risk of introducing unsupported or false information.

An important aspect of the Report Author's work is placing references to sources. In order to increase the accuracy and consistency of these citations, the Report Author works closely with the *Source Manager*. This cooperation is aimed at ensuring the integrity of the answer with previously prepared documents and the possibility of verifying the information presented in the final report. The Source Manager, working in synergy with the Report Author, performs a specialized function in the process of citation verification and formatting. It effectively complements the Report Author's competences by using the advanced capabilities of large language models. It should be noted, however, that this approach is not without its drawbacks – it can lead to increased system latency and increased consumption of computational resources, expressed in the number of tokens processed.

The *Format Validator* plays a key role in the quality assurance process of the system, acting as the first of three specialized agents whose goal is to guarantee a high standard of the generated response. Despite some analogies with the Source Manager in terms of verifying the availability of source data, the Format Validator function is much more complex and decision-making in nature. Its role goes beyond simply confirming the presence of sources to include comprehensive evaluation and validation of the structure and format of the presented data. The main function of this agent goes beyond simply checking the format of sources in the generated response – it is tasked with making the crucial determination whether the generated content meets the required standards to proceed further in the workflow or whether additional corrections are required. In the event that the generated content does not meet the established criteria, the Format Validator provides detailed feedback explaining the specific reasons for rejecting the generated content. This information serves as a valuable guide for the Report Author, indicating the direction and scope of subsequent iterations. The precision of the feedback is crucial to streamlining the revision process, enabling targeted improvements that address specific issues with missing sources, formatting, or structure. In turn, when the generated content meets the Format Validator criteria, the agent updates the system state with a decision to proceed. A positive evaluation indicates that the generated content has

passed the initial quality checkpoint and is ready for further evaluation by subsequent quality assurance agents.

The *Hallucination Validator* serves as another quality control mechanism in the multi-agent system workflow, aimed at ensuring the highest level of content integrity and factual accuracy. This agent performs detailed cross-checking between the generated text and its cited sources and checks the content for instances of false or unsubstantiated information.

The *Response Verifier* is the last quality check point in the proposed system workflow, placed immediately before the generated response is presented to user. The primary function of this agent goes beyond the format and factual accuracy checks performed by earlier validators – its job is to ensure that the response generated directly and comprehensively addresses the user's original query or the conversation summary established at the very beginning of the process.

In the process of selecting appropriate large language models for the proposed system, a comprehensive evaluation of available models based on several key criteria was carried out. The analysis of benchmark results, such as LLMU, GPQA, DROP, MGSM, MATH, HumanEval, MMMU, revealed that individual LLM models show different levels of proficiency in different types of tasks. This required a thorough process of validation of the effectiveness of individual LLMs in order to identify the models most suitable for the specific requirements of the proposed solution.

The main criteria for selecting LLM models included: open-source availability: to ensure transparency and adaptability of the system, hardware compatibility (models must run on 32 GB of RAM, balancing performance with resource efficiency), JSON mode support: essential to maintain consistent output formats and prevent errors during agent-specific tasks, cost, efficiency, and speed: prioritizing models that offer optimal performance within reasonable computational and financial constraints.

These criteria led us to focus on a specific subset of models, deliberately excluding some popular options. Although models from OpenAI, Anthropic (Claude), and Google are known for their accuracy, conveniently exposed via API, their proprietary nature, high cost, or lack of robust JSON mode support made them inappropriate for the proposed agent system. The following models were selected for detailed evaluation: Llama2, Llama3, Gemma2.

The module responsible for information retrieval uses several powerful tools to enhance the capabilities of selected language models. Key to the information gathering process are two advanced search engines: Exa and Tavily, and the FireCrawl web page parsing tool. Exa enables agents to perform comprehensive web searches, providing access to a huge amount of up-to-date information. Tavily offers an alternative search method, optimized for large language models and autonomous agents. FireCrawl is key to in-depth analysis of specific web content, searching specific URLs and converting the content to pure markdown. The synergistic combination of these tools provides selected language models with access to comprehensive, diverse and well-structured knowledge.

In order to maintain the high level and reliability of selected LLM models, the LangSmith tool was used in the project as the main debugging tool. This platform allows monitoring, testing and improving the application of the language model in agents. LangSmith's ability to track, log, and analyze model results is key to identifying and resolving issues with LLMs, optimizing performance, and ensuring agent reliability across tasks and scenarios.

To seamlessly integrate different components and create a coherent system, the proposed system relies heavily on two popular libraries: LangChain and LangGraph. LangChain serves as a foundation, connecting different language models, tools, and data sources into a unified workflow. LangGraph is used to design and manage agent workflows, allowing the creation, modification, and optimization of processes that drive their actions. Both libraries integrate seamlessly with LangSmith, further simplifying the development process.

V. Evaluation of the multi-agent system and analysis of results

In order to evaluate the performance of created autonomous agents system powered by large language models, an evaluation method was developed and tests were carried out on the system in order to optimize complex research processes. The comprehensive evaluation process was developed using a specially designed dataset. The dataset consists of 128 questions and their corresponding answers, divided into three subjective difficulty levels: easy, medium, and hard. The difficulty classification is based on the expected complexity of the information retrieval process required to accurately answer each question, as determined by the dataset authors.

The questions cover a variety of domains, including general knowledge, academic papers, complex definitions, current events, and time-sensitive information (e.g., daily sunset times or restaurant opening hours). Sample questions are presented din figure 5. Additionally, the dataset includes queries about future events and intentionally false or paradoxical questions to assess the system's ability to detect and cope with user-provided misinformation or logical inconsistencies.


```
{
    "ID": "0023",
    "question": "What is the concept of 'zero-shot learning' in LLMs?",
    "answer": "Zero-shot learning refers to an LLM's ability to perform tasks it was not explicitly       trained
on, by leveraging its general language understanding from pretraining, without requiring       additional task-
specific data.",
    "difficulty": "hard",
},
{
    "ID": "0049",
    "question": "In which city would you find the historic Colosseum?",
    "answer": "The Colosseum is located in Rome, Italy.",
    "difficulty": "easy",
    "type": "general"
},
{
    "ID": "0068",
    "question": "When is the next full moon this month?",
    "answer": "The next full moon will occur on the 12th of this month.",
    "difficulty": "medium",
    "type": "time"
},
{
    "ID": "0128",
    "question": "When OJ Simpson murdered George Washington?",
    "answer": "George Washington was not killed by OJ Simpson but died on December 14, 1799, at his plantation home,
Mount Vernon, in Virginia. His death was the result of a severe throat infection, believed to be acute epiglottitis
or possibly acute bacterial pneumonia.",
    "difficulty": "hard",
    "type": "fake"
},
```


Fig. 5. Sample items in prepared set of test questions.

Given the emphasis on evaluating small, open-source models such as Llama2 (6.74 billion parameters), Llama3 (8.03 billion parameters), and Gemma2 (9.24 billion parameters), a special testing methodology was developed—each model runs a specially prepared set of 128 questions 5 times, which allows for accurate consistency assessment and eliminates uncertainties arising from causes other than models themselves.

To ensure an objective and comprehensive evaluation, three advanced large-scale language models – GPT-4o, Gemini 1.5 Pro, and Claude 3.5 Sonnet – were used as validators and evaluators. These models were carefully configured with specific evaluation guidelines and examples to standardize the evaluation process. The scores of all three models are averaged to obtain a final score for each response.

The evaluation method based on a multi-model, multi-iteration approach to evaluating responses generated by a multi-agent system provides an autonomous method for evaluating the performance of smaller, open-source language models. By leveraging the capabilities of more advanced models in the evaluation process, we aim to provide a detailed and comprehensive analysis of the performance of the developed system across different query types and difficulty levels in an automated manner.

The results show that Gemma2 performs best in the source attribution and hallucination detection categories, indicating its factual consistency, but requires significantly more processing time than Llama2 and slightly more than Llama3. In general, when sufficiently precise data is provided—that is, when the question is complete, does not require guesswork, and the knowledge it refers to is widely available on the Internet—all models generate satisfactory and helpful answers according to the evaluative models. The models tested also perform well in handling questions containing false information, typically informing the user that the statement requires personal confirmation because it is false according to available knowledge (Table I).

TABLE I
PERFORMANCE COMPARISON OF TESTED MODELS IN 3 CATEGORIES

Model/Category	Response Adequacy	Source Attribution	Factual Consistency
LLAMA2:7B	34.78%	48.43%	75.46%
LLAMA3:8B	46.10%	73.28%	81.10%
GEMMA2:9B	60.97%	89.96%	85.16%

Gemma2 outperforms other models in all three evaluation categories: response adequacy, source attribution, and factual consistency. Response adequacy assesses how helpful the response generated by the multi-agent system is to the user, comparing it with a model response prepared by a human. Source attribution checks whether each piece of information in the response is properly annotated with a source, and also assesses whether the format of these references is in accordance with guidelines, i.e. the numbering and list of sources are at the end of the response. Factual consistency, on the other hand, concerns verification whether the response does not contain hallucinations, false information, or information that is not found in the sources provided.

On average, Gemma2 uses about 5500 tokens for the full response generation process, Llama3 about 6300 tokens, while Llama2 uses less than 3000 tokens (this is also due to the smaller context window). Llama2, having a context window half as small as Llama3 and Gemma2, performs much worse and is prone to critical errors that can lead to system failures. Examples of such errors include misunderstanding JSON output formats or misinterpreting instructions, resulting in failure to complete tasks step by step. The limited context window of Llama2 is especially problematic in combination with the Exa tool, which usually delivers long documents, sometimes exceeding four thousand tokens per document.

As can be seen from the analysis presented in the diagram (Fig. 6), the most failure-prone agent turned out to be the Format Validator, responsible for 48.7% of all system problems. The cause of these failures was in most cases endless loops, which resulted from a lack of agreement between the Format Validator, the Source Manager and the Report Author on the source annotations. The Format Validator consistently rejected the generated text for correction, while the Source Manager considered that all information was in accordance with the requirements. Similar problems occurred with the agents responsible for Hallucination Validation (6.2%) and Response Validation (7.9%), where there were also disagreements with the Report Author. These problems also resulted from low quality documents or sources. The Content Validator encountered difficulties in assessing some documents, which occurred in 5.2% of cases. These problems were caused by the inappropriate format of documents or their insufficient number – when the number of documents dropped below three or 80% of the initial value, the agent rejected documents indefinitely. The main agent of the system, the Researcher, also encountered many problems during its work, recorded in as many as 27.7% of cases. He mainly encountered difficulties in working with the tools, especially in the case of the Llama2 model, which was the weakest among the tested ones. In the case of the Researcher, the tools were often used incorrectly, which led to the rapid reaching of the maximum number of attempts to complete the task by a given agent, which was ten. Additionally, some questions still contained many complex assumptions, and the search tools used had difficulties, especially in the case of time-dependent queries. The least failure-prone agent turned out to be the Context Translator, whose role was relatively simple, and the task itself was small, which limited the potential for error. Nevertheless, in 4.3% of cases, problems were noted related to understanding the proper format of the response, especially the JSON format. These problems were particularly noticeable in the case of the Llama2 and Llama3 models. Furthermore, no failures were reported for the User Assistant.

The in-depth failure analysis provides valuable information about the weak points of the system and potential areas for improvement. This allows for a better understanding of the limitations of individual system components and their

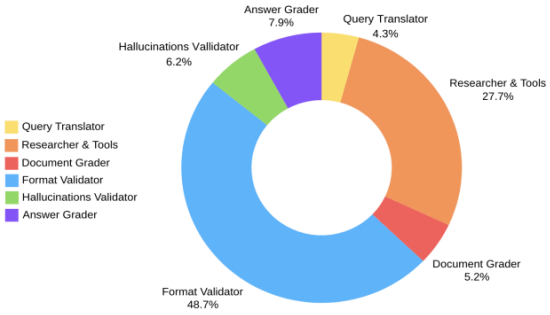

Fig. 6. The most failure-prone agents in multi-agent system created.

interactions, which is crucial for further development and optimization of the created solution. A significant number of failed processes are largely due to the suboptimal configuration of the system for specific LLM models, which accounts for about 44% of all 1920 attempts. This results in infinite loops resulting from the lack of consensus between individual agents and edge cases that lead to system failures. The results of the agent failure analysis emphasize the need for future versions of the system to put more emphasis on experiments in smaller scopes for each agent and their mutual interactions, which will allow for the identification of optimal communication solutions.

The analysis of the most failing agents revealed that most problems occur in the module responsible for the quality of the responses (Format Validator, Hallucinations Validator, Answer Validator) and the module generating the responses (Report Writer, Source Manager), where misunderstandings between agents often lead to deadlocks. The most common failure scenario involves one agent rejecting a generated response and redirecting it, only for the receiving agent to find it acceptable and retransmit the same response to the next agent in the queue, thus creating an unnecessarily iterative and unproductive process. This failure case for multi-agent systems highlights the critical need for more sophisticated agent communication protocols and decision-making mechanisms.

VI. Conclusions

The paper presented proposed approach to automating knowledge acquisition by developing and building a system using autonomous agents with LLMs as their cognitive and operational cores. The created framework represents a significant step forward in the field of information retrieval, offering a new approach to how machines can assist humans in knowledge acquisition and analysis.

Experimental results obtained from testing the system reveal both its significant potential and challenges. Although the system showed promising capabilities in handling a variety of research tasks, it also revealed areas requiring further optimization and improvement. These findings underline the complexity of creating truly autonomous and efficient research assistants and emphasize the continuous need for progress in this field. Solving these challenges could lead to more robust and efficient autonomous systems capable of handling even more complex research tasks.

The developed multi-agent system, although showing promising possibilities, has numerous areas requiring further development. Potential improvements concern both the modernization of key functions and the improvement of the user interface, which can significantly increase the efficiency and usability of the system. The proposed modifications are aimed at developing a comprehensive optimization process that will increase the competences of agents and the entire platform. Thanks to this, the system could become more advanced, versatile and ergonomic, offering more effective support in the analysis and solving of complex research queries.

References

[1] Chen, F., and Ren, W.: On the control of multi-agent systems: A survey. In: Found. Trends Syst. Control., Vol. 6, pp. 339–499 (2019)
[2] Erduran, O. I.: Machine learning algorithms for cognitive and autonomous BDI agents. In: Lernen, Wissen, Daten, Analysen (2022)
[3] Ossowski, S.: Co-ordination in artificial agent societies: Social structures and its implications for autonomous problem-solving agents. (1998)
[4] Grabowski, L. M., Luciw, M. D., and Weng, J.: A system for epigenetic concept development through autonomous associative learning. In: Proceedings of IEEE 6th International Conference on Development and Learning, pp. 175–180 (2007)
[5] Kaber, D. B.: A conceptual framework of autonomous and automated agents. In: Theoretical Issues in Ergonomics Science, Vol. 19, pp. 406–430 (2017)
[6] Geekforgeeks, Learning agents in AI. [Online]. Available: https://www.geeksforgeeks.org/learning-agents-in-ai/, Last accessed: 21.03.2024
[7] Chu, Z., Wang, Y., Zhu, F., Yu, L., Li, L., and Gu, J.: Professional agents – evolving large language models into autonomous experts with human-level competencies. In: ArXiv: 2402.03628 (2024)
[8] Yao, Y., Wang, P.. Tian, R., Cheng, S., Li, Z., Deng, S., Chen, H., and Zhang, N.: Editing large language models: Problems, methods, and opportunities. In: Proceedings of 2023 Conference on Empirical Methods in Natural Language Processing, pp. 10222–10240, ACL (2023)
[9] Xia, H., Yang, Z., Dong, Q., Wang, P., Li, Y., Ge, T., Liu, T., Li, W., and Sui, Z.: Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding. In: Findings of ACL: ACL 2024, pp. 7655–7671 (2024)
[10] Yang, Z., Du, V., Mao, R., Ni, J., and Cambria, E.: Logical reasoning over natural language as knowledge representation: A survey. In: ArXiv: 2303.12023 (2024)
[11] Quere, M. A. L. et al., LLMs as research tools: Applications and evaluations in HCI data work. In: Proceedings of CHI Conference on Human Factors in Computing Systems p. 479, pp. 1–7 (2024)
[12] Nejjar, M., Zacharias, L., Stiehle, F., and Weber, I.: LLMs for science: Usage for code generation and data analysis. In" Software: evaluation and Process (2024)
[13] Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Dou, Z., and Wen, J.: Large language models for information retrieval: A survey. In" ArXiv: 2308.07107 (2024)
[14] GPT researcher, GPT Researcher. [Online]. Available: https://github.com/assafelovic/gpt-researcher/activity, Last accessed: 21.03.2024
[15] Y. Nakajima, Babyagi. [Online]. Available: https://yoheinakajima.com/birth-of-babyagi/, Last accessed: 21.03.2024
[16] Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S.: Reflexion: Language agents with verbal reinforcement learning. In: ArXiv: 2303.11366 (2023)