Bridging Human and AI Decision-Making with LLMs: The RAGADA Approach

Tapio Pitkäranta[1] and Leena Pitkäranta[2]
[1] *Department of Computer Science and Engineering, Aalto University, Finland*
[2] *Department of Industrial Engineering and Management, Aalto University, Finland*

http://www.aalto.fi

Keywords: RAG: Retrieval Augmented Generation, RAGADA: Retrieval Augmented Generation Algorithmic Decision Alignment, LLM Large Language Models, IR Information Retrieval, Multi-Agent Systems (MAS).

Abstract: The Retrieval Augmented Generation Algorithmic Decision Alignment (RAGADA) architecture is an advancement in AI-augmented decision-making for corporate environments. This paper discusses RAGADA's innovative architecture that merges RAG and Multi-Agent System (MAS) with sophisticated business algorithms and dynamic interfaces, enhancing natural language interaction between AI systems and users. This fusion extends AI's reach, facilitating adaptable decision-making tools for leaders, in line with evolving business strategies and ethical standards. Experimental validation of RAGADA within the banking sector, involving diverse stakeholder groups ranging from customers to business and ethical managers, confirms its effectiveness. The system adeptly translates natural language inquiries into actionable insights, thereby improving the user experience and decision-making transparency. This validation underscores RAGADA's potential to transform stakeholder engagement and demonstrates a leap in utilizing AI for strategic and ethical business management.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The recent enhancements of Large Language Models (LLMs) like ChatGPT has marked a significant shift in artificial intelligence (AI), establishing a new paradigm in human-machine interaction. These LLMs have transitioned from academic innovations to a 'killer application in AI' with wide variety of use cases across various industries, offering intuitive and adaptable interfaces for diverse applications. Their advancement in natural language processing has revolutionized machine comprehension, enabling complex dialogue and task execution.

LLMs are evolving beyond sophisticated chatbots to offer a platform that transforms interaction with machines, democratizing AI access for users with varied technical expertise. A pressing research question is aligning these algorithms with human values and objectives (Christian, 2020), highlighting the need for a multidisciplinary approach in AI (Wilson, 1999). AI now merges data science and mathematics with ethics and leadership, ensuring a holistic development and deployment of technologies like LLMs.

This paper presents RAGADA, an innovative software architecture integrating Retrieval Augmented Generation with dynamic, user-friendly interfaces for both customers and executives. RAGADA aims to revolutionize AI systems' user interaction, focusing on natural language processing to enhance customer experience and allow executives to transparently adjust algorithmic decisions, thereby improving user satisfaction and strategic agility in corporate environments.

2 RELATED RESEARCH STUDIES

2.1 LLMs and RAG

The advent of the Transformer model (Vaswani et al., 2017) catalyzed significant advancements in Natural Language Processing (NLP) and Large Language Models (LLM), as seen in models like BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020), and GPT-4. These LLMs have demonstrated remarkable text generation abilities, applicable in various domains including automated decision-making (Brown et al., 2020).

Pitkäranta, T. and Pitkäranta, L.
Bridging Human and AI Decision-Making with LLMs: The RAGADA Approach.
DOI: 10.5220/0012705000003690
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024) - Volume 1*, pages 812-819
ISBN: 978-989-758-692-7; ISSN: 2184-4992
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

The integration of LLMs with Retrieval Augmented Generation (RAG) represents a key development in enhancing NLP and AI. RAG effectively combines pre-trained language models with external knowledge retrieval, significantly improving LLMs' adaptability and accuracy (Karpukhin et al., 2020). It mitigates the constraints of LLMs' static knowledge bases by dynamically incorporating external data, thus boosting the models' contextual relevance (Lewis et al., 2020).

Research underscores the utility of LLMs in complex decision-making (Petroni et al., 2019), and RAG's ability to update these models with current information is especially valuable in dynamic corporate scenarios (Guu et al., 2020), (Borgeaud et al., 2022). The LLM-RAG synergy augments automated decision-making in businesses by addressing challenges in managing real-time, context-specific data.

2.2 AI Safety and Management Literature

The integration of large language models in decision-making processes necessitates careful consideration of ethical aspects (Christian, 2020). Studies have critically examined the risks associated with these models, emphasizing ethical dilemmas, potential biases, and environmental concerns due to their scale and complexity (Bender et al., 2021). The call for a responsible and deliberate approach in the development and application of these models is driven by their extensive societal implications (Bender et al., 2021). This discussion underscores the need for balancing technological progress with ethical responsibility in AI advancements (Christian, 2020).

Applying AI and ML for decision making rationalizes decision making criteria and needs objectives in digital format. Balanced Scorecard is one framework for performance measurement that integrates financial and non-financial metrics (Kaplan, 1992).

In our study, we provide details on an experimental implementation concerning decision-making within the banking sector. The implementation in the banking sector, particularly from an organizational capability perspective, is further elaborated in (Dash et al., 2021).

2.3 Metadata Management and Data Catalogues

Effective metadata management is crucial for organizations handling extensive data assets (Olesen-Bagneux, 2023). It consolidates data storage, search, and management, thereby facilitating efficient access and comprehension of organizational data. By employing machine learning and AI, these systems enhance data governance and quality through automated classification and improved dataset accessibility. Such tools play a key role in democratizing data within organizations, ensuring regulatory compliance and preserving data integrity with transparent audit trails. As data complexities grow, the strategic implementation of data catalogs becomes increasingly important for maximizing data resource utilization and meeting regulatory standards.

A wide array of both open-source and commercial data catalog software is available. Examples of open-source software include OpenMetadata: https://open-metadata.org/, Apache Atlas: https://atlas.apache.org/, Datahub: https://datahub.io/, Metacat: https://github.com/Netflix/metacat, Amundsen: https://www.amundsen.io/, and CKAN: https://ckan.org/.

2.4 Multi-Agent Systems

Multi-agent systems (MAS) are collections of autonomous, interacting agents that work together to solve complex problems that are beyond the capabilities of individual agents acting alone (Wooldridge, 2009). Research in MAS covers a wide range of topics including coordination mechanisms, communication protocols, and collective decision-making strategies. Notable advancements in the field have demonstrated that MAS can effectively handle dynamic and uncertain environments, which are common in real-world scenarios such as traffic management, robotic teams, and distributed control systems (Stone and Veloso, 2000).

One of the fundamental challenges in MAS is the design of cooperation strategies that enable agents to work together harmonously. This requires sophisticated negotiation protocols and conflict resolution techniques, which are crucial for maintaining coherent behavior among agents (Rosenschein and Zlotkin, 1994).

2.5 Agent Interaction Protocols and Chat Templates

Software agents adhere to interaction protocols, notably those established by FIPA standards (http://www.fipa.org/repository/ips.php3), focusing on communication-centric design for context-aware responses. Similarities are evident in the way LLM-based chatbots manage conversational flow. They blend preset and adaptive responses, evolving with user preferences and external data integration, which results in enhanced accuracy and improved customer interaction experiences.

3 METHODOLOGY AND SYSTEM DESIGN

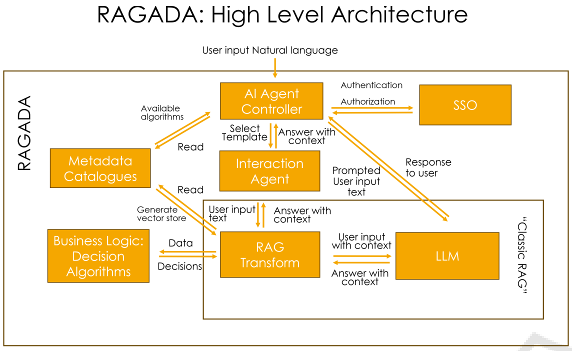


Figure 1: RAGADA High Level Architecture: interactions between human input and software components.

This section describes Innovative Software Architecture for Retrieval Augmented Generation Algorithmic Decision Alignment (RAGADA).

3.1 Architecture: Interaction Model

The Figure 1 describes the high-level components of the RAGADA architecture. In this section we discuss the main roles and interactions of the components in this architecture.

In the RAGADA architecture, the AI Agent Controller plays a pivotal role in orchestrating interactions between users and the decision-making logic. It begins by assimilating the initial text provided by the user and then identifies the specific chat template or interaction protocol the user is engaging with. The AI Agent Controller carefully selects the most suitable interaction agent, based on the user's context and intent, and ensures that the user has the necessary permissions for it. This careful selection is crucial for ensuring that the system's responses are coherent and consistent, adhering to a uniform interaction paradigm.

In the RAGADA architecture, the Interaction Agent serves as the central conversational AI interface that facilitates the dialogue between the human user and the system. This agent is responsible for interpreting user input in natural language and generating appropriate responses that are contextually relevant to the user's requests or inquiries. By leveraging a selection template and a repository of available algorithms, the Interaction Agent dynamically constructs replies that not only provide information but also guide the user through the system's functionalities. The Interaction Agent can handle a variety of inputs, transforming user queries into structured data that can be processed by the Business Logic: Decision Algorithms. Moreover, it ensures that user interactions are intuitive and efficient, enhancing user engagement and satisfaction with the system.

The middle level components in this structure are RAG Transform and the Language Model (LLM), key components of the classic Retrieval-Augmented Generation (RAG) model that elevates response quality by weaving in external data. The RAG Transform is pivotal, acting as the analytic engine where user inputs undergo initial processing and enhancement. It ingests raw user queries and augments them with context derived from the Catalogues—a repository replete with domain-specific data. This context-enrichment empowers the RAG Transform to interpret the queries with precision, significantly refining the system's output in terms of relevance and specificity.

Next to the RAG Transform resides the Language Model (LLM), presumably a Large Language Model akin to GPT. The LLM employs the enriched inputs from the RAG Transform to create nuanced and contextually informed responses. It taps into a vast reserve of learned patterns to formulate replies that exhibit a high degree of linguistic finesse, akin to human conversation.

The system's decision-making prowess is further enhanced by integrating Business Logic with Decision Algorithms, which operate in concert with the RAG components. They leverage both the foundational data and the context processed by the AI to implement domain-specific directives and heuristics. These algorithms are indispensable in scenarios that necessitate not just information retrieval but also autonomous decision-making, such as in financial services or healthcare.

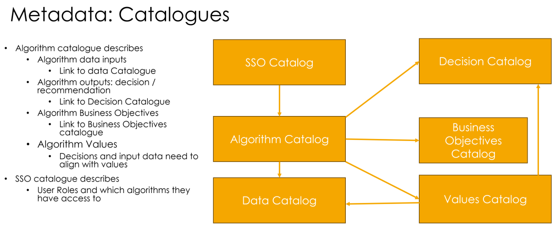
In essence, RAGADA's architecture is a harmonious fusion of traditional RAG elements with cutting-edge AI decision-making mechanisms. It combines the RAG Transform's contextual acuity, the LLM's generative capabilities, and the AI Agent Controller's ability to select the right interaction agent, all underpinned by robust Business Logic. This confluence results in a system adept at addressing complex user inquiries with remarkable precision and efficiency, positioning RAGADA as a formidable tool across a multitude of automated, intelligent interaction domains.

A very important component is the catalogue,

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

which consists of multiple sub-components and will be discussed in the subsequent section. The catalogue serves both as an input for the RAG Transform and as a modeling instrument to bridge the concepts.

3.2 Architecture: Catalogue Layer


Metadata: Catalogues

- Algorithm catalogue describes
  - Algorithm data inputs
    - Link to data Catalogue
  - Algorithm outputs: decision / recommendation
    - Link to Decision Catalogue
  - Algorithm Business Objectives
    - Link to Business Objectives catalogue
  - Algorithm Values
    - Decisions and input data need to align with values
- SSO catalogue describes
  - User Roles and which algorithms they have access to

Figure 2: RAGADA Architecture: Metadata catalogue relationships and high level description.

The depicted Figure 2 illustrates an interconnected framework of metadata catalogues that serve as the backbone for a data-driven decision-making system. This systematic arrangement ensures that every step in the algorithmic process is underpinned by a structured repository of information, spanning from raw data to core business values.

Central to this framework is the 'Data Catalog', which meticulously details the datasets available for use. This catalogue is instrumental in providing the foundational data inputs to the 'Algorithm Catalog', which, in turn, specifies the algorithms that will process the input data. The Algorithm Catalog is a comprehensive directory that not only lists the algorithms but also delineates their data requirements, expected outputs, and the relevant business objectives they are designed to achieve. It ensures a seamless flow of information by creating explicit links to the Data Catalog and the Decision Catalog.

The 'Decision Catalog' is a critical component that collates the various decisions or recommendations generated by the algorithms. It acts as a reference point for the end outputs of the decision-making process, encapsulating the inferences drawn from the data through the algorithms. This catalogue is directly informed by the outputs of the Algorithm Catalog, ensuring that decisions are traceable back to their algorithmic origins.

Moreover, the 'Business Objectives Catalog' aligns the algorithmic decisions with the strategic goals of the organization. It provides a clear linkage between the algorithms' functions and the overarching objectives they are meant to serve, guaranteeing that algorithmic outputs are not only data-driven but also strategically focused.

Lastly, the 'Values Catalog' upholds the ethical and value-based considerations that should guide every decision. This catalogue ensures that the data inputs, algorithmic processes, and business objectives are all in alignment with the organization's core values. It acts as a moral compass, steering the decision-making process towards outcomes that are not only effective but also ethically sound.

Together, these catalogues create a robust and ethically-grounded architecture that integrates diverse sets of information into a coherent system, designed to facilitate informed and value-aligned decision-making in a corporate environment.

4 EXPERIMENTAL VALIDATION

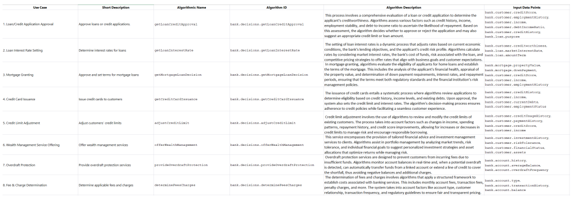
We are not aware of any standard conversational AI RAG benchmark dataset suitable for evaluating RAGADA architecture. Our research evaluates the RAGADA model, specifically within the banking sector, known for its complex decision-making processes. We also compare the performance to fine tuned GPT-4 and custom GPT-4 implementations.

Experimental Implementation. The experimental validation of RAGADA within the banking sector involved a series of user stories designed to assess the system's decision-making capabilities. We constructed a comprehensive algorithm catalogue from banking sector use cases portion of which is illustrated in Figure 3.

The experimental phase focused on the detailed exploration of the getLoanDecision algorithm, a pivotal component of our algorithmic catalogue, as depicted in Figure 3. This choice was made to thoroughly illustrate the customer interaction process from multiple stakeholder perspectives. The 'getLoanInterestRate' algorithm stands out due to its complexity in balancing diverse inputs, such as credit scores and market rates, and its direct impact on customer satisfaction and financial product performance. Our implementation simulated various real-world scenarios to evaluate the algorithm's robustness and adaptability, scrutinizing its decision-making process in dynamically adjusting interest rates in response to fluctuating economic indicators and customer credit profiles.

4.1 Focus on One Algorithm

Focusing on a single decision algorithm, we explored the model's capabilities in a controlled yet intricate environment, representing broader banking decision frameworks. This approach facilitated an in-depth



Use Case	Short Description	Algorithmic Name	Algorithm ID	Algorithm Description	Input Data Points

1. Loan/Credit Application Approval	Approve loans or credit applications.	`getLoanCreditApproval`	`bank.decisions.getLoanCreditApproval`	This process involves a comprehensive evaluation of a loan or credit application to determine the applicant's creditworthiness. Algorithms assess various factors such as credit history, income, employment stability, and debt-to-income ratio to ascertain the likelihood of repayment. Based on this assessment, the algorithm decides whether to approve or reject the application and may also suggest an appropriate credit limit or loan amount.	`bank.customer.creditScore,`
					`bank.customer.employmentHistory,`
					`bank.customer.income,`
					`bank.customer.debtIncomeRatio,`
					`bank.customer.creditHistory,`
					`bank.loan.purpose`

2. Loan Interest Rate Setting	Determine interest rates for loans	`getLoanInterestRate`	`bank.decisions.getLoanInterestRate`	The setting of loan interest rates is a dynamic process that adjusts rates based on current economic conditions, the bank's lending objectives, and the applicant's credit risk profile. Algorithms calculate rates by considering market interest rates, the bank's cost of funds, risk associated with the loan, and competitive pricing strategies to offer rates that align with business goals and customer expectations.	`bank.customer.creditworthiness,`
					`bank.loan.marketInterestRate,`
					`bank.loan.amountTerm`

3. Mortgage Granting	Approve and set terms for mortgage loans	`getMortgageLoanDecision`	`bank.decisions.getMortgageLoanDecision`	In mortgage granting, algorithms evaluate the eligibility of applicants for home loans and establish the terms of the mortgage. This includes the analysis of the applicant's financial health, appraisal of the property value, and determination of down payment requirements, interest rates, and repayment periods, ensuring that the terms meet both regulatory standards and the financial institution's risk management policies.	`bank.mortgage.propertyValue,`
					`bank.mortgage.downPayment,`
					`bank.customer.creditScore,`
					`bank.customer.income,`
					`bank.customer.employmentHistory`

4. Credit Card Issuance	Issue credit cards to customers	`getCreditCardIssuance`	`bank.decisions.getCreditCardIssuance`	The issuance of credit cards entails a systematic process where algorithms review applications to determine eligibility based on credit history, income levels, and existing debts. Upon approval, the system also sets the credit limit and interest rates. The algorithm's decision-making process ensures adherence to credit policies while facilitating a seamless customer experience.	`bank.customer.creditHistory,`
					`bank.customer.income,`
					`bank.customer.currentDebts,`
					`bank.customer.employmentStatus`

5. Credit Limit Adjustment	Adjust customers' credit limits	`adjustCreditLimit`	`bank.decisions.adjustCreditLimit`	Credit limit adjustment involves the use of algorithms to review and modify the credit limits of existing customers. The process takes into account factors such as changes in income, spending patterns, repayment history, and credit score improvements, allowing for increases or decreases in credit limits to manage risk and encourage responsible borrowing.	`bank.customer.creditUsageHistory,`
					`bank.customer.paymentHistory,`
					`bank.customer.creditScore,`
					`bank.customer.income`

6. Wealth Management Service Offering	Offer wealth management services	`offerWealthManagement`	`bank.decisions.offerWealthManagement`	This service encompasses the provision of tailored financial advice and investment management services to clients. Algorithms assist in portfolio management by analyzing market trends, risk tolerance, and individual financial goals to suggest personalized investment strategies and asset allocations that optimize returns while managing risk.	`bank.customer.investmentHistory,`
					`bank.customer.riskTolerance,`
					`bank.customer.financialStatus,`
					`bank.customer.assets`

7. Overdraft Protection	Provide overdraft protection services	`provideOverdraftProtection`	`bank.decisions.provideOverdraftProtection`	Overdraft protection services are designed to prevent customers from incurring fees due to insufficient funds. Algorithms monitor account balances in real-time and, when a potential overdraft is detected, can automatically transfer funds from a linked account or extend a line of credit to cover the shortfall, thus avoiding negative balances and additional charges.	`bank.account.history,`
					`bank.account.averageBalance,`
					`bank.account.overdraftFrequency`

8. Fee & Charge Determination	Determine applicable fees and charges	`determineFeesCharges`	`bank.decisions.determineFeesCharges`	The determination of fees and charges involves algorithms that apply a structured framework to establish costs associated with banking services. This includes monthly account fees, transaction fees, penalty charges, and more. The system takes into account factors like account type, customer relationship, transaction frequency, and regulatory guidelines to ensure fair and transparent pricing.	`bank.account.type,`
					`bank.account.transactionHistory,`
					`bank.account.balance`

Figure 3: Example algorithm catalogue.

analysis and provided insights into stakeholder interactions within real-world banking scenarios.

In this paper, we will explain how we integrated the 'getLoanDecision' algorithm into the Algorithm Catalog, along with the necessary additional catalog definitions as defined in Figure 2, and how stakeholders interact with the algorithm. Although we created several other similar algorithms for the catalog (getCreditCardDecision, getBankAccountDecision, getMortgageLoanDecision, etc.), we will not focus on those in this paper.

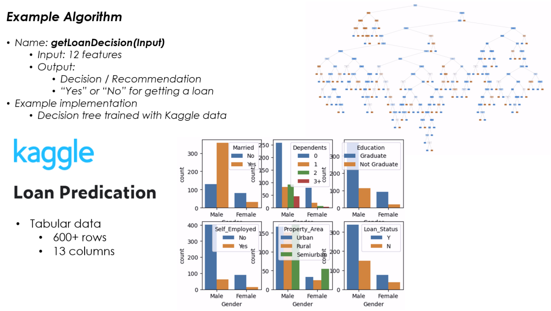
We assume that the bank employs role-based authentication and authorization for users, and we will omit these aspects from this experiment.

The study's next section examines user stories from bank employees, customers, and management, revealing how each group interacts with and perceives the algorithm. This analysis aims to showcase the algorithm's functionality and its impact on diverse stakeholders, underscoring RAGADA's potential to transform decision-making in complex organizational contexts.

4.2 User Stories

User stories used in this implementation:

- As a bank user, I would like to apply for a short-term loan.
- As a bank user, I would like to complain to the ethical value alignment algorithm that I am being asked ethically questionable information while applying for a loan.
- As a bank compliance department manager, I would like to remove ethically questionable data points from loan decision making.
- As a bank loan business department manager, I would like to change the business target for short-term loan decisions from minimizing credit problems to acquiring new customers.



Example Algorithm

- *Name: **getLoanDecision(Input)***
  - *Input: 12 features*
  - *Output:*
    - *Decision / Recommendation*
    - *"Yes" or "No" for getting a loan*
- *Example implementation*
  - *Decision tree trained with Kaggle data*

kaggle

Loan Predication

- Tabular data
  - 600+ rows
  - 13 columns

Figure 4: Example Algorithm: getLoanDecision(Input): Decision.

4.3 Example Algorithm: getLoanDecision(Input): Decision

In the modern corporate environment, particularly within large organizations, the plethora of decision-making processes constitutes a complex landscape, often involving hundreds or thousands of decisions ripe for algorithmic transformation. Our research delved into this realm, focusing on the experimental application of algorithms in a practical setting. We developed 'getLoanDecision', an algorithm tailored to streamline the loan approval process. This algorithm processes various customer profile inputs to generate loan decisions.

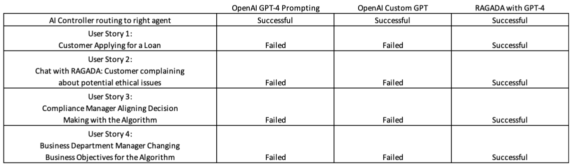
For a realistic implementation, we utilized a Kaggle dataset specifically designed for bank loan scenarios (https://www.kaggle.com/datasets/ninzaami/loan-predication/data). This dataset, encompassing a wide range of customer scenarios typical in the financial sector, served as the training ground for our algorithm, ensuring it reflects real-world banking decision-making.

We trained a decision tree logic from the data, as illustrated in Figure 4, along with some exploratory analysis. We opted for decision trees because their logic can be relatively easily understood by humans, which aids in analyzing ethical compliance and understanding how the algorithm aims to achieve the business targets.

4.4 RAGADA Experimental Implementation

In our research, we combined advanced technological tools and programming languages to operationalize our algorithm and user stories. We used Apache Atlas for data management and cataloging, a tool crucial for maintaining data integrity and compliance in the banking sector. For processing at the Retrieval Aug-

SCIENCE AND TECHNOLOGY PUBLICATIONS



	OpenAI GPT-4 Prompting	OpenAI Custom GPT	RAGADA with GPT-4
AI Controller routing to right agent	Successful	Successful	Successful
User Story 1: Customer Applying for a Loan	Failed	Failed	Successful
User Story 2: Chat with RAGADA: Customer complaining about potential ethical issues	Failed	Failed	Successful
User Story 3: Compliance Manager Aligning Decision Making with the Algorithm	Failed	Failed	Successful
User Story 4: Business Department Manager Changing Business Objectives for the Algorithm	Failed	Failed	Successful

Figure 5: Comparison between: Propted LLM, Custom GPT and RAGADA.

mented Generation (RAG) level, Python Langchain was employed, known for its effectiveness in handling complex language models. This choice allowed for nuanced interpretation of language patterns, enhancing user-algorithm interaction. Additionally, Python Scikit-learn was utilized for training the algorithm, selected for its comprehensive data analysis tools and user-friendly interface. Its capability to manage large datasets and diverse algorithms ensured our model's accuracy and adaptability, aligning with our objective to develop an efficient, user-friendly, and adaptable algorithm.

4.5 RAGADA Comparison

The performance of RAGADA, integrated with GPT-4, was compared against other AI configurations, including those OpenAI's GPT models. The results, as summarized in Figure 5, illustrate RAGADA's superior performance, achieving successful outcomes across various scenarios. These ranged from customer interactions, such as loan applications and complaints about ethical considerations, to internal managerial decisions affecting the system's business logic. The success of RAGADA in these user stories confirms its potential to revolutionize decision-making processes by leveraging advanced natural language understanding and contextually aware AI interactions.

4.6 Role Playing Users Stories with the System

In developing our system, we focused on natural language dialogue for intuitive interaction, ensuring user-friendliness across various technical skills. Comprehensive testing confirmed the interface's effectiveness and adaptability, demonstrating the system's ability to dynamically respond to diverse user inputs and offer tailored solutions in complex scenarios. Next we will go through a couple of previously described user stories as chat prompts with our RAGADA.

4.7 Discussion: Customer Applying for a Loan

Client: I would like to apply for a short term loan.

RAGADA: May I confirm that you are interested in applying for a short-term loan and wish to proceed with this application? The interest rate for this loan would be 12 month EURIBOR + 0.5 percent as bank margin. I can make an automatic decision once I receive the necessary input data from you. First, I will verify the information we have on file from the CRM system to ensure it is current.

Client: Sounds good!

RAGADA: Certainly, based on our CRM system, here are the details we have on record for your loan application: 1.) Gender: Male 2.) Married: Yes 3.) Education: Bachelor's Degree 4.) SelfEmployed: No 5.) ApplicantIncome: \$4,500 per month 6.) CreditHistory: Yes (indicating a positive credit history) 7.) PropertyArea: Zip code 10007 (New York)
Please confirm if these details are accurate and up-to-date.

Client: These are correct!

RAGADA: Excellent, thank you for confirming. Now, let's proceed with collecting the remaining information needed for your loan application. Could you please provide the following details? 1.) Dependents: How many individuals rely on your income for support? 2.) CoapplicantIncome: If applicable, what is the monthly or annual income of the co-applicant? 3.) LoanAmount: How much are you looking to borrow? 4.) LoanAmountTerm: Over how many months or years do you wish to repay this loan?
Once you provide this information, we'll be able to complete your loan application process.

Client: Here is the information: 1.) Dependents: I have one boy who is seven years old 2.) CoapplicantIncome: my wife makes \$50,000 per year 3.) LoanAmount: I need \$27,500 for new Tesla Model 3 car 4.) LoanAmountTerm: We were thinking three years.

RAGADA: Thank you for providing the necessary details. I have now updated your loan application with the following information:
1.) Dependents: 1 (a seven-year-old boy)

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

2.) CoapplicantIncome: $50,000 per year (as contributed by your wife) 3.) LoanAmount: $27,500 (for the purchase of a new Tesla Model 3 car) 4.) LoanAmountTerm: 3 years Could you confirm this data is correct? After that I will now call the automatic decision logic with this information.

Client: Yes, information is correct.

RAGADA: We are pleased to inform you that your loan application has been successfully processed and approved through our advanced automatic decision-making algorithm. As a result of this approval, the requested funds have been deposited directly into your bank account. Should you have any questions or require further assistance, please do not hesitate to contact us.

Client: Thank you! This was excellent customer service!!!

4.8 Discussion: Customer Complaining About Potential Ethical Issues

Client: I was recently granted a short term loan. Thank you for that. However, what bothers me is that it seems you also used my address and Zip code in the decision making. Could you confirm this.

RAGADA: Yes, I can confirm that our Bank Automatic Algorithm Decision Catalogue mentions that Zip Code is used as an input for short term loan decision making.

Client: I think there are ethical problems using ZIP code in the automatic decision making. Zip code might contain racial signals and discriminating people who apply for a loan based on their race.

RAGADA: Thank you for the comment. In our bank algorithms are not allowed to change input datasets to decision making automatically. Would you like me to post this ethical concern to our ethical committee to process?

Client: Yes. Can you also ask your committee to let me know what they decided?

RAGADA: Of course! I have now created a ticket for the ethical committee and added you into the distribution list of the outcome.

Client: Thank you!

4.9 Discussion: Compliance Manager Aligning Decision Making with the Algorithm

Compliance Manager: The short-term loan algorithm currently uses the customer's Zip code in the decision-making process. This data point cannot be used due to potential ethical issues. Please remove it from the decision-making process.

RAGADA: I have added a filter to the 'getLoanDecision' algorithm that excludes the Zip code from the input parameters. This change has been documented as an audit trail in the Algorithm Catalog. Currently, I am not authorized to retrain the model automatically. If you have concerns about the impact of this change, I can create a ticket for our data science department to analyze the effect.

Compliance Manager: Thank you. Please proceed with that and include me in the ticket so I can be informed about the outcome.

RAGADA: Done!

4.10 Discussion: Business Department Manager Changing Business Objectives for the Algorithm

Business Department Manager: I have changed the business objectives of the short-term loan algorithm from minimizing credit losses to including a weighted customer acquisition objective. Could you retrain the model?

RAGADA: Currently, I am not authorized to retrain the model automatically. However, I can create a ticket for our data science department to analyze the effect.

Business Department Manager: Thank you. Please proceed with that and include me in the ticket so I can be informed about the outcome.

RAGADA: Done!

5 DISCUSSION

Our research on the RAGADA model marks a significant advancement in bridging gap between human stakeholders communicating with natural language with automatic algorithmic decision-making.

Focusing on the banking 'getLoanDecision' algorithm, we've unveiled the complexities of algorithmic decisions and different stakeholders in finance. This model handles diverse scenarios, from loan applications to ethical dilemmas, exemplifying its ability to streamline customer service and operational efficiency. However, it also raises ethical issues, like the use of ZIP codes in decision-making, highlighting the importance of ethics in algorithm development. Addressing such concerns proactively, including referring them to an ethical committee, demonstrates RAGADA's potential in managing sensitive matters, crucial for trust and transparency in banking. Overall, RAGADA's nuanced decision-making approach offers promising prospects for evolving financial services.

6 CONCLUSION AND FUTURE WORK

This research introduces and validates the RAGADA model, a novel AI-based software architecture enhancing corporate decision-making. Successfully implemented in the banking sector, RAGADA leverages natural language interfaces to facilitate complex decisions, demonstrating significant advancements in AI and corporate governance. It has shown potential in improving decision accuracy, user satisfaction, and strategic adaptability. Future exploration includes extending RAGADA's applicability to various sectors and enhancing its capabilities for more complex scenarios. Emphasizing ethical AI and user-centric design, the model presents a substantial progression in AI-integrated corporate decision-making, offering a blueprint for future advancements in AI-augmented governance.

REFERENCES

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G. B., Lespiau, J.-B., Damoc, B., Clark, A., et al. (2022). Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Christian, B. (2020). *The alignment problem: Machine learning and human values*. WW Norton & Company. ISBN: 9781786494306.

Dash, R., Kremer, A., and Petrov, A. (2021). Designing next-generation credit-decisioning models. *McKinsey & Company, White paper, Dec.*

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. (2020). Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Kaplan, R. S. (1992). *The balanced scorecard measures that drive performance*. Harvard business review.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Olesen-Bagneux, O. (2023). *Enterprise Data Catalog*. O'Reilly Media. ISBN: 149209871X.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. (2019). Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Rosenschein, J. S. and Zlotkin, G. (1994). *Rules of encounter: designing conventions for automated negotiation among computers*. MIT press.

Stone, P. and Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8:345–383.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wilson, E. O. (1999). Consilience: The unity of knowledge. *Vintage; Reprint edition*. ISBN: 978-0679768678.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.