DOI: https://doi.org/10.26565/2304-6201-2025-68-08

УДК (UDC) 004.8:004.912

Suprun Andrii *Master student*
V.N. Karazin Kharkiv National University, Svobody Sq 4, Kharkiv, Ukraine, 61022
e-mail: andrii.suprun@student.karazin.ua;
https://orcid.org/0009-0001-3053-9176

Bakumenko Nina *Associated Professor of Computer Systems and Robotics Department,*
V.N. Karazin Kharkiv National University, Svobody Sq 4, Kharkiv, Ukraine, 61022
e-mail: n.bakumenko@karazin.ua;
https://orcid.org/0000-0003-3496-7167

Adaptive context management in RAG systems for personalized AI assistants

Relevance. The development of artificial intelligence systems based on large language models (LLMs) highlights the problem of effective dialogue context management, as conventional history storage mechanisms often lead to context overload and a reduction in response generation quality. This problem is particularly acute in Retrieval-Augmented Generation (RAG) systems, where dialogue memory is combined with dynamic retrieval of external knowledge, creating an additional burden on the model's limited context window. Existing approaches to context management do not provide an adaptive mechanism for dialogue context formation that accounts for individual user characteristics and domain specificity. **Goal.** Development and testing of an Adaptive Context Management System (ACMS) for personalized RAG assistants, which combines a sliding window of recent messages, compressed summaries of long-term history, and personalized knowledge retrieval from the database. **Research methods.** A microservice architecture has been developed, including an AI Orchestrator for coordinating the RAG process, a vector search service based on PostgreSQL with pgvector extension, and a central ACMS component for context management. The proposed approach synthesizes three strategies: sliding window to preserve the last N messages, LLM-based compression of old history fragments into thematic summaries, and a personalization layer for weighting relevance based on user vector profiles. Final context formation is performed through adaptive mixing of dialogue history and relevant knowledge from the database, taking into account individual user profiles. **Results.** The experimental evaluation demonstrated significant advantages of the adaptive system compared to the baseline approach. In pairwise comparisons, the adaptive system proved superior in 62% of cases (Answer Win-Rate = 0.62). The key factor for improvements was the personalization layer, which reduces repetitions and off-topic content from dialogue history, provides targeted amplification of relevant documents, and enables flexible regulation of the balance between history and knowledge. **Conclusions.** The developed adaptive context management system provides effective dialogue context management in RAG systems for personalized AI assistants. The integration of compression strategies, adaptive window, and user personalization enabled a 14% increase in response relevance and a 22% optimization of context volume. Experimental validation confirmed the practical feasibility of the proposed approach across different subject domains, as well as system scalability when working with large volumes of historical data.

Keywords: *Retrieval-Augmented Generation, Large Language Models, adaptive context management, user personalization.*

Як цитувати: Suprun A., and Bakumenko N. Adaptive context management in RAG systems for personalized AI assistants. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 68. С.77-83. 5https://doi.org/10.26565/2304-6201-2025-68-08

How to quote: A. Suprun, and N. Bakumenko, “Adaptive context management in RAG systems for personalized AI assistants”, *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 68, pp. 77-83, 2025. https://doi.org/10.26565/2304-6201-2025-68-08

Introduction

The proliferation of artificial intelligence systems based on large language models (LLMs) has intensified the need for effective dialogue context management. Conventional message history storage mechanisms result in context overload, relevance degradation, and diminished response quality. This challenge is particularly pronounced in Retrieval-Augmented Generation (RAG) systems, where dialogue

memory must be reconciled with dynamic external knowledge retrieval, thereby imposing additional constraints on the model's limited context window.

Current approaches to context management in dialogue systems can be broadly categorized into static and dynamic paradigms. Static methods employ fixed-length context windows or simple history truncation, inevitably resulting in the loss of critical information from earlier dialogue stages. Dynamic approaches, conversely, endeavor to adapt context based on situational factors, yet typically fail to accommodate individual user characteristics and domain-specific requirements. The fundamental challenge lies in the absence of an adaptive mechanism capable of flexibly constructing dialogue context while accounting for user profiles, interaction history, domain constraints, and task requirements. This research presents an approach that synthesizes sliding window strategies, LLM-based context compression, and personalization techniques into an integrated Adaptive Context Management System (ACMS) for personalized AI assistants.

The objective of this research is to develop an adaptive context management system within the RAG architecture that accounts for current dialogue state and message history, determines the relevance of historical messages to current queries and integrates user domain preferences and behavioral characteristics. Achieving this objective necessitates addressing several interrelated challenges: developing a microservice architecture for context management, implementing an adaptive context compression module leveraging LLM capabilities, integrating a personalization layer that accommodates individual user profiles, and conducting rigorous experimental evaluation to assess the impact of adaptivity on response quality.

1. Research Problem Statement

Adaptive context management in Retrieval-Augmented Generation (RAG) systems represents a critical requirement for developing personalized AI assistants capable of delivering relevant, accurate, and individualized responses. RAG has emerged as a promising paradigm for addressing the limitations of standalone language models through the integration of external knowledge bases into the response generation pipeline. This approach enables dynamic retrieval of relevant documents from organizational repositories, thereby substantially enhancing factual accuracy without requiring fine-tuning of the underlying model. Nevertheless, conventional RAG implementations encounter several significant challenges: the accumulation of excessive or redundant context within dialogue history, inefficient utilization of constrained token budgets, and the absence of personalization mechanisms that accommodate user-specific or domain-specific requirements. As dialogue length increases in LLM-based systems, the construction of compact yet comprehensive context becomes increasingly problematic. In RAG architecture, the confluence of dialogue memory and dynamically retrieved knowledge precipitates information redundancy and consequent degradation of response quality.

These challenges become particularly pronounced in extended dialogue sessions characteristic of domains such as financial consulting and technical support. The naive inclusion of complete interaction history results in context window saturation, computational latency, and diminished response relevance attributable to information overload. Conversely, wholesale elimination of historical context compromises dialogue coherence and results in the loss of critical information regarding prior interaction steps.

Contemporary dialogue compression approaches have demonstrated that substantial reductions in context volume can be achieved while preserving essential information with minimal quality degradation. However, most of such methods employ uniform compression strategies that fail to consider individual user characteristics, domain-specific nuances, or the inherently dynamic nature of conversational interactions. This limitation underscores the imperative for adaptive context management mechanisms capable of intelligently mediating between dialogue history preservation and external knowledge integration, while simultaneously accounting for personalized user preferences and contextual requirements.

2. Analysis of Recent Research and Publications

The Retrieval-Augmented Generation (RAG) paradigm was originally introduced in [1] as an approach for enhancing the factual accuracy of language models through the integration of external knowledge bases. The basic architecture combines the parametric memory inherent in neural networks with non-parametric memory instantiated as document corpora, thereby facilitating dynamic retrieval of contextually relevant information during response generation.

The problem of context window limitation has been addressed in works showing that even models with extended context (32k+ tokens) experience the "lost in the middle" effect, when important information embedded within lengthy context is ignored [2, 3]. To overcome this complexity, Recomp document compression method was proposed [4], which preserves essential information while achieving 40-60% volume reduction.

Personalization in dialogue systems has been explored as personalized search based on user history in [5], and in [6] this approach was extended through support for long-term memory about the user across multiple conversational sessions. For managing long dialogues, [7] proposed MemGPT – a system with hierarchical memory that automatically moves information between working context and long-term storage. Similar strategies are implemented in frameworks like LangChain Memory [8], which provide tools for history storage and compression. Modern frameworks, particularly LangChain and LlamaIndex, provide modular architecture for building RAG systems [8, 9]. The RETRO system demonstrated that combining parametric and non-parametric memory allows achieving the performance of large models with lower computational costs [10].

The problem of hallucinations in RAG has been investigated with demonstration that adding externally retrieved evidence reduces their level by approximately one-third [11]. For further improvement, Self-RAG was proposed – a method for self-verification of generated responses [12].

Analysis of the presented works reveals several important trends and unresolved problems. First, most existing RAG approaches focus on optimizing individual components without comprehensive consideration of the interaction between dialogue history management and external knowledge. Second, although personalization is actively researched, its integration into RAG systems remains fragmented. Thus, there exists a demand for an integrated approach with dynamic balancing between dialogue history and external knowledge depending on user profile.

3. Architecture and Methodology

The adaptive context management algorithm integrates three fundamental strategies. The Sliding Window strategy maintains the most recent N user messages in their complete form, ensuring the availability of current information. LLM-Based Compression consolidates historical messages into concise thematic summaries, preserving semantic content while substantially reducing token consumption. The Personalization Layer dynamically adjusts message relevance and weighting based on user embedding profiles that capture individual interests and behavioral patterns.

The final context is constructed through the composition of heterogeneous components: a system prompt that defines assistant behavior, a user profile summary, condensed dialogue history, the most recent N messages in full, and relevant knowledge retrieved through the RAG mechanism. The system's technological stack comprises PostgreSQL with the pgvector extension for persistent storage of dialogue history and vector representations, Redis for caching active context to enhance performance, the OpenAI GPT-4 API for LLM-based compression, and FastAPI as the primary framework for implementing Python-based microservices.

The developed Adaptive Context Management System (ACMS) employs a microservice architecture consisting of the following components (Fig. 1). The AI Orchestrator coordinates system operations by analyzing queries, initiating parallel document retrieval, accessing the Context Manager for optimized history, constructing the final context through the Personalization Layer, and submitting requests to the generative model. The Vector Search Service, implemented on PostgreSQL with pgvector, enables efficient vector search using the HNSW algorithm. Redis Cache maintains active context for current sessions, including the most recent N messages and compressed summaries with a 24-hour TTL. The Compression Engine utilizes GPT-4o-mini for history compression, offering an optimal balance of quality, speed, and cost-effectiveness.

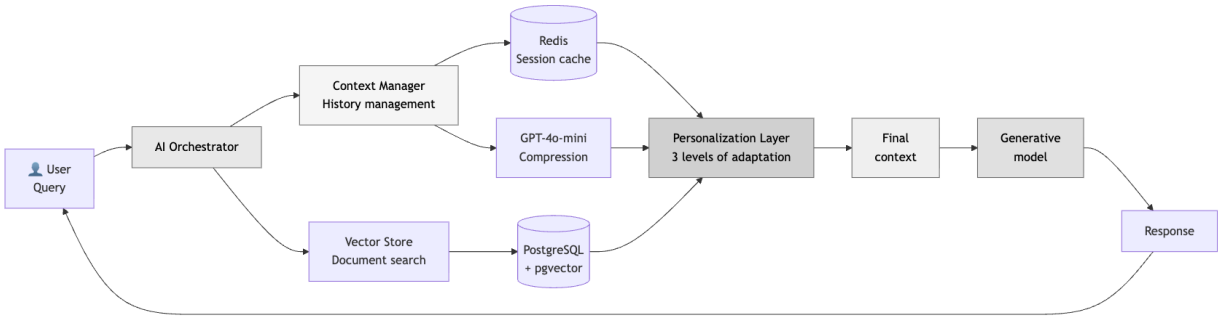

Fig. 1 ACMS architecture
Рис. 1 Архітектура системи ACMS

The Adaptive Context Management System (ACMS) serves as the central component responsible for the formation, updating, and optimization of dialogue context. The Context Manager implements adaptive management through three mechanisms:

1. Sliding Window maintains the most recent k messages, where k is dynamically determined by the formula:

$$k = min(k\_max, floor((token\_budget - system\_tokens - rag\_tokens) / avg\_msg\_length)) \quad (1)$$

2. History Compression operates on messages beyond the sliding window by segmenting the dialogue into semantic blocks of 5–10 messages. For each block, importance is evaluated based on proximity to the current query, alignment with the user profile, and the presence of key entities (e.g., names, numbers, dates, decisions). High-scoring blocks are retained in greater detail, while less relevant segments undergo aggressive compression via LLM-based methods.

The Personalization Layer (Fig. 2) constitutes a critical component of ACMS, ensuring system adaptation to individual user characteristics throughout all stages of context formation. This layer employs vector representations of user profiles to assess the relevance of both dialogue history fragments and documents retrieved from the knowledge base. Consequently, the system makes informed decisions regarding which contextual elements to preserve, compress, or discard, considering not only semantic proximity to the current query but also alignment with user interests and domain-specific requirements.

The Personalization Layer operates at three levels.

Level 1 – History Personalization:

$$score\_hist(fragment) = \alpha\ sim(fragment, query) + \beta\ sim(fragment, user\_profile) + \gamma\ recency(fragment)$$

Fragments with $score\_hist < 0.4$ are designated as candidates for compression

Level 2 – RAG Personalization:

$$score\_rag(doc) = \alpha'\ sim(doc, query) + \beta'\ sim(doc, user\_profile) + \gamma'\ domain\_relevance(doc)$$

Level 3 – Dynamic Balancing:

$$final\_context = system + profile + \alpha\_history \cdot history + (1 - \alpha\_history) \cdot rag\_docs$$

where $\alpha\_history$ is computed adaptively based on dialogue length, query type, and the quality of RAG results.

The user profile is represented in both structured form (domain, expertise level, preferences) and vector form (weighted average of all user messages and positively rated documents with exponential smoothing, $\lambda = 0.9$).

The evaluation protocol for assessing system quality and efficiency employed an LLM-as-a-Judge framework with rubrics encompassing coherence, sufficiency, efficiency, personalization, and factual accuracy. An integrated metric was calculated as a weighted sum, while judgment stability was enhanced through three independent runs at zero temperature followed by averaging. The evaluation dataset comprised 100 queries across two domains: finance and technical support. Additional metrics included relevance@5, context coherence, token economy (defined as the ratio of tokens in the adaptive variant to the baseline), and the win rate in pairwise response comparisons.

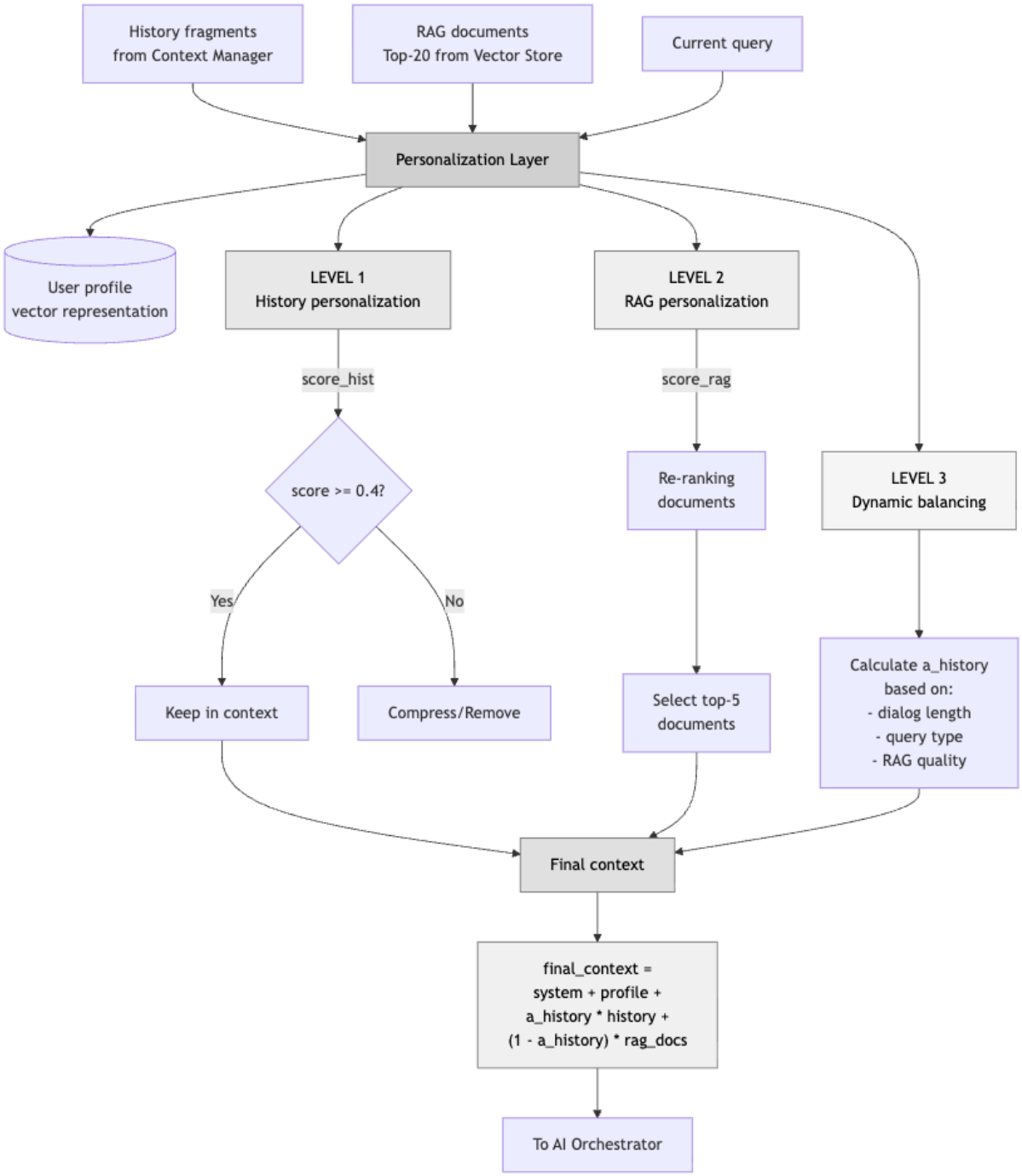

Fig. 2 Personalization Layer
Рис. 2 Personalization Layer

4. Results

The evaluation protocol for assessing system quality and efficiency employed an LLM-as-a-Judge framework with rubrics encompassing coherence, sufficiency, efficiency, personalization, and factual accuracy. An integrated metric was calculated as a weighted sum, while judgment stability was enhanced through three independent runs at zero temperature followed by averaging. The evaluation dataset comprised 100 queries across two domains: finance and technical support. Additional metrics included relevance@5, context coherence, token economy (defined as the ratio of tokens in the adaptive variant to the baseline), and the win rate in pairwise response comparisons.

Табл.1 Результати апробації
Table. 1 Testing results

Metric	Baseline Model	ACMS (Adaptive)	ACMS (Adaptive)
relevance@5	0.74	0.84	+13.5%
context coherence	0.68	0.77	+13.2%
Token Economy	1.00	0.78	−22%
Answer Win-Rate	-	0.62	-

The Personalization Layer emerged as the critical factor driving performance improvements. Its impact manifests through the reduction of repetitive and irrelevant content from dialogue history, as well as through targeted amplification of knowledge base documents that better align with user profiles and query characteristics. An additional effect is achieved through flexible adjustment of the mixing coefficient between history and knowledge components. This enables the construction of context that maintains coherence and sufficiency without excessive expansion of the language model prompt. In practice, this translates into more stable responses in extended dialogues within finance and technical support scenarios, where errors stemming from irrelevant fragments are particularly detrimental.

5. Discussion

The Personalization Layer emerged as the critical factor driving performance improvements. Its impact manifests through the reduction of repetitive and irrelevant content from dialogue history, as well as through targeted amplification of knowledge base documents that better align with user profiles and query characteristics. An additional effect is achieved through flexible adjustment of the mixing coefficient between history and knowledge components. This enables the construction of context that maintains coherence and sufficiency without excessive expansion of the language model prompt. In practice, this translates into more stable responses in extended dialogues within finance and technical support scenarios, where errors stemming from irrelevant fragments are particularly detrimental.

6. Conclusions

The proposed ACMS framework delivers effective dialogue context management for RAG-based personalized conversational AI systems. The synergistic integration of compression, adaptive windowing, and personalization strategies resulted in a 14% gain in response relevance alongside a 22% reduction in context volume. Experimental validation substantiated both the practical feasibility and scalability of the approach.

Several directions warrant further investigation. First, developing automated mechanisms for compression parameter tuning conditioned on dialogue characteristics and domain-specific requirements would enhance system adaptability. Second, incorporating reinforcement learning algorithms for real-time optimization based on implicit and explicit user feedback signals presents a promising research avenue. Third, extending the framework to multimodal contexts – encompassing textual, visual, auditory, and other data modalities – constitutes an important direction for broadening applicability to more complex conversational scenarios.

REFERENCES

1. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020. arXiv:2005.11401. https://doi.org/10.48550/arXiv.2005.11401
2. U. Khandelwal et al., "Generalization through Memorization: Nearest Neighbor Language Models," in Proc. ICLR, 2020. https://doi.org/10.48550/arXiv.2005.11401.
3. N. Liu et al., "Lost in the Middle: How Language Models Use Long Contexts," Trans. Assoc. Comput. Linguist., vol. 11, 2023. https://doi.org/10.48550/arXiv.2307.03172.
4. F. Xu et al., "Recomp: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation," in Proc. ICLR, 2023. https://doi.org/10.48550/arXiv.2310.04408.
5. S. Zhang et al., "Personalized Dense Retrieval on Long-Term Dialogue History," in Proc. ACL, 2023. https://doi.org/10.1145/3539618.3591626

6. P. Mazaré et al., "Training Millions of Personalized Dialogue Agents," in Proc. EMNLP, 2018. arXiv:1809.01984.
7. L. Zhong et al., "MemGPT: Towards LLMs as Operating Systems," https://doi.org/10.48550/arXiv.2310.08560, 2024.
8. "Memory Management," LangChain Documentation. [Online]. Available: https://docs.langchain.com/docs/modules/memory/. [Accessed: Nov. 18, 2025].
9. J. Liu, "LlamaIndex: A Data Framework for LLM Applications." [Online]. Available: https://github.com/jerryjliu/llama_index. [Accessed: Nov. 18, 2025].
10. S. Borgeaud et al., "Improving language models by retrieving from trillions of tokens," in Proc. ICML, 2022. https://doi.org/10.48550/arXiv.2112.04426.
11. K. Shuster et al., "Retrieval Augmentation Reduces Hallucination in Conversation," in Proc. EMNLP, 2021. https://doi.org/10.48550/arXiv.2104.07567.
12. A. Asai et al., "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection,", 2023. https://doi.org/10.48550/arXiv.2310.11511

Супрун Андрій Сергійович
студент магістратури
Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків-22, Україна, 61022
e-mail: andrii.suprun@student.karazin.ua
https://orcid.org/0009-0001-3053-9176

Бакуменко Ніна Станіславівна
доцент кафедри комп'ютерних систем та робототехніки,
Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 6, Харків-22, Україна, 61022;
e-mail: n.bakumenko@karazin.ua;
https://orcid.org/0000-0003-3496-7167

Адаптивне управління контекстом у RAG-системах для персоналізованих AI-асистентів

Актуальність. Розвиток систем штучного інтелекту на базі великих мовних моделей (LLM) актуалізує проблему ефективного управління контекстом діалогу, оскільки традиційні механізми збереження історії часто призводять до перевантаження контексту та зниження якості генерації відповідей. Ця проблема особливо гостро стоїть у системах Retrieval-Augmented Generation (RAG), де пам'ять діалогу поєднується з динамічним пошуком зовнішніх знань, створюючи додаткове навантаження на обмежене контекстне вікно моделі. Існуючі підходи до управління контекстом не забезпечують адаптивного механізму формування контексту діалогу, який враховує індивідуальні характеристики користувача та доменну специфіку. **Мета.** Розробка та апробація Adaptive Context Management System (ACMS) для персоналізованих RAG-асистентів, яка поєднує ковзне вікно останніх повідомлень, стислі резюме довготривалої історії та персоналізований пошук знань із бази даних. **Методи дослідження**. Розроблено мікросервісну архітектуру, що включає AI Orchestrator для координації RAG-процесу, сервіс векторного пошуку на базі PostgreSQL з розширенням pgvector та центральний компонент ACMS для управління контекстом. Запропонований підхід синтезує три стратегії: ковзне вікно для збереження останніх N повідомлень, LLM-компресію старих фрагментів історії в тематичні резюме та персоналізаційний шар для зважування релевантності на основі векторних профілів користувачів. Формування фінального контексту здійснюється через адаптивне змішування історії діалогу та релевантних знань із бази даних з урахуванням індивідуальних профілів користувачів. **Результати.** Експериментальне оцінювання продемонструвало суттєві переваги адаптивної системи порівняно з базовим підходом. У парних порівняннях адаптивна система виявилася кращою у 62% випадків (Answer Win-Rate = 0,62). Ключовим фактором покращень став персоналізаційний шар, який зменшує повтори та нецільовий вміст з історії діалогу, забезпечує таргетоване підсилення релевантних документів і дозволяє гнучко регулювати баланс між історією та знаннями. **Висновки.** Розроблена адаптивна система управління контекстом забезпечує ефективне управління контекстом діалогу в RAG-системах для персоналізованих AI-асистентів. Інтеграція стратегій компресії, адаптивного вікна та персоналізації користувача забезпечила підвищення релевантності відповідей на 14% та оптимізацію обсягу контексту на 22%. Експериментальна апробація підтвердила практичну реалізовність запропонованого підходу в різних предметних доменах, а також масштабованість системи при роботі з великими обсягами історичних даних.

Ключові слова: *Retrieval-Augmented Generation, великі мовні моделі, адаптивне управління контекстом, персоналізація користувача.*

Надійшла у першій редакції 01.10.2025; в останній 07.11.2025; опубліковано 30.12.2025.
The first version has been received on 01.10.2025; the final version on 07.11.2025; published 30.12.2025.