BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 31 No. 3. 2026

UDC 004.738.5:001.8
DOI: 10.62660/bcstu/3.2026.58

ISSN: 2306-4412
E-ISSN: 2708-6070

Model of a scientist's personalised information service

Volodymyr Liubchak
PhD in Physical and Mathematical Sciences, Associate Professor
Sumy State University
40000, 116 Kharkivska Str., Sumy, Ukraine
https://orcid.org/0000-0002-7335-6716
Maksym Shovkoplias*
Postgraduate Student
Sumy State University
40000, 116 Kharkivska Str., Sumy, Ukraine
https://orcid.org/0009-0007-4192-4743

Abstract. The article considers the issue of improving the efficiency of scientific activity by creating a personalised intelligent service. Given the rapid growth in the number of publications, traditional keyword-based search methods are becoming insufficiently effective. The aim of the work was to develop the architecture and mathematical models of a system capable of automating the selection of relevant literature based on content rather than simple word matching. The development was based on a formal five-component model for formalising the researcher's information environment, which integrates modules for profiling, intelligent services, data management, interface interaction, and analytics. The Gemini large language model was used for deep semantic analysis of texts and their transformation into vector representations. The results of the study included the developed three-tier architecture based on the Flutter framework and the Firebase cloud platform, which ensures cross-platform compatibility and instant synchronisation of personalised data. The paper detailed the algorithms for vectorising scientific interests and ranking publications using the cosine similarity method. It was established that the use of vector representations with a dimensionality of 768 components allows for the identification of relevant sources based on semantic features, ignoring the absence of a direct lexical match. The effectiveness of the dynamic profiling mechanism was substantiated, which distinguishes between a researcher's stable scientific interests and their short-term situational queries through a system of weight coefficients. A module for the automatic recognition of bibliographic metadata based on NER technology has been developed and described, enabling the automated structuring of data from unsystematised text arrays. The proposed approach allows for the creation of an adaptive system that adjusts to changes in the researcher's scientific interests and significantly saves time when working with sources

Keywords: personalised scientific service; Gemini AI; Flutter; Firebase; semantic search; text vectorisation; research automation

INTRODUCTION

In the context of global digitalisation, the number of scientific publications is increasing exponentially, leading to the phenomenon of information overload. This creates a critical problem for the modern researcher: the time required to monitor new sources, filter content, and search for relevant data often exceeds the

Article's History: Received: 29.04.2026; Revised: 19.08.2026; Accepted: 21.09.2026; Published: 07.10.2026

Suggested Citation:
Liubchak, V., & Shovkoplias, M. (2026).Model of a scientist's personalised information service. *Bulletin of Cherkasy State Technological University*, 31(3), 58-67. doi: 10.62660/bcstu/3.2026.58.

*Corresponding author (shovkoplyasmaks02@gmail.com)

Copyright © The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/)

time allocated directly to analytical and experimental work. Traditional search tools, such as Google Scholar, Scopus, or library catalogs, are mostly based on lexical matching of keywords. However, this approach has significant limitations: it does not take context into account, ignores synonymous relationships, and often produces a large amount of "noise" information that formally contains the requested words but does not correspond to the user's scientific interests in terms of content. The relevance of this work is determined by the need to create an effective tool that combines the power of modern generative models (specifically, Gemini) with the convenience of mobile interfaces to automate the routine processes of scientific search.

The problem of individualising the information service of scientists has become a subject of active discussion in the scientific community. In particular, S.M. Kashlot *et al*. (2026) in their work considered strategies for overcoming information overload using AI. Semantic approaches to query expansion are detailed in the study by A. Allahim *et al*. (2025). The use of artificial intelligence in recommender systems is highlighted in the research by Q. Zhang *et al*. (2021). The development of large language models (LLMs) and their architectural features are thoroughly analysed by Y. Zhu *et al* (2023) and H. Naveed *et al*. (2023). The specifics of named entity recognition (NER) in scientific texts were studied by Z. Nasar *et al*. (2021) and X. Huang *et al*. (2024). In previous studies by the authors V.O. Liubchak & M.O. Shovkoplias (2024), a comprehensive review of existing models and personalisation methods was conducted. The analysis showed that most existing systems operate according to static templates and are unable to dynamically adapt to changes in the researcher's scientific interests in real-time. The identified shortcomings indicate the need to transition from passive search engines to active intelligent assistants. The conceptual foundations for building such a system were proposed in M. Shovkoplias (2025), where an AI-based researcher support model was justified. It was shown that the use of advanced natural language processing (NLP) technologies and LLM makes it possible to bridge the semantic gap between the user's query and the content of a scientific article. However, the theoretical concept requires practical concretisation in the form of a formalised model architecture and algorithmic support.

Analysis indicates that despite a significant number of developments, most existing personalisation systems are based on static algorithms and are unable to dynamically adapt to changes in a researcher's interests in real-time. In particular, the issue of automating semantic content differentiation when processing large arrays of unstructured scientific data remains insufficiently studied. Based on this, the aim of the work was the development and scientific justification of the operational algorithms for a personalised service, ensuring a qualitative transition from traditional lexical search to deep semantic differentiation of content.

MATERIALS AND METHODS

This study is a theoretical and applied research work aimed at formalising and optimising information support processes for researchers. To substantiate the proposed models and methods, the following methodological approaches were employed: analysis of scientific sources to identify the current challenges of information overload; formalisation to define the system's conceptual framework; mathematical modeling to represent semantic data structures; structural-functional modeling to design the service architecture; algorithmisation to develop the data processing workflows; and comparative analysis to evaluate the proposed semantic approaches against traditional lexical methods.

Dataset preparation and text processing. To ensure the reliable functioning of the semantic search and to test the proposed algorithms, a specialised dataset of scientific publications was formed. The dataset was aggregated using open-access APIs from academic databases, specifically CrossRef and arXiv, targeting the domains of computer science, information technology, and artificial intelligence. In total, a corpus of 15,000 scientific abstracts and their corresponding bibliographic metadata was collected. The selection of materials was carried out based on strict criteria: the presence of peer review, publication within the last five years (2020-2025), and English as the primary language of the text. This ensured the relevance and linguistic consistency of the analysed material. To transform the unstructured textual data from the dataset into a format suitable for semantic analysis, the system utilises the Gemini embedding model. This model maps textual input into a dense continuous vector space. During the transformation process, the raw text of an abstract or a user query undergoes tokenisation and is processed through deep neural network layers to capture complex semantic relationships, context, and synonymy. The output is a semantic vector (embedding) with a fixed dimensionality of 768 components.

Mathematical representation of semantic data. To process scientific information, the system transforms natural language text into a structured mathematical form. Each document or user query was represented as a high-dimensional semantic vector (embedding):

$$V = [x_1, x_2, \ldots, x_n], \quad (1)$$

where V is a vector in a multidimensional space, $n = 768$ (the dimensionality of the Gemini model), and each x_i is a numerical value representing a specific semantic feature of the text. This vector captures the "meaning" of the content, allowing the system to compare documents based on their conceptual similarity rather than just keyword matches.

Calculation of semantic relevance. The core of the recommendation engine is the calculation of the "distance" between the researcher's interests and the content of scientific articles. The system utilises Cosine Similarity, which measures the cosine of the angle between two vectors (V_{user} and $V_{article}$):

$$Relevance = cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}, \quad (2)$$

where A_i and B_i are the components of the user profile and article vectors respectively. The result is a value from 0 to 1, where 1 indicates perfect thematic alignment and 0 indicates no semantic connection. This method is optimal for scientific search because it focuses on the direction of the vector (the topic) rather than its magnitude (the length of the text), ensuring that short abstracts and long articles are ranked fairly.

Dynamic user profiling. To account for both long-term research and current tasks, the total user profile (P_{total}) was calculated as a weighted sum:

$$P_{total} = (w_1 \cdot P_{core}) + (w_2 \cdot P_{context}). \quad (3)$$

The system distinguishes stable interests (P_{core}, derived from saved articles and publications) from short-term needs ($P_{context}$, generated from current search queries and session history). The weight coefficients are strictly normalised ($w_1 + w_2 = 1$). Their values are dynamically adjusted using an exponential time decay function: $w2(t) = winit \cdot e^{-\gamma t}$). During an active search session, the contextual weight w_2 is prioritised. As time t passes since the last interaction, w_2 exponentially degrades, proportionally increasing w_1 to return the system's focus to core interests. The profile is updated in real-time; every user action (querying, viewing, or saving a document) triggers an instant recalculation of the weights and vectors via cloud functions.

NER-approach and metadata structuring. Given that scientific texts are often unstructured, the system employs a Named Entity Recognition (NER) approach to automatically extract and structure bibliographic metadata. The system classifies and recognises specific entity types: [PER] for author names, [ORG] for academic affiliations, [DATE] for publication years, [TITLE] for the names of articles or journals, and [ID] for unique identifiers such as DOIs. The extraction relies on a hybrid algorithm: standard regular expressions (regex) are applied for rigid formats (e.g., parsing DOIs and dates), while a customised prompt utilising the Gemini LLM is deployed to recognise context-dependent entities like author names and titles. Once extracted, the normalisation process converts the data into unified formats (e.g., formatting dates to ISO 8601 standard (International Organization for Standardization, 2024) and standardising author name abbreviations) before writing them to the database.

Software implementation and architecture. The methodological framework is practically realised through a robust Three-Tier Architecture, which strictly separates the presentation, business logic, and data management layers to ensure system scalability and maintainability. The presentation layer, or client layer, is implemented using the Flutter framework. This technology was selected because it guarantees cross-platform compatibility, allowing the system to deploy a native-like client interface across iOS, Android, and web environments from a single codebase. It provides the user with an intuitive UI to input queries and view recommendation feeds, while seamlessly running background processes to capture behavioral signals, such as reading time and bookmarking, for profile adaptation.

Operating entirely in a serverless environment via Cloud Functions, the business logic layer acts as the computational core. It seamlessly manages the integration of the Gemini API into the semantic analysis workflow. When a text query is received from the client, the Cloud Function securely authenticates the request, forwards the text to the Gemini API for vectorisation, and processes the returned 768-dimensional semantic vectors. This server-side processing offloads heavy computational tasks from the user's mobile device, ensuring high performance. Finally, for cloud storage and data management, the data layer utilises Firebase Firestore. As a NoSQL document-oriented database, Firestore is uniquely suited to store both the structured bibliographic metadata generated by the NER module and the complex arrays of semantic vectors. A critical functional advantage of using Firestore is its built-in real-time synchronisation capabilities. Whenever the business logic layer updates a user's dynamic profile (Pcore or Pcontext) based on new behavioral data, Firestore instantly syncs these changes across all the user's connected devices, ensuring continuous and uninterrupted scientific research.

RESULTS

Formalisation of the service structural model components. To design the architecture of an intelligent system, it is necessary to clearly define its components and the principles of their interaction. The service model is formalised as a five-element tuple:

$$F = (P, S, D, I, U)$$

This approach allows for the decomposition of the complex scientific search process into distinct functional blocks. Component P (Profile) is a digital model of the user's interests. In most existing systems, a profile is a static set of keywords (tags), such as "Artificial Intelligence" or "Medicine". However, this approach is imprecise due to the inherent ambiguity of words. In the proposed model, P is represented as a dynamic set of vectors. As established in the methodology, each scientific interest and text is encoded as a high-dimensional

semantic vector (embedding). This representation enables the system to search for information based on conceptual meaning rather than literal character matching.

Structurally, it is advisable to divide the profile into two components: the base profile (P_{core}), which reflects stable scientific interests (e.g., a dissertation topic) that change slowly and are formed based on the analysis of the author's previously published works; and the situational profile ($P_{context}$), which reflects current, short-term tasks (e.g., searching for literature for a specific article), where such interests tend to gradually fade over time. Component S (Services) acts as the system's core and consists of a set of software algorithms that provide information processing. Within the proposed architecture, a key role is played by the use of the Gemini large language model, which enables functions such as vectorisation (transforming a user's query text into a vector format), semantic search (finding relevant documents by comparing vectors), and entity extraction (NER) (automatic text analysis to identify authors, titles, and bibliographic data). Component D (Data) describes how data storage is organised within the system. It handles various types of information, combining structured metadata with unstructured article texts. To ensure high performance and real-time operational capability, the architecture provides for the use of a cloud-based NoSQL database (specifically, the Firebase ecosystem). Each document d_j in the database is described by a set:

$$d_j = (M_j, T_j, \vec{v_j})$$

where M_j – metadata (title, authors, year, DOI); T_j – text content (abstract); $\vec{v_j}$ – pre-calculated semantic vector of the document. Storing the ready vector allows for significant acceleration of the search process, avoiding repetitive text processing by the neural network.

Interface and Analytics (I, U) components provide the system's interaction with the researcher: the interface (I) is responsible for data visualisation and, considering the need for access to the service from various devices such as smartphones, tablets, or the web, is implemented using the Flutter cross-platform framework, which allows creating adaptive applications with a single codebase; analytics (U) implements a feedback mechanism by analysing user actions, specifically viewing and saving articles, and automatically adjusting the weight coefficients of the vectors in profile P, thereby adapting the output results to the researcher's current needs. The structural-functional diagram of the model is presented in Figure 1.

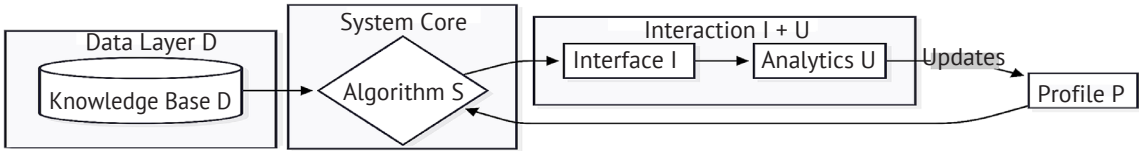


Figure 1. Structural-functional diagram of the model

Source: created by the authors

Thus, the presented structural-functional diagram reflects the integrity of the proposed approach, where each component performs a strictly defined role in the process of intelligent support for the researcher. The interaction between the dynamic profile and semantic services, supported by a scalable cloud infrastructure, creates a foundation for an adaptive information environment. This formal structure ensures the consistent transformation of raw scientific data into personalised recommendations, which serves as the basis for the subsequent algorithmic implementation of the system

Data processing and ranking algorithms for enabling system intelligence. The formalisation of the structural components creates a theoretical foundation for the system. The practical transition to a functioning service is achieved through the implementation of the data processing algorithms defined in the methodology. Following Formula (1), unstructured scientific texts and researcher interests are represented as 768-dimensional semantic vectors generated using the Gemini model family (Anil *et al.*, 2023; Naveed *et al.*, 2023). This vector space provides a common basis for comparing user interests and scientific publications according to semantic similarity rather than lexical coincidence.

Publication ranking is performed using cosine similarity, as defined in Formula (2), which measures the proximity between the researcher's profile vector and article vectors (Huang *et al.*, 2024). This approach enables the proposed model to prioritise publications according to thematic relevance and provides the algorithmic basis for reducing irrelevant search results. Furthermore, the dynamic profiling mechanism practically integrates long-term research interests with immediate tasks. By applying the weighted sum model defined in Formula (3), the service maintains a fine balance between the user's core scientific focus (P_{core}) and situational queries ($P_{context}$). This results in an adaptive profile that evolves in real-time based on user interaction markers such as abstract viewing time or article saving, ensuring that the recommendation feed remains relevant even as the researcher's focus shifts between different projects.

A critical function of the data component (D) is the capability to transform unstructured text arrays

into structured databases. This task is addressed by a module based on Named Entity Recognition (NER) technology, adapted for the nuances of scientific texts (Nasar *et al.*, 2021; Keloth *et al.*, 2024). Unlike simple pattern-based search methods, the algorithm employs contextual analysis to distinguish semantic roles. For example, the system correctly identifies whether a name is an author's surname or part of a scientific term. A comparative analysis of the developed approach and traditional methods is presented in Table 1.

Table 1. Comparative characteristics of the traditional and developed algorithmic approaches

Comparison criterion	Traditional lexical search (keyword-based))	Developed semantic algorithm (S)
Basic unit of analysis	Keyword (token)	Semantic vector (embedding v)
Sensitivity to synonyms	Low (requires exact match)	High (understands context via semantic vectors)
Profile model (P)	Static set of tags	Dynamic weighted interest graph
Time consideration	Sorting by date (DESC)	Time Decay function
Ranking principle	Word frequency (TF-IDF)	Cosine vector similarity + interest weight
Cold start problem	High (requires manual query input)	Medium (based on author's publication analysis)

Source: created by the authors

Thus, the comparative analysis demonstrates the qualitative advantage of the developed semantic algorithm over traditional lexical search methods. By utilising vector representations instead of static tags, the system overcomes the issue of low sensitivity to synonyms and provides a deeper understanding of the scientific context. The implementation of a dynamic interest graph and a time decay function allows the model to adapt to the researcher's real needs in real-time, making the proposed approach an effective tool for addressing the "cold start" problem and information overload in the modern scientific environment.

Architecture and interaction of model components. To transform the developed algorithms and the formal model into a functioning system, it is necessary to define the overall logical scheme of the interaction between its components. The architecture of the proposed model is considered as a continuous cyclic process of information exchange between the user and the intelligent knowledge base (Fig. 2).

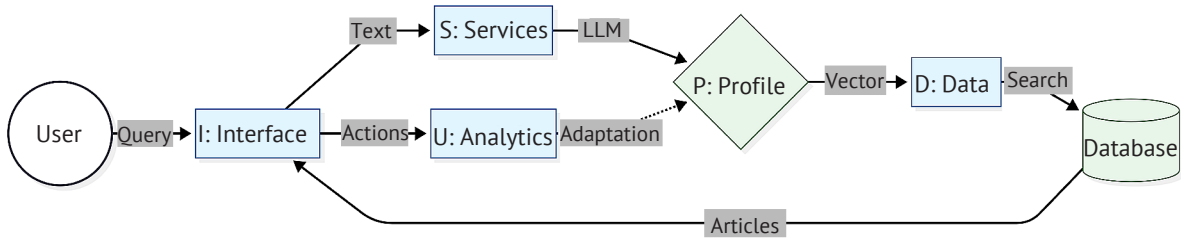


Figure 2. Architecture and component interaction diagram of the model

Source: created by the authors

The process begins with the request initialisation stage, which is handled by the interface interaction component (I). At this step, the researcher formulates their scientific query in a free-form text format or provides the system with a base article of interest. The primary goal of this stage is to collect raw input data for further analysis in the most natural format possible for a human. Next, the collected information is passed to the intelligent services module (S), where deep semantic processing takes place. The resulting text is vectorised using a large language model, transforming into a multidimensional numerical array. Based on this vector, the system dynamically forms or updates the user's situational profile (P), which accurately reflects their current, local scientific interest.

The next step is to access the data management space (D). Using the generated query vector, the algorithm searches the database of stored scientific publications and metadata. By applying the cosine similarity mathematical method, the system calculates the semantic distance between the user's interest vector and the vectors of all available articles. As a result, documents whose content most closely matches the researcher's needs are selected, filtering out "information noise". The iterative cycle concludes with the feedback and adaptation stage. Selected relevant documents are returned to the user interface for visualisation. Simultaneously, the analytics module (U) captures all researcher actions regarding these documents in the background – detailed views, additions to their own library, clicking DOI links, or ignoring the output. These behavioral markers are automatically used to adjust the weight coefficients in the base profile (P). This enables the architecture to provide continuous self-learning for the system: every action of the scientist makes subsequent recommendations even more accurate and personalised.

Software architecture and information exchange for the intelligent system. The developed theoretical model (1) served as the basis for designing the service's software architecture. To ensure system flexibility and scalability, a Three-Tier Architecture was chosen, which allows for a clear distinction between data processing logic, the user interface, and information storage (Donglan *et al.*, 2024).

Presentation layer (client layer) – component I. The presentation layer, which corresponds to component I in the formal model, acts as the single point of entry for the user into the intelligent system. Since modern scientific activity is characterised by high mobility, a critical requirement for the interface is its adaptability and availability on various types of devices – from smartphones to web browsers (Cheon & Chavez, 2021). The technical implementation of this level is based on the use of the Flutter cross-platform framework, which ensures native performance and a unified visual style for iOS and Android operating systems using a single codebase (Kishore *et al.*, 2022; Shanmugam *et al.*, 2023). Functionally, the client application not only visualises the results of semantic search as a convenient recommendation feed but also acts as an active data collection tool: it records the researcher's behavioral patterns (abstract viewing time, clicking a DOI link, adding an article to a personal library). These interaction signals are automatically transmitted to the data layer for an immediate update of the user profile, allowing the system to dynamically adjust the output to current scientific interests.

Business logic layer (application layer) – components S, P. The business logic layer, which structurally combines the services (S) and profiling (P) components, acts as the "intellectual core" of the architecture, where the system's main computational processes are concentrated. The technical implementation of this level is based on the Serverless computing concept using Cloud Functions (Thomas *et al.*, 2026), which ensures flexible scalability and seamless integration with the Gemini large language model API. It is at this stage that key algorithmic transformations take place: the input text query from the user is transmitted to the neural network to generate a high-dimensional semantic vector (embedding), based on which the content ranking procedure is performed using the cosine similarity method. Offloading these resource-intensive mathematical operations to a cloud environment allows for a "thin client" architecture, guaranteeing high service speed and immediate processing of large data sets regardless of the end-user's smartphone processing power.

Data layer – components D, U. The data layer, responsible for the functioning of the data management (D) and analytics (U) components, serves as the foundation for the reliable storage of the user's digital profile and the cataloging of scientific materials. The technological basis for this level is the Firebase Firestore (NoSQL) cloud document-oriented database, which is ideal for storing heterogeneous data: structured bibliographic metadata, multidimensional vector representations of articles, and the history of the researcher's activity (Mudrikah *et al.*, 2022). A key architectural advantage of this solution is the built-in support for Real-time database synchronisation: any state change, such as saving an article on a smartphone, is immediately replicated on the server and becomes available on the user's other devices (tablet or web version), ensuring the continuity of the scientific process. The overall interaction between the architectural layers and data exchange mechanisms is illustrated in Figure 3.

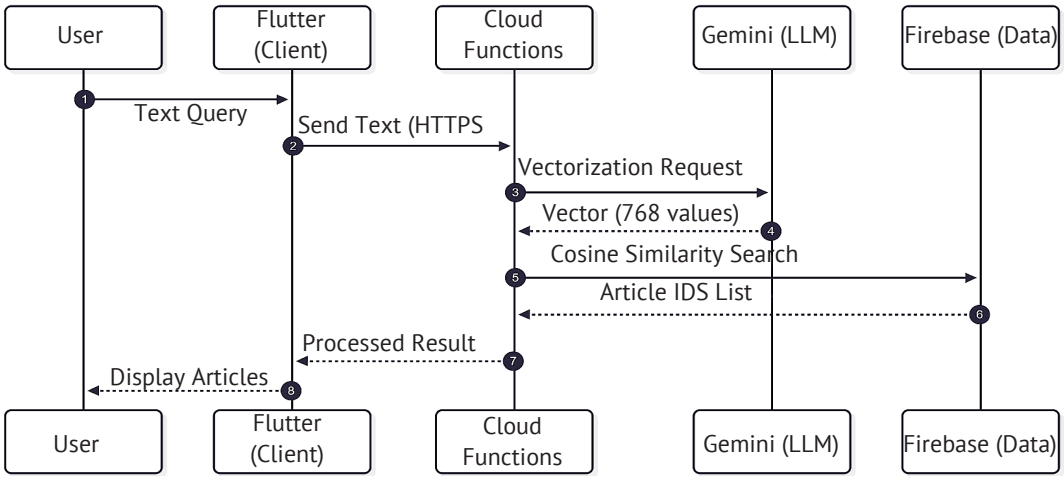


Figure 3. Information exchange diagram

Source: created by the authors

The practical interaction between the described layers is realised through a sequential data processing algorithm. The process is initiated by the user in the mobile application (Flutter), from where the generated

query is transmitted via a secure cryptographic channel (HTTPS) to the processing server. On the server side, a request is made to the Gemini model API to transform the query text into a 768-dimensional vector. Subsequently, the database searches for the nearest vectors within the publication array and returns a list of relevant article identifiers, which is visualised via the application interface. The proposed hybrid architecture is optimal for a specialised scientific service as it effectively combines the computational power of cloud-based artificial intelligence with the ergonomics and mobility of the client interface.

DISCUSSION

The obtained results demonstrated the effectiveness of transitioning from traditional lexical search methodologies to a fully dynamic semantic environment for scientific information retrieval. The implementation of the proposed model directly addresses the limitations of keyword-based systems, offering a more nuanced, context-aware approach to processing scientific literature. This research strongly aligns with the conclusions drawn by S.M. Kashlot *et al.* (2026), who indicated the critical need to utilise generative artificial intelligence to overcome modern information overload. In their systematic review, S.M. Kashlot *et al.* (2026) established that researchers spend an unsustainable amount of time filtering out irrelevant search results. However, their proposed strategies focused primarily on web-based platforms and static AI filters. The architecture developed in this study expands upon their foundational concepts by integrating these capabilities into a cross-platform mobile environment. By moving the intelligent assistant directly to the researcher's smartphone or tablet with real-time syncing, the proposed model effectively transforms information overload management from a localised desktop task into a continuous, mobile-friendly process. The proposed architecture addresses the computational overhead associated with real-time embedding generation by utilising pre-calculated 768-dimensional semantic vectors stored in Firebase Firestore. By shifting the computational workload to the data-ingestion phase, the system enables instantaneous semantic matching via cosine similarity without requiring real-time neural network processing for the document corpus.

When evaluating semantic ranking algorithms, it is highly relevant to compare the proposed dynamic profiling method with the work of A. Allahim *et al.* (2025). Allahim's research utilised vector representations to expand search queries, successfully capturing synonyms and related concepts that traditional TF-IDF (Term Frequency-Inverse Document Frequency) algorithms missed. While their methodology yielded high precision for isolated queries, it inherently lacked temporal adaptability; the system treated all user interests as static over time. To resolve this critical flaw, the current work proposes a unique mechanism for dynamically dividing the user profile into a base profile (Pcore) and a situational profile (Pcontext). The absolute necessity of incorporating temporal dynamics into user profiles was highlighted by L.-W. Pan *et al.* (2026). Their proposed solution to drifting user interests involved complex chronological recalculations of the researcher's entire search history. This approach, while accurate, is computationally expensive and difficult to scale. In contrast, the methodology developed in this study applies a mathematically simpler but highly effective time decay function specifically to the situational profile (Pcontext). This allows the system to seamlessly adapt to short-term, situational research tasks without losing the user's core expertise (Pcore), thereby drastically reducing algorithmic complexity while maintaining high recommendation relevance.

The challenge of utilising artificial intelligence in recommender systems, particularly in specialised domains, was extensively considered by Q. Zhang *et al.* (2021). Their research placed heavy emphasis on collaborative filtering techniques. However, collaborative filtering inevitably suffers from the "cold start" problem-it struggles to recommend newly published, highly specialised papers that have not yet been heavily interacted with by other users. The current research proves that a content-based filtering approach, driven by deep semantic embeddings and cosine vector similarity, provides a superior solution for academic environments. Because the system analyses the semantic content of the article itself rather than user interaction history, it can accurately recommend a paper the moment it is published. This conclusion is robustly supported by S. Kuznetsov & P. Kordík (2023), who argued that dense retrieval methods (like semantic embeddings) fundamentally outperform sparse retrieval (like keyword matching) in cross-domain scholarly searches. The model proposed in this article actively advances their theoretical concept by introducing an interactive feedback loop. The query vector is not static; it is actively modified in the background based on interaction analytics (such as read time or bookmarking), continuously refining the dense retrieval process.

Regarding the automated structuring of bibliographic data, Z. Nasar *et al.* (2021) and X. Huang *et al.* (2024) thoroughly investigated the application of Named Entity Recognition (NER) models utilising contextual self-attention. Academic literature analysis compiled by S.O. Semerikov *et al.* (2022) indicates that traditional approaches to NER involve delayed batch processing on local servers or heavy edge-device computation. In contrast, the solution proposed in this article takes a different path. By integrating the NER process directly into a serverless cloud infrastructure (Cloud Functions), the system ensures instant metadata extraction without draining the mobile device's battery or processing power. The critical importance of data

normalisation prior to database insertion was heavily emphasised by J. Priem *et al.* (2022). They noted that metadata inconsistencies across different scientometric databases cause massive failures in accurate citation tracking and content retrieval. The NER algorithm developed in this study resolves this issue at the point of ingestion by automatically transforming varied, unstructured inputs into unified standards (such as formatting all temporal data to ISO 8601 and normalising author credentials) during the extraction phase.

Finally, evaluating the technological stack against contemporary research confirms the viability of the chosen architecture. Z. Donglan & M.Y. Darus (2024) concluded in their comparative analysis that the Flutter framework provides near-native rendering speeds, which is an absolute requirement for data-intensive applications displaying complex scientific feeds. Similarly, S. Kot & J. Smołka (2023) confirmed the unmatched efficacy of Firebase for managing real-time data synchronisation in mobile applications. The successful theoretical and structural integration of these technologies in the proposed model proves that NoSQL document-oriented databases paired with serverless functions are highly efficient for storing, updating, and instantly syncing high-dimensional vector arrays. Thus, comparing the results of this study with contemporary publications by international scientists confirms both the scientific novelty and the tangible practical advantages of the developed model. Ultimately, the significance of this proposed architecture extends far beyond localised software optimisation, offering a robust foundational blueprint for the broader development of next-generation semantic recommendation systems. By seamlessly bridging dense vector spaces with low-latency cross-platform deployment, this structural framework significantly advances the capabilities of adaptive scientific information retrieval. Consequently, it lays a scalable foundation for the evolution of truly intelligent research environments that dynamically adjust to the shifting cognitive demands and specialised workflows of the global scientific community.

CONCLUSIONS

In this work, the urgent task of providing information support to researchers facing the challenge of finding relevant data amidst the rapid growth of publications has been addressed. The primary result of the study is the development and substantiation of a formal model (1), which allowed for the decomposition of a complex information system into distinct components: from a digital user profile to the interaction interface. Drawing upon contemporary research in the field of natural language processing and recommender systems, an approach to representing data in the form of dynamic vectors was adapted. The application of vector models instead of static sets of keywords enables the system to recognise the semantic content of texts. Within the developed model, this allowed for the implementation of a mechanism where the system automatically distinguishes between a researcher's stable interests (e.g., a dissertation topic) and current short-term queries, adapting recommendation output in real-time. The adaptation of a classic three-tier architecture (Three-Tier Architecture) to the specifics of a scientific service is proposed. Using proven engineering solutions from other software engineering domains, the integration of a modern technology stack was proposed to address the assigned information retrieval tasks. Specifically, the use of the Flutter mobile framework and the Firebase cloud database is based on best practices for creating cross-platform applications with real-time data synchronisation. At the same time, a key feature of the developed architecture is the integration of the Gemini large language model, which provides the system with the necessary intelligence for deep text analysis.

Operational service algorithms have been formalised in this work, automating routine processes-from transforming a text query into a numerical vector to identifying named entities (NER) in bibliographic lists. The proposed component interaction scheme proves that the transition from simple lexical search to semantic analysis can significantly reduce the cognitive load on the researcher and enhance the quality of scientific work. The results obtained create a foundation for the software implementation of the service as a mobile application and its subsequent practical integration. Prospects for future research include the practical software implementation of the proposed architecture by creating a basic functional prototype of the system. This will allow for empirical testing of the effectiveness of the developed semantic search algorithms in the real-world conditions of researchers' work.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None

REFERENCES

[1] Allahim, A., Cherif, A., & Imine, A. (2025). Semantic approaches for query expansion: Taxonomy, challenges, and future research directions. *PeerJ Computer Science*, 11, article number e2664. doi: 10.7717/peerj-cs.2664.

[2] Anil, R., *et al.* (2023). Gemini: A family of highly capable multimodal models. *ArXiv*. doi: 10.48550/arXiv.2312.11805.

[3] Cheon, Y., & Chavez, C. (2021). Converting Android native apps to Flutter cross-platform apps. In *2021 international conference on computational science and computational intelligence* (pp. 1898-1904). Las Vegas: IEEE. doi: 10.1109/CSCI54926.2021.00355.
[4] Donglan, Z., & Darus, M.Y. (2024). A comparative analysis of cross-platform mobile development frameworks. *In 2024 IEEE 6th symposium on computers & informatics (ISCI)* (pp. 84-90). Kuala Lumpur: IEEE. doi: 10.1109/ISCI62787.2024.10667693.
[5] Donglan, Z., Darus, M.Y., & Ramli, A.B. (2024). Investigating developer experiences with UI components in Flutter: Challenges and implications. In *2024 international visualization, informatics and technology conference (IVIT)* (pp. 128-133). Kuala Lumpur: IEEE. doi: 10.1109/IVIT62102.2024.10692626.
[6] Huang, X., Peng, H., Zou, D., Liu, Z., Li, J., Liu, K., Wu, J., Su, J., & Yu, P.S. (2024). CoSENT: Consistent sentence embedding via similarity ranking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32, 2800-2813. doi: 10.1109/TASLP.2024.3402087.
[7] International Organization for Standardization. (2024). *ISO/TS 7552-2:2024*. Retrieved from https://www.iso.org/standard/82828.html.
[8] Kashlot, S.M., Ruthven, I., & Moshfeghi, Y. (2026). Strategies for managing information overload: A systematic review. *Journal of Documentation*, 82(2), 249-265. doi: 10.1108/JD-09-2025-0257.
[9] Keloth, V., *et al*. (2024). Advancing entity recognition in biomedicine via instruction tuning of large language models. *Bioinformatics*, 40(4), article number btae163. doi: 10.1093/bioinformatics/btae163.
[10] Kishore, K., Khare, S., Uniyal, V., & Verma, S. (2022). Performance and stability comparison of React and Flutter: Cross-platform application development. In *2022 international conference on cyber resilience* (pp. 1-4). Dubai: IEEE. doi: 10.1109/ICCR56254.2022.9996039.
[11] Kot, S., & Smołka, J. (2023). A performance analysis of a cloud database on mobile devices. *Journal of Computer Sciences Institute*, 29, 360-365. doi: 10.35784/jcsi.3798.
[12] Kuznetsov, S., & Kordík, P. (2023). Overcoming the cold-start problem in recommendation systems with ontologies and knowledge graphs. In *New trends in database and information systems* (pp. 591-603). Cham: Springer. doi: 10.1007/978-3-031-42941-5_52.
[13] Liubchak, V.O., & Shovkoplias, M.O. (2024). Review of models and methods for individual customization of a scientist's information service. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 2, 88-96. doi: 10.32782/IT/2024-2-11.
[14] Mudrikah, F.Z.A., Istikmal, I., & Aditya, B. (2022). Design of a geographic information system for forest and land fires based on a real-time database on microservices infrastructure. In *2022 IEEE international conference on internet of things and intelligence systems (IoTaIS)* (pp. 1-6). Bali: IEEE. doi: 10.1109/IoTaIS56727.2022.9975953.
[15] Nasar, Z., Jaffry, S.W., & Malik, M.K. (2021). Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys*, 54(1), 1-39. doi: 10.1145/3445965.
[16] Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2023). A comprehensive overview of large language models. *ArXiv*. doi: 10.48550/arXiv.2307.06435.
[17] Pan, L.-W., Pan, W.-K., Wei, M.-Y., Yin, H.-Z., & Ming, Z. (2026). A survey on sequential recommendation. *Frontiers of Computer Science*, 20, article number 2003606. doi: 10.1007/s11704-025-41329-w.
[18] Priem, J., Piwowar, H., & Orr, R. (2022). OpenAlex: A fully-open index of scholarly works. *arXiv*. doi: 10.48550/arXiv.2205.01833.
[19] Semerikov, S.O., Vakaliuk, T.A., Kanevska, O.B., Ostroushko, O.A., & Kolhatin, A.O. (2025). Edge intelligence unleashed: A survey on deploying large language models in resource-constrained environments. *Journal of Edge Computing*, 4(2), 179-233. doi: 10.55056/jec.1000.
[20] Shanmugam, G., Pappa, C.M.K., Tamilvizhi, T., Nelson, L., & Surendran, R. (2023). Intelligent analysis on frameworks for mobile app development. In *2023 5th international conference on artificial intelligence and smart energy*. Tirunelveli: IEEE. doi: 10.1109/ICSSIT55814.2023.10060902.
[21] Shovkoplias, M.O. (2025). AI-based model of a researcher support service. *Bulletin of Cherkasy State Technological University*, 30(4), 82-96. doi: 10.62660/bcstu/4.2025.82.
[22] Thomas, A., Adebayo, H., James, P., & Eniola, A. (2026). *Serverless computing for AI-based NLP applications: Scalability and cost efficiency analysis*.
[23] Zhang, Q., Lu, J., & Jin, Y. (2021). Artificial intelligence in recommender systems. *Complex & Intelligent Systems*, 7, 439-457. doi: 10.1007/s40747-020-00212-w.
[24] Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Chen, H., Liu, Z., Dou, Z., & Wen, J.-R. (2023). Large language models for information retrieval: A survey. *arXiv*. doi: 10.48550/arXiv.2308.07107.

Модель персоналізованого інформаційного сервісу науковця

Володимир Любчак
Кандидат фізико-математичних наук, доцент
Сумський державний університет
40000, вул. Харківська, 116, м. Суми, Україна
https://orcid.org/0000-0002-7335-6716
Максим Шовкопляс
Аспірант
Сумський державний університет
40000, вул. Харківська, 116, м. Суми, Україна
https://orcid.org/0009-0007-4192-4743

Анотація. У статті розглядається питання підвищення ефективності наукової діяльності шляхом створення персоналізованого інтелектуального сервісу. В умовах стрімкого зростання кількості публікацій традиційні методи пошуку за ключовими словами стають недостатньо ефективними. Метою статті була розробка архітектури та математичних моделей системи, здатної автоматизувати добір релевантної літератури на основі змісту, а не простого збігу слів. В основу розробки покладено формальну п'ятикомпонентну модель формалізації інформаційного середовища науковця, яка інтегрує модулі профілювання, інтелектуальних сервісів, управління даними, інтерфейсної взаємодії та аналітики. Застосовано велику мовну модель Gemini для глибокого семантичного аналізу текстів та їх перетворення у векторні представлення. Результатами дослідження є розроблена трирівнева архітектура на базі фреймворку Flutter та хмарної платформи Firebase, яка забезпечує кросплатформеність та миттєву синхронізацію персоналізованих даних. У роботі детально описано алгоритми векторизації наукових інтересів та ранжування публікацій методом косинусної подібності. Встановлено, що використання векторних представлень розмірністю 768 компонентів дозволяє ідентифікувати релевантні джерела за семантичними ознаками, ігноруючи відсутність прямого лексичного збігу. Обґрунтовано ефективність механізму динамічного профілювання, який через систему вагових коефіцієнтів розрізняє стабільні наукові інтереси дослідника та його короткочасні ситуативні запити. Розроблено та описано модуль автоматичного розпізнавання бібліографічних метаданих на основі технології NER, що дозволяє автоматизувати структурування даних із несистематизованих текстових масивів. Запропонований підхід дозволяє створити адаптивну систему, яка підлаштовується під зміни наукових інтересів дослідника та значно економить час на роботу з джерелами

Ключові слова: персоналізований науковий сервіс; Gemini AI; Flutter; Firebase; семантичний пошук; векторизація тексту; автоматизація досліджень