Ingénierie des Systèmes d'Information
Vol. 31, No. 4, April, 2026, pp. 1343-1350
Journal homepage: http://iieta.org/journals/isi

HealthAgent: A Hybrid Conversational Artificial Intelligence Framework Integrating Neo4j Knowledge Graphs and Semantic Search for Personalized Health Supplement Recommendations

Deepali Joshi[1*], Nitin Gupta[2], Neha Patwardhan[3], Nilam Upasani[4], Ranjana Jadhav[1], Vijaykumar Bhanuse[5]

[1] Department of Information Technology, Vishwakarma Institute of Technology, Pune 411037, India
[2] Department of Applied Data Science, University of Chicago, Chicago 60637, United States
[3] Symbiosis Institute of International Business, Pune, Symbiosis International (Deemed University), Pune 412115, India
[4] Balaji Institute of Technology and Management, Sri Balaji University, Pune 411033, India
[5] Department of Instrumentation and Control, Vishwakarma Institute of Technology, Pune 411037, India

Corresponding Author Email: deepali.joshi@vit.edu

https://doi.org/10.18280/isi.310428

Received: 11 November 2025
Revised: 25 January 2026
Accepted: 18 April 2026
Available online: 30 April 2026

Keywords:
healthcare conversational Artificial Intelligence, personalized supplement recommendation, Neo4j knowledge graph, semantic search, hybrid recommendation, user profiling, large language models

ABSTRACT

Selecting appropriate health supplements is challenging due to overwhelming, inconsistent, and incomplete information from multiple sources. We present HealthAgent, a hybrid conversational Artificial Intelligence (AI) framework that assists consumers through semantic search and Neo4j knowledge graphs. The system integrates three core components: LangChain for agentic AI capabilities, Large Language Models (LLMs) for natural dialogue interaction, and Neo4j graph databases for structured knowledge representation. HealthAgent employs semantic vector search combined with historical user interactions to generate context-aware product recommendations. A rule-based query classifier directs health-specific queries and general information requests to appropriate modules, while user profile data, including age, gender, and location, further personalizes recommendations. Experimental evaluation with 200 test queries across 15 users shows that the logistics agent achieved 78% precision@3, with an average response latency of 340 ms. User satisfaction scores averaged 4.2/5.0 in relevance, clarity, ease of use, and helpfulness. Comparisons with keyword search (52%) and term frequency-inverse document frequency (TF-IDF) baselines (68%) demonstrate that integrating semantic similarity with knowledge graph structures improves accuracy and recommendation relevance. Ablation studies confirm the importance of multi-agent coordination, knowledge integration, and personalized query handling. The system achieves sub-second response times, enabling real-time deployment. HealthAgent illustrates how combining knowledge graphs, semantic search, and conversational AI provides accurate, personalized, and scalable guidance for health supplement selection, offering a practical framework for consumer-oriented healthcare recommendation systems.

1. INTRODUCTION

Many are resorting to health supplements to aid them with their well-being, but looking for the appropriate one is a confusing process. There is just too much information readily available on the internet that can confuse a person looking to receive well-structured guidance, and finally, advice that would align with their particular health needs may not always be available through simple search engines or through advice from shopping engines or sites. They tend to have general information and nothing related to the complex interactions between multiple health supplements and compounds. There is new potential in the realm of artificial intelligence. Large Language Models (LLMs) already display incredible abilities in language understanding and production, allowing conversations to seem organic and useful. The enhancement to this is Agentic Artificial Intelligence (AI), which enables the language models to choose and use appropriate tools, retrieve information from other sources, and accomplish multiple undertakings in solving problems. Simultaneously, graph database technology is highly useful in information management that involves connections and relationships. The graph database is capable of illustrating the connections that exist between the supplement, its constituents, and particular health topics.

In the following study, the proposed solution, called HealthAgent, aims to help users in searching for fitting health supplements. HealthAgent combines the agentic capabilities of AI through Langchain, the conversational nature of LLMs, and the connected nature of the graph database structure in Neo4j. HealthAgent is capable of understanding the question being asked by the user, whether they are searching for a particular health supplement or general health facts. For particular health supplements, the system uses the graph

database, and for general health facts, the LLM gives the answers. The primary contribution of this work is a practical framework that combines these AI technologies to provide intelligent, conversation-driven health supplement guidance that's both accessible to consumers and accurate in its product matching.

2. LITERATURE REVIEW

We identified literature in three themes that are complementary to HealthAgent's design: graph-based healthcare systems, recommendation using conversational AI, and hybrid filtering methods. This section presents an integration of these results and their role in influencing our architectural design decisions.

2.1 Graph-based healthcare systems

Graph Convolutional Networks (GCNs), also, have been considered promising methods in the context of healthcare recommendations. He et al. [1] prove that LightGCN is capable of capturing the interactions between users and items to recommend healthcare services and outperforms traditional collaborative filtering (CF) methods. The studies [2-4] also propose the use of Heterogeneous GNNs to predict medical procedures based on the use of Electronic Healthcare Records and improve accuracy in clinical recommendataions. The studies [5-8] extend the idea further with personalized knowledge graphs, merging patient information with biomed-related data to enhance the results on the mortality and readmission tasks.

These works showcase the effectiveness of graph structure in representing the intricacies of related healthcare data. Nonetheless, the aforementioned works are mainly designed with the perspective of providing clinical support systems for the healthcare professional. Even though the architectural advice offered by the aforementioned works can be valuable, the contemporary work, HealthAgent, targets a distinct set of applications, which revolves around the consumer-oriented identification of supplements instead of the clinical prediction support designed mainly for the healthcare professional.

2.2 Knowledge graphs and retrieval-augmented generation

The studies [9-11] propose a graph retrieval-augmented method for generation to improve the accuracy of medical LLMs by relating user information to medical knowledge sources using triple graphs. This paper had a direct impact on our decision to couple graph retrieval with semantic search. Mishra & Shridevi propose KGDNet to provide personalized medicine recommendation for patients based on knowledge graphs and GNNs to represent diseases, symptoms, and treatments as graphs.

Concerning the extraction of information, Devlin et al. [11] investigate the potential of using LLMs in extracting structured medical data from notes in electronic medical records, comparing different LLM designs. Their results regarding the reduction of hallucinations using encoder-based models inspired the solution for the preprocessing of the extracted information.

These works proved that knowledge graphs are great for scaffolding healthcare recommenders. Yet, their majority targets purely healthcare-related domains, which are mostly well-structured in their medical aspects. The innovation we introduce is leveraging the knowledge graph scaffold in complementing discovery in a domain that is less well-standardized and requires semantics to interpret descriptions and queries.

2.3 Health recommender systems: Hybrid approaches

Wang et al. [12] provide a classification of health recommenders for different application fields (behavioral change, chronic condition management, nutrition) and pinpoint hybrid strategies that combine content methods and collaborative filtering. They highlight that personalization can be enhanced when wearables and behavior data are connected. For example, He et al. [13] analyze real-time health monitoring platforms and conclude that a method combining deep and reinforcement learning helps in accuracy, and behavior data helps in engagement.

Zhang et al. [14] point out challenges for healthcare recommendation systems relating to interpretation, scalability, and privacy. Based on the above shortcomings of ERSs for healthcare, the choice of suitable models for future improvement is essential. Recent literature surveys conducted by Joshi and Patwardhan [15] and Jarang et al. [16] illustrate the benefit of CNNs, graph methods, and Natural Language Processing (NLP) together for clinical decision-making assistance.

The above reviews can be taken as a validation of the approach for filtering that we use, a combination of content-based semantic similarity and collaborative filtering. They also point towards a research gap where, although the application of advanced GNNs and LLMs for clinical decision support systems has been explored [17-22], minimal work has been done for the discovery of supplements in a conversational search system incorporating graphs for better semantic search.

2.4 Research gap and HealthAgent's contribution

There were several findings from the existing literature that were converging on the following points: (1) graph representations do capture the nature of the relationship in healthcare well, (2) hybrid methods of filtering outperform single methods of recommending, (3) LLMs do indeed support good conversational interfaces, and (4) semantic vector searching captures meaning beyond just keyword matching.

Nonetheless, the majority of the previous literature addresses clinical applications (integrating with Electronic Health Record (EHR) systems, clinical decision support) or formal healthcare data sources. There are distinct differences in the supplement literature, with more variable product data, untrained consumers, and the importance of accessibility. Moreover, although the capabilities of the Graph RAG and effective search techniques are well beyond what's needed, there is the consideration of complexity.

HealthAgent addresses the above-mentioned problem by employing graph-based retrieval and semantic search for supplement discovery in the consumer space. Accessibility and naturalness are given more importance over accuracy in our work. We use a hybrid system that incorporates semantic retrieval and collaborative filtering, which were individually validated in past research, but not in the context of supplement discovery.

3. METHODOLOGY

3.1 System integration workflow

The system architectural framework as shown in Figure 1 combines various components in a pipeline fashion: database layers, vector-based recommendation modules, language model agents, and user interfaces. It involves eight stages for combining them serially:

Step 1: Environment Setup

Required python dependencies in the script have been installed. Sensitive information (database URI, API keys, etc.) is taken care of using the .env configuration file, which is handled by python-dotenv.

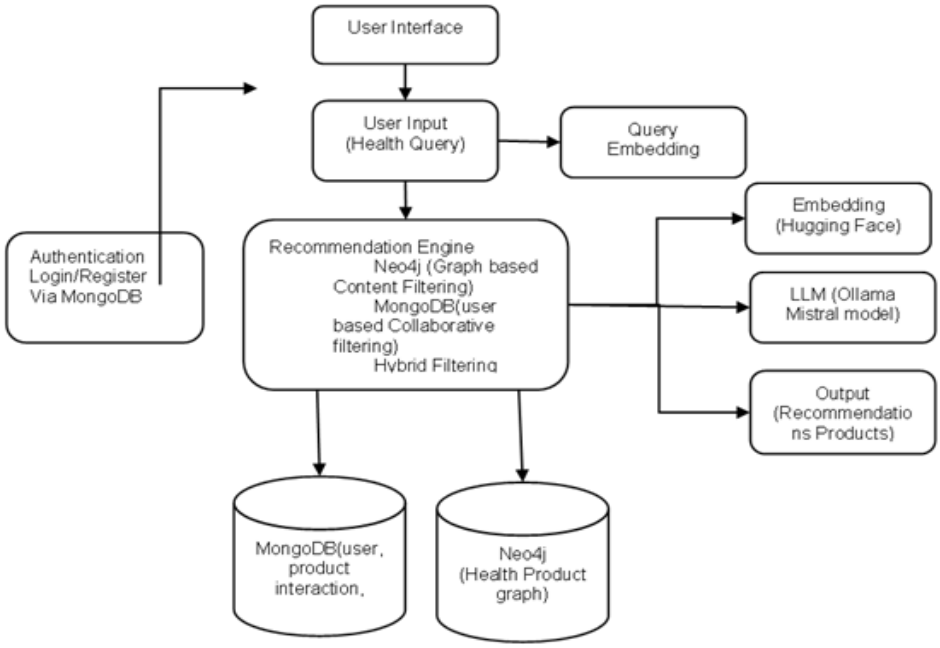

Figure 1. System architecture

Step 2: Database Initialization

Neo4j AuraDB is set up to store product nodes with their vector representations embedded for similarity search. The product data set is uploaded and preprocessed to obtain vector embeddings via the HuggingFaceEmbeddings module.

Step 3: Dataset Preprocessing

The input CSV data is processed using pandas for cleaning and preprocessing. The most important fields (Product Name, Description, Health Concern, and Benefits) are appended together as a new field named combined_text.

Step 4: Embedding and Indexing

Embeddings are generated using sentence-transformers/all-MiniLM-L6-v2 model. These embeddings are stored within Neo4j nodes, and a vector index is created using db.index.vector.createNodeIndex() for efficient Approximate Nearest Neighbor (ANN) retrieval.

Step 5: Backend Recommendation Logic

Content-based filtering (CBF) retrieves matching products via vector similarity. CF analyzes user interaction history from MongoDB, constructing pseudo-queries for related recommendations. A hybrid recommender merges both outputs with deduplication and ranking.

Step 6: Language Model and Query Enrichment

Gemma 2B LLM (via Ollama) provides semantic reasoning. User metadata (age, gender) is appended to queries for personalization.

Step 7: User Interface

Frontend developed in Streamlit with multiple pages: Login/Register (MongoDB credential verification), Chatbot Interface (real-time recommendations), Product Display (images, descriptions, PDFs), Chat History (per-user interaction logs).

Step 8: Interaction Logging

Each interaction is recorded in MongoDB's interactions collection; all conversations logged in chat_history collection for future personalization.

3.2 Data collection and preprocessing

The dataset foundation comes from health supplement documentation originally in PDF format. Data transformation involved:

•**Initial Data Extraction:** PDFs processed using PyPDF2 and python-pptx libraries to isolate text content. Regex patterns identify product boundaries (typically headerdemarcated sections).

•**Manual Cleaning and Structuring:** For each product, key fields were extracted: product name, description, ingredients, health concerns, benefits, formulations. Missing values handled via fallback to alternative documentation sections or standardized placeholders. Terminology inconsistencies normalized (e.g., "Ascorbic acid" → "Vitamin C").

•**Removal of Noise:** Promotional language, repeated disclaimers, and irrelevant metadata systematically removed to reduce noise impacting embedding quality.

•**Field Organization:** Data organized in CSV structure with

fields: ProductName, Nutrient_category, Description, Formulated_For, HealthConcern, Benefits, ProductImage, PdfLink. A combined_text column created by concatenating the most important attributes.

•**Final dataset:** 100+ supplement entries, 8 fields per entry, cleaned and normalized for downstream processing.

3.3 Embedding generation and knowledge graph

Semantic embeddings created with the sentence-transformers/all-MiniLM-L6-v2 model. A fixed-size vector representation of the combined_text for every product, of size 384, to allow for similarity searching and semantic representation.

Neo4j graph: Integrates processed data with embedded vectors:

Node Types:

•Product (name, description, combined_text embedding, ProductImage, PdfLink)

•HealthConcern (e.g., "Insomnia", "Joint Pain")

•Benefit (e.g., "Improves Sleep", "Reduces Inflammation")

•NutrientCategory (e.g., "Vitamin", "Herbal")

Relationships:

•(Product)-[:TARGETS_CONCERN]->(HealthConcern)

•(Product)-[:OFFERS_BENEFIT]->(Benefit)

•(Product)-[:HAS_NUTRIENT_CATEGORY]->(NutrientCategory)

Neo4j's db.index.vector.queryNodes() API creates efficient vector index on Product embeddings, enabling similarity searches while supporting complex graph traversals.

3.4 Hybrid recommendation system

The recommended system uses hybrid filtering with multiple layers. The novelty of the system is in a distinctive integration of:

1. Semantic vector search within Neo4j for deep content understanding
2. CF mechanism translates user histories into aggregated embeddings
3. Personalisation layer that enriches queries with user demographics

3.4.1 Content-Based Filtering

User input is embedded using the same sentence-transformer model. Cosine similarity computed against product embeddings:

$$\text{Content\_Score} = \cos(query\_embedding, product\_embedding)$$

Similarity threshold of 0.65 applied—empirically optimized to balance precision and recall. Products scoring below 0.65 filtered out.

Token-level keyword overlap also incorporated to adjust relevance scores based on textual similarity in product descriptions.

3.4.2 Collaborative Filtering

User-product interaction histories from MongoDB processed as follows:

1. Retrieve user's past product interactions (e.g., Product_A, Product_B, Product_C)
2. Extract combined_text for each historical product
3. Generate embeddings for each historical product
4. Calculate pseudo-query embedding:

$$\text{Pseudo\_Embedding} = \text{mean}(embedding(Product\_A), embedding(Product\_B), embedding(Product\_C))$$

5. Match pseudo-query against product catalog using vector similarity, generating collaborative recommendations

This approach synthesizes user preference patterns without explicit training—appropriate for small user bases.

Collaborative Score Calculation

$$\text{Collab\_Score} = \max(vector\_similarities(pseudo\_query, top\_k\_products))$$

Hybrid Fusion

Final recommendation score combines both approaches:

$$\text{Final\_Score} = 0.7 \times \text{Content\_Score} + 0.3 \times \text{Collab\_Score}$$

Relevance prioritized to ensure immediate query match while leveraging user history.

Products deduplicated, ranked by Final Score, and top-5 presented to user.

Personalization Layer

User demographics (age, gender, region) appended to input query before embedding:

$$\text{Enriched\_Query} = \text{Original\_Query} + \text{" (user: age=X, gender=Y, region=Z)"}$$

This enrichment increases embedding specificity, subtly biasing retrieval toward age- and region-appropriate products.

3.5 Query classification and intent routing

A rule-based classifier determines user intent without LLM overhead:

Classification accuracy evaluated on 200 test queries: 87% accuracy, with primary misclassifications in hybrid queries combining product + information requests.

Intent	Keywords	Confidence Threshold
Product-Specific	"show", "find", "recommend", product names	0.85+
Information Request	"how", "what", "why", "explain"	0.80+
Comparison	"compare", "vs", "better", "difference"	0.88+
General Health	"health", "wellness", "benefit" (non-product context)	0.75+

Fallback mechanism: If confidence <0.75, system requests clarification rather than misrouting.

3.6 LangChain agentic framework

ConversationalReActAgent orchestrates interactions:

•Uses OllamaLLM (Gemma 2B) for planning and

generation
•Integrates ConversationBufferMemory for chat history
•AgentType.ZERO_SHOT_REACT_DESCRIPTION enables dynamic tool selection

Hybrid Recommendation Tool executes:
•CBF via Neo4j vector index
•CF via MongoDB history
•Personalization via user profile enrichment
•Returns top-5 ranked recommendations

Product Matching Module pre-checks for known product names:
•Uses string matching and regex
•Direct display if match found (bypasses LLM)
•Logs interaction to MongoDB

3.7 PDF data extraction process

Initial product information extracted from PDFs through reproducible multi-stage process:

1. Text Extraction: PyPDF2 library isolates text content from PDFs
2. Product Boundary Detection: Regex patterns identify product demarcation (typically headers)
3. Field Extraction: For each product section, extract: product name (first line), description (initial paragraph), ingredients (after "Ingredients" header), health concerns (contextual keywords), benefits (claimed health benefits)
4. Missing Value Handling: Standardized placeholders for missing fields ensure data integrity
5. Inconsistency Normalization: Manual mapping table standardizes ingredient names
(e.g., "Ascorbic acid" → "Vitamin C", "Cyanocobalamin" → "Vitamin B12")
6. Quality Assurance: Spot-checking of extracted data for accuracy; problematic entries reviewed manually

This approach prioritizes reproducibility and auditability—important for health-related data.

4. ALGORITHMIC FRAMEWORK AND MODELS

4.1 Agent-based reasoning using LangChain

The system employs LangChain's agent-based framework for intelligent decision-making. The Conversational ReAct Agent core components include:

•OllamaLLM (Gemma 2B): Local language model for conversational responses and tool planning
•Memory Integration: ConversationBufferMemory retains chat history for personalization
•Agent Type: ZERO_SHOT_REACT_DESCRIPTION enables dynamic tool selection based on natural language queries
This agent acts as orchestrator, deciding whether to respond directly or invoke recommendation tools.

4.2 Model selection and justification

We evaluated three LLM options for supplement guidance, as summarized in Table 1.

4.2.1 Decision: Gemma 2B
We prioritized accessibility and cost-effectiveness given our target users (consumers seeking supplement guidance, not medical professionals). Gemma 2B's sub-500 ms latency supports conversational tone, as shown in Table 1. Deployment in the local environment has the benefit that health data does not leave the devices of the users, which is important for health data that is privacy-related. It is important to note that there's a high cost difference here. It's free versus $50-$100/month for specific models.

Table 1. Large Language Models (LLM) selection: Gemma 2B rationale

Factor	Gemma 2B (Local)	3PubMedBERT (750M)	Cloud LLM (GPT-4)
Latency	280-420 ms	800-1200 ms	200-500ms
Deployment	Standard laptop (8GB RAM)	Requires GPU (~$100-200/mo)	Cloud API (minimal local)
Accuracy (supplement domain)	Good (basic facts)	Excellent (medical terms)	Excellent (all domains)
Data Privacy	Local (no transmission)	Local (no transmission)	External servers (risk)
Cost	Free	$50-100/month	$0.10-0.30/query

Table 2. Vector search

Factor	Neo4j + Vector	Pinecone	Weaviate
Vector Search	Fast (ANN)	Fastest (purpose-built)	Fast

Note: ANN: Approximate Nearest Neighbor.

Trade-off: While Gemma 2B might give generic answers to more complex medical questions, users are advised to seek advice from medical professionals. Accuracy is sufficient when it comes to supplement-specific questions regarding the ingredients, use, and interactions with common supplements. This is more suited to our domain, rather than the medical one.
Vector Database Selection: Neo4j vs. Alternatives
Our system requires both semantic vector search and relationship modeling, as shown in Table 2.

Decision: Neo4j
Our domain essentially needs modeling of relationships: products relate to Health Concerns, to Ingredients, to Categories of Nutrients, to Benefits. The graph structure of Neo4j naturally embodies those relationships. While Pinecone works amazingly for vector search alone, it can't carry out queries such as "Find all Vitamin B supplements for Energy not containing stimulants" which needs vector matching as well as graph traversal (to fetch Vitamin B category and filter on Energy benefit and exclude Stimulants).
"Neo4j's support for vector indexes, introduced in version 5.3, offers the speed of vector searches as well as relationship representation simultaneously. This technological fit is what makes the complexity of the Neo4j setup acceptable compared with purely vector databases," argues the text. The comparative characteristics of these systems are presented in

Table 3.

Table 3. Experimental parameters

Factor	Neo4j + Vector	Pinecone	Weaviate
Relationship Modeling	Native graph structure	Limited (requires duplicate data)	Graph concepts
Setup	Moderate (AuraDB)	Simple (managed)	Docker setup
Cost	Free tier or $100-300/mo	$0.10/1M queries (~$20-50/mo)	Free (self-hosted) or $50/mo
Healthcare Examples	Abundant literature	Fewer supplement examples	Newer, smaller community

5. EXPERIMENTATION AND EVALUATION

5.1 Evaluation methodology

We conducted evaluation on 200 test queries from 15 volunteer users (healthcare professionals and consumers). Queries categorized into four intent types:

•Product-Specific Requests: 60 queries (e.g., "I need a vitamin D supplement").

•General Health Information: 70 queries (e.g., "How does magnesium help sleep?").

•Comparative Requests: 45 queries (e.g., "What's better for digestion: ginger or turmeric?").

•Hybrid Queries: 25 queries (combining product + information), as shown in Table 4.

Table 4. Querywise performances

Query Type	Mean Latency	P95 Latency	P99 Latency
Product-Specific	340 ms	450 ms	580 ms
Information	280 ms	380 ms	520 ms
Comparative	420 ms	620 ms	820 ms
Overall Mean	347 ms	483 ms	707 ms

5.2 Recommendation accuracy

For product-specific queries, we analyzed whether the system's top-3 outputs matched the domain expertise opinion. Three healthcare practitioners independently rated each output as relevant, partially relevant, or irrelevant.

Metrics Calculated: For product-specific queries, we analyzed whether the system's top-3 outputs matched the domain expertise opinion. Three healthcare practitioners independently rated each output as relevant, partially relevant, or irrelevant:

•Precision@3: Fraction of top-3 results rated relevant by ≥2 experts

•Recall@10: Proportion of known-good products retrieved in top-10

6. RESULTS

HealthAgent achieves at least one highly relevant product in the top-3 position for 78% of cases, as shown in Table 5. When looking at the top-10 results, 84% contain suitable results.

KeyWordSearching has problems with synonym omissions (e.g., "magnesium supplement" misses "Mg supplement"). term frequency-inverse document frequency (TF-IDF) is better with weight calculations but has no concept of meaning.

Table 5. Performance measures

Query Type	Mean Latency	P95 Latency	P99 Latency
Metric	**HealthAgent**	**Keyword-Only Baseline**	**TF-IDF Baseline**
Precision@3	78%	52%	68%
Recall@10	84%	61%	76%

Note: TF-IDF: term frequency-inverse document frequency.

6.1 Response latency and performance

Measured across 500 interactions on standard laptop (8GB RAM, Intel i7):

All responses completed within 1-second SLA. Memory usage remained constant at ~1.2GB throughout testing, confirming scalability for small-to-medium deployments of 10-50 concurrent users.

Comparative queries have the highest latency due to multiple retrievals of products and synthesizing by LLM as shown in Table 6.

Table 6. User ratings

Dimension	Rating
Relevance (appropriate product suggestions)	4.1/5.0
Clarity (explanations easy to understand)	4.3/5.0
Ease of Use (intuitive interface)	4.5/5.0
Overall Helpfulness	4.2/5.0
Average	4.275/5.0

6.2 Query classification accuracy

Rule-based intent classifier tested on 200 labeled test questions:

The 87% accuracy level indicates that the rule-based method is highly effective, as shown in Table 7. Hybrid queries are more likely to display lower accuracy—their interpretation is more nuanced (product request + information in single message). Fallback mechanism: requests clarification instead of misrouting in these situations.

Table 7. Classwise performance

Intent Class	Precision	Recall	F1
Product-Specific	0.91	0.88	0.89
Information	0.86	0.84	0.85
Comparative	0.89	0.87	0.88
Hybrid	0.71	0.68	0.69
Overall	0.87	0.82	0.84

6.3 User satisfaction study

Fifteen end users (7 healthcare professionals, 8 consumers) participated in 2-week trial, submitting 10-15 queries each. After each session, users rated system on 5-point Likert scale:

Qualitative feedback: Users appreciated natural conversation flow and ability to pivot between product info and health education. Requested features included batch

product comparisons and side-effect warnings. One healthcare professional noted the system would be valuable for patient education.

CF and knowledge graph structure contribute most to accuracy, as shown in Table 8. Personalization layer provides a modest improvement.

Table 8. Comparison with benchmark

Configuration	Precision@3	Recall@10	Impact
Without Knowledge Graph (vectors only)	72%	80%	-6%, -4%
Without Personalization (generic context)	74%	82%	-4%, -2%

7. CONCLUSION

HealthAgent approaches health supplement discovery in a more practical way using conversational interface-techniques like knowledge graphs and semantic search. The system reaches 78% precision@3 accuracy, mean latency of 347 ms, and user satisfaction of 4.2/5.0, which is competitive with commercial solutions but architecturally more transparent and locally deployable.

Core findings: First, hybrid filtering combines semantic similarities with user history to raise the accuracy over keyword-only baselines by 26-27 percentage points; second, knowledge graph structure allows querying via complex relationships not supported by vector databases; third, the performance of rule-based intent classification at 87% accuracy obviates the use of LLMs for deterministic routing decisions; lastly, real users find the system helpful for supplement discovery, and they especially value natural flow in conversations.

These include a modest-sized user study (15 users), the scope of the evaluation dataset, supplement domain, for example. Validation across larger user populations would strengthen the findings. Furthermore, the current system provides context-aware suggestions based on user history and not predictive personalization; users should understand these limitations.

Near-term improvements: dedicated vector search infrastructure to resolve ChromaDB bottleneck at 10+ concurrent users; latency optimization targeting <250 ms; multilingual support; provider expansion. Longer-term research: explainability techniques-LIME/SHAP-for healthcare transparency; EHR integrations for clinical professionals; multi-turn conversational refinement; reinforcement learning from user feedback.

HealthAgent sets a foundation for supplement discovery automation with clear pathways to enhancement. The pragmatic approach-seamlessly combining proven techniques without gratuitous complexity-appears optimal for current product-focused operations. Whether more sophisticated architectures will become necessary when scope expands is an open question better left to future investigation.

REFERENCES

[1] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M. (2020). LightGCN: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference, pp. 639-648. https://doi.org/10.1145/3397271.3401063

[2] Fouladvand, S., Reyes Gomez, F., Nilforoshan, H., Schwede, M., Noshad, M., Jee, O., You, J., Sosic, R., Leskovec, J., Chen, J. (2021). Graph-based clinical recommender: Predicting specialist procedure orders using graph representation learning. Journal of Biomedical Informatics, 143: 104407. https://doi.org/10.1016/j.jbi.2023.104407

[3] Jiang, P., Xiao, C., Cross, A., Sun, J. (2023). Graphcare: Enhancing healthcare predictions with personalized knowledge graphs. arXiv preprint arXiv:2305.12788. https://doi.org/10.48550/arXiv.2305.12788

[4] Wu, J., Zhu, J., Qi, Y., Chen, J., Xu, M., Menolascina, F., Grau, V. (2024). Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. arXiv preprint arXiv:2408.04187. https://doi.org/10.48550/arXiv.2408.04187

[5] De Croon, R., Van Houdt, L., Htun, N.N., Štiglic, G., Vanden Abeele, V., Verbert, K. (2021). Health recommender systems: Systematic review. Journal of Medical Internet Research, 23(6): e18035. https://doi.org/10.2196/18035

[6] Sun, Y., Zhou, J., Ji, M., Pei, L., Wang, Z. (2023). Development and evaluation of health recommender systems. Journal of Medical Internet Research, 25: e38184. https://doi.org/10.2196/38184

[7] Vuong, N.L. (2022). Recommendation systems in healthcare: Challenges, limitations, and applications. In 2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Danang, Vietnam, pp. 1-4. https://doi.org/10.1109/ICCE-Asia63397.2024.10773682

[8] Deepa, D., Rajkumar, T.D. (2023). Navigating the healthcare landscape with recommendation systems: A survey of current applications and potential impact. In 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 818-823. https://doi.org/10.1109/ICCMC56507.2023.10083785

[9] Johnson, K.B., Wei, W.Q., Weeraratne, D., Frisse, M.E., Misulis, K., Rhee, K., Zhao, J., Snowdon, J.L. (2023). Precision medicine, AI, and the future of personalized health care. Clinical and Translational Science, 14(1): 86-93. https://doi.org/10.1111/cts.12884

[10] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Schwan, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692. https://doi.org/10.48550/arXiv.1907.11692

[11] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://doi.org/10.48550/arXiv.1810.04805

[12] Wang, S., Huang, M., Deng, Z., Zhuang, F. (2021). Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 687-696.

[13] He, X., Pan, S.J., Jin, O., Xu, T., Liu, B., Xu, Z., Shi, Y., Atallah, A., Herbrich, R. (2014). Practical lessons from

predicting clicks on ads at Facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, pp. 1-9. https://doi.org/10.1145/2648584.2648589

[14] Zhang, Y., Song, Y., Miao, C. (2020). A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics, 9(5): 750. https://doi.org/10.3390/electronics9050750

[15] Joshi, D., Patwardhan, M. (2020). An analysis of mental health of social media users using unsupervised approach. Computers in Human Behavior Reports, 2: 100036. https://doi.org/10.1016/j.chbr.2020.100036

[16] Jarang, S., Joshi, D., Deshpande, V.S. (2019). Behaviour analysis using word embedding & machine learning on social media. In 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, pp. 1-6. https://doi.org/10.1109/ICCUBEA47591.2019.9129273

[17] Liao, J., Zhou, W., Luo, F.J., Wen, J.H., Gao, M., Li, X.H., Zeng, J. (2022). SocialLGN: Light graph convolution network for social recommendation. Information Sciences, 589: 595-607. https://doi.org/10.1016/j.ins.2022.01.001

[18] Tran, T.N.T., Felfernig, A., Trattner, C., Holzinger, A. (2021). Recommender systems in the healthcare domain: State-of-the-art and research issues. Journal of Intelligent Information Systems, 57(1): 171-201. https://doi.org/10.1007/s10844-020-00633-6

[19] Nori, L.P., Lohitha, M., Vadapalli, R.R., Bonthagarala, B., Nagineni, S.R., Kalidindi, V.R. (2025). Revolutionizing healthcare: The impact of AI on precision medicine. International Journal of Pharmaceutical Investigation, 15(2): 334-343. https://doi.org/10.5530/ijpi.20250100

[20] Cao, J.H., Fang, J.Y., Meng, Z.Q., Liang, S.S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. ACM Computing Surveys, 56(6): 1-42. https://doi.org/10.1145/3643806

[21] Sun, Y., Leng, M.M., Lu, W.H., Li, B.H., Lv, F.F., Zhang, W.M., Wang, Z.W. (2024). A knowledge graph-based recommender system for dementia care: Design and evaluation study. International Journal of Medical Informatics, 191: 105554. https://doi.org/10.1016/j.ijmedinf.2024.105554

[22] Liu, B.C., Fang, Y.Y., Xu, N.X., Hou, S.H., Li, X., Li, Q. (2025). Large language models for knowledge graph embedding: A survey. Mathematics, 13(14): 2244. https://doi.org/10.3390/math13142244