International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V74I6P103

Volume 74 Issue 6, 22-32, June 2026

Original Article

Hierarchical Agentic RAG Framework for Intelligent Product Discovery in Distributed E-Commerce Microservices Using MCP and Dynamic Context-Aware Vector Intelligence

Dhruv Kumar Seth[1], Karan Kumar Ratra[2]

[1,2]*Walmart Global Tech, Sunnyvale, CA, USA.*

[1]*Corresponding Author : er.dhruv08@gmail.com*

Received: 18 April 2026 Revised: 24 May 2026 Accepted: 13 June 2026 Published: 30 June 2026

Abstract *- Modern e-commerce platforms increasingly rely on distributed microservice architectures in which product catalog management, search, recommendation, pricing, inventory, promotions, and order fulfillment operate as independent services. Product information and operational signals are distributed across these services and must be coordinated during query processing. Consequently, product discovery extends beyond traditional search by integrating semantic relevance, user context, and real-time operational constraints across multiple services. This paper presents HAAF, a Hierarchical Agentic RAG Framework for intelligent product discovery in distributed e-commerce environments. The framework introduces collaborative multi-agent orchestration, integrated with Model Context Protocol (MCP)-based inter-service communication and Dynamic Context-Aware Vector Intelligence (DCVI). DCVI continuously captures contextual information from multiple dimensions, including user behavior, session activity, product relationships, inventory conditions, and real-time events, and combines them using an adaptive weighting mechanism. HAAF integrates Reciprocal Rank Fusion (RRF) to combine dense semantic retrieval and sparse keyword-based retrieval within a unified architecture. The framework is designed to support flexible, context-sensitive product discovery across distributed e-commerce environments and serves as an architectural foundation for future empirical evaluation and large-scale implementation.*

Keywords *- Agentic AI, Retrieval-Augmented Generation, E-Commerce Microservices, Intelligent Product Discovery, Model Context Protocol, Vector Databases, Multi-Agent Systems, Dynamic Context Intelligence, Reciprocal Rank Fusion, Distributed Systems.*

1. Introduction

Large-scale e-commerce platforms have evolved into distributed ecosystems composed of independently deployable microservices responsible for catalog management, semantic and keyword-based retrieval, personalization, pricing, inventory management, customer analytics, and order fulfillment [7][8].

Within such environments, product discovery becomes a cross-service coordination problem in which retrieval relevance, operational constraints, and contextual user signals must be evaluated collectively. The quality of product discovery directly influences user engagement, conversion behavior, and overall platform efficiency.

Conventional product discovery pipelines rely predominantly on keyword matching or collaborative filtering derived from historical interaction logs. Although these techniques remain effective in broad recommendation settings, they exhibit known deficiencies under dynamic conditions: response to real-time intent shifts is limited, adaptation to live inventory changes is slow, and integration of heterogeneous contextual signals from independent services is architecturally constrained.

Retrieval-Augmented Generation (RAG) [1] has demonstrated that dynamically retrieved contextual evidence can improve the quality of reasoning and factual consistency in knowledge-intensive tasks. Dense Passage Retrieval (DPR) [2] and related dense retrieval architectures have further improved semantic retrieval capabilities, while Self-RAG [3] introduced adaptive retrieval strategies for generative systems. However, these approaches generally assume a single retrieval pipeline operating over a relatively centralized knowledge source and do not address coordinated retrieval across distributed microservice environments.

Recent advances in multi-agent systems demonstrate that hierarchical task decomposition, agent specialization, and tool-augmented reasoning improve decision-making in complex distributed environments. The Model Context Protocol (MCP) provides a standardized interface for schema-driven communication among agents, tools, and

services, enabling structured context propagation across service boundaries. The integration of hierarchical agent orchestration, MCP-based inter-service communication, and context-aware retrieval establishes a unified architectural foundation for intelligent product discovery at scale. This architecture enables retrieval systems to adapt to shifting user intent, live operational constraints, and heterogeneous service environments while preserving contextual consistency across distributed services throughout the product discovery pipeline. Despite recent advances in Retrieval-Augmented Generation, multi-agent systems, and context-aware retrieval, existing literature has not reported a unified framework that combines hierarchical agent orchestration, MCP-based service coordination, dynamic context-aware vector intelligence, and hybrid retrieval for distributed e-commerce product discovery.

This gap motivates the proposed Hierarchical Agentic RAG Framework (HAAF), which integrates these capabilities within a single architectural model for distributed microservice environments. Accordingly, the objective of this paper is to define a unified conceptual architecture for distributed e-commerce product discovery by integrating hierarchical agent orchestration, MCP-based service coordination, Dynamic Context-Aware Vector Intelligence (DCVI), and hybrid retrieval, and to establish a foundation for future implementation and empirical evaluation.

1.1. Contributions

This paper makes four primary contributions. First, it proposes HAAF, a six-role hierarchical agent framework that integrates MCP-based inter-service communication to support intelligent product discovery in distributed e-commerce microservices. Second, it introduces Dynamic Context-Aware Vector Intelligence (DCVI), a five-component contextual representation model that combines user, session, product, inventory, and event information and uses adaptive weighting to enable efficient context updates.

Third, it presents a hybrid retrieval architecture based on Reciprocal Rank Fusion (RRF) [23], enabling dense semantic retrieval and sparse keyword retrieval to operate within a unified retrieval pipeline. Finally, we present a conceptual analysis and four case studies that show how the architecture of HAAF supports adaptive retrieval, inventory-aware ranking, and decision-making in distributed microservices for e-commerce.

The key contribution of our framework, compared to existing RAG architectures, is the integration of hierarchical agent orchestration, MCP-based service coordination, dynamic, context-aware vector intelligence, and a hybrid retrieval architecture. In contrast to most RAG architectures, which employ a centralized retrieval pipeline with isolated knowledge sources for search, HAAF provides a unified architecture for distributed search across heterogeneous microservice environments. Consequently, HAAF efficiently supports context propagation, inventory-aware ranking, and adaptive product discovery in complex distributed e-commerce systems.

1.2. Paper Organization

Section II reviews related work. Section III defines the conceptual problem and identifies key challenges in distributed e-commerce product discovery. Section IV presents the HAAF architecture and agent hierarchy. Section V introduces the framework formulation, including adaptive contextual representations and hybrid retrieval mechanisms. Section VI discusses the conceptual evaluation perspective and comparison dimensions. Section VII presents illustrative case studies. Section VIII discusses scalability considerations, operational limitations, and ethical concerns. Section IX concludes the paper and outlines future research directions.

2. Background And Related Work

2.1. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) [1] introduced a framework that combines neural retrieval with generative language models to improve factual grounding in knowledge-intensive tasks. Dense Passage Retrieval (DPR) [2] further advanced dense semantic retrieval through bi-encoder architectures optimized for open-domain question answering. Subsequent work extended retrieval-aware reasoning capabilities through adaptive and self-reflective retrieval strategies. Self-RAG [3], for example, incorporated retrieval-aware generation and self-evaluation mechanisms to improve contextual reasoning during inference.

Subsequent work explored alternative integration strategies. REALM [9] incorporated retrieval directly into the language model pre-training phase, while REPLUG [18] demonstrated that external retrieval modules can improve large language models without task-specific retraining.

Despite these advances, existing RAG architectures generally assume centralized retrieval pipelines operating over relatively static knowledge sources. These approaches do not explicitly address coordinated retrieval across heterogeneous microservices, real-time operational constraints, or hierarchical multi-agent orchestration within distributed e-commerce environments.

2.2. Multi-Agent and Tool-Augmented Systems

Yao et al. [4] proposed ReAct, interleaving chain-of-thought reasoning with tool invocations, demonstrating improved multi-step task completion. Wu et al. [5] introduced AutoGen for multi-agent conversational workflows with role-differentiated agents. Schick et al. [6] showed in Toolformer that language models can learn to invoke APIs through self-supervised annotation. Park et al. [19] simulated multi-agent social environments and demonstrated emergent collaborative behaviors. These frameworks are not designed for the latency, throughput, and real-time data constraints of production e-commerce retrieval systems.

2.3. E-Commerce Recommendation

He et al. [7] replaced the inner product of matrix factorization with a multi-layer perceptron in Neural Collaborative Filtering, establishing a neural baseline for

user-item preference modeling. Covington et al. [8] described YouTube's two-stage deep neural network recommendation pipeline combining candidate generation with a ranking network trained on watch time signals. Session-based approaches use recurrent models [20] or transformers (BERT4Rec [22]) to capture within-session intent. Sun et al. [21] evaluated large language models as zero-shot rankers for recommendation, finding they can match supervised baselines with appropriate prompting. These works do not address the cross-service, inventory-aware retrieval setting.

2.4. Dense Retrieval and Vector Databases

Devlin et al. [10] introduced BERT, which provides contextualized representations that underpin modern dense retrievers. Johnson et al. [11] developed FAISS, which provides GPU-accelerated Approximate Nearest-Neighbor (ANN) search using inverted-file indexes and product quantization. Malkov and Yashunin [12] characterized HNSW, which provides ANN search with O(log n) query complexity and theoretical convergence guarantees. Modern vector databases—Milvus [24], Pinecone, Weaviate, ChromaDB—extend these foundations into production-scale systems that support real-time upserts, filtering, and hybrid search.

2.5. Reciprocal Rank Fusion

Cormack et al. [23] introduced RRF as a parameter-free rank aggregation method that consistently outperforms Condorcet methods and learned combination approaches on ad hoc retrieval benchmarks. The method requires no calibration of system score scales, making it suitable for combining heterogeneous retrieval systems. HAAF uses RRF to fuse dense and sparse retrieval results.

Although prior work has explored RAG, dense retrieval, recommendation systems, and multi-agent reasoning independently, limited work has examined their integration within distributed e-commerce microservice environments. HAAF addresses this gap by proposing a unified conceptual architecture for agentic product discovery with MCP-based coordination, context-aware vector intelligence, and inventory-aware retrieval.

3. Problem Definition

3.1. Conceptual Problem Statement

Many large-scale e-commerce sites today are developed as a collection of distributed microservices. For product discovery, typical signals include product information displayed on product detail pages, search queries, product recommendations generated by recommendation components, and real-time information on inventory, prices, product eligibility for a given price, and other regional information.

These signals can drive product discovery, and product discovery is not just search anymore; search and product recommendations are two of the signals. Search and product recommendations are two of the signals, and in addition to these static signals, product discovery must also take into account in real-time information regarding product availability, whether a product is eligible for a given price, and other regional information.

In a distributed microservice architecture, these signals are often fragmented across independent services with different data models, update frequencies, and access patterns. A product may satisfy semantic relevance criteria yet fail inventory or eligibility checks managed by separate services. Conversely, user intent may shift during an active session, rendering statically computed retrieval results less accurate over time.

This paper defines the core problem as follows: how can an e-commerce product discovery system coordinate multiple microservices and contextual signals to generate more relevant, inventory-aware, and session-aware product results without relying on a single static retrieval pipeline?

Rather than presenting an experimentally validated optimization model or production deployment study, this paper proposes a conceptual framework for intelligent product discovery in distributed e-commerce environments. The proposed framework combines hierarchical agent orchestration, MCP-based service integration, hybrid retrieval, and dynamic context-aware vector intelligence to address the challenges of fragmented information, session intent drift, real-time operational constraints, and cross-service context propagation. The objective is to provide an architectural foundation for future implementation and empirical evaluation.

3.2. Identified Challenges

Five key challenges motivate the HAAF design:

(C1) Fragmented Information: Relevant signals are distributed across heterogeneous microservices with differing schemas and data representations.

(C2) Session Intent Drift: User intent may evolve during a browsing session, making static retrieval and recommendation models less effective.

(C3) Real-Time Constraint Propagation: Dynamic factors such as inventory status, pricing, and availability may change continuously and require timely propagation across services.

(C4) Context Coherence: Conventional service-to-service interactions can lose contextual information across boundaries, reducing consistency in downstream decision-making.

(C5) Scalable Adaptive Retrieval: Continuously adapting retrieval mechanisms and embedding representations at a large scale introduces computational and system efficiency challenges.

4. Haaf Architecture

4.1. Overview

HAAF organizes its components into five functional layers: (L1) User Interaction, (L2) Hierarchical Agents, (L3) MCP Communication, (L4) Microservices, and (L5) DCVI with Vector Storage. Figure. 1 shows the full architecture. Figure. 3 shows the inter-agent sequence for a single query.

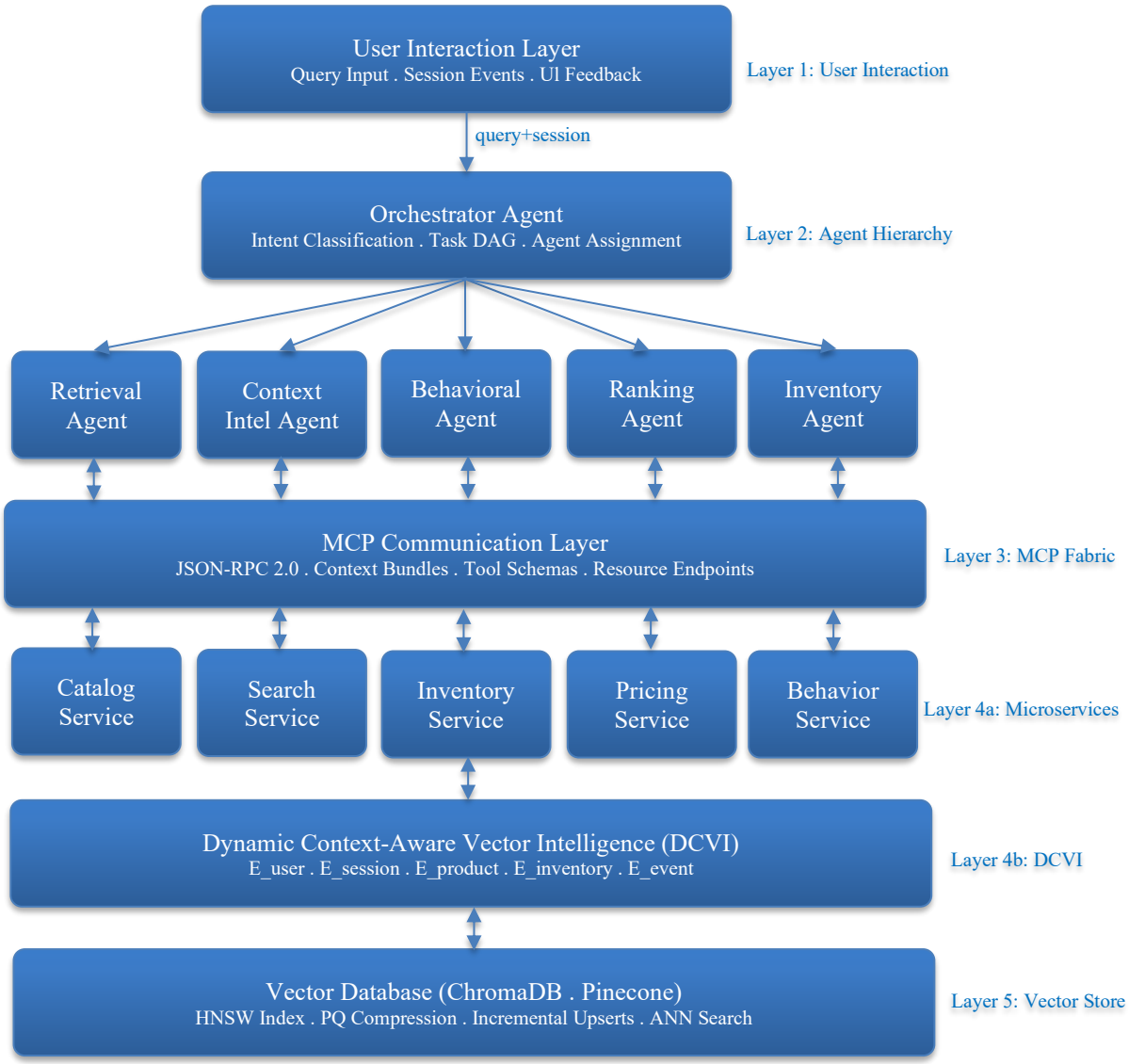

Fig. 1 HAAF five-layer system architecture

4.2. Agent Roles

The proposed HAAF architecture adopts a two-tier agent hierarchy consisting of an Orchestrator Agent and specialized worker agents. This design separates coordination logic from domain-specific reasoning, enabling modular scalability.

4.2.1. Orchestrator Agent

The Orchestrator Agent receives a user query Q and a session context S as input. The query intent is first identified by a lightweight language-understanding component. Next, the query Q is decomposed into a set of sub-tasks, and a Directed Acyclic Graph (DAG) of tasks is generated.

Each task in the graph is then dispatched to the corresponding worker agents of the conversational system with MCP-based task contracts. A worker agent then aggregates the intermediate results of a task and updates the contextual state representations for the subsequent tasks in the conversational flow.

4.2.2. Retrieval Agent

The Retrieval Agent performs a hybrid search for candidate solutions, using a dense semantic search and a sparse keyword search. The results are then fused by ranking aggregation techniques, such as Reciprocal Rank Fusion (RRF), to produce a set of candidate solutions for the following agents.

4.2.3. Context Intelligence Agent

Computes contextual relevance between candidate representations and accumulated user context. It may further exploit product relationship graphs to identify related or complementary products beyond the initial retrieval set.

4.2.4. Behavioral Analysis Agent

Analyzes the interaction on the session level to extract information on the user's behavior, such as his/her browsing behavior and level of engagement, as well as his/her time-evolving intent. The behavioral analysis produced in this process is used to improve the ranking and to adapt the context produced by other retrieval agents.

4.2.5. Ranking Agent

Generates final ranking scores by integrating relevance signals, contextual alignment, behavioral indicators, and domain-specific quality measures. Learning-to-rank or adaptive optimization strategies may be employed to improve ranking effectiveness.

4.2.6. Inventory and Pricing Agent

Interacts with inventory and pricing services to incorporate real-time operational constraints such as availability and pricing conditions into the retrieval and ranking pipeline.

4.3. Dynamic Context-Aware Vector Intelligence

DCVI maintains five embedding components in a shared vector database.

Figure 2 shows the embedding lifecycle and adaptive combination.

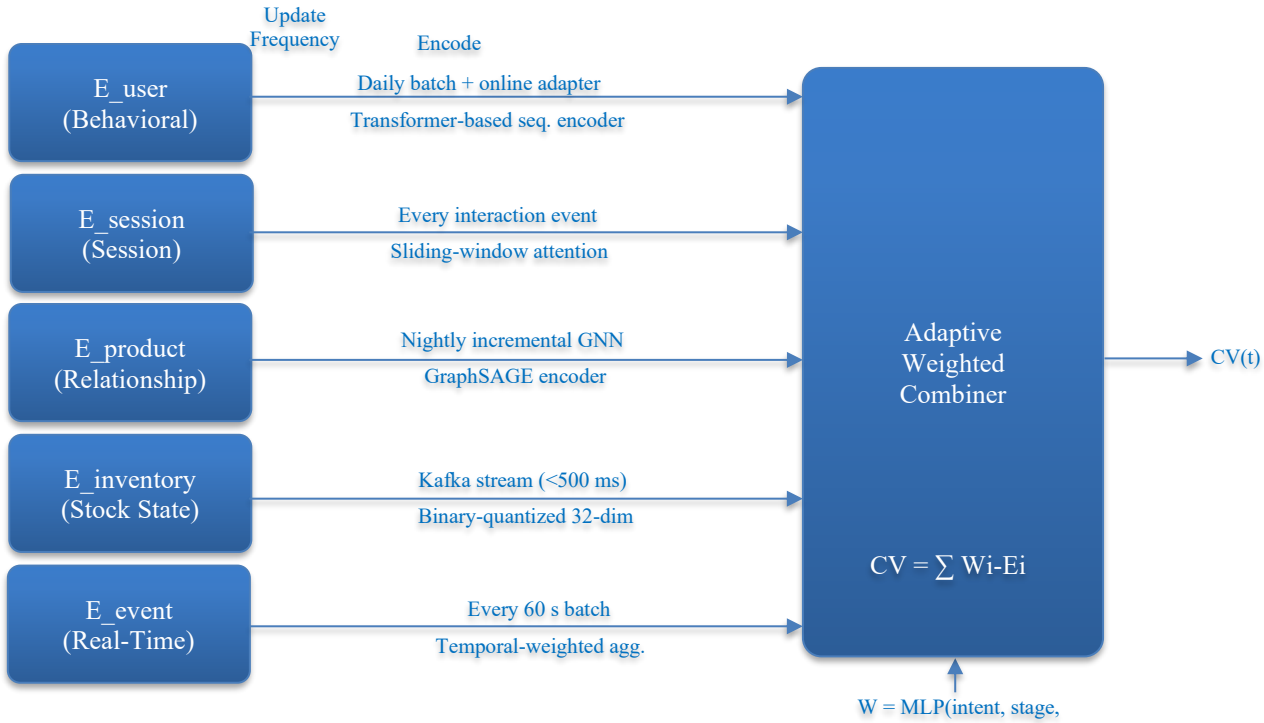

Fig. 2 DCVI five-component embedding lifecycle and adaptive combination

4.3.1. E_user

Represents Long-term user preferences of registered users from past interactions. Using sequential representation learning, long-term user preferences can be updated periodically as new interactions are incorporated.

4.3.2. E_session

Captures short-term session context of recent interaction events in the form of a sequence. This encodes the current user's browsing intent and can be updated incrementally.

4.3.3. E_product

The product information is represented as a graph, where links between products reflect certain relations (e.g., co-purchase, complementary products, substituting products, etc.).

4.3.4. E_inventory

This entity represents the operational state of the online shop. The information necessary to retrieve product information includes, for example, product availability, up-to-date stock information, and order processing (e.g., shipping, picking). Because this entity's information changes frequently, it must be taken into account when retrieving results.

4.3.5. E_event

Captures temporally evolving signals, including active promotions, trending products, and changes in user demand patterns that may influence retrieval relevance.

4.4. MCP Communication Layer

Each microservice exposes an MCP server defining resource endpoints, tool schemas, and context propagation headers. Each request to an MCP server contains a context bundle with the user's session ID, the PCA-compressed context vector (64 dimensions for efficient transmission), the user's current intent, and the accumulated candidate set for which finalization is performed. Moreover, the context bundle contains a priority score computed by the Orchestrator.

Fig. 3 depicts the sequence of inter-agent interactions via the MCP protocol.

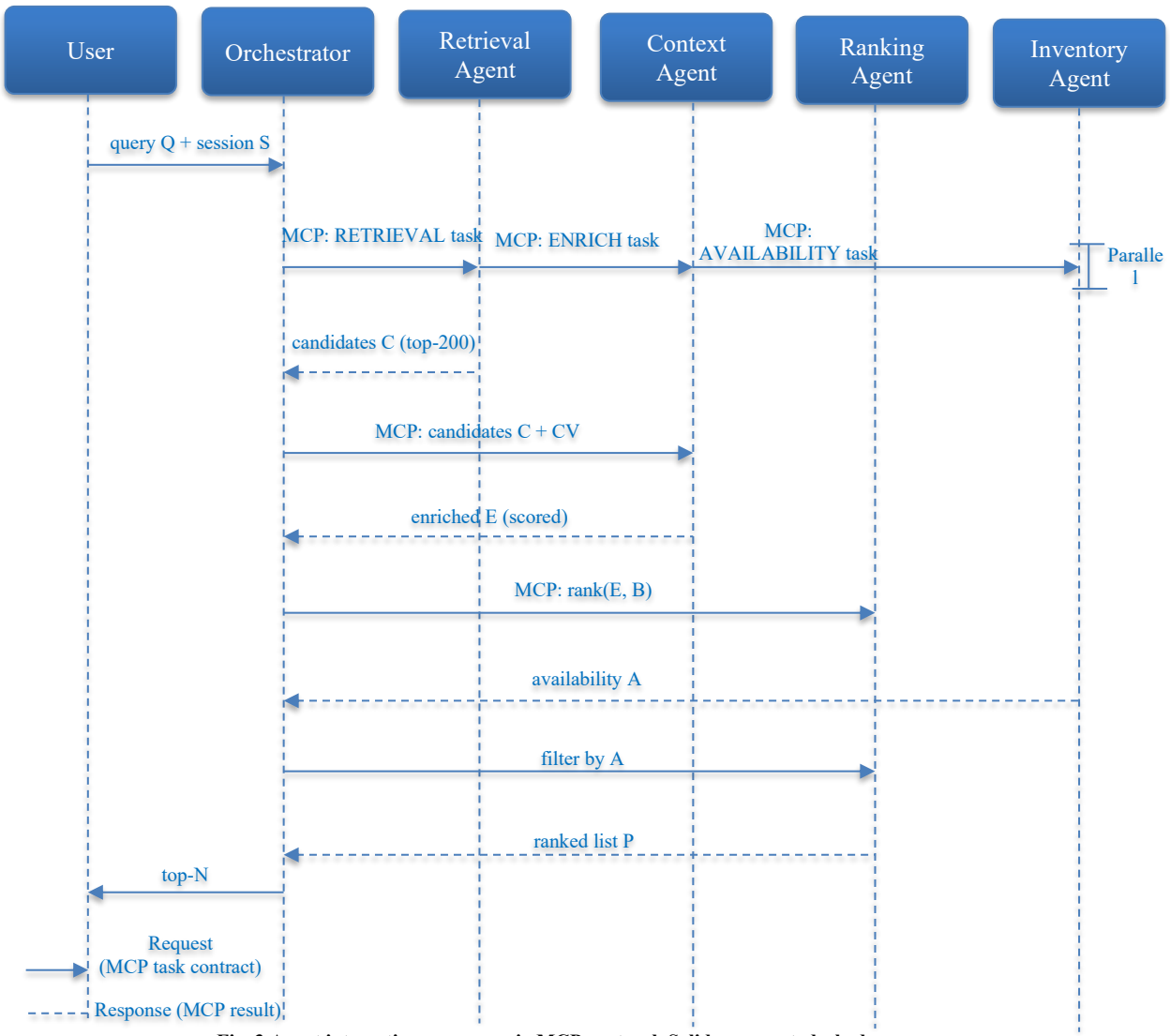

Fig. 3 Agent interaction sequence via MCP protocol. Solid = request, dashed = response

5. Framework Formulation

5.1. Adaptive DCVI Weighting

The Dynamic Context Vector integrates multiple contextual dimensions, including long-term behavioral preferences, short-term session activity, product relationships, inventory conditions, and temporally evolving event signals. The relative importance of these contextual dimensions is dynamic and can change significantly based on factors such as the user's intent, the session's progress, and operational changes. For example, during an active browsing session, session-level information would receive more weight than long-term information from past sessions. In addition, during periods of significant change in operational circumstances (e.g., stockout information), local inventory and event information would be given greater weight.

5.2. Incremental DCVI Update

User and session representations are updated on the fly as more interaction events are observed. The corresponding behavioral or session representation is updated via an exponential moving average for each observed interaction event. Thus, recent user behavior is taken into account, while the most relevant contextual information is kept up to date.

$$E\_user \leftarrow E\_user + \eta \cdot (enc(e_t) - E\_user) \qquad (2)$$

where η represents a session-level learning rate and $enc(e_t)$ denotes the encoded representation of the current interaction event.

This formulation uses an exponential moving-average approach that gradually updates user representations as new interactions occur. Compared with complete re-indexing, incremental updates reduce computational overhead because only the affected embedding dimensions are updated. The per-event computational complexity is $O(d)$, where d is the embedding dimensionality, rather than $O(n \cdot d)$ for a full recomputation of the embedding across the product catalog.

The incremental update mechanism is proposed as a design strategy to support scalability and responsiveness in large-scale distributed environments while maintaining context continuity across user sessions.

5.3. Reciprocal Rank Fusion

HAAF conceptually combines dense semantic retrieval and sparse keyword-based retrieval using Reciprocal Rank Fusion (RRF) [23]. By ranking aggregates of results rather than individual documents, RRF can merge rankings from multiple differently distributed scores without first calibrating them to a common scale. This makes RRF well-suited for densely semantically aware retrieval, as well as for sparse keyword-aware retrieval (e.g., using BM25), and thus for retrieval that combines many different semantic methods with many sparse retrieval methods.

Hybrid retrieval combines the semantic strengths of dense retrieval with the lexical precision of sparse retrieval, improving robustness across diverse product search scenarios in distributed e-commerce environments. It achieves the best results in context and semantics through the densest retrieval methods and the most accurate keyword search. On the one hand, hybrid retrieval improves the robustness of Product Search and supports different search scenarios for distributed e-commerce environments. The combined retrieval strategy is intended to improve robustness and adaptability across heterogeneous search scenarios in distributed e-commerce environments.

5.4. Context-Aware Ranking Strategy

HAAF, as part of the product discovery flow within the product retrieval, enrichment, and ranking component, ranks products using a multifactorial decision process. The decision process considers query-product relevance, session context, historical user behavior, product relationships, product stock, and other e-commerce-specific ranking constraints.

Unlike traditional search ranking approaches that rely mostly on fixed scores within a result set (e.g., query-product relevance scores), context-aware ranking in the e-commerce setting takes into account an increasing number of dynamic factors as the user interacts during a session. For example, the most recent user interactions with products (e.g., products that have gone out of stock vs. those that have just come back in stock), active promotions and discount events, user preferences, etc.

By developing HAAF as a conceptual framework for ranking, the process is viewed as a multi-factor decision. When products are retrieved and enriched by the corresponding services, all signals are aggregated to maximize the ranking of recommended products.

The objective is to provide product results that remain responsive to both user intent and operational conditions within distributed e-commerce environments.

5.5. Adaptive Agent Coordination

An Orchestrator Agent within the HAAF framework controls multiple worker agents. The process of orchestrating single-worker agents is not strictly linear and can vary for many reasons (e.g., due to queries, session state, or system state).

This adaptive coordination is the key to making the distributed system as flexible as possible. The services might be available or unavailable; the workload can change dramatically, and the context of a query can be completely different from one query to the next. In heterogeneous microservice ecosystems, the framework thus enables efficient service retrieval and the necessary contextual reasoning through dynamic coordination of specialized agents.

6. Conceptual Evaluation

6.1. Evaluation Perspective

HAAF is proposed as a conceptual framework, not as a usable production system. Therefore, the objective of evaluating HAAF is to analyze its architecture and assess its ability to overcome the limitations of current information retrieval and recommendation systems. Since HAAF is a conceptual framework, an evaluation of HAAF would typically involve comparing concepts, analyzing trade-offs among alternative design decisions, and examining possible use cases. Such an evaluation does not typically involve a measurement of performance.

6.2. Comparison Dimensions

The proposed framework is analyzed across several dimensions that are commonly discussed in the retrieval, recommendation, and agentic AI literature:

- Multi-agent coordination
- Cross-service communication
- Dynamic contextual adaptation
- Inventory-aware retrieval
- Session-aware personalization
- Scalability within distributed microservice environments
- Extensibility through standardized service interfaces

These dimensions are used to compare HAAF against representative approaches discussed in the related work section, including traditional retrieval systems, recommendation systems, Retrieval-Augmented Generation architectures, and multi-agent reasoning frameworks.

6.3. Scenario-Based Assessment

This functionality for distributed adaptive information access can be illustrated using several scenarios for e-commerce applications. To provide more in-depth insights into these scenarios for personalized shopping, for inventory-aware search, as well as for very large numbers of users accessing promotional content, the individual functional components of HAAF are described in more detail.

This analysis is used to provide architectural insights and to begin implementing and evaluating HAAF in detail. To this end, thorough benchmarking, large-scale experiments in distributed settings, and quantitative performance analysis will be conducted. Future empirical validation of HAAF can be performed using public e-commerce and retrieval datasets such as Amazon Product Review, RetailRocket, and ESCI. Candidate baselines may

include BM25, dense retrieval, standard RAG, Self-RAG, and non-agentic hybrid retrieval. Evaluation metrics may include Recall@K, MRR, NDCG@K, Precision@K, latency, and throughput. Ablation analysis can evaluate the contribution of DCVI, RRF-based fusion, inventory-aware ranking, and MCP-based context propagation. This validation plan is intended to guide future implementation without presenting unsupported experimental claims.

Table 1. Comparison of HAAF and Representative Approaches for Intelligent Product Discovery

Capability	Traditional Retrieval	RAG	Multi-Agent Systems	HAAF
Semantic Retrieval	Partial	✓	Partial	✓
Multi-Agent Coordination	✗	✗	✓	✓
MCP-Based Communication	✗	✗	✗	✓
Dynamic Context Adaptation	Limited	Partial	Partial	✓
Inventory-Aware Retrieval	✗	✗	Partial	✓
Session-Aware Personalization	Limited	Limited	Partial	✓
Distributed Microservice Integration	Limited	Limited	Partial	✓

Table 1 conceptually compares HAAF with representative retrieval, recommendation, and multi-agent approaches discussed in the related work section. While existing approaches address individual aspects of retrieval, recommendation, or agent coordination, HAAF integrates hierarchical agent orchestration, MCP-based communication, dynamic contextual intelligence, and inventory-aware retrieval within a unified framework for distributed e-commerce environments.

Table 2. Limitations of Representative Approaches and How Haaf Addresses them

Existing Approach	Primary Limitation	HAAF Design Response
Standard RAG	Assumes centralized retrieval over relatively static knowledge sources	Hierarchical agents coordinate retrieval across distributed microservices using MCP-based context propagation.
ReAct	General-purpose reasoning and tool use without domain-specific retrieval orchestration	Specialized retrieval, ranking, behavioral, and inventory agents support structured product discovery workflows.
AutoGen	Multi-agent collaboration, but no architecture for inventory-aware retrieval or product ranking	HAAF introduces domain-specific orchestration integrated with retrieval and operational services.
Traditional Recommendation Systems	Limited support for real-time operational constraints such as inventory, pricing, and fulfillment	DCVI combines behavioral, session, product, inventory, and event signals for adaptive retrieval and ranking.

While Table 1 compares the capabilities of representative approaches, Table II summarizes their principal architectural limitations and highlights how HAAF addresses each limitation through hierarchical agent orchestration, MCP-based coordination, and dynamic contextual retrieval.

7. Case Studies

7.1. Personalized Product Discovery

A user searches for "running shoes" during a period of uneven inventory availability across fulfillment centers. Traditional retrieval systems return results ranked by relevance scores computed at index time, without access to live stock data. HAAF addresses this through the Inventory and Pricing Agent, which queries the inventory microservice in real time and injects E_inventory signals into the DCVI weighting mechanism. Products confirmed as available at the user's regional fulfillment center receive elevated ranking scores, while out-of-stock items are deprioritized subject to configurable policy thresholds. The result is a retrieval output reflecting both semantic relevance and operational reality, reducing add-to-cart failures caused by stale inventory data.

7.2. Inventory-Aware Search

A user searches for "office chair" while browsing from a geolocation served by multiple distribution centers. A typical retrieval model would return a ranked list of products based on standard measures for ranking search results, such as text relevance, historical popularity, and past user

behavior. However, the most relevant products may be out of stock in local stores, or available for delivery that is too slow for the user's needs. In HAAF, during retrieval, the Inventory and Pricing Agent issues queries to the relevant inventory and pricing services, and the resulting information is used by the DCVI layer to weight semantic relevance based on availability for delivery to a user's geolocation and other fulfillment constraints.

Products relevant to a user's search and available for immediate delivery are ranked highest, while out-of-stock products and those with delivery that is too slow are ranked lower and, in the worst case, replaced by similar products that are available. Thus, a retrieval that is aware of inventory can produce much better results for user searches in a distributed e-commerce system.

7.3. Flash Sale and High-Traffic Events

During large-scale promotional events, the quality of returned products, context-sensitive processing, and the performance of the product discovery system are critical. HAAF addresses these challenges through hierarchical agent management, adaptive candidate selection, and distributed microservice management. The tasks are distributed among the different agents to preserve context between services. At the same time, HAAF can scale up quickly in response to changes in usage and peak loads.

8. Discussion

8.1. Scalability

HAAF is designed to scale both agents and microservices. The MCP-based communication within HAAF is decoupled from the actual service deployment. Thus, each service can be scaled individually. The DCVI vector layer can be extended to support sharded deployment across multiple distributed vector stores. The orchestration logic can be extended to support distributed coordination in future releases.

8.2. Operational Considerations

An additional major feature of the proposed framework is the selective deployment of modules to reduce the computational overhead of agent coordination, contextual embedding management, and distributed search. These points shall be elaborated in more detail as part of the planned implementation study to enable a comprehensive analysis of the trade-offs (quality of search results vs. search time, as well as scalability vs. infrastructure costs).

8.3. Failure Modes and Limitations

Several potential failure modes and limitations should be considered for the proposed framework.

(F1) Cold-Start Users: Cold-start users have only a very limited interaction history, which may not be sufficient to generate an accurate behavioral representation of the user. For such users, the framework might have to rely on a combination of session-based information, popularity-based priors, and category-based behavioral information from past interactions within the same category. Such information could be used to build a behavioral representation of the user until enough interaction data is collected to produce a more accurate behavioral representation for that specific user and category in the future.

(F2) Rapid Inventory Churn: This scenario mimics a highly dynamic environment with fast-changing products and services in the retailer's inventory. As the services and components are distributed across multiple sites worldwide, there may be delays in distributing the latest information to all services in the system. Therefore, temporarily unavailable products should be represented in retrieval and ranking results for a short period until the framework has synchronized with the updated information from the corresponding services.

(F3) LLM Non-Determinism: In the context of the Orchestrator Agent, the LLM used for intent classification and task decomposition in a query is non-deterministic, i.e. for same input in different runs of the query (identical inputs, identical queries), different results are obtained (e.g. different task decomposition in the Task Scheduler), which makes results less reproducible and harder to debug in distributed setups.

(F4) Fault Tolerance: In deployment-oriented extensions of HAAF, fault tolerance can be supported through MCP request timeouts, retry policies, circuit breakers, and degraded-mode ranking. When vector retrieval is temporarily unavailable, sparse keyword retrieval can serve as a fallback to preserve service availability. Similarly, delayed inventory updates may be handled using cached availability signals with reduced confidence until synchronization is completed. Trace identifiers propagated through MCP interactions can further improve observability and debugging across distributed agents and microservices.

(F5) Concept Drift and Context Freshness: Product discovery systems operate in environments where user behavior, product catalogs, inventory levels, and seasonal demand continuously evolve. To preserve retrieval quality, future implementations of HAAF may periodically refresh contextual embeddings, incrementally update session representations, monitor retrieval effectiveness, and adapt DCVI weighting over time. These mechanisms can improve contextual freshness while reducing the need for complete re-indexing of vector representations.

Several limitations also remain. First, the proposed framework is conceptual and has not yet been validated through large-scale production deployment or empirical benchmarking. Second, the adaptive weighting strategy for DCVI may require additional refinement to effectively model complex multi-modal behavioral and contextual signals.

Third, adaptive agent coordination strategies may require careful optimization to balance retrieval quality, orchestration complexity, and operational efficiency in large-scale distributed environments.

8.4. Ethical Considerations

Components of the behavioral analysis within the HAAF's ranking model can strengthen existing user biases (e.g., brand preference, price-band anchoring) when historical interaction data exhibit systematic biases. The commercial signal features in the ranking model's retrieval function (e.g., margin scores, promotional incentives) must be governed to avoid degrading user retrieval quality in commercial optimization. As ranked retrieval functions are, the operator has to perform fairness audits regularly to check product visibility equity for vendors of different sizes, as well as across different product categories.

9. Conclusion

This paper introduces HAAF, a Hierarchical Agentic RAG Framework for Intelligent Product Discovery in Distributed E-Commerce Microservices. The framework integrates hierarchical agent orchestration, MCP-based inter-service communication, Dynamic Context-Aware Vector Intelligence (DCVI), hybrid retrieval through Reciprocal Rank Fusion (RRF), and context-aware ranking within a unified architecture for distributed e-commerce product discovery.

Retrieval and ranking within isolated services are insufficient to address the challenges of product discovery in distributed e-commerce systems. Thus, HAAF constitutes a unified framework for Intelligent Product Discovery by combining hierarchical agent orchestration and MCP-based inter-service communication to enable contextual reasoning and adaptive retrieval in distributed microservice environments. By using behavioral signals, session context, product relationships, stock information, and real-time events to retrieve products, HAAF supports Intelligent Product Discovery in e-commerce systems.

Future work will focus on implementing HAAF as a production-ready prototype and evaluating its effectiveness using representative e-commerce retrieval benchmarks. Planned studies include comparisons with state-of-the-art retrieval and recommendation approaches, large-scale deployment experiments in distributed microservice environments, and analysis of trade-offs among retrieval quality, latency, scalability, and orchestration complexity. The DCVI model will also be extended to incorporate multimodal contextual representations, including visual product features and additional behavioral signals, enabling richer context modeling for intelligent product discovery.

References

[1] Patrick Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459-9474, 2020. [Google Scholar] [Publisher Link]

[2] Vladimir Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, EMNLP, pp. 6769-6781, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Akari Asai et al., "Self-RAG: Learning to Retrieve, Generate, and Critique Through Self-Reflection," *International Conference on Learning Representations Proceedings*, vol. 2024, pp. 1-30, 2024. [Google Scholar] [Publisher Link]

[4] Shunyu Yao et al., "React: Synergizing Reasoning and Acting in Language Models," *arXiv Preprint*, pp. 1-33, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Qingyun Wu et al., "Autogen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," *arXiv Preprint*, pp. 1-43, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Timo Schick et al., "Toolformer: Language Models can Teach Themselves to use Tools," *Advances in Neural Information Processing Systems*, vol. 36, pp. 1-13, 2023. [Google Scholar] [Publisher Link]

[7] Xiangnan He et al., "Neural Collaborative Filtering," *arXiv Preprint*, pp. 1-10, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Paul Covington, Jay Adams, and Emre Sargin, "Deep Neural Networks for Youtube Recommendations," *Proceedings of the 10th ACM Conference on Recommender Systems*, Association for Computing Machinery, New York, NY, United States, pp. 191-198, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Kelvin Guu et al., "REALM: Retrieval-Augmented Language Model Pre-Training," *Proceedings of the 37th International Conference on Machine Learning, PMLR*, vol. 119, pp. 3929-3938, 2020. [Google Scholar] [Publisher Link]

[10] Jacob Devlin et al., "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, NAACL, vol. 1, pp. 4171-4186, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jeff Johnson, Matthijs Douze, and Hervé Jégou, "Billion-Scale Similarity Search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535-547, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Yu A. Malkov, and D.A. Yashunin, "Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824-836, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive Representation Learning on Large Graphs," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1-11, 2017. [Google Scholar] [Publisher Link]

[14] Petar Veličković et al., "Graph Attention Networks," *arXiv Preprint*, pp. 1-12, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Model Context Protocol, What is the Model Context Protocol (MCP)?, Model Context Protocol, 2024. [Online]. Available: https://modelcontextprotocol.io/docs/getting-started/intro

[16] Jianmo Ni, Jiacheng Li, and Julian McAuley, "Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, pp. 188-197, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Stephen Robertson, and Hugo Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 4, no. 1-2, pp. 1-59, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[18] Weijia Shi et al., "REPLUG: Retrieval-Augmented Language Model Pre-Training," *arXiv Preprint*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Joon Sung Park et al., "Generative Agents: Interactive Simulacra of Human Behavior," *arXiv Preprint*, pp. 1-22, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Balázs Hidasi et al., "Session-based Recommendations with Recurrent Neural Networks," *arXiv Preprint*, pp. 1-20, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[21] Junling Liu et al., "Is ChatGPT a Good Recommender? A Preliminary Study," *arXiv Preprint*, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Fei Sun et al., "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer," *arXiv Preprint*, pp. 1-10, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[23] Gordon V. Cormack, Charles L.A. Clarke, and Stefan Buettcher, "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods," *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, New York, NY, United States, pp. 758-759, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[24] Christopher J. Burges, Robert Ragno, and Quoc V. Le "Learning to Rank with Nonsmooth Cost Functions," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1-8, 2006. [Google Scholar] [Publisher Link]

[25] Eugene Ie et al., "RecSim: A Configurable Simulation Platform for Recommender Systems," *arXiv Preprint*, pp. 1-23, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Victor Sanh et al., "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter," *arXiv Preprint*, pp. 1-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[27] Jianguo Wang et al., "Milvus: A Purpose-Built Vector Data Management System," *Proceedings of the 2021 International Conference on Management of Data*, Association for Computing Machinery, New York, NY, United States, pp. 2614-2627, 2021. [CrossRef] [Google Scholar] [Publisher Link]