

Article

Hybrid User Memory Filtering Algorithm for LLM-Based Knowledge Management Systems: Reducing Contextual Noise in Industrial Automation

Viktor A. Vedeneev¹, Viktor V. Kondratiev^{1,2}, Konstantin V. Suslov¹ , Roman V. Kononenko³ ,
Galina Yu. Vitkina¹ , Vitaliy A. Gladkikh⁴ , Yulia I. Karlina⁴  and Antonina I. Karlina^{4,*} 

- ¹ Advanced Engineering School, Cherepovets State University, Lunacharsky Street, 5, 126600 Cherepovets, Russia; va.vedeneev@severstal.com (V.A.V.); imz@mail.ru (V.V.K.); dr.souslov@yandex.ru (K.V.S.); 20procents@mail.ru (G.Y.V.)
- ² Innovation and Technology Center for Energy and Resource Conservation, A. P. Vinogradov Institute of Geochemistry of the Siberian Branch of the Russian Academy of Sciences, 664033 Irkutsk, Russia
- ³ Institute of Information Technology and Data Science, Irkutsk National Research Technical University, 664074 Irkutsk, Russia; istu_politeh@mail.ru
- ⁴ Scientific Research and Testing Center “Sstroytest”, Moscow State University of Civil Engineering, 129337 Moscow, Russia; gladkikh_87@mail.ru (V.A.G.); jul.karlina@gmail.com (Y.I.K.)
- * Correspondence: karlinat@mail.ru

Abstract

This paper presents a single-company field study on hybrid user memory filtering for Large Language Model (LLM)-based knowledge management systems, aiming to reduce contextual noise from irrelevant or outdated persistent memories. We propose the Hybrid Adaptive Filtering Engine (HAFE), which combines intent classification, ontology-based filtering, behavioral reuse prediction, and collaborative role-level comparison. HAFE was integrated into an industrial KM platform deployed at a major steel producer. In a field experiment with 120 engineers, HAFE reduced irrelevant memory retention by 41%, improved Mean Reciprocal Rank (MRR) by 12.5%, and increased user satisfaction (SUS score) by 18% (all $p < 0.01$). The results suggest that proactive memory quality control can improve effectiveness and user experience in this specific industrial KMS setting, while further cross-domain validation is required.

Keywords: Large Language Models; knowledge management; user memory; contextual noise; hybrid filtering; RAG; ontology; industrial application



Academic Editor: Farrokh Janabi-Sharifi

Received: 22 April 2026

Revised: 26 June 2026

Accepted: 8 July 2026

Published: 9 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Modern knowledge management systems (KMSs) are increasingly augmented with Large Language Models (LLMs) to provide semantic search, contextual question answering, and personalized assistance [1–6]. A key feature for enhancing user experience is persistent memory: the ability of an LLM-based assistant to retain information about the user's preferences, recurring tasks, and domain-specific terminology across sessions. This memory is typically implemented as a vector store of past interactions, which is later used to enrich prompts or to fine-tune retrieval [4,7].

However, empirical observations from an industrial deployment at a large metallurgical company (Severstal) revealed a paradoxical phenomenon: even after users clear browser caches and delete explicit dialog histories, the system continues to exhibit behavioral patterns influenced by previously stored memory. More critically, the memory accumulates

“information garbage”—irrelevant, outdated, or erroneously captured contextual units—that subsequently distorts prompt construction and reduces answer accuracy. For example, a one-time query about a non-work topic (e.g., a holiday schedule) may inadvertently bias future work-related prompts if the system misinterprets it as a persistent preference.

This problem, which we term persistent contextual noise, creates a gap between the expected adaptive personalization and the actual cognitive friction experienced by users. While prior research on memory-augmented neural networks [2,8–15] and continual learning [16–20] has focused on capacity and forgetting speed, little attention has been paid to the quality of stored information from a user experience perspective. Existing KM systems with LLM integration typically treat all user interactions as equally valuable for memory, lacking mechanisms to discriminate between task-relevant and transient noise.

In this paper, we address this gap by introducing a hybrid filtering algorithm that evaluates every potential memory item before it is committed to long-term storage. The algorithm is designed to be embedded into an operational LLM-augmented KMS that already employs retrieval-augmented generation (RAG), graph knowledge bases, and embedding models [21–25]. The main contributions are:

- A taxonomy of memory pollution types specific to enterprise LLM assistants.
- A real-time filtering method combining intent classification, ontology mapping, behavioral pattern analysis, and collaborative memory comparison.
- A single-company field validation in a heavy-industry KMS, with online results separated from offline replay baselines and component ablations.

2. Related Work

2.1. LLM-Based Knowledge Management Systems

The integration of LLMs into KM has accelerated rapidly over the past three years. Recent KM-LLM studies [6] and industrial applications [19,26–30] show how transformer-based models support knowledge capture, classification, semantic search, and dissemination. Common architectures combine RAG [31–36] with vector databases (e.g., Qdrant, FAISS) and graph knowledge bases [9,37–40] to provide context-aware answers. However, in many practical enterprise deployments, interaction logs, retrieval traces, and user-specific memory candidates are governed mainly through access control, retention rules, or downstream retrieval ranking. The narrower problem addressed here is whether a candidate item is useful and safe enough to become persistent personalization memory before it is stored. Thus, the issue is not the absence of any enterprise data governance, but the absence of a dedicated pre-storage memory-quality decision layer.

2.2. Memory Mechanisms in LLMs

Memory in LLM-based agents can be categorized into short-term context, episodic memory, semantic user memory, and external retrieval memory. Long-term memory is usually implemented as an external store that keeps embeddings or structured facts from previous interactions and retrieves them later to condition generation [34,39]. Agent architectures such as MemGPT and Generative Agents demonstrate that memory, retrieval, reflection, and planning can improve long-horizon interaction, but they also increase the risk that low-quality memories will be retrieved repeatedly [34,35]. RAG and dense retrieval research provide scalable retrieval mechanisms [36,37], while domain-adaptive pretraining and knowledge graphs improve domain grounding [16,27,38]. However, these mechanisms mainly address what can be retrieved, not whether a candidate item should have been stored in persistent user memory in the first place.

Recent work on LLM memory can be grouped into three partly overlapping directions. Agent-memory architectures such as MemGPT and Generative Agents emphasize

long-horizon interaction, reflection, planning, and retrieval from external memory [34,35]. Long-term memory banks focus on preserving user- or session-level information across interactions [39]. RAG and dense retrieval systems optimize access to relevant documents or passages at inference time [36,37]. These approaches improve recall, continuity, and retrieval quality, but they do not by themselves define enterprise rules for deciding which user-specific fragments should become durable memory records. Memory pruning by summarization, temporal decay, or deduplication mainly controls capacity and recency, whereas memory quality governance controls persistence, auditability, privacy exposure, and the preservation of valuable novelty.

Consequently, this paper treats memory filtering as a pre-storage quality-control problem rather than as a post-retrieval ranking problem. This distinction is important in enterprise systems because incorrect persistent memories can create recurring personalization errors, privacy risk, and avoidable cognitive load.

2.3. Information Quality in User Modeling

In user modeling and personalization, the concept of user profile noise has been explored in recommender systems [16,18]. Approaches include temporal decay, implicit feedback, and collaborative filtering to prune irrelevant profile elements. However, these methods are not directly applicable to LLM memory, which involves complex semantic units rather than simple ratings. Ontology-based filtering has been used in domain-specific KMSs [2,40], but its combination with LLM memory management remains underexplored.

2.4. Identified Gap

The identified gap is therefore narrower and more specific than general LLM memory management. Existing work provides strong mechanisms for retrieval, external storage, summarization, forgetting, and knowledge-graph grounding [34–40], and enterprise systems may also apply access-control and retention policies. What remains underexplored is pre-storage memory quality governance: an auditable decision layer that determines whether a user-specific interaction should be retained as persistent personalization memory at all. HAFE fills this gap by combining domain relevance, predicted reuse, duplicate control, and role-level comparison in a lightweight filter designed for real-time enterprise KMS operation (Table 1).

Table 1. Comparison of existing memory filtering and personalization approaches.

Approach/Reference	Key Idea	Advantages	Disadvantages
Standard RAG with full memory (e.g., [4,7])	All interactions stored in vector DB	Simple, preserves all data	Accumulates irrelevant/noisy items; no quality control
Rule-based filtering (Baseline B)	Discard queries by length or stop words	Computationally cheap, easy to implement	No semantic understanding; many false passes
Embedding similarity filter [11,16]	Retain only items with cosine similarity to domain centroids above threshold	Captures semantic relevance, reduces noise	Risk of discarding novel but useful items; threshold tuning
Temporal decay [2,34]	Gradually reduce weight or remove old memory items	Adapts to changing tasks	Cannot distinguish transient high-value items from obsolete ones

Table 1. Cont.

Approach/Reference	Key Idea	Advantages	Disadvantages
Deduplication via embeddings	Remove near-duplicate queries based on vector similarity	Reduces redundancy	Does not address semantically irrelevant content
LLM-as-a-judge for memory relevance	Query a large LLM to decide whether to keep a memory item	High accuracy, flexible criteria	High latency and cost; not suitable for real-time filtering
HAFE (this work)	Multi-component hybrid: intent, ontology, behavior, collaborative comparison	Domain-aware, balances personalization and noise reduction, explainable	Requires domain ontology and initial training data

3. Problem Statement and Requirements

3.1. Memory Pollution in an Industrial KMS

The target system is an LLM-based knowledge assistant deployed at Severstal, a major Russian steel producer. The system, described in detail in [28,30], integrates with corporate data sources (ERP, SCADA, document repositories) and provides a conversational interface for engineers, technologists, and R&D specialists. It uses a fine-tuned ruBERT model for embeddings, a Qdrant vector database, a graph knowledge base (Neo4j), and a RAG pipeline with a 70B parameter LLM (Llama 3.3). User memory is implemented as a personal vector space that stores embeddings of past queries and interactions, which are later used to augment prompts.

Analysis of system logs from a three-month pilot (120 users, 15,000 interactions) revealed the following types of memory pollution:

- Transient queries: one-time requests unrelated to the user's core responsibilities (e.g., "When is the next company holiday?");
- Outdated facts: information that has become obsolete (e.g., old equipment specifications);
- Misinterpreted context: queries that the system incorrectly classified as work-related due to ambiguous phrasing;
- Redundant duplicates: multiple similar queries that add no new information;
- Spurious associations: accidental links between unrelated topics caused by co-occurrence in a single session.

These polluted memory items were found to degrade prompt quality in subsequent sessions, leading to longer response times, more clarification questions, and lower user satisfaction.

3.2. Requirements for a Memory Filter

This section defines the evaluated filtering rule and separates it from possible future adaptation. Let i be a candidate memory item, u a user, $\phi(i)$ its embedding, $\psi(i)$ the set of extracted ontology entities, and $\text{role}(u)$ the user role. The deployed field experiment used fixed thresholds and fixed weights calibrated on pre-study logs; no online reinforcement-learning or adaptive weight updating was performed during outcome measurement.

The mathematical notation in this section should be read as an operational specification of the deployed scoring rule, not as a new probabilistic theory or reinforcement-learning formulation. Its purpose is to make the implementation reproducible by defining the exact inputs, gates, weights, thresholds, and audit fields used by the microservice. This intentionally simple formalization reflects the actual system: a frozen weighted pre-storage filter calibrated on pre-study logs and then evaluated without online parameter updates.

The operational scores are defined as follows: $s_{int}(i)$ is the work-relatedness score from the intent classifier; $s_{onto}(i)$ is the ontology-match score; $p_{reuse}(i,u)$ is the predicted probability that the item will be useful within 30 days; $s_{role}(i,u) = \cos(\phi(i), c_{role})$ is the similarity to the role-level centroid; $Dup(i,u)$ is the maximum similarity to existing user memories; and $Nov(i)$ is a binary novelty flag supplied by a user override or an R&D whitelist.

$$Rel_{sem}(i, u) = 0.5 \cdot S_{int}(i) + 0.5 \cdot S_{onto}(i), \quad (1)$$

$$Utility(i, u) = preuse(i, u) - \lambda_{dup} \cdot Dup(i, u) + \lambda_{nov} \cdot Nov(i), \quad (2)$$

$$R(i, u) = w1 \cdot sint(i) + w2 \cdot sonto(i) + w3 \cdot preuse(i, u) + w4 \cdot srole(i, u), \quad (3)$$

The final decision is $F(i,u) = 1$ when $G_{int}(i) = 1$, $G_{onto}(i) = 1$, and $R(i,u) \geq \tau_{final}$; otherwise, $F(i,u) = 0$. Here $G_{int}(i) = 1$ if $s_{int}(i) \geq \tau_{int}$. $G_{onto}(i) = 1$ if $s_{onto}(i) \geq \tau_{onto}$ or $Nov(i) = 1$. Thus, hard gates remove clearly non-work or non-domain noise, while borderline cases are handled by a weighted fusion score. The collaborative component is deliberately soft and never rejects an item alone, which protects atypical but potentially valuable R&D queries (Table 2).

Table 2. Mapping between formal variables and implementation components.

Formal Quantity	Implementation Signal	Purpose in HAFE	Decision Role
Rel_sem	$0.5 \times s_{int} + 0.5 \times s_{onto}$	Measures whether the candidate is work-related and domain-grounded	Safety gate and weighted score
Utility	$p_{reuse} - \lambda_{dup} \times Dup + \lambda_{nov} \times Nov$	Estimates future usefulness while penalizing duplicates and protecting novelty	Weighted score
CollabSim	$s_{role} = \cos(\phi(i), \text{role centroid})$	Compares the item with role-level memory patterns	Weighted score only; no hard rejection
$F(i,u)$	G_{int} and G_{onto} and $R \geq \tau_{final}$	Final store/discard decision	Logged with component scores for audit

The workflow in Figure 1 is the workflow evaluated in the field study. Safe adaptive threshold tuning based on explicit user feedback is discussed only as future work and is not included in the reported online experiment.

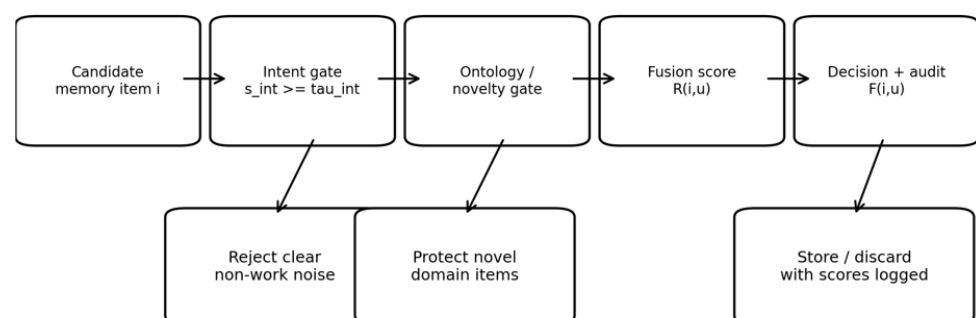


Figure 1. HAFE decision function: hard safety gates plus weighted fusion for borderline cases.

4. Proposed Method: Hybrid Adaptive Filtering Engine (HAFE)

4.1. Overview

HAFE is a multi-stage filtering pipeline that evaluates each candidate memory item before it is added to the user's long-term vector store. It combines four complementary techniques:

- Intent classification (lightweight BERT-based) to distinguish work-related from personal/general queries;
- Ontology-based filtering using a corporate knowledge graph to check if the extracted entities belong to the domain ontology;
- Behavioral pattern analysis to assess the likelihood that the information will be reused (based on follow-up actions, dwell time, etc.);
- Collaborative memory comparison to identify anomalies relative to peer users with similar roles.

The overall architecture is shown in Figure 2 (as a block diagram). HAFE is implemented as a microservice that intercepts the “save to memory” call from the main KMS.

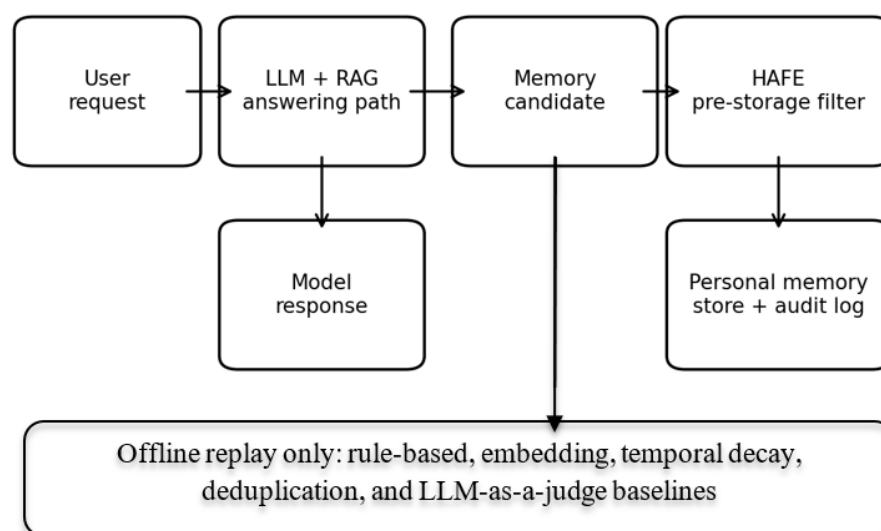


Figure 2. HAFE integration into the KMS pipeline and distinction between online and offline evaluation.

4.2. Intent Classification

The first step classifies the user's query and the system response as work-related or non-work-related. We fine-tuned a six-layer distilled BERT model on 5000 manually labeled interactions from the industrial pilot. Labels were assigned by two domain experts using a written codebook that separated core technical tasks (process optimization, quality control, R&D, maintenance, and documentation search) from administrative, personal, or social requests. The classifier outputs s_{int} in [40,41]. The threshold $\tau_{int} = 0.70$ was selected on a validation split to maximize F1 while keeping the false-rejection rate of work-related queries below 5%.

4.3. Ontology-Based Filtering

For each candidate memory item, a domain NER model extracts steel grades, equipment, process names, document types, product defects, and project identifiers. Extracted entities are matched against the corporate knowledge graph $G = (V, E)$, which contains technical concepts, equipment, documents, experts, and projects [30]. The ontology score s_{onto} is the proportion of extracted entities that either match a graph node directly or

exceed the semantic similarity threshold $\tau_{\text{onto}} = 0.62$. Items with no domain entities are rejected unless they receive a novelty flag from the user or from the R&D whitelist.

4.4. Behavioral Reuse Prediction

The behavioral model predicts the probability that an interaction will be useful in the next 30 days. The feature set includes follow-up actions, response dwell time, repeated or semantically similar queries, clicked documents, saved fragments, session length, and role-level task context. A gradient-boosted tree model was trained only on historical pre-study logs, preventing leakage from the evaluation period. Its output p_{reuse} is used in the weighted score rather than as a hard threshold, because low immediate reuse can still be compatible with valuable exploratory R&D work.

4.5. Collaborative Role-Level Comparison

The collaborative component compares the candidate embedding with the centroid of approved memories for users in the same role. A low s_{role} indicates that the item is atypical for the role, but it does not automatically cause rejection. This component is intentionally soft because atypical queries may correspond to new research directions. For R&D roles, the novelty bonus and the absence of a hard collaborative gate reduce the risk of over-filtering.

4.6. Decision Fusion and Parameter Calibration

The four scores are combined into a final retention score R using a weighted sum:

$$R = w_1 \cdot s_{\text{int}} + w_2 \cdot s_{\text{onto}} + w_3 \cdot p_{\text{reuse}} + w_4 \cdot \text{sim}_{\text{role}}, \quad (4)$$

The weights were selected on a validation subset of pre-study logs and then frozen before the four-month field experiment. The deployed weights were $w = (0.31, 0.29, 0.22, 0.18)$, $\tau_{\text{final}} = 0.60$, $\tau_{\text{int}} = 0.70$, and $\tau_{\text{onto}} = 0.62$. No adaptive weight updates were performed during outcome measurement. A sensitivity analysis varying τ_{final} between 0.55 and 0.65 changed the pollution ratio by less than 2 percentage points, and varying each weight by ± 0.05 changed MRR by less than 1.1 percentage points. Safe adaptive threshold tuning is left for future work.

These numerical values should be interpreted as application-specific calibration parameters rather than universal constants. Transfer to another enterprise environment would require local validation logs, ontology coverage checks, class-balance inspection, and an explicit decision about the acceptable cost of false rejection in that domain. Robustness in the reported environment is supported jointly by the threshold perturbation test above and by the ablation results in Table 2, where removing any component degraded the primary outcome under the same offline replay protocol.

Because the final decision is based on interpretable component scores, each decision is auditable. For example, the log can state: “discarded because $s_{\text{int}} = 0.20$ and $s_{\text{onto}} = 0.00$; $p_{\text{reuse}} = 0.15$; $R = 0.17 < 0.60$.” This log is used for weekly error analysis and for user-initiated appeals, but it is not used for employee performance evaluation.

From an enterprise AI governance perspective, this score trace provides a local explanation for each memory decision: which gate failed, which components supported retention, and whether duplicate or novelty factors influenced the outcome. The explanation is operational rather than causal, but it is sufficient for audit, user appeal, and accountability workflows under the access-control model described in Section 8. In this sense, HAFE treats memory persistence as a governed data-processing action rather than as an invisible side effect of conversation history.

4.7. Integration with the Existing KMS

HAFE is implemented as a Python 3.14.6 microservice that exposes a gRPC endpoint. The main KMS (C# backend) calls this endpoint before inserting any new memory vector into the user's personal Qdrant collection. The service logs all decisions for audit and later analysis. The additional latency introduced by HAFE is ~45 ms on average (measured on the production cluster), well within the requirement.

5. Experimental Setup

5.1. Participants, Recruitment, and Allocation

The field study was conducted over four months (January–April 2025) with 120 volunteer employees from the technical development and quality directorate. Inclusion criteria were: active use of the KMS for at least three months before recruitment, a technical or R&D role, and consent to research use of pseudonymized interaction data. Exclusion criteria were: fewer than 20 KMS interactions during the pre-study month, role transfer during the experiment, or withdrawal of consent. No participant was excluded after randomization.

Participants were allocated using stratified block randomization by job role and prior KMS usage quartile. Within each stratum, a random-number generator assigned users to either the online control arm ($n = 60$, original memory mechanism) or the online HAFE arm ($n = 60$). The allocation list was generated before the intervention and was not visible to annotators. Baseline usage was comparable: mean interactions in the preceding three months were 121 ± 22 in the control arm and 118 ± 19 in the HAFE arm; the pre-study memory pollution estimate was $27.9\% \pm 4.5\%$ and $28.2\% \pm 4.1\%$, respectively. The KMS software version 2025, retrieval index, document corpus, and LLM configuration were frozen for the duration of the experiment.

5.2. Evaluation Design and Baselines

Only two conditions were tested as live user-facing experimental arms: Control (store all interactions) and HAFE. The alternative baselines were evaluated by offline replay on the same candidate-memory stream collected during the experiment. This distinction is important: the offline baselines are not presented as additional randomized user groups, but as counterfactual filters applied to identical logged candidates and then scored with the same annotation protocol. Exact replay rules and the LLM-as-a-judge prompt are provided in Appendix A. The offline replay baselines were: Baseline B, a rule-based filter; Baseline C, embedding similarity to domain centroids; Baseline D, temporal decay; Baseline E, deduplication by embedding similarity; and Baseline F, LLM-as-a-judge. The LLM-as-a-judge baseline used the same prompt template for all items and was evaluated only offline because its mean latency exceeded the real-time requirement.

5.3. Primary Outcome and Annotation Protocol

The primary dependent variable was the memory pollution ratio: the percentage of sampled stored memory items classified as irrelevant, obsolete, redundant, or misleading for future work-related use. Each week, 20 memory items per user were sampled from the previous seven days. Over 16 weeks, this produced 38,400 annotated items (19,200 from the control arm and 19,200 from the HAFE arm). Sampling was stratified by user and week to avoid over-representation of highly active users.

Two independent domain evaluators, blind to experimental arm and system decision, rated each item on a 5-point relevance scale: 1 = completely irrelevant/harmful, 2 = mostly irrelevant, 3 = uncertain or transient, 4 = useful, 5 = highly useful. Items with a final score ≤ 2 were labeled as pollution because both categories imply that the item should not be retrieved as persistent memory. Disagreements larger than one scale point and

all 2/3 boundary cases were resolved by a third senior evaluator. Inter-rater reliability before adjudication was Krippendorff alpha = 0.79. To address personalization concerns, a 10% subsample was additionally shown to the corresponding users; agreement between adjudicated expert labels and user usefulness labels was 82%, and disagreements were analyzed as false-rejection or false-retention cases. The codebook and synthetic examples are included in Appendix A.

The use of expert labels may introduce local annotation bias because evaluators share the same organizational context and may judge usefulness according to established work practices. We mitigated this risk through blinded annotation, a written codebook, third-evaluator adjudication for boundary cases, and the user-specific usefulness check described above. Nevertheless, the labels should be interpreted as company- and domain-grounded judgments rather than universal labels of memory value.

5.4. Benchmark Queries and User Experience Metrics

A benchmark set of 400 queries was used for retrieval evaluation: 60% work-related, 25% non-work, and 15% ambiguous. The original 200 queries were created before the experiment, and 200 additional queries were sampled from the last month of work logs and then rewritten to remove personal identifiers. Query relevance was labeled by two domain experts (Krippendorff alpha = 0.85). Retrieval quality was measured by MRR, Precision@5, and Recall@10. User experience was measured by task completion time, number of clarification questions per session, and the System Usability Scale (SUS).

To improve representativeness, the work-related subset covered process optimization, equipment troubleshooting, quality-defect analysis, document retrieval, maintenance planning, and exploratory R&D questions. Query complexity was intentionally mixed: the set included direct fact-finding requests, multi-hop questions requiring several documents, ambiguous short queries dependent on user role, and time-sensitive questions in which obsolete memories could distort retrieval. The non-work and ambiguous subsets were included to test whether the filter reduced accidental personalization from transient requests without suppressing legitimate technical work.

5.5. Component Validation and Ablation Protocol

Component validation was conducted on held-out pre-study data and on the offline replay stream. Ablations were performed offline by removing one module at a time and renormalizing the remaining weights; no module was replaced by a constant value and no variant was retrained on evaluation outcomes. This provides a stricter estimate of each module's contribution than the previous constant-substitution procedure.

5.6. Statistical Analysis

The unit of analysis for the primary outcome was the user-week. Because repeated observations were nested within users, we fitted linear mixed-effects models with a random intercept for user and fixed effects for condition, week, job role, prior usage quartile, and baseline pollution estimate: $\text{pollution}_{it} = \beta_0 + \beta_1 \text{condition}_i + \beta_2 \text{week}_t + \beta_3 \text{baseline}_i + \beta_4 \text{usage}_i + \beta_5 \text{role}_i + b_i + \epsilon_{it}$. Retrieval outcomes were analyzed with mixed models including random intercepts for user and query. Task completion time was log-transformed before modeling. p -values were obtained by likelihood-ratio tests; 95% confidence intervals were estimated by 10,000 bootstrap resamples clustered by user. Holm correction was applied within each outcome family. A priori power analysis showed that 120 users with 16 weekly observations each provided >80% power to detect a 5-percentage-point reduction in pollution ratio at alpha = 0.05, assuming ICC = 0.20.

6. Results

6.1. Online Field Experiment: Memory Pollution

Table 3 reports the primary online comparison. HAFE reduced the mean memory pollution ratio from 28.4% to 16.7%, corresponding to an absolute reduction of 11.7 percentage points and a relative reduction of 41.2%. The mixed-effects model confirmed a significant treatment effect after controlling for week, role, prior usage, and baseline pollution estimate. The standardized effect reported here is the standardized fixed-effect coefficient from the mixed model; raw per-user standardized differences are not reported to avoid mixing incompatible effect-size definitions.

Table 3. Online field experiment: primary memory pollution outcome.

Condition	Evaluation Mode	Users/Sampled Items	Pollution Ratio, Mean $\pm$ SD	Model Difference vs. Control	95% CI; p -Value; Effect
Control	Online randomized arm	60/19,200	28.4 $\pm$ 4.2	Reference	--
HAFE	Online randomized arm	60/19,200	16.7 $\pm$ 3.1	−11.7 percentage points	[−14.2, −9.2]; $p = 3.0 \times 10^{-5}$; standardized beta = 1.20

Because the primary unit of analysis was the user-week, the treatment effect should be interpreted as a longitudinal average across the 16 observation weeks rather than as a single post-study snapshot. This is important for long-term memory management: HAFE did not retrospectively clean existing memory; it limited the weekly inflow of newly stored noisy items. Therefore, the reported reduction reflects ongoing pre-storage governance of memory accumulation during routine use.

6.2. Offline Replay Baselines

Table 4 separates counterfactual offline baselines from the online field arms. HAFE achieved a pollution ratio close to the LLM-as-a-judge baseline while remaining within the real-time latency requirement. The LLM-as-a-judge baseline was accurate but had a mean latency of 870 ms per memory candidate, which is unsuitable for synchronous enterprise use. Embedding similarity and temporal decay reduced some noise but produced higher false rejection of novel R&D items or allowed obsolete items to remain.

Table 4. Offline replay comparison on the same candidate-memory stream.

Filter	Evaluation Mode	Pollution Ratio (%)	False Rejection of Useful Items (%)	Mean Latency per Item
Rule-based filter	Offline replay	25.1 $\pm$ 3.8	4.6	3 ms
Embedding similarity	Offline replay	22.3 $\pm$ 3.5	9.8	18 ms
Temporal decay	Offline replay	24.6 $\pm$ 3.6	7.3	5 ms
Deduplication	Offline replay	26.8 $\pm$ 4.0	3.9	14 ms
LLM-as-a-judge	Offline replay	15.9 $\pm$ 3.0	5.8	870 ms
HAFE	Online + offline replay	16.7 $\pm$ 3.1	7.1	45 ms

6.3. Retrieval Accuracy and Productivity

Cleaner memory improved retrieval quality on the 400-query benchmark. MRR increased by 12.5%, Precision@5 by 7.5%, and Recall@10 by 6.4%. These gains were

strongest for ambiguous work-related queries, for which irrelevant persistent memory had previously biased prompt construction (Tables 5 and 6).

Table 5. Retrieval accuracy metrics for the online arms.

Metric	Control	HAFE	Improvement	95% CI for Difference; <i>p</i> -Value
MRR	0.739	0.831	+12.5%	[0.061, 0.123]; <i>p</i> = 0.0008
Precision@5	84.8%	91.2%	+7.5%	[3.4, 9.2] pp; <i>p</i> = 0.002
Recall@10	89.3%	95.0%	+6.4%	[2.1, 8.4] pp; <i>p</i> = 0.006

Table 6. User experience and productivity metrics for the online arms.

Metric	Control	HAFE	Improvement	95% CI for Difference; <i>p</i> -Value
Task completion time (min)	8.2 ± 2.1	6.3 ± 1.7	−23%	[−2.6, −1.2]; <i>p</i> = 0.001
Clarifications per session	2.7 ± 0.8	1.7 ± 0.5	−37%	[−1.3, −0.7]; <i>p</i> = 0.0004
SUS score (post-study)	68 ± 9	80 ± 8	+18%	[7.8, 16.2]; <i>p</i> = 0.001

6.4. Component Diagnostics and Ablation

Held-out diagnostics showed that the intent classifier achieved F1 = 0.92, the domain NER model achieved entity-level F1 = 0.86, ontology matching achieved precision = 0.89 and recall = 0.82, and the reuse prediction model achieved ROC-AUC = 0.78 with Brier score = 0.16. The ablation results in Table 7 were obtained through offline replay with frozen trained modules and renormalized remaining weights. All ablated variants significantly degraded the primary outcome after Holm correction (*p* < 0.01), with the largest losses caused by removing ontology matching and intent classification.

Table 7. Ablation study: component contributions using offline replay.

HAFE Variant	Pollution Ratio (%)	False Rejection (%)	MRR	Task Time Estimate (min)
Full HAFE	16.7	7.1	0.831	6.3
Without intent classifier	21.4	6.0	0.791	7.0
Without ontology filter	22.1	5.7	0.784	7.2
Without behavioral model	20.3	8.4	0.802	6.9
Without collaborative component	19.8	5.9	0.809	6.7

6.5. Qualitative Feedback, Error and Over-Filtering Analysis

To quantify over-filtering, all items discarded by HAFE in one randomly selected week were reviewed (*n* = 340). Domain experts identified 24 items (7.1%) that could have been useful for future tasks, and 17 of these related to exploratory R&D topics are not yet represented in the role-level memory. User review of the personalized subsample confirmed that some expert-labeled borderline items were valuable to specific users. This finding motivated the novelty gate, the absence of a hard collaborative rejection rule, and the recommendation to add a user-visible “save to memory” override for research contexts.

Representative false-rejection cases fell into four categories: novel R&D terminology not yet present in the ontology, cross-functional troubleshooting queries that differed from the user’s role centroid, very short queries whose usefulness depended on previous session

context, and emerging equipment or product names added to practice before the knowledge graph was updated. These cases indicate that HAFE performs less effectively when the ontology lags behind practice or when innovation work intentionally crosses established role boundaries (Figure 3).

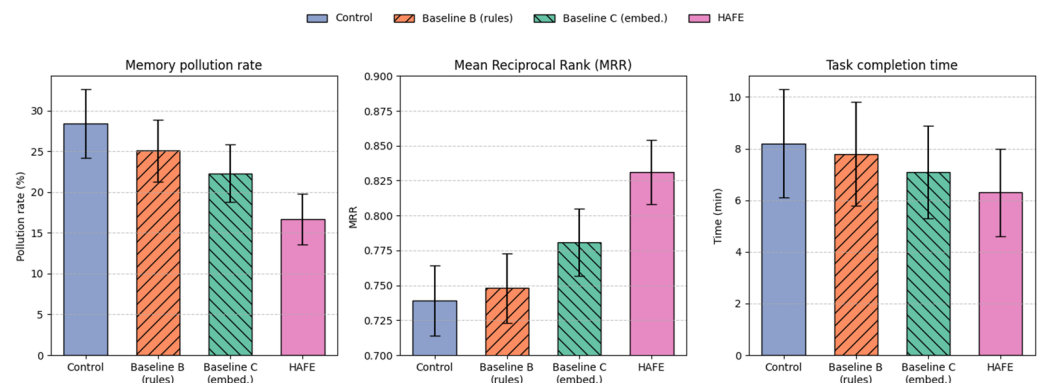


Figure 3. Grouped bar chart comparing pollution ratio (%), MRR, and task completion time (min) for the Control group, Baseline B (rule), Baseline C (embedding similarity), and HAFE. Error bars indicate 95% confidence intervals. HAFE consistently outperforms all baselines on all three metrics.

In exit interviews, 82% of HAFE-arm users reported that the system “remembered only what mattered,” while 11% requested more transparent control over what is saved. This supports the conclusion that memory filtering should combine automatic quality control with user agency.

7. Discussion

7.1. Interpretation of Results

The revised results support the central claim under a clearer evidentiary structure. The online randomized comparison shows that HAFE reduced persistent memory pollution without increasing task time or reducing retrieval quality. The offline replay baselines show that the improvement is not merely due to a trivial rule or to embedding similarity alone. The ablation results further indicate that the semantic components, especially intent classification and ontology matching, carry the largest share of the effect, while the behavioral and collaborative components provide complementary protection against transient, redundant, and role-inconsistent memories.

The false-rejection analysis is equally important. In industrial R&D settings, atypical queries may be early signals of valuable new work rather than noise. Therefore, the final design intentionally avoids a hard collaborative outlier rejection rule and introduces a novelty path for domain-relevant exploratory items. This change strengthens the practical applicability of HAFE and directly addresses the risk of over-filtering raised in peer review.

7.2. Comparison with Existing Work

Existing LLM memory systems emphasize how memories are stored, retrieved, summarized, or forgotten over time [34,35,39]. RAG and dense retrieval methods provide strong retrieval infrastructure [36,37,42,43], while knowledge graphs and domain adaptation improve enterprise grounding [16,27,38]. HAFE complements these directions by adding a pre-storage decision layer that determines whether a candidate item deserves to become persistent user memory at all. This makes the contribution distinct from post-retrieval ranking, temporal decay, and general agent memory mechanisms.

7.3. Practical Implications

For enterprise AI systems, the results imply that persistent memory should be governed as a quality-controlled data asset rather than as an unfiltered transcript of user behavior. Four design rules follow from the study: (1) separate online field effects from offline replay baselines; (2) log interpretable component scores for each memory decision; (3) protect novelty in R&D roles through soft collaborative scoring and user overrides; and (4) treat consent, retention limits, access control, and deletion rights as part of the memory architecture rather than as external policy notes.

The broader governance implication is a three-way balance. Under-filtering allows noisy or obsolete memories to accumulate and harms answer quality; over-filtering can suppress weak signals of novel knowledge, especially in R&D; and opaque filtering can reduce user trust by making the assistant appear unpredictable. The practical goal is therefore not aggressive deletion, but reversible, logged, user-controllable filtering with explicit novelty exceptions and clear deletion rights.

7.4. Limitations and Future Work

Several limitations remain. First, the online experiment was conducted within one company, one industrial domain, one KMS architecture, and one user population, so external validity must be tested in other sectors before broader claims are made. Second, ontology matching and NER are domain-specific and require retraining when HAFE is transferred to another industry. Third, although offline replay is useful for comparing baselines and ablations, it cannot fully capture behavioral adaptation by users. Fourth, the user-specific usefulness check covered a 10% subsample. Finally, confidentiality constraints prevent release of raw industrial logs; therefore, Appendix A provides an anonymized codebook, synthetic examples, and implementation parameters rather than the production dataset. Planned extensions include a user-visible “save/forget memory” interface, cross-industry evaluation, and a prospective online trial of safe adaptive threshold tuning.

In addition, the thresholds and weights reported here should not be transferred unchanged to another company or industry. They depend on local ontology maturity, the distribution of work and non-work interactions, and the business tolerance for false rejection of exploratory knowledge. Expert-based labels also reflect local norms of technical usefulness, which is why user-facing override and appeal mechanisms are important in future deployments.

8. Ethical Considerations

The study was conducted under an internal research protocol approved by the company’s data protection and information security functions. Participation was voluntary, and employees were informed that declining or withdrawing would not affect employment evaluation, access to the KMS, or managerial assessment. Consent covered pseudonymized research use of interaction metadata, memory candidates, and evaluation outcomes; users could request deletion of their personal memory contents at any time.

Personal identifiers were replaced by salted hashes before analysis. The mapping table was stored separately from the research dataset and was accessible only to a designated data protection officer. Raw interaction logs were retained for 90 days for audit and troubleshooting; after that period, only aggregated metrics and anonymized annotation records were retained. Dwell time, clicks, and follow-up actions were used solely as model features for memory quality prediction and were not exposed to line managers or HR systems.

Access to memory logs and annotations was controlled through role-based and attribute-based access controls, with audit trails for every read operation. HAFE decisions

were logged with component scores to support explainability and appeal, but individual-level profiles were not used for performance monitoring. The design follows GDPR principles of purpose limitation, data minimization, storage limitation, integrity/confidentiality, and data subject rights, and is aligned with enterprise AI governance practices such as auditability, transparency, and human oversight.

9. Conclusions

This paper presented HAFE, a hybrid pre-storage filtering algorithm for reducing persistent contextual noise in LLM-augmented knowledge management systems. The manuscript clarifies the decision function, aligns the formal notation with the implementation, separates online field evidence from offline replay baselines, and expands the annotation, statistical, ablation, error, and privacy protocols.

In a four-month online field experiment with 120 users from one industrial company, HAFE reduced the memory pollution ratio from 28.4% to 16.7%, improved MRR from 0.739 to 0.831, reduced task completion time by 23%, and increased SUS from 68 to 80. Offline replay suggests that the effect cannot be explained by simple rule-based filtering, embedding similarity, temporal decay, deduplication, or LLM-as-a-judge alone. These findings are promising, but they should be interpreted as evidence for one industrial KMS architecture rather than as proof of universal effectiveness across enterprise LLM systems.

The main practical conclusion is that persistent user memory in enterprise LLM systems should not be treated as an automatic by-product of interaction history. It requires explicit quality control, transparent audit logs, user deletion and override rights, and role-sensitive safeguards for novelty. Broader generalization requires replication in other companies, domains, and KMS architectures.

At a broader systems level, the study positions memory quality management as an emerging architectural challenge for enterprise LLM applications. As industrial assistants move from single-session question answering toward persistent, role-aware collaboration, adaptive memory governance is likely to become a core component alongside retrieval, access control, monitoring, and human oversight.

Author Contributions: Conceptualization, V.A.V. and K.V.S.; Methodology, K.V.S.; Software, R.V.K. and Y.I.K.; Validation, R.V.K.; Formal Analysis, K.V.S., G.Y.V. and V.A.G.; Investigation, V.V.K. and G.Y.V.; Resources, V.V.K. and G.Y.V.; Data Curation, V.A.V., R.V.K. and G.Y.V.; Writing—Original Draft, V.A.V.; Writing—Review and Editing, V.A.V. and A.I.K.; Visualization, V.V.K., R.V.K., V.A.G. and Y.I.K.; Supervision, V.A.G., Y.I.K. and A.I.K.; Project Administration, Y.I.K. and A.I.K.; Funding Acquisition, A.I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: All participants in this study were volunteer employees/users and did not undergo any invasive procedures, human sampling, medical intervention, or invasive experiments. The study used pseudonymized interaction data only. Therefore, ethical approval was not required for this non-invasive study.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The original contributions of this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABAC	Attribute-Based Access Control
BERT	Bidirectional Encoder Representations from Transformers
ERP	Enterprise Resource Planning
GPT	Generative Pre-trained Transformer
HAFF	Hybrid Adaptive Filtering Engine
KM	Knowledge Management
KMS	Knowledge Management System
LLM	Large Language Model
MRR	Mean Reciprocal Rank
NLP	Natural Language Processing
R&D	Research and Development
RAG	Retrieval-Augmented Generation
RBAC	Role-Based Access Control
SCADA	Supervisory Control and Data Acquisition
SUS	System Usability Scale

Appendix A Reproducibility Details (Anonymized)

The production interaction logs, user profiles, and corporate ontology cannot be published because they contain confidential industrial and employee-related information. To support reproducibility without disclosing sensitive data, this appendix provides the annotation codebook, synthetic examples, model parameters, baseline replay rules, and the LLM-as-a-judge prompt used in the offline analysis.

Table A1. Annotation codebook and synthetic examples.

Score	Codebook Criterion	Synthetic Memory Candidate	Label
1	Completely irrelevant or harmful for future work-related personalization	User asks: “What is the weather in Sochi this weekend?”	Pollution
2	Mostly irrelevant, obsolete, redundant, or likely to mislead future prompts	Old equipment setting superseded by a newer operating instruction	Pollution
3	Transient or ambiguous; useful only with user confirmation	“Prepare a report for last month” without project or plant context	Borderline
4	Useful for the user role and likely to support future work	Query about causes of surface defects in hot-rolled strip	Retain
5	Highly useful, domain-grounded, reusable knowledge or preference	User repeatedly requests standard calculations for steel grade 09G2S	Retain

Table A2. Implementation parameters and offline replay rules.

Element	Specification Used in the Reported Study
Intent classifier	Six-layer distilled BERT; 5000 labeled interactions; 70/15/15 train/validation/test split; max length 128 tokens; AdamW, learning rate 2×10^{-5} , batch size 32, four epochs; threshold $\tau_{\text{int}} = 0.70$ selected on validation F1 with false rejection $< 5\%$.
NER and ontology matching	Domain NER detects steel grades, equipment, processes, document types, defects, and project identifiers. Direct graph-node match or embedding similarity $\geq \tau_{\text{onto}} = 0.62$ counts as a match; no-domain items are rejected unless the novelty flag is present.
Reuse prediction	Gradient-boosted tree model trained on pre-study logs only; features: follow-up actions, dwell time, repeated/semantically similar queries, clicked documents, saved fragments, session length, and role context; target: similar query within 30 days.
LLM-as-a-judge prompt	“Given the user role, candidate memory, extracted entities, and current task context, decide whether this item should be stored as persistent work memory. Return KEEP or DISCARD and one concise reason. Do not use private employee attributes.”
Offline replay baselines	All baselines used the same logged candidate-memory stream. Rule-based, embedding, temporal-decay, deduplication, and LLM-as-a-judge filters were applied counterfactually and scored with the same annotation protocol; they were not additional online arms.
Ablation protocol	For each ablation, the corresponding module was removed, the remaining weights were renormalized, and the frozen model was replayed offline. No ablated variant was retrained on evaluation outcomes.

References

- Alavi, M.; Leidner, D.E. Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Q.* **2001**, *25*, 107–136. [\[CrossRef\]](#)
- Bengio, Y.; Courville, A.; Goodfellow, I. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
- Ershov, V.A.; Kondratiev, V.V.; Karlina, A.I.; Kolosov, A.D.; Sysoev, I.A. Selection of control system parameters for production of nanostructures concentrates. *J. Phys. Conf. Ser.* **2018**, *1118*, 012014. [\[CrossRef\]](#)
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
- Davenport, T.H.; Prusak, L. *Working Knowledge: How Organizations Manage What They Know*; Harvard Business School Press: Boston, MA, USA, 1998.
- Al Atrachi, O.M.A.; Abdulmuhsin, A.A.; Rehman, S.U.; Mjlae, S.A. Developing a knowledge management-large language model (KM-LLM) application in higher education: A UTAUT2 perspective. *VINE J. Inf. Knowl. Manag. Syst.* **2026**, *ahead-of-print*. [\[CrossRef\]](#)
- Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [\[CrossRef\]](#)
- Drucker, P.F. *The Age of Discontinuity*; Harper & Row: New York, NY, USA, 2007.
- Hamilton, W.L. *Graph Representation Learning*; Morgan & Claypool: San Rafael, CA, USA, 2020.

10. ISO 30401:2018; Knowledge Management Systems—Requirements. International Organization for Standardization: Geneva, Switzerland, 2018.
11. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **2021**, *7*, 535–547. [[CrossRef](#)]
12. Kaplan, R.S.; Norton, D.P. *The Balanced Scorecard*; Harvard Business Review Press: Boston, MA, USA, 1996.
13. Liu, B. *Sentiment Analysis and Opinion Mining*; Morgan & Claypool: San Rafael, CA, USA, 2012.
14. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
15. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
16. Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* **2016**, *104*, 11–33. [[CrossRef](#)]
17. Nonaka, I.; Takeuchi, H. *The Knowledge-Creating Company*; Oxford University Press: Oxford, UK, 1995.
18. Kondratiev, V.V.; Nebogin, S.A.; Sysoev, I.A.; Gorovoy, V.O.; Karlina, A.I. Description of the test stand for developing of technological operation of nano-dispersed dust preliminary coagulation. *Int. J. Appl. Eng. Res.* **2017**, *12*, 12809–12813.
19. Petrov, A.V.; Sidorova, L.K. *Artificial Intelligence in Corporate Systems*; INFRA-M: Moscow, Russia, 2024. (In Russian)
20. Polanyi, M. *The Tacit Dimension*; University of Chicago Press: Chicago, IL, USA, 1966.
21. Solodusha, S.; Suslov, K.; Gerasimov, D. Applicability of Volterra integral polynomials in the control systems of electric power facilities. In Proceedings of the 2016 International Conference Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy's Conference), Moscow, Russia, 1–3 June 2016.
22. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
23. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [[CrossRef](#)]
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008. [[CrossRef](#)]
25. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv* **2023**, arXiv:2312.10997.
26. Wiig, K.M. *Knowledge Management Foundations*; Schema Press: Arlington, TX, USA, 1993.
27. Buchgeher, G.; Gabauer, D.; Martinez-Gil, J.; Ehrlinger, L. Knowledge Graphs in Manufacturing and Production: A Systematic Literature Review. *IEEE Access* **2021**, *9*, 55537–55554. [[CrossRef](#)]
28. Vedeneev, V.A. Intelligent corporate search in a large metallurgical company. *Autom. Ind.* **2021**, *8*, 49–52. (In Russian)
29. Vedeneev, V.A.; Ershov, E.V. Method and algorithms for information processing in a corporate knowledge management system of a metallurgical company. In Proceedings of the XVII International Conference on Optical-Electronic Devices and Systems, Kursk, Russia, 12–15 September 2023; pp. 79–81. (In Russian)
30. Vedeneev, V.A.; Ershov, E.V.; Kovyrshin, R.G. Knowledge management model for a metallurgical enterprise based on embedding models. *Bull. Cherepovets State Univ.* **2025**, *3*, 7–16. (In Russian)
31. Vedeneev, V.A. Software for an intelligent search and analytical system based on scientific and technical literature. In Proceedings of the All-Russian Conference “Regional Science”, Cherepovets, Russia, 1 April–30 May 2022. (In Russian)
32. Xu, R.; Wunsch, D.C. Survey of Clustering Algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [[CrossRef](#)] [[PubMed](#)]
33. Zhuravlev, Y.I.; Lbov, G.S. *Methods for Processing Unstructured Data*; Nauka: Novosibirsk, Russia, 2021. (In Russian)
34. Packer, C.; Wooders, S.; Lin, K.; Fang, V.; Patil, S.G.; Stoica, I.; Gonzalez, J.E. MemGPT: Towards LLMs as Operating Systems. *arXiv* **2023**, arXiv:2310.08560.
35. Park, J.S.; O'Brien, J.C.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative Agents: Interactive Simulacra of Human Behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST), San Francisco, CA, USA, 29 October–1 November 2023; pp. 1–22. [[CrossRef](#)]
36. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-T.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
37. Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.-T. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6769–6781. [[CrossRef](#)]
38. Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8342–8360. [[CrossRef](#)]
39. Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; Wang, Y. MemoryBank: Enhancing Large Language Models with Long-Term Memory. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 19724–19731. [[CrossRef](#)]

40. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [[CrossRef](#)] [[PubMed](#)]
41. European Parliament and Council. Regulation (EU) 2016/679 of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation). *Off. J. Eur. Union* **2016**, *L119*, 1–88.
42. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; NIST: Gaithersburg, MD, USA, 2023. [[CrossRef](#)]
43. *ISO/IEC 42001:2023; Information Technology—Artificial Intelligence—Management System*. International Organization for Standardization: Geneva, Switzerland, 2023.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.