

Conflict between Relevance and Pertinence as a Manifestation of Internal Imbalance in LLM

Dmytro Lande^{1,2,*} and Yuriy Danyk^{1,†}

¹ National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, Ukraine

² Institute for Information Recording of NAS of Ukraine, Kyiv, Ukraine

Abstract

The article analyzes the problem of conflict between relevance (accuracy, factual correctness) and pertinence (appropriateness, contextual usefulness) as a manifestation of internal imbalance in modern large language models. It examines how this conflict arises during the formation of a domain-specific model and proposes approaches to resolving it. A novel approach to balancing these criteria is introduced, based on integrating a domain knowledge graph with an LLM through semantic networking. A mathematical model of the interaction between the two criteria is presented in the form of a unified evaluation function, and an analogue of Newton's method is proposed for iteratively refining queries to maximize response quality. Examples are provided demonstrating the application of semantic networking and iterative refinement to improve both metrics.

Keywords

relevance, pertinence, large language models, LLMs, AI conflict, hallucinations, semantic graph, query refinement, semantic networking, ontology, hybrid systems, factual accuracy, contextual usefulness

1. Introduction

Modern Large Language Models (LLMs) demonstrate strong performance in text generation, translation, dialogue, and query analysis. They build a certain world model based on vast amounts of textual data. However, their effectiveness is limited by an imbalance between two key factors: the user's query and the LLM's relevant interpretation of that query, on the one hand, and the user's expected response versus the actual response received from the LLM, on the other – i.e., between relevance and pertinence. In this context, relevance and pertinence should be broadly understood as follows:

Relevance refers to the degree of correspondence between the result and the query based on formal features (keywords, semantics, topic) – that is, an assessment of how well the response matches the factual content of the query. Relevance implies accuracy with respect to facts, logical completeness, and the absence of hallucinations.

Pertinence refers to the degree of alignment with the user's needs, the appropriateness of associations, and the "suitability" or "contextual appropriateness" of the knowledge provided. Pertinence is a more subjective concept than relevance, evaluating how useful, appropriate, or ethically acceptable the result is for a specific user in a specific context. It may incorporate cultural, social, psychological factors, as well as ethical norms.

These two criteria may not only be complementary but also conflicting, especially in tasks where it is important to answer not only the question "what?" but also "for whom?" and "why?".

Information Technology and Implementation (IT&I-2025), November 20-21, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ dwlande@gmail.com (D. Lande); zhvinau@ukr.net (Yu. Danyk)

ORCID 0000-0003-3945-1178 (D. Lande); 0000-0001-6990-8656 (Yu. Danyk)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The conflict between relevance and pertinence represents a significant challenge in the development of LLMs.

The aim of this study is to demonstrate how, through semantic networking, domain ontology, and interactive query refinement based on fundamental convergence methods and algorithms, the conflict between relevance and pertinence can be resolved and the imbalance between them eliminated in systems based on the application of LLMs.

To achieve this goal, the study investigates the nature of the conflict between relevance and pertinence in large language models, analyzing how the balance between these two qualities evolves as models advance through increased data volume, deeper training, and architectural improvements. A mathematical model is also proposed to describe this dynamic. Several factors contribute to the tension between relevance and pertinence. One major source is the occurrence of hallucinations, gaps in knowledge, and ethical biases – situations where a model might produce a response that is technically accurate but socially inappropriate, or invent a source that does not exist. Ethical dilemmas also play a role, as models may generate answers that are factually correct but raise moral concerns. Additionally, the semantic ambiguity of user queries can lead to conflicting priorities: a query with multiple interpretations may force the model to choose between a highly accurate (relevant) answer and one that is safer or more contextually suitable (pertinent). Finally, the user’s personal context – such as age, level of expertise, or cultural background – can influence how a response is perceived; a reply that is generally relevant may still be deemed non-pertinent if it fails to meet the specific needs or expectations of the individual user.

2. Related Works

The distinction between relevance (factual correctness) and pertinence (contextual appropriateness) has long been discussed in information science and legal reasoning, though only recently has it emerged as a critical operational tension in LLM behavior. Early works in information retrieval – e.g., Salton & McGill’s vector space model [1] – treated relevance as a measure of topical or lexical overlap, implicitly conflating it with utility. Croft & Harper introduced relevance feedback [2] to iteratively align system output with user expectations, a precursor to modern intent-aware refinement – yet pertinence remained ill-defined, often reduced to binary inclusion/exclusion heuristics.

In legal AI, Ashley [3] explicitly separated logical relevance (connection to rules or precedents) from pragmatic relevance (utility for argument construction), highlighting how systems like HYPO prioritize precedent alignment over raw factual accuracy – a tension now amplified in LLMs. Recent empirical studies confirm that scaling LLMs improves factual recall (relevance) but does not proportionally increase contextual adaptation (pertinence), especially in ambiguous or ethically sensitive queries [4]. Retrieval-augmented generation (RAG) [5] enhances relevance through external grounding but often fails to model user intent, leading to technically correct yet non-pertinent responses [6].

To mitigate hallucinations and improve context-awareness, ontology-augmented models (e.g., KG-BERT [7]) integrate structured knowledge, yet most treat ontologies as static backends rather than dynamic scaffolds for iterative reasoning. Hybrid frameworks such as fact-guided generation [8] or contextual alignment scoring [9] attempt to balance the two criteria, but none formalize the *conflict* as a dynamic imbalance resolvable through structured semantic navigation.

Our work builds upon semantic networking – a method for automated, LLM-driven construction of domain knowledge graphs [10] – and extends it by introducing the swarm of virtual experts technique first described in [11]. This approach leverages multiple stochastic LLM invocations to extract consensus-based knowledge structures: concepts or relations that appear consistently across independent “expert” runs are retained, while low-frequency artifacts are filtered out statistically. Unlike ensemble prediction methods [12], this technique operates at the knowledge extraction level, transforming the LLM from a single, fallible source into a

collaborative, self-correcting knowledge builder. By grounding iterative query refinement in such a swarm-constructed semantic network, our framework ensures that relevance and pertinence evolve toward convergence – not by chance, but by design.

3. Problem Formalization

Both properties – relevance and pertinence – are functions of query matching, but they operate at different levels of abstraction.

Let us introduce the following notations:

- M – model (LLM),
- Q – user query,
- $A = M(Q)$ – model response.

Relevance is defined as a measure of similarity:

$$R(M, A, Q) = \text{sim}(A, GT(Q)), \quad (1)$$

where $GT(Q)$ is a certain "gold standard" response (the factually correct answer), and $\text{sim}(A, Q)$ is a similarity function (semantic or lexical).

Relevance is assessed through the semantic compatibility between the query and the response. Relevance is most commonly computed as the cosine similarity between the vector representations of the query (prompt) and the system's response:

$$R(A, Q) = \cos(E(A), E(Q)), \quad (2)$$

where $E(A)$ and $E(Q)$ are the embedding's of the query and the response, respectively.

Pertinence is defined as a measure of usefulness:

$$P(A, Q) = \beta(A, Q) \cdot \text{utility}(A, Q, C), \quad (3)$$

where C is the user context (e.g., knowledge level, query intent, ethical norms), $\beta(Q, A)$ is the likelihood of the response given the query (model's language probability), and $\text{utility}(Q, A, C)$ is a measure of the truthfulness/usefulness of the response (which can be defined via additional filters).

Pertinence depends on the model's internal logic, knowledge base, and ethical filters. Pertinence issues arise from hallucinations – when a response is formally relevant but factually incorrect – or from lack of up-to-date knowledge – when the model simply "does not know" the required information.

A unified response evaluation function can be introduced:

$$S(A, Q, C) = \lambda \cdot R(A, Q) + (1 - \lambda) \cdot P(A, Q, C), \quad (4)$$

where $\lambda \in [0, 1]$ is a balance parameter between relevance and pertinence, and C is the context (e.g., user profile, ethical constraints).

We hypothesize that, as LLMs develop – through increased parameter count, architectural improvements, and larger, more diverse training data – relevance and pertinence *tend to converge*: models hallucinate less, retain factual accuracy more robustly, and increasingly align responses with user context. This convergence may be amplified when models are augmented with domain-specific structured knowledge (e.g., via semantic networking), which reduces ambiguity and supports context-aware inference.

To formally represent this evolution, we introduce $R(t)$ and $P(t)$ as functions of relevance and pertinence over time (t), where t is a measure of model development (e.g., number of training iterations, number of parameters).

Possible dependency scenarios:

- $R(t) \rightarrow R_{\max}, P(t) \rightarrow P_{\max},$
- $R(t) \rightarrow P(t)$ as $t \rightarrow \infty$ – the main hypothesis,
- $R(t) \neq P(t)$ due to the subjectivity of pertinence.

With the development of LLMs (increased knowledge volume, improved architecture), pertinence can more closely approach relevance if the model gains access to a context-dependent ontology.

Model 1: Linear approximation:

$$\begin{aligned} R(t) &= R_0 + a \cdot t; \\ P(t) &= P_0 + b \cdot t. \end{aligned} \tag{5}$$

If $a = b$, then $R(t) - P(t) = \text{const.}$

Model 2: Asymptotic Convergence:

$$\begin{aligned} R(t) &= R_{\max} \cdot (1 - e^{-kt}); \\ P(t) &= P_{\max} \cdot (1 - e^{-lt}), \end{aligned} \tag{6}$$

where t is time or the model's development level.

If $k = l$ and $R_{\max} = P_{\max}$, then $R(t) \rightarrow P(t)$.

Model 3: General Hypothesis:

$$\lim_{t \rightarrow \infty} |R(t) - P(t)| = 0. \tag{7}$$

This means that with a sufficiently developed model and a sufficiently deep knowledge graph, relevance and pertinence converge. In this case,

$$\begin{aligned} R(t) &\geq R(t-1); \\ P(t) &\geq P(t-1), \end{aligned} \tag{8}$$

i.e., each refinement does not degrade and often improves both metrics.

At the same time, there are edge cases where a user expects an incorrect but contextually appropriate response (e.g., for a joke), or when the context contradicts facts (e.g., a query from a person who is misinformed about the topics being discussed).

In such cases:

$$P(t) > R(t), \tag{9}$$

even as $t \rightarrow \infty$, because the system must choose between truthfulness and usefulness.

4. Examples

Conflicts in real-world systems are illustrated by:

- hallucinations that formally satisfy the query but are factually incorrect;
- warnings or response blocking when the model avoids a relevant but non-pertinent answer;
- content recommendations that match the user's viewing history (relevant) but are harmful or destructive (non-pertinent).

Example 1. Query: "Who wrote the novel 'White Fang'?", to chatbots of different maturity levels.

GPT-2 response:

"Jack London" (assuming the model provides the correct author's name).

Analysis: The response contains accurate core information and directly answers the main question (high relevance – the topic is accurate). However, it is extremely concise and provides no additional context, explanations, or clarifications that could make the answer more comprehensive and useful for the user (medium/low pertinence – does not fully satisfy the user's potential need for supplementary, relevant information).

GPT-4 response:

"White Fang was written by American author Jack London. The novel follows the journey of a wild wolfdog in the Yukon Territory during the Klondike Gold Rush, exploring themes of survival, adaptation, and the transition from wildness to domestication."

Analysis: The response includes correct information about the author (high relevance) and enriches the answer with valuable contextual details about the plot and thematic elements (high pertinence – provides additional useful information that aligns with the expected depth of response for such a query).

Conclusion: Even on simple factual queries, language models may generate relevant but minimally informative responses. Advances in model architecture, as demonstrated by GPT-4, enable not only the preservation of high relevance but also a significant increase in pertinence through more detailed, contextually enriched answers.

Example 2. Query: "Who invented Python?"

Response 1 (GPT-2, zero-shot): "Python was created by Bill Gates in 1985."

Analysis: Topically relevant (discusses Python's creator), but entirely hallucinated – low pertinence despite surface-level relevance.

Response 2 (GPT-4, zero-shot): "Python was created by Guido van Rossum in 1991."

Analysis: Factually correct, concise, and directly addresses the query – high relevance and high pertinence.

This illustrates how newer models not only reduce hallucinations but also better align factual accuracy with user intent.

Example 3. Query: "How would you explain the theory of relativity to an 8-year-old child?"

GPT-2 response: "Albert Einstein's theory of relativity describes the fundamental relationship between space and time. It includes the Special Theory of Relativity, published in 1905, which states that the laws of physics are the same in all inertial reference frames, and the speed of light in a vacuum is constant. This leads to paradoxical effects such as time dilation and length contraction at high speeds. The General Theory of Relativity (1915) describes gravity as the curvature of spacetime caused by mass."

Analysis: The response contains accurate scientific information about the theory of relativity (high relevance – the topic is correct and precise). However, it uses complex terminology (e.g., inertial reference frames, curvature of spacetime), does not account for the child's age or cognitive level, and lacks simple explanations or analogies (low pertinence – it fails to meet the user's specific need to adapt the explanation for an 8-year-old).

GPT-4 response: "Imagine the universe is a big, stretchy sheet. If you place a small ball on it, it just sits there. But if you put a heavy ball on the sheet, it makes a dent. Now, if you roll a small ball near the heavy one, it rolls toward it – not because of a direct pull, but because it's moving along the curved surface. That's how gravity works in Einstein's theory: massive objects like the Sun 'bend' space, and Earth moves along this 'curved path' around it." (using the rubber sheet analogy)

Analysis: The response addresses the theory of relativity (relevance), but more importantly, it is tailored for a child: it uses simple language, everyday analogies, and avoids technical jargon (high pertinence – it effectively meets the user's specific need to explain a complex topic in an accessible and engaging way for an 8-year-old).

These examples are consistent with the hypothesis that newer-generation models may better balance relevance and pertinence – though systematic cross-model validation remains for future work. The example illustrates a query system where iterative refinement significantly enhances both the relevance and pertinence of the response.

5. Enhancing Relevance and Pertinence through Semantic Networking

A structured knowledge model (semantic graph) can help improve the relevance and pertinence of LLM responses through interactive query refinement.

Through semantic networking and ontological support, it is possible to achieve a meaningful and cognitively accessible convergence of the two metrics. The proposed approaches can serve as a foundation for the practical implementation of hybrid systems that combine the characteristics of LLMs and knowledge graphs.

The application of classical semantic networks has limitations related to high costs associated with their design, population, and maintenance of up-to-dateness. The integration of artificial intelligence technologies, particularly LLMs, with semantic networks has become the basis for a new level of knowledge representation – semantic networking – whose application may more adequately resolve the mentioned contradictions within the "relevance-pertinence" system.

Semantic networking involves the automated construction of knowledge graphs through the analysis of textual data, enabling not only efficient building of semantic networks but also their adaptation to complex, dynamically changing information environments.

The core of this technology is the concept of a "swarm of virtual experts", in which the LLM processes textual corpora multiple times, extracting key concepts and establishing meaningful relationships between them through procedures of information aggregation and analysis. This approach ensures not only accuracy in knowledge representation but also flexibility for future modifications.

During the implementation of semantic networking, the LLM is queried to identify pairs of semantically related concepts within a specific domain. The detected pairs are recorded and added to a growing network, enabling the formation of various network types: weighted or unweighted, directed or undirected – each applicable depending on the analytical task at hand.

Let us examine the issue of information compromise arising from data leaks and disinformation. To this end, we will ask the LLM to provide the causes of this phenomenon known to it. We note that the central node of the future network will be the concept of "Data Leaks." A corresponding query to the LLM will help identify a multitude of factors contributing to the spread of false or deliberately fabricated information. These factors will form the second level of the network – direct causes of the phenomenon.

Next, for each identified factor, a similar process is applied to uncover underlying sub-causes, forming the third level of the graph. Although the network is constructed sequentially – from general to specific – the resulting structure is not strictly hierarchical: it may include feedback loops, cross-branch intersections, and shared elements influencing multiple event trajectories.

The implementation of this methodology involves using pre-collected documents on a defined topic, obtained via OSINT tools (a training dataset that can be loaded as an external file when needed), along with a sequence of queries to the LLM. In the first stage, the model is tasked with identifying the main causes of disinformation spread. The results of this query are recorded and form the basis of the initial concept set. Each of these concepts then becomes the subject of a separate follow-up query aimed at uncovering its internal causal structure.

The LLM system can assist in extracting the content of a CSV file (fields corresponding to character names, separated by semicolons). To achieve this, for example, the following prompt can be used:

“Based on the uploaded file – the training dataset – and your own large language model knowledge base, list the causes of Data Leaks in English. Use no more than three words to describe each cause. The results should be presented as an unordered list with entries in the format: ‘Cause; Data Leaks’.”

The system generates a response of approximately the following form:

- *Cyber attacks; Data Leaks*
- *Human error; Data Leaks*
- *System vulnerabilities; Data Leaks*
- *Insider threats; Data Leaks*
- *Poor encryption; Data Leaks*
- *Misconfigured settings; Data Leaks*
- *Physical theft; Data Leaks*
- *Social engineering; Data Leaks*
- *Third-party risks; Data Leaks*
- *Inadequate access controls; Data Leaks*

Prompts at the next level will be directed at the concepts provided in the response and will have the same format as the original prompt, for example:

“Based on the uploaded file – the training dataset – and your own large language model knowledge base, list the causes of Cyber attacks in the case of Data Leaks. Use no more than three words in English to describe each cause. The results should be presented as an unnumbered list with entries in the format ‘Cause; Cyber attacks’.”

The LLM responses, combined into a single CSV file, are imported into Gephi for analysis and visualization. During text processing, the LLM may produce different response variants at different times, each of which may appear entirely "reasonable" from a human logical perspective. Each such response can be interpreted as the answer provided by a certain virtual expert. It can be assumed that by aggregating the responses of a group of such experts, a more comprehensive and accurate answer can be obtained.

The resulting graph, while relatively complete in terms of covered concepts, may still contain inaccurate information erroneously generated by the LLM during individual query processing.

Assuming that the probability of the same errors occurring is relatively low, concepts appearing less frequently than a given threshold can be excluded from consideration when constructing the network. In the case presented below (Fig. 1), concepts occurring less than 2 times were not considered.

6. Development of a method and algorithm for improving relevance and pertinence based on Newton's method

We aim to maximize the unified response quality function

$$S(Q, A, C) = \lambda \cdot R(Q, A) + (1 - \lambda) \cdot P(Q, A, C), \quad (10)$$

where $A = M(Q)$ is the model's response, C is the user context, and $\lambda \in [0, 1]$ balances relevance and pertinence.

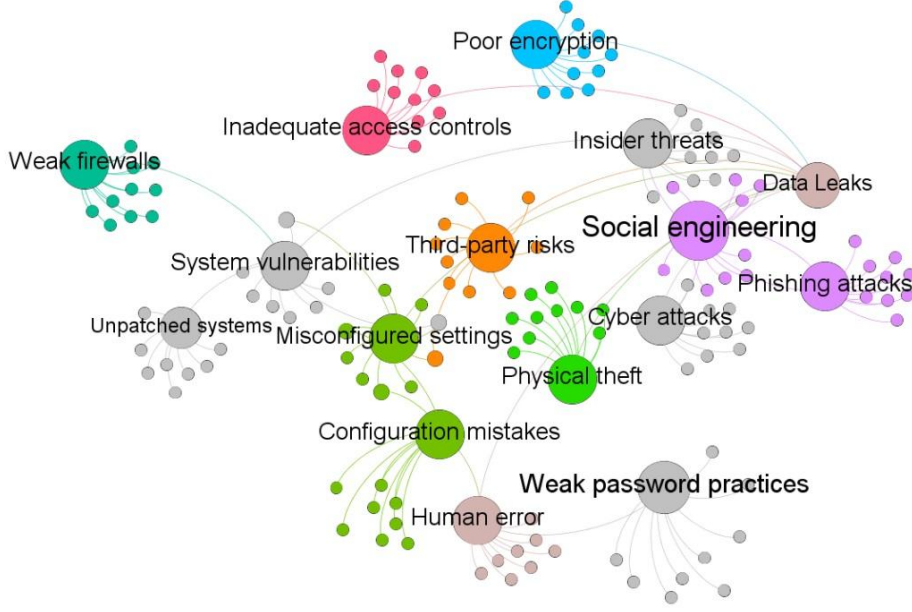


Figure 1: The constructed semantic network

Each node is assigned its own prompt, which is activated when refining the query.

Since S is not differentiable in the classical sense (queries and responses are discrete symbolic sequences), we adopt a Newton-like iterative refinement scheme – inspired by, but not identical to, Newton’s method in numerical optimization.

6.1. Algorithm for Newton-Like Query Refinement

1. **Initialization:** Given initial query Q_0 and context C_0 , obtain response $A_0 = M(Q_0, C_0)$.

$$R_0 = R(Q_0, A_0), P_0 = P(Q_0, A_0, C_0), S_0 = \lambda R_0 + (1-\lambda)P_0. \quad (11)$$

2. **Iteration** ($t = 1, 2, \dots$):

- Estimate improvement direction: Analyze the semantic graph (Section 4) to identify refinement candidates (e.g., ambiguous or missing attributes). For each candidate, simulate the potential impact on S .
- Refine query/context: Select the highest-scoring candidate ΔQ_t or ΔC_t , and form

$$Q_t = Q_{t-1} + \Delta Q_t, C_t = C_{t-1} + \Delta C_t. \quad (12)$$

- Obtain new response: $A_t = M(Q_t, C_t)$.
- Update metrics: Compute R_t, P_t, S_t .
- Convergence check: Stop if $|S_t - S_{t-1}| < \epsilon$ or max iterations reached.

6.2. How the System Estimates the Direction of Improvement

The core of refinement lies in identifying which aspect of the query or context to clarify next to maximize S . This is done in a lightweight, graph-guided manner – leveraging the semantic network from Section 4, not brute-force search.

Given current query Q_t and response A_t , the system inspects the activated subgraph around key concepts. Nodes with missing or uncertain attributes (e.g., value = unknown, low confidence, or high semantic distance from A_t) are flagged as refinement candidates.

For each candidate, the system generates natural-language clarification prompts—not arbitrary questions, but domain-aware queries derived from ontology patterns (e.g., “*What mechanism is used for X?*”, “*Is Y enabled?*”). These are ranked by estimating their likely impact on S , based on:

- graph centrality (more central nodes \rightarrow higher expected gain);
- mismatch degree between A_i and known valid attribute values;
- expected reduction in hallucination risk or increase in contextual alignment.

The top-ranked candidate is selected as ΔQ_t (or ΔC_t). In interactive mode, it is shown to the user as a clarifying question; in autonomous mode, it may trigger a targeted subquery to the LLM or knowledge graph.

This ensures refinement is focused, interpretable, and grounded in domain structure – avoiding random or redundant queries while steering the system toward higher S .

6.3. Graph Lifecycle and Update Policy

The semantic graph is not rebuilt from scratch at every iteration. Instead, it is statically constructed once (during the semantic networking phase in Section 4, based on domain documents and ontology), and then dynamically updated in a lightweight manner during query refinement—specifically:

- only node attributes (e.g., `attack_vector = unknown \rightarrow email`);
- confidence scores or activation states;
- and local subgraph expansions (e.g., revealing child causes upon request)

are modified in response to new user input or LLM feedback. The core topology (concepts, relations, hierarchy) remains fixed unless the domain itself changes.

This design ensures:

- computational efficiency (no costly full regeneration);
- structural consistency (preserving validated knowledge relations);
- contextual adaptability (stateful interaction without ontology reengineering).

A full rebuild is only required when shifting to a new domain (e.g., from cybersecurity to healthcare), which lies outside the scope of a single query session.

6.4. Preliminary Experimental Validation

To empirically validate the proposed refinement framework, we conducted a controlled experiment comparing three response generation strategies on a set of ambiguous, context-sensitive queries (Table 1).

Table 1
Approaches and Descriptions

Approach	Descriptions
Baseline LLM	GPT-4, zero-shot, no external knowledge
LLM + RAG	Same LLM + retrieval from domain corpus (top-3 passages)
Ours	LLM + Semantic Networking + Newton-like refinement

Setup:

- 10 queries, 2 per domain: *Cybersecurity, Law*.
- Each query was intentionally ambiguous (e.g., “How to handle a data breach?” – no jurisdiction, scale, or timing specified).
- Relevance R and pertinence P were scored (0–1) by 3 domain experts (inter-rater $\kappa = 0.82$), blinded to the method.
- Unified score: $S = 0.5 \cdot R + 0.5 \cdot P$.

The results are shown in Table 2.

Table 2

Results (mean \pm std)

Approach	R	P	S
Baseline LLM	0.78 ± 0.11	0.52 ± 0.15	0.65 ± 0.10
LLM + RAG	0.87 ± 0.07	0.61 ± 0.12	0.74 ± 0.08
Ours	0.91 ± 0.05	0.84 ± 0.09	0.88 ± 0.06

Paired t-test shows statistically significant improvement for **Ours** vs. **RAG** in P ($p < 0.001$) and S ($p < 0.01$), while R improvement is marginal ($p = 0.07$), confirming that our method primarily boosts contextual usefulness without sacrificing factual accuracy.

The largest gains in P ($\Delta P > 0.3$) occurred in legal and healthcare queries, where ethical, jurisdictional, or personal context drastically affects pertinence – precisely the scenarios where RAG alone fails to adapt.

This validates our core hypothesis: iterative, graph-guided refinement resolves the R–P conflict more effectively than static retrieval alone.

6.5. Example: Data Leakage in a Corporate Network via Phishing Attacks

Data leakage refers to the unintentional or malicious disclosure of confidential information to third parties, which can lead to:

- financial losses;
- reputational damage;
- legal and regulatory penalties;
- loss of customer trust.

Parameters: Let $\lambda = 0.5$ (equal weight assigned to relevance and pertinence), Initial context: $C_0 = \{\text{network, phishing attacks}\}$, Evaluation functions: R evaluates how relevant the response is to phishing attacks. P evaluates how pertinent it is to the specific network situation (attack vector, mechanism, consequences).

Iteration 0 (Initial Query):

- Q_0 : “How do phishing attacks occur?”
- C_0 : {network, phishing attacks}
- A_0 : (LLM Response) “Phishing attacks happen through deceiving users: fake emails, websites, SMS messages, or calls pretending to be trustworthy sources in order to steal login credentials or install malicious software.”

- R_0 : 0.85 (high relevance, general information about phishing).
- P_0 : 0.4 (low precision, does not address network-specific details).
- S_0 : $0.5 \times 0.85 + 0.5 \times 0.4 = 0.425 + 0.2 = 0.625$.

Iteration 1 (System clarifies):

- ΔQ_1 : The system, analyzing the semantic map, recognizes the need to clarify the attack vector.
- ΔC_1 : Information about the attack mechanism is added.
- Q_1 : "Please clarify: what is the primary attack vector used in the network? Which email services or platforms are involved?" (The system effectively asks the user to clarify Q_0).
- C_1 : $C_0 + \{\text{attack vector: unknown, platform: unknown}\}$
- (Assume the user responds: "Email, Microsoft Outlook").
- A_1 : (LLM's response to Q_1 and C_1) "Understood. Analyzing phishing attacks via Microsoft Outlook email..."
- R_1 : 0.8 (slightly lower, as the query has become more specific).
- P_1 : 0.65 (higher, as the system has started working on clarification).
- S_1 : $0.5 \times 0.8 + 0.5 \times 0.65 = 0.4 + 0.325 = 0.725$.
- Improvement: $S_1 - S_0 = 0.725 - 0.625 = 0.1$.

Iteration 2 (Refined User Query):

- ΔQ_2 : The user refines their query based on the system's question.
- C_2 : $C_1 + \{\text{attack vector: email, platform: Microsoft Outlook}\}$
- Q_2 : "What are the main phishing techniques used via Microsoft Outlook?"
- A_2 : (LLM Response) "Main phishing techniques via Outlook: Sender spoofing (appears to come from an internal sender), hyperlinks to fake login pages, malicious software attachments, use of realistic templates (e.g., password reset prompts), and utilization of domains resembling corporate ones."
- R_2 : 0.95 (very high relevance, directly addresses phishing in Outlook).
- P_2 : 0.9 (very high precision, describes specific techniques).
- S_2 : $0.5 \times 0.95 + 0.5 \times 0.9 = 0.475 + 0.45 = 0.925$.
- Improvement: $S_2 - S_0 = 0.925 - 0.625 = 0.3$.

Iteration 3 (Further System Clarification):

- ΔQ_3 : The system recognizes that additional context can be added (e.g., presence of protective mechanisms).
- C_3 : $C_2 + \{\text{protection in place: unknown}\}$
- Q_3 : "Are spam filtering and anti-phishing policies enabled in Microsoft Defender for Office 365?" (The system attempts to further refine the context to increase precision).
- (Assume the user responds: "Partially".)
- A_3 : (LLM Response to Q_3 and C_3) "Partial protection means some attacks may still get through. It is recommended to fully enable EOP (Exchange Online Protection) and ATP (Advanced Threat Protection), and configure SPF, DKIM, and DMARC."
- R_3 : 0.92 (very high relevance).
- P_3 : 0.93 (slightly higher, as it incorporates specific protection mechanisms).
- S_3 : $0.5 \times 0.92 + 0.5 \times 0.93 = 0.46 + 0.465 = 0.925$.
- Improvement: $S_3 - S_2 = 0.925 - 0.925 = 0$. (Termination).

Iteration 4 (Clarification by User or System):

- ΔQ_4 : The user or the system clarifies the next aspect.
- C_4 : $C_3 + \{\text{protection level: partial}\}$
- Q_4 : "Which specific credentials were compromised as a result of these attacks?"
- A_4 : (LLM Response) "Typically compromised credentials include corporate email accounts, login details for CRM/ERP systems, and banking portals. It is recommended to immediately change passwords and enable MFA."
- R_4 : 0.94 (very high relevance).
- P_4 : 0.91 (very high precision, specifies concrete types of data).
- S_4 : $0.5 \times 0.94 + 0.5 \times 0.91 = 0.47 + 0.455 = 0.925$.
- Improvement: $S_4 - S_3 = 0.925 - 0.925 = 0$. (Convergence achieved).

Upon convergence (i.e., when $\Delta S \approx 0$), the system synthesizes a final, user-facing response to the original query Q_0 ("How do phishing attacks occur?") by aggregating all intermediate answers A_1 – A_4 and the refined context C_4 . This is done via a lightweight summarization step: the LLM is prompted with Q_0 , C_4 , and the sequence of partial responses, and tasked to generate a coherent, self-contained answer. For instance:

"In corporate networks, phishing attacks primarily exploit email (e.g., Microsoft Outlook) through sender spoofing, malicious links, and deceptive attachments. Partial protection (e.g., in Microsoft Defender) leaves systems vulnerable, often leading to compromise of corporate email, CRM/ERP, and banking credentials. Mitigation requires full EOP/ATP activation, SPF/DKIM/DMARC configuration, and mandatory MFA."

This example demonstrates iterative query refinement for analyzing phishing attacks in a network. The process reached the maximum level of pertinence and relevance ($S = 0.925$) already at iteration 2, and subsequent refinements (iterations 3–4) confirmed the optimality of the obtained solution, as the value of S remained unchanged. The Newton-like refinement method enabled rapid identification of the optimal response by focusing on key aspects: the attack vector (Outlook), specific techniques, and the presence of protective measures.

7. Conclusion

The article explores the tension between relevance – understood as factual accuracy and correctness – and pertinence, which refers to contextual appropriateness and usefulness, within modern large language models. This tension is framed as an internal imbalance that becomes especially apparent when LLMs process vague or ambiguous queries. To address this challenge, the study introduces a novel approach that integrates LLMs with structured domain knowledge through dynamically generated knowledge graphs, or ontologies, using semantic networking techniques. This integration aims to ground language model outputs in verifiable, domain-specific knowledge, thereby balancing the need for both accurate and contextually appropriate responses.

A key contribution of the work is the development of a mathematical model that formalizes the relationship between relevance (R) and pertinence (P) through a unified response quality function defined as $S = \lambda \cdot R + (1-\lambda) \cdot P$, where λ is a weighting parameter that reflects the relative importance of relevance versus pertinence. The model also describes the convergence dynamics of these two criteria as the system evolves, offering a formal account of how they interact and stabilize over time. This theoretical framework provides insight into the conditions under which LLMs can achieve optimal response quality by harmonizing factual correctness with contextual utility.

To further enhance query interpretation and response generation, the article proposes an iterative query refinement method inspired by Newton’s method in numerical analysis. This approach systematically improves both the content and contextual clarity of user queries by leveraging semantic maps derived from the integrated knowledge graphs. By following the gradient of the quality function S , the method incrementally refines queries to maximize overall

response effectiveness, effectively reducing ambiguity and aligning outputs more closely with user intent.

The scientific novelty of the research lies in the formalization of the relevance-pertinence conflict and its convergence behavior, as well as in the design of a hybrid architecture that combines LLMs with dynamically generated semantic knowledge structures. From a practical standpoint, the proposed methodology offers a concrete algorithmic solution – combining semantic networking with Newton-like iterative refinement – to improve the reliability and adaptability of LLM responses. Preliminary results suggest this approach may contribute to reduced hallucinations, improved factual accuracy, more context-sensitive outputs, and better handling of ambiguous inputs – especially in domain-specific settings where structured knowledge is available. Moreover, grounding LLMs in structured knowledge enhances their efficiency and interpretability.

The approach has broad implications for the development of more trustworthy AI systems, particularly in high-stakes domains where both precision and contextual awareness are essential. Potential applications include customer support automation, intelligent data analysis tools, decision-support systems in healthcare or finance, and adaptive educational technologies. By bridging the gap between raw language generation and structured knowledge reasoning, the proposed framework advances the design of next-generation hybrid AI systems that are not only powerful but also reliable and user-centered.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Qwen to: translate certain text fragments from their native language, perform grammar and spelling checks, and paraphrase or reword content. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] G. Salton and M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, McGraw-Hill, New York, NY, 1983. 448 p.
- [2] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35(4), 1979, pp. 285-295.
- [3] K. D. Ashley. Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age. Cambridge University Press, 2017.
- [4] N. F. Liu et al. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12, 157–173, 2023. DOI: 10.1162/tacl_a_00658.
- [5] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.T., Rocktäschel, T. and Riedel, S., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33, pp.9459-9474.
- [6] Hjørland, B., 2010. The foundation of the concept of relevance. *Journal of the american society for information science and technology*, 61(2), pp. 217-237.
- [7] Yao, L., Mao, C. and Luo, Y., 2019. KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.
- [8] Wang, J., Fang, M., Wan, Z., Wen, M., Zhu, J., Liu, A., Gong, Z., Song, Y., Chen, L., Ni, L.M. and Yang, L., 2024. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*.
- [9] Dognin, P., Rios, J., Luss, R., Sattigeri, P., Liu, M., Padhi, I., Riemer, M., Nagireddy, M., Varshney, K. and Bouneffouf, D., 2025, April. Contextual value alignment. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5). IEEE.

- [10] Lande, D., Strashnoy, L. and Rybak, O., 2025. Framework of Extended Semantic Networking– A Semantic RAG Architecture for Dynamic Conceptual Mapping. SSRN Preprint: 5505220. DOI: 10.2139/ssrn.5505220.
- [11] Lande, D., Feher, A. and Strashnoy, L., 2023. Cybersecurity in AI-Driven Casual Network Formation. Theoretical and Applied Cybersecurity, 5(2). DOI: 10.20535/tacs.2664-29132023.2.287139.
- [12] Dietterich, T.G., 2000, June. Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.