MSc Artificial Intelligence
Master Thesis

Automated Topic Page Generation Using Multi-Agent LLMs

by
Danilo Toapanta
14077566

June 30, 2025

36 ECs
02/12/2024 – 30/06/2025

Supervisor:
MSc A Artemis Capari

Other Reader:
MSc A Zahra Abbasiantaeb

Second reader:
Dr A Hosein Azarbonyad

Universiteit van Amsterdam

Elsevier

Contents

Abstract

The exponential growth of scientific literature has created unprecedented challenges for researchers seeking to access specialized knowledge. While there exist platforms that address this challenge, such as ScienceDirect's Topic Pages, these platforms primarily offer fragmented information consisting of isolated snippets without cohesive synthesis. This fragmentation imposes a significant cognitive burden on users who must manually connect disparate pieces of information to develop a comprehensive understanding of a complex topic.

To address these limitations, we present Apollo, a multi-agent framework for creating Wikipedia-like topic pages from scientific literature. Apollo employs an iterative knowledge curation process that constructs knowledge graphs from scientific snippets, uses collaborative agents to systematically expand topic exploration, and generates well-referenced content through specialized writer and reviewer agents. We introduce SciWiki-2k, a benchmark dataset of 2,000 high-quality Wikipedia articles spanning 20 scientific domains, to evaluate automated scientific content generation.

We evaluate Apollo against state-of-the-art baselines including STORM, OmniThink, and oRAG using comprehensive methodologies incorporating automatic metrics, LLM-as-judge assessments, and human expert evaluations. Our graph-based knowledge curation approach retrieved significantly more unique snippets and achieved greater information diversity compared to baseline methods. Apollo demonstrated superior outline quality with advantages in coherence and logical organization, validated through preference tests across multiple LLM evaluators. For article generation, Apollo produced content that closely resembles human-written articles, as evidenced by ROUGE scores and entity recall metrics, with particular strengths in interest, depth, and relevance.

Most significantly, Apollo achieved substantially lower hallucination rates (5.70% compared to STORM's 34.34%) while maintaining superior citation coverage. The critical reviewer agent proved essential for factual grounding, with ablation studies confirming that iterative refinement significantly enhances content reliability. Human expert evaluation validated strong alignment with automated assessments, with subject matter experts rating Apollo higher on 8 out of 9 metrics, particularly in content depth, coverage, and factual verifiability.

These findings demonstrate that multi-agent frameworks with structured knowledge curation can significantly improve automated scientific content generation, offering a promising approach for addressing information synthesis challenges in rapidly expanding scientific domains.

Chapter 1

Introduction

The exponential growth of scientific literature has created unprecedented challenges for researchers, educators, and students seeking to access specialized knowledge [1]. With millions of research articles published annually, finding reliable and relevant sources among this vast amount of information has become increasingly complex and time-consuming [2]. Researchers often struggle to identify the most pertinent information for understanding specific concepts, particularly when encountering unfamiliar topics outside their primary expertise.

Recognizing this challenge, ScienceDirect, Elsevier's scientific database platform, introduced Topic Pages [3]. This knowledge base helps users understand scientific concepts across 20 domains by presenting snippets from peer-reviewed journals, articles, and books alongside concept definitions and related topics [4]. For example, the machine learning Topic Page displays a definition and relevant snippets covering various aspects about this concept[1]. However, while these pages successfully address the problem of presenting reliable sources, they primarily offer fragmented information consisting of isolated snippets without cohesive synthesis. As a result, users may struggle to (1) develop a holistic understanding of the topic, and (2) gain in-depth knowledge due to the cognitive burden of connecting disparate pieces of information [5]. Ideally, users would benefit from topic pages that synthesize sources into coherent narratives, as research has shown improved learning outcomes with synthesized information [6, 7]. Unfortunately, creating such integrated content remains highly labor-intensive and time-consuming [8], making it impractical to produce at large scale.

Recent advances in large language models (LLMs) have opened possibilities for automating content generation tasks [9]. However, applying these technologies to the creation of scientific topic pages introduces unique challenges beyond simple text generation. Scientific content requires high factual accuracy, proper grounding in peer-reviewed sources, and navigation of complex domain-specific terminology [10]. Existing automated article generation methods face limitations when applied to scientific content creation [11]. Many struggle with maintaining factual grounding throughout long-form content, often producing articles with unsupported claims or hallucinated information [12]. Additionally, these methods typically employ simple retrieval strategies that may miss important related concepts or fail to explore a topic with sufficient depth [13].

To address these limitations, this thesis presents Apollo, a multi-agent framework designed for generating comprehensive, Wikipedia-like topic pages from scientific literature. Apollo employs an iterative knowledge curation process that systematically explores topics through structured information gathering, constructs detailed outlines based on discovered relationships, and generates well-referenced content through collaborative agent interactions. The framework addresses key challenges in automated content generation, including information diversity, factual grounding, and content organization.

[1] See example at: https://www.sciencedirect.com/topics/computer-science/machine-learning

In order to assess how well Apollo performs across the different stages of topic page creation, we present SciWiki-2k, a comprehensive benchmark dataset of high-quality Wikipedia articles focused on scientific concepts. This dataset comprises 2,000 articles spanning 20 distinct domains, systematically constructed by selecting the most popular topics from ScienceDirect. To our knowledge, SciWiki-2k is the first dataset specifically designed for evaluating the automation of topic pages across such a broad range of scientific domains.

Using this benchmark dataset, we assess the performance of Apollo and comparable state-of-the-art methods. We begin with traditional automatic metrics to measure similarity to human-written pages through word-overlapping approaches. However, recognizing that lexical metrics cannot capture semantic meaning and content quality [14], we leverage large language models as evaluators. These models have shown strong correlation with human judgment and allow us to conduct cost-effective assessment across multiple quality dimensions [15, 16, 17]. Finally, since LLM-based evaluations alone may not capture all complexities of content quality [18], we also conduct validation through human expert evaluations. To this end, we work with subject matter experts (SMEs) to assess the generated topic pages using the same evaluation criteria employed by our LLM judges.

Given this context, our investigation is structured around the following research questions:

RQ1: How do different components of our iterative knowledge curation process and collaborative content generation approach contribute to improvements in automatic metrics and content quality assessments compared to existing baseline methods?

RQ2: How does our collaborative agent system improve factual grounding and citation quality, and which components of our iterative refinement process are essential for maintaining well-referenced and verifiable content?

RQ3: To what extent do automated LLM evaluations align with human expert assessments when both use the same set of rubrics to evaluate the quality of a topic page?

The contributions of our work include: (1) Apollo, a novel multi-agent framework for automatically generating scientific topic pages; (2) SciWiki-2k, a publicly available benchmark dataset for evaluating automated scientific content creation; (3) A comprehensive evaluation of state-of-the-art long-form content generation methods on writing tasks; (4) Two novel evaluation metrics adapting methodologies from fact-checking literature to assess the factual grounding of generated content; (5) An evaluation comparison examining the correlation between expert human judgments and automated assessment metrics across multiple quality dimensions.

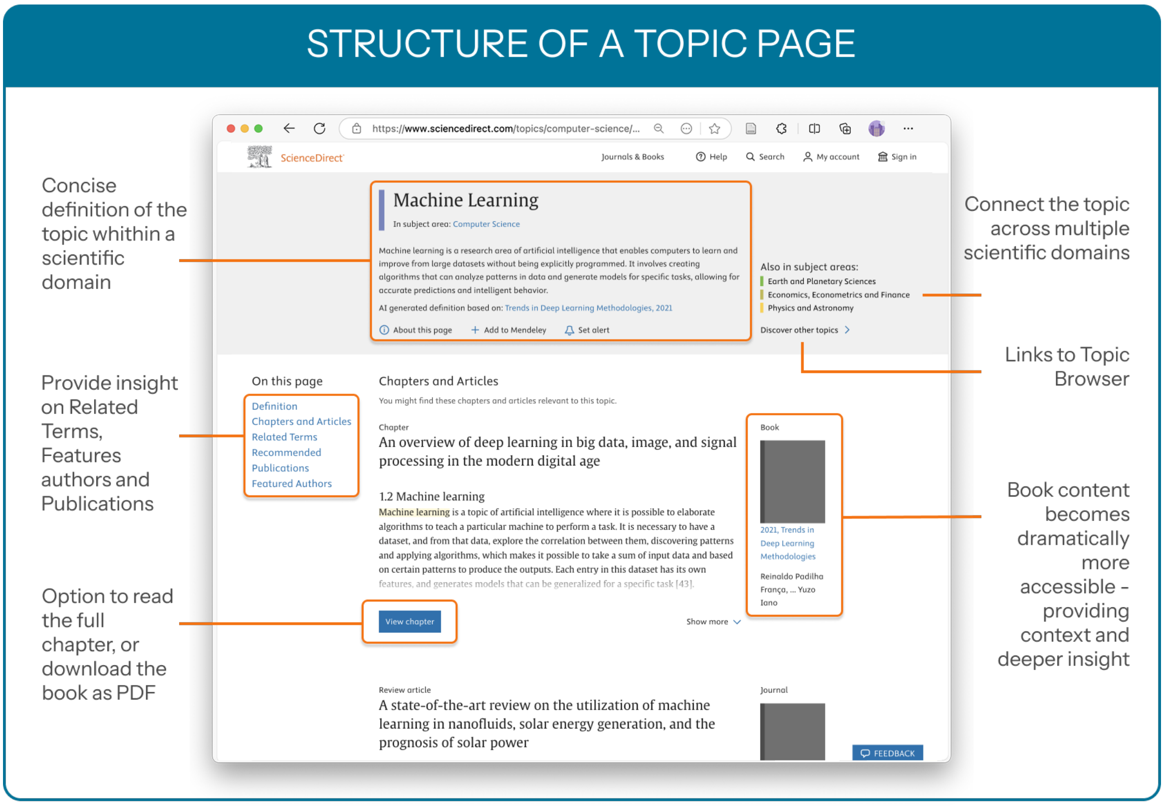
The remaining of this thesis is structured as follow. Chapter 2 provides background information and establishes the theoretical foundation for this research. Chapter 3 describes our Apollo framework, including the iterative knowledge curation process and collaborative content generation methodology. Chapter 4 outlines the experimental setup, evaluation metrics, and baseline comparisons used to assess our approach. Chapter 5 presents and analyses the results across knowledge curation quality, content generation performance, and factual grounding metrics. Finally, Chapter 6 summarizes our findings, discusses limitations, and suggests future research directions.

Chapter 2

Related work

2.1 Topic Pages

Topic Pages represent a specialized knowledge base developed by ScienceDirect to address the challenge of navigating complex scientific terminology and concepts across diverse research domains. Introduced by Elsevier as part of their ScienceDirect platform, Topic Pages serve as curated collections of scientific concepts sourced directly from peer-reviewed scholarly documents [3]. Unlike collaborative platforms such as Wikipedia, Topic Pages maintain scientific rigour by exclusively presenting content from academic sources.


STRUCTURE OF A TOPIC PAGE
Concise definition of the topic whithin a scientific domain
Provide insight on Related Terms, Features authors and Publications
Option to read the full chapter, or download the book as PDF
Connect the topic across multiple scientific domains
Links to Topic Browser
Book content becomes dramatically more accessible - providing context and deeper insight
https://www.sciencedirect.com/topics/computer-science/...
ScienceDirect
Journals & Books
Help
Search
My account
Sign in
About this page
Add to Mendeley
Set alert
Discover other topics
On this page
Definition
Chapters and Articles
Related Terms
Recommended
Publications
Featured Authors
View chapter
Show more
FEEDBACK

Machine Learning

In subject area: Computer Science

Machine learning is a research area of artificial intelligence that enables computers to learn and improve from large datasets without being explicitly programmed. It involves creating algorithms that can analyze patterns in data and generate models for specific tasks, allowing for accurate predictions and intelligent behavior.

AI generated definition based on: Trends in Deep Learning Methodologies, 2021

Also in subject areas:
Earth and Planetary Sciences
Economics, Econometrics and Finance
Physics and Astronomy

Chapters and Articles

You might find these chapters and articles relevant to this topic.

Chapter

An overview of deep learning in big data, image, and signal processing in the modern digital age

1.2 Machine learning

Machine learning is a topic of artificial intelligence where it is possible to elaborate algorithms to teach a particular machine to perform a task. It is necessary to have a dataset, and from that data, explore the correlation between them, discovering patterns and applying algorithms, which makes it possible to take a sum of input data and based on certain patterns to produce the outputs. Each entry in this dataset has its own features, and generates models that can be generalized for a specific task [43].

Book

2021, Trends in Deep Learning Methodologies

Reinaldo Padilha França, ... Yuzo Iano

Review article

A state-of-the-art review on the utilization of machine learning in nanofluids, solar energy generation, and the prognosis of solar power

Journal

Figure 2.1: Topic Page on Machine Learning

As illustrated in Figure 2.1, the current structure of Topic Pages consists of three primary components: a concise definition providing a brief yet comprehensive description of the scientific concept, a collection of up to ten text snippets extracted from relevant books and articles published within the ScienceDirect database, and a set of related topics to facilitate knowledge exploration and discovery. The definition component is generated through a combination of BERT-based sentence classification and Retrieval Augmented Generation (RAG) pipelines, while the snippet ranking employs fine-tuned dense retrieval models optimized for scientific content using Generative Pseudo Labeling techniques [3]. These components are designed to provide researchers with immediate access to crucial information, particularly when encountering unfamiliar concepts.

However, while Topic Pages successfully present reliable sources and maintain scientific credibility, their current format primarily offers fragmented information consisting of isolated snippets without cohesive synthesis. Recent advances in generative artificial intelligence [9] present an opportunity to automatically transform these curated scientific sources into coherent, Wikipedia-style topic pages.

2.2 Existing multi-agent systems

The emergence of large language models has transformed how we approach complex writing tasks [19, 20], particularly in expository writing where the goal is to present factual information in an organized, neutral manner. This includes generating Wikipedia-like articles, literature reviews, and scientific summaries.

Traditional approaches using single LLMs face fundamental limitations when generating long-form content [21]. Context window constraints force systems to process information in chunks, often breaking semantic continuity across sections [22]. More critically, without proper grounding mechanisms, LLMs tend to hallucinate or rely on their internal knowledge rather than retrieved sources, leading to unsupported claims in the final output [23].

To address these challenges, researchers have developed multi-agent frameworks that decompose the writing process into specialized roles [24]. These systems mirror human writing workflows, which typically involve distinct phases of research, planning, drafting, and revision [25]. By assigning specific tasks to different agents, these frameworks can maintain better quality control while leveraging the strengths of each component [26, 27].

Below we present three foundational methods that relate directly to the generation of long-form content:

1. **Outline-driven RAG (oRAG)** [11]: a simple multi-agent approach which follows a straightforward two-stage process. First, the system generates a structured outline using an LLM based on the input topic. Second, it processes each section independently by retrieving relevant information specific to that section's heading. Lastly the retrieved content is summarized into a section. While computationally efficient, oRAG suffers from limited exploration during the research phase. The system only performs retrieval based on the initial topic and section headings, missing potentially valuable related concepts that could emerge through more iterative search processes.

2. **STORM** [11]: a more sophisticated approach based on simulated conversations. The system first identifies different perspectives on a topic by analysing related Wikipedia articles, then assigns these perspectives to different LLM agents. These perspective-guided agents engage in simulated conversations where one agent asks questions while another provides answers grounded in retrieved web sources.

   The key innovation in STORM lies in its perspective-driven research methodology. For example, when researching ”the 2022 Winter Olympics opening ceremony,” an event planner perspective might ask about transportation arrangements and budgets, while a cultural analyst might focus on symbolic elements and artistic choices. This multi-perspective approach helps STORM discover more diverse information compared to generic question-asking strategies. However, STORM faces significant challenges with factual grounding.

The system's conversation-based approach can lead to high hallucination rates, as agents may generate round of conversation where different perspective may be unaware of what it has been already discussed thereby looping into conversation with no extra new retrieved information.

3. **OmniThink** [28]: a recent multi-agent workflow which employs a hierarchical tree-based exploration strategy. The system introduces two core components: an Information Tree that organizes retrieved information hierarchically, and a Conceptual Pool that maintains extracted insights and guides further exploration. The system starts with a root topic, retrieves initial information, and extracts key concepts into a conceptual pool. It then iteratively expands the information tree by generating targeted queries based on the explored branched of the tree. This process continues until sufficient information is gathered. The strength of OmniThink lies in its systematic approach to knowledge expansion. Unlike STORM's conversation-based method, OmniThink uses a structured tree exploration to ensure comprehensive coverage. However, while this method introduces a novel system, the exploration of the tree may lead to explore already covered areas. This happens because the strategy used by this framework relies on independent branch exploration. In concrete terms, it falls back to the STORM problem were different perspective are unaware of what has been discussed by the other agents. In this case, this happens due to the unawareness between isolated nodes of the tree structure. Similarly, the method does not explore any grounding strategy which makes the generation of content prone to hallucinated content.

Recognizing these limitations, our proposed method described in the following sections utilizes a novel approach which focuses on a mechanism to iterative explore a scientific topic while ensuring the generation of the content remains factual.

2.3 Knowledge Graph related literature

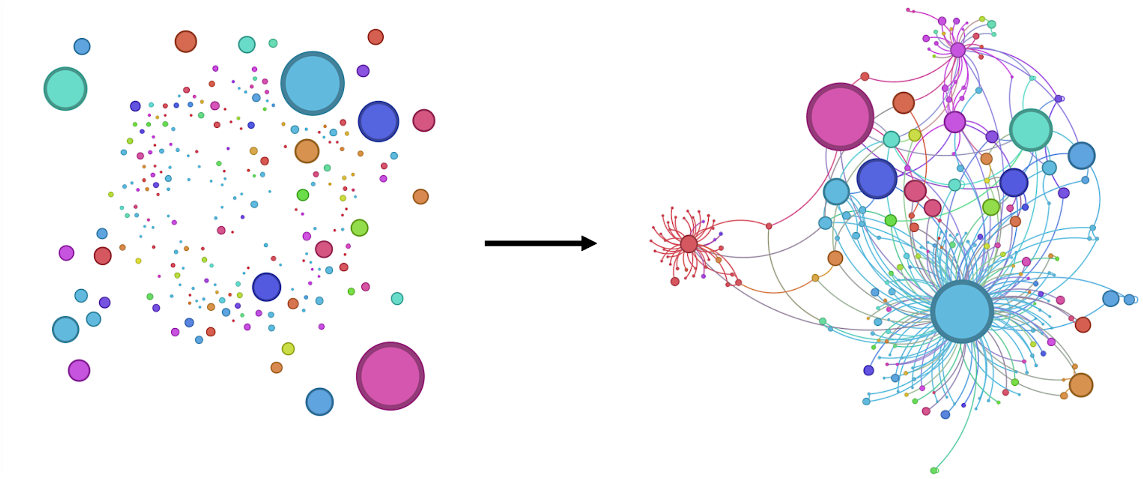


Figure 2.2: From unorganized information to KG [29]

Knowledge graphs (KGs) have emerged as an essential component in enhancing the performance of LLM-based content generation tasks [30, 31, 32]. Recent studies emphasize that presenting information to LLMs in structured relational triples can significantly improve their ability to retrieve and reason with factual information compared to traditional prose formats

[33, 34]. This structured representation allows models to focus clearly on precise factual relationships, reducing the likelihood of hallucinations and improving the reliability of generated content [35, 36].

Graph-enhanced retrieval mechanisms, such as GraphRAG [37], leverage KGs to address the limitations inherent to traditional retrieval-augmented generation (RAG) methods, particularly for complex multi-hop queries requiring detailed reasoning and fact chaining [32, 38]. Studies indicate that KGs are especially beneficial for question-answering and explanatory tasks, as they facilitate the precise retrieval of relevant facts and structured reasoning paths, significantly enhancing the depth and accuracy of generated answers [39, 40].

Additionally, recent work explore advanced methods for improving KG construction and utilization, such as employing LLMs to dynamically expand and refine KGs based on retrieved snippets [41, 42]. Such methods highlight the flexible and generalizable nature of LLMs in constructing KGs without dependence on fixed ontologies or fine-tuned extraction models [43].

For knowledge graph-to-text generation, research has identified optimal formats for presenting structured information to LLMs. Studies comparing different serialization approaches found that JSON format produces the best results, allowing LLMs to better understand the factual nature of relationships compared to other structured formats [34]. This finding influences how knowledge graphs should be formatted when used as input for content generation systems. The process of transforming knowledge graphs back into coherent text requires careful consideration of information density and organization. Research has shown that unordered triplet structures facilitate precise retrieval of relevant facts, making them valuable for outline generation and content planning [33]. However, for final content generation, the structured information often needs to be combined with contextual details from original sources to produce natural, engaging text.

Building upon these findings, we hypothesise that the benefit of using knowledge graphs in automated scientific writing extends beyond simple fact retrieval methods. Specifically, we propose that by constructing knowledge graphs from retrieved scientific documents, we can identify conceptual gaps and relationships that guide iterative information gathering, ultimately leading to more comprehensive and well-structured articles. This is the core mechanism of our proposed method, details of which are explained in Section 3.3.

2.4 LLMs Evaluations

Evaluating the quality of automatically generated content presents significant challenges, particularly for long-form articles where traditional metrics like BLEU or ROUGE capture only surface-level similarities [14]. Human evaluation, while considered the gold standard, is expensive, time-consuming, and difficult to scale for comprehensive assessments across multiple quality dimensions [44]. This has led researchers to explore large language models as automated evaluators.

Unlike traditional metrics that rely on lexical overlap, LLM evaluators can assess semantic meaning, coherence, factual accuracy, and stylistic appropriateness. Research has demonstrated that well-designed LLM judges can achieve strong correlations with human evaluators across various tasks [15, 16, 17]. This capability is particularly valuable for evaluating scientific writing, where quality depends not just on fluency but on factors like factual accuracy, appropriate use of technical terminology, logical organization, and proper citation practices.

The development of specialized evaluation models has advanced significantly beyond general-purpose LLMs used for evaluation. Prometheus [45, 46] represents a notable example of this specialization, being specifically fine-tuned for text quality assessment the model is able to achieve assessments that closely match those of human evaluators. More recently, work has shown that different models excel in different evaluation contexts. Research comparing various LLM judges found that GPT-4 Turbo outperforms Prometheus in certain scenarios [47], while specialized models like FLASK [48] offer advantages in terms of cost-effectiveness and accessibility for researchers with limited computational resources.

Research on long-form document evaluation has led to the development of specialized frameworks like LR-1 [49], which focuses specifically on assessing whether generated outputs correctly address complex questions based on reference materials. For scientific writing evaluation, researchers have adapted methodologies from fact-checking literature to assess factual grounding more systematically. Approaches like atomic claim verification [50] and coverage assessment [51] provide more granular evaluation of how well generated content is supported by source materials.

Given these capabilities, LLM evaluators exhibit critical limitations that researchers must address. Self-preference bias represents a major concern, where models disproportionately favor content generated by themselves or related model families over external or human-written content [52, 15, 53]. Position bias also affects evaluation quality, as the order in which content is presented can substantially influence judgments [37]. Furthermore, despite high correlation in many aspects, LLMs occasionally fail to capture subtle qualitative dimensions such as redundancy or nuanced interpretability that human evaluators discern more effectively [53].

Addressing these challenges, we incorporate LLM-as-judge evaluations in our systematic approach while validating these assessments through human expert evaluations. The details of this evaluation methodology are explained in Section 4.5.

Chapter 3

Method

3.1 SciWiki-2k Dataset

To evaluate the performance of our proposed topic generation framework, we developed SciWiki-2k, a comprehensive dataset of high-quality Wikipedia articles focused on scientific concepts. We constructed this dataset by identifying top trends on the ScienceDirect website [4] and systematically selected the 50 most popular topics. To ensure diversity and further broaden the scope of our dataset, we added an additional 50 topics from each domain. The final dataset consists of 2,000 Wikipedia articles spanning over 20 domains that cover scientific-related concepts. We make SciWiki-2k publicly available on HuggingFace[1].

3.1.1 Dataset Construction

We curated SciWiki-2k by collecting scientific and high quality English Wikipedia articles. For retrieving scientific articles we follow a 3 step process.

First, we match URLs of the respective Wikipedia articles to the corresponding concept at hand. For instance:

- `Carbon-14` → https://en.wikipedia.org/wiki/Carbon-14.

Second, to ensure that the selected articles meet a high-quality threshold, we leverage the Prediction API of the Objective Revision Evaluation Service (ORES), a machine learning-based assessment tool developed by the Wikimedia Foundation to asses the quality of an article [54]. Using the retrieved URLs, we obtain `predictquality` scores for each article. For instance:

- `Carbon-14` → GA:
  `argmax{B: 0.12, C: 0.09, FA: 0.16, GA: 0.78, Start: 0.012, Stub: 0.004}`

In SciWiki-2k, we only consider articles rated as "B" (B-Class), "GA" (Good Article), or "FA" (Featured Article) and filter out articles with lower ratings, such as stubs or start-class pages, as they often lack depth and require review to address citation issues, potential false claims, and other quality concerns as indicated by the Wikipedia Content Assesment [55]. For a detailed overview of the Wikipedia quality grading scheme, refer to Table 3.1.

Finally, we fetch the selected high-quality articles and parse their contents by extracting the section headings along with the corresponding textual content. Other metadata, such as figures and tables, are omitted during parsing.

[1] https://huggingface.co/SciWiki

Class	Criteria	Reader's Experience	Editing Suggestions
FA	The article has attained **featured article** status by passing an in-depth examination by impartial reviewers.	Professional, outstanding, and thorough; a definitive source for encyclopedic information.	No further content additions should be necessary unless new information becomes available; prose quality improvements are possible.
GA	The article meets all **good article** criteria, reviewed by one or more impartial reviewers.	Useful to nearly all readers, with no obvious problems; approaching professional publication quality.	Some expert editing may help; comparison with an existing featured article may highlight missing content.
B	The article meets all **B-Class** criteria. It is mostly complete but requires further work.	Readers are not left wanting, but content may not be comprehensive for a serious student or researcher.	Some content and style improvements are needed. Supporting materials and compliance with guidelines should be considered.

Table 3.1: Wikipedia Article Quality Ratings

3.1.2 Filtering Stage

After the initial dataset construction, we implement a filtering stage to remove ambiguous or irrelevant topics. This step ensures that each Wikipedia article accurately corresponds to the specific, domain-focused topic presented in Science Direct.

a) **Unrelated Articles**

We implement a verification process to identify and remove instances where Wikipedia URLs linked to ScienceDirect topics lead to unrelated content. First, we retrieve the actual destination URL for each Wikipedia link and check whether it represents a 1:1 match with the original topic. When the retrieved URL does not match the expected topic, we manually review these instances. For example, the topic `Moral philosophy` redirects to the Wikipedia page https://en.wikipedia.org/wiki/Ethics, which was flagged for manual checking and subsequently removed after review. However, other instances such as `Polygonum cuspidatum`, which leads to the Wikipedia page https://en.wikipedia.org/wiki/Reynoutria_japonica, are preserved in our dataset since the topic is also known as `Polygonum cuspidatum` as cited in the Wikipedia page itself.

b) **Interdisciplinary Coverage**

We exclude Wikipedia articles that address a topic broadly across multiple domains. For instance, Science Direct might present the concept `Postmodernism` solely within the context of `Psychology` [56], whereas the corresponding Wikipedia article may cover the same concept across various fields such as philosophy, literature, or social theory as written in the actual Wikipedia article. This discrepancy where Wikipedia articles span multiple domains, makes fair evaluation difficult. Consequently, we exclude such domain-overlapping, ambiguous or completely unrelated articles from the SciWiki-2k dataset. An example of this scenario is shown in Appendix B.

3.2 Scientific Source Collection

To generate high-quality, trustworthy, and scientifically accurate topic pages, our approach relies on retrieving relevant snippets from Elsevier's ScienceDirect corpus. These snippets form the foundational knowledge source from which the multi-agent LLM framework generates synthesized, coherent content.

3.2.1 Preprocessing

Documents in the collection undergo a preprocessing pipeline to enhance their quality, manage context-window limitations, and eliminate redundancy. Specifically, we perform the following operations on all retrieved snippets:

a) **Chunking**

Snippets that exceed the maximum token length of 512 tokens are split into smaller chunks to preserve their semantic integrity. We achieve this by first encoding the text snippets into tokens using the embedding model `snowflake-arctic-embed-m-v2.0`[2]. If a snippet surpasses the 512-token threshold, we split it at sentence boundaries, ensuring no sentence is broken mid-word. Short snippets containing fewer than 20 words are excluded from our dataset, as they typically lack sufficient informational value.

b) **Deduplication**

Due to potential overlaps across retrieved snippets, we remove those snippets where identical content is found. To account for minor textual variations, we identify cases where snippets appear different due to small modifications at the beginning of their text. Specifically, the addition of reference numbers, chapter numbers, or section headings that precede otherwise identical content. For example, a snippet beginning with "3.1 Plant genetics involves..." versus "Plant genetics involves..." contains the same substantive information. These instances are manually reviewed and filtered to prevent redundancy.

3.2.2 Vector Store Creation

Model Name	# dim	BEIR	MIRACL	CLEF (Full)
snowflake-arctic-m-v2.0	768	55.4	55.2	53.9
snowflake-arctic-m	768	54.9	24.9	29.1
me5 base	1024	51.2	48.8	48.1
bge-m3 (BAAI)	1024	48.8	56.8	—
gte (Alibaba)	768	51.1	52.3	53.1

Table 3.2: Comparison of models across various retrieval tasks. Source at [57].

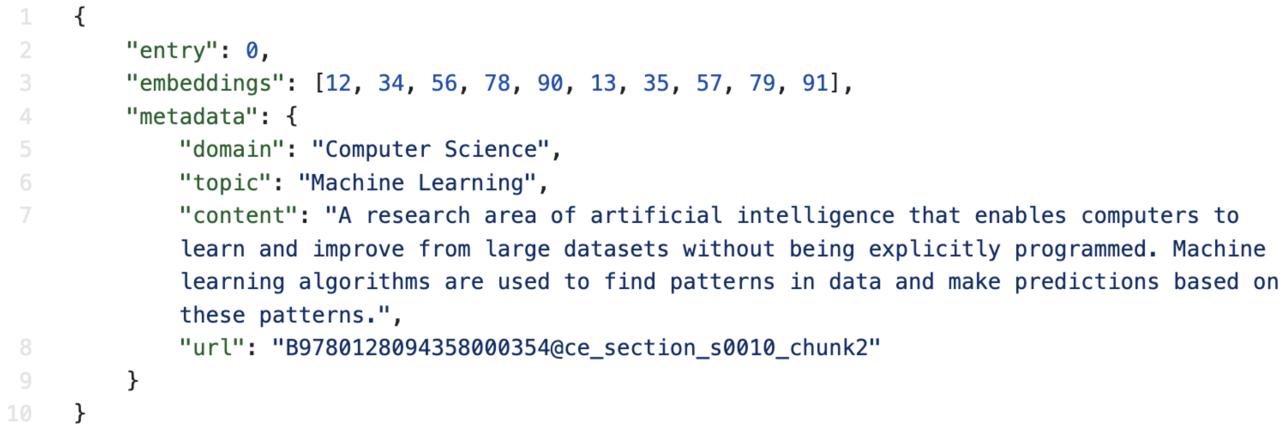
After preprocessing, the cleaned snippets are transformed into high-dimensional vector embeddings and indexed in Qdrant, an open-source vector database built for high-performance vector search with advanced filtering capabilities [58].

Metadata Storage and Filtering

To enhance retrieval efficiency, each snippet stored in Qdrant is enriched with metadata to facilitates structured querying and filtering. As shown in Figure 3.1, the metadata includes:

- **URL Identifiers**: these are internal identifiers that map each snippet to its source article, journal, or chapter. As shown in Figure 3.1, they consist of three components: an internal identifier (e.g., `B9780128094358000354`), a unique code describing the source type (`ce_section_s0010`), and a suffix indicating the chunking strategy explained earlier (`chunk2`). These URL identifiers are used throughout the topic page's content as in-line citations, providing groundedness for later verification purposes (see Section 4.5).

[2] https://huggingface.co/Snowflake/snowflake-arctic-embed-m-v2



```
{
    "entry": 0,
    "embeddings": [12, 34, 56, 78, 90, 13, 35, 57, 79, 91],
    "metadata": {
        "domain": "Computer Science",
        "topic": "Machine Learning",
        "content": "A research area of artificial intelligence that enables computers to learn and improve from large datasets without being explicitly programmed. Machine learning algorithms are used to find patterns in data and make predictions based on these patterns.",
        "url": "B9780128094358000354@ce_section_s0010_chunk2"
    }
}
```

Figure 3.1: Example entry at Qdrant

- **Domains and Topics**: these are the metadata fields that organize the vector store and control the scope of information retrieval. Concretely, each snippet belongs to a specific domain and topic. When generating a topic page, the framework searches only within the relevant domain collection rather than the entire corpus. For instance, when creating content about Machine Learning, the system retrieves information exclusively from the Computer Science domain. The complete list of topics per domain can be found in Tables 1–3.

Figure 3.2 illustrates the complete embedding and storage workflow described above. The diagram shows how preprocessed snippets are processed by the embedding model to generate vector representations after completing the chunking and deduplication operations. These vectors are then stored in Qdrant along with their associated metadata including domain, topic, and URL identifiers.

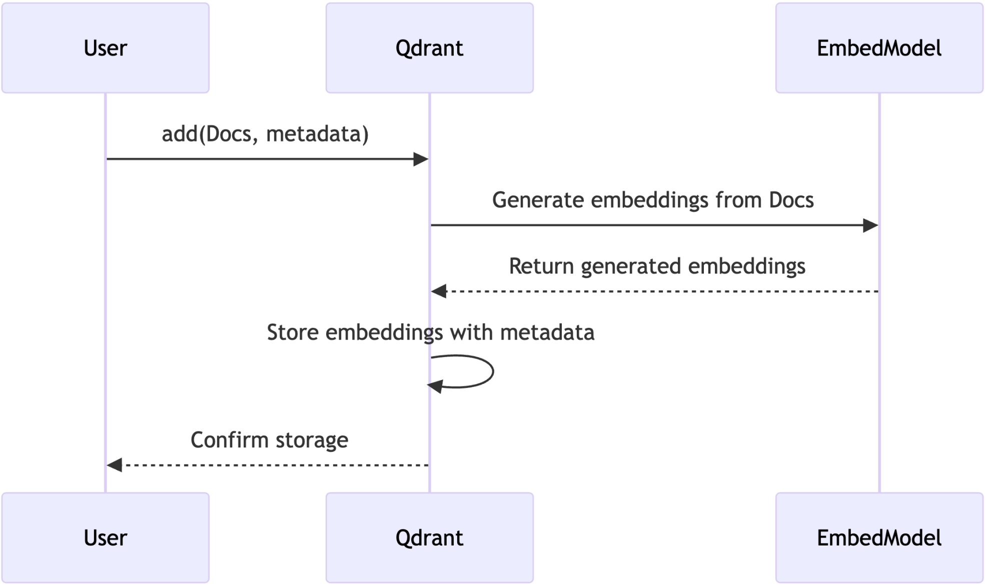

User
Qdrant
EmbedModel
add(Docs, metadata)
Generate embeddings from Docs
Return generated embeddings
Store embeddings with metadata
Confirm storage
User
Qdrant
EmbedModel

Figure 3.2: Sequence diagram for the embedding process

Having constructed this vector database containing embedded snippets with their associated metadata, we now explain how our framework utilizes this knowledge base to retrieve relevant information for generating topic pages.

3.3 Apollo Framework

We present `apollo`, an autonomous LLM-based framework capable of automating the generation of Topic Pages. Apollo emulates the slow-thinking process how humans would write an article e.g. by (i) searching relevant information for the given topic, (ii) organizing this information into a coherent outline and (iii) iteratively refining the sections that compose the article. An overview of the building blocks of our framework is provided in Figure 3.3.

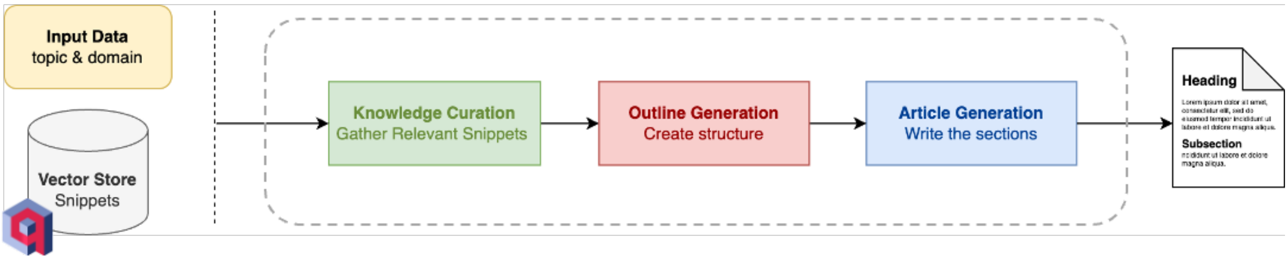


Figure 3.3: Overview of `apollo`, our three-step pipeline for topic-page creation.

3.3.1 Problem Formulation

Let a user supply a topic $T \in \mathcal{T}$ and a domain $D \in \mathcal{D}$, where T can be any of the scientific topics within the 20 academic domains as shown in Tables 1–3. Let the domain-specific vector collection be:

$$\mathcal{C}_D = \{s_1, s_2, \ldots, s_N\},$$

where each snippet s_i is a entry point (see Figure 3.1) defined as:

$$s_i = \left\langle c_i,\ \mathbf{e}_i,\ m_i \right\rangle, \qquad \begin{array}{ll} c_i & : \text{raw text}, \\ \mathbf{e}_i & \in \mathbb{R}^d \text{ (embedding vector)}, \\ m_i & : \langle d_i,\ t_i,\ u_i \rangle \text{ (metadata: domain, topic, URL)}. \end{array}$$

The goal is to generate a coherent topic page $\mathcal{A}$ that explains topic T (within domain D) by drawing evidence exclusively from the domain-specific vector store $\mathcal{C}_D$. Formally, we create a topic page using a three-step process:

(i) **Knowledge Curation.** Retrieve relevant information to construct the topic page

$$\mathcal{I} = \text{Retrieve}(T, \mathcal{C}_D),$$

(ii) **Outline Generation.** Construct an outline conditioned on the retrieved information, topic and domain

$$\mathcal{O} = \text{Construct}(\mathcal{I}, T),$$

(iii) **Article Generation.** Write the full article based on the outline and all the information gathered:

$$\mathcal{A} = \text{Write}(\mathcal{O}, \mathcal{K}),$$

where $\mathcal{K} \subseteq \mathcal{C}_D$ is the *curated knowledge base* obtained by the knowledge-curation module (Section 3.3.2).

The detailed implementation of these components is explained in the following sections and the mathematical notation and symbols used throughout this framework are summarized in Appendix C.

3.3.2 Knowledge Curation

Analogous to a systematic research process [59, 60], `apollo` starts the creation of a topic page by first gathering relevant information for the given topic. While searching information might seem a straightforward process (e.g., querying the vector store with query q and retrieving the top-k most relevant snippets to create the article $\mathcal{A}$), this approach has a fundamental limitation. Such a simple retrieval strategy may miss valuable related information that could be discovered through more exploratory search processes, similar to how researchers iteratively refine their understanding by following citation trails and exploring interconnected concepts [61, 62].

To address this problem, we propose a novel system that explores and expands the current understanding of topic T through an iterative process. Our approach organizes retrieved information into knowledge graphs, which are then used to generate new, focused queries that retrieve additional relevant snippets. Figure 3.4 illustrates this process.

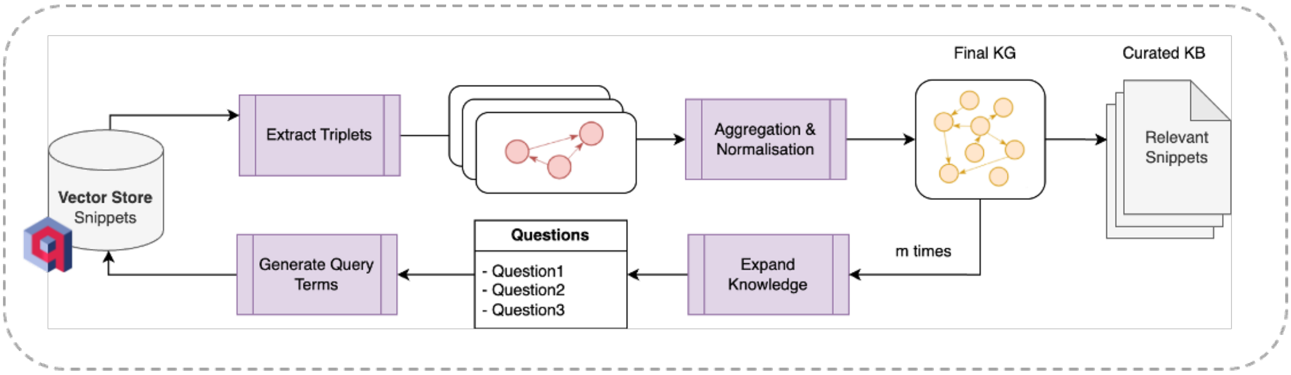


Figure 3.4: Knowledge-curation pipeline. Starting from an initial query, the system extracts relational triplets, merges them into a knowledge graph, generates new query terms from that graph, and repeats the process for m iterations.

Initialization Stage

Step 1: Initial Query Processing. Given topic T, we start the process of gathering relevant information by querying the vector store with $q := T$. We perform retrieval from the domain-specific collection:

$$\mathcal{I}_0 = \text{Retrieve}(T, \mathcal{C}_D), \tag{3.1}$$

where $\mathcal{I}_0 = \{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\} \subseteq \mathcal{C}_D$ represents the top-k most relevant snippets based on cosine similarity between the topic embeddings and the snippet embeddings $\{\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \ldots, \mathbf{e}_{i_k}\}$.

Step 2: Knowledge Graph Construction. For each retrieved snippet $s_i \in \mathcal{I}_0$, we apply an extraction operator

$$\Phi : s_i \longmapsto \Big\{ (h, r, t) \;\Big|\; h, t \in \mathcal{E},\ r \in \mathcal{R} \Big\}, \tag{3.2}$$

where an LLM (Prompt 1) extracts triplets of the form (h, r, t) comprising of a *head* entity h, a *relation* label r, and a *tail* entity t; $\mathcal{E}$ is the universe of entities and $\mathcal{R}$ is the set of relation types. The extracted triplets define a snippet-level sub-graph $G_i = (V_i, E_i)$ where:

- $V_i \subseteq \mathcal{E}$ – entities (vertices) found in s_i,
- $E_i = \{(h, r, t) \mid h, t \in V_i,\ r \in \mathcal{R}\}$ – edges linking those entities.

Essentially, each G_i captures the core conceptual structure of s_i; its vertices and edges serve later to formulate follow-up queries that either (i) deepen our understanding of a concept present in the snippet or (ii) broaden the overall coverage of topic T by exploring related concepts and entities identified in the knowledge graph (see Expansion Stage 3.3.2)

Step 3: Graph Aggregation and Normalization. Having extracted individual subgraphs from each retrieved snippet, we now combine these into a unified knowledge representation. We construct the initial knowledge graph by aggregating all subgraphs:

$$\mathcal{G}_0 = \bigcup_{i=1}^{k} G_i = \left(\bigcup_{i=1}^{k} V_i, \bigcup_{i=1}^{k} E_i \right), \tag{3.3}$$

Because this aggregation may result in duplicate or semantically equivalent entities across different subgraphs, we apply a normalization function η:

$$\mathcal{G}_0^* = \eta(\mathcal{G}_0) \tag{3.4}$$

where η is an LLM-based normaliser that merges semantically equivalent entities and their associated edges (e.g. *"LLM"* and *"large-language model"*).

Expansion Stage

Up to this point we have produced a *shallow* knowledge graph $\mathcal{G}_0^*$ that captures only the information reachable from the initial query $q{:=}T$. In practice, human investigators would now "zoom-in" on promising concepts and "zoom-out" to related areas that have not yet been covered. We emulate this behaviour with two co-operating LLM agents that iterate over the graph, identify knowledge gaps, and issue new query, retrieval cycles. The loop is repeated until a user-defined maximum depth m is reached (we use $m{=}4$ in all experiments).

Agent 1 – *Post-doc Researcher.* Inspired by the "role-playing" strategy of [63], the first agent takes the normalised graph at depth m, $\mathcal{G}_m^*$, and produces a set of focused research questions that would deepen and broaden the current understanding of topic T:

$$\mathbb{Q}_m \;=\; \Psi\Big(\mathcal{G}_m^*,\, \mathcal{M}_Q\Big) \;=\; \{\, (q_j,\ \rho_j)\}_{j=1}^{n}, \tag{3.5}$$

where Ψ is an LLM (Prompt 2) that (i) inspects structural signals in $\mathcal{G}_m^*$, (ii) selects underexplored or high-impact entities/relations, and (iii) formulates $n{=}10$ questions: $n_d{=}5$ "in-depth" questions that target specific underexplored concepts and $n_b{=}5$ "breadth" questions that branch into adjacent areas. Each question q_j is accompanied by a rationale ρ_j that justifies why pursuing this direction could be fruitful. The memory set $\mathcal{M}_Q$ stores all questions asked so far, preventing repetitions across iterations.

Agent 2 – *Reflective Query Synthesiser.* The second agent receives $\mathbb{Q}_m$ and, after reflecting on every (q_j, ρ_j) pair, synthesises a small and diverse list of query terms:

$$\mathbb{L}_m \;=\; \Lambda\Big(\mathbb{Q}_m,\, \mathcal{M}_L\Big) \;=\; \{\ell_1, \ell_2, \ldots, \ell_t\}, \tag{3.6}$$

where Λ is an LLM-based operator (Prompt 3) that (i) decomposes each question into its salient entities, relations, and context, (ii) paraphrases or expands those elements into concrete search strings, and (iii) filters out any term that already appears in the query-memory $\mathcal{M}_L$.

In our implementation we set $t \leq 10$ to balance between depth-oriented and breadth-oriented terms. Similarly, we store all the query terms generated by this agent to prevent repetition.

Retrieval & Graph Update. Using the generated query terms $\mathbb{L}_m$ we perform retrieval operations to obtain new information:

$$\mathcal{I}_{m+1} = \text{Retrieve}\Big(\mathbb{L}_m,\ \mathcal{C}_D\Big) \setminus \Big(\bigcup_{j=0}^{m} \mathcal{I}_j\Big), \tag{3.7}$$

$$\Big\{G_i\Big\}_{s_i \in \mathcal{I}_{m+1}} = \Phi\Big(\mathcal{I}_{m+1}\Big), \tag{3.8}$$

$$\mathcal{G}^*_{m+1} \ = \ \eta\Big(\mathcal{G}^*_m \cup \bigcup_{s_i \in \mathcal{I}_{m+1}} G_i\Big), \tag{3.9}$$

where (1) already-seen snippets are filtered out to guarantee novel evidence, (2) Φ extracts triplets from every new snippet as in (3.2), and (3) η normalises and merges the enlarged graph exactly as described in Section 3.3.2. Both $\mathcal{M}_Q$ and $\mathcal{M}_L$ are updated after each iteration:

$$\mathcal{M}_Q \leftarrow \mathcal{M}_Q \cup \{q_j\}_{j=1}^{n}, \qquad \mathcal{M}_L \leftarrow \mathcal{M}_L \cup \{\ell_j\}_{j=1}^{t}. \tag{3.10}$$

Stopping Criterion. The expansion process continues iteratively until reaching a predefined maximum depth m. The resulting knowledge graph $\mathcal{G}_m = (V_m, E_m)$ captures entities and relations relevant to (T, D), and the union of all retrieved snippets:

$$\mathcal{K} \ = \ \bigcup_{j=0}^{m} \mathcal{I}_j,$$

constitute the curated knowledge base $\mathcal{K}$ that is handed to the outline and article-generation phases (cf. Eq. (iii) in Section 3.3).

Pseudo-code. Algorithm 1 (Appendix D) lists the pseudo-code for the complete expansion procedure, including the interplay between the agents, the memory-aware query generation, batched retrieval, triplet extraction, and incremental graph normalisation.

3.3.3 Outline Generation

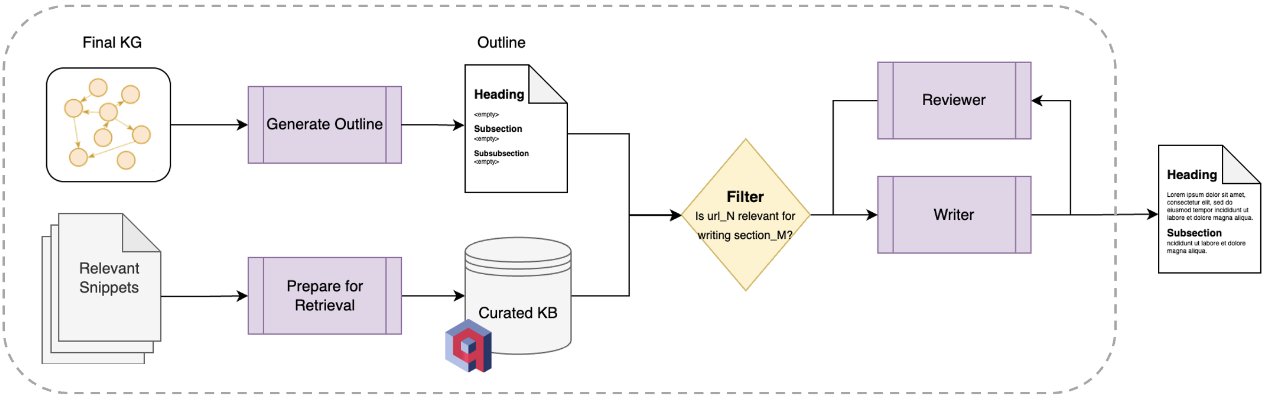

Final KG
Outline
Generate Outline
Heading
<empty>
Subsection
<empty>
Subsubsection
<empty>
Relevant Snippets
Prepare for Retrieval
Curated KB
Filter
Is url_N relevant for writing section_M?
Reviewer
Writer
Heading
Lorem ipsum dolor sit amet, consectetur elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Subsection
ncididunt ut labore et dolore magna aliqua.

Figure 3.5: Outline and article generation pipeline. The final knowledge graph $\mathcal{G}^*_m$ is transformed into a hierarchical outline, while the curated knowledge base $\mathcal{K}$ is prepared for section-specific retrieval. The Writer and Reviewer agents collaborate iteratively to produce factual, well-referenced content for each section.

Having constructed a comprehensive knowledge graph $\mathcal{G}^*_m$ that encodes hierarchical relationships between entities relevant to topic T, we now transform this structured representation into a coherent article outline. The knowledge graph provides a structural foundation through its entity relationships and hierarchical patterns.

Outline Construction. We generate the article outline by applying an LLM-based transformation to the final knowledge graph:

$$\mathcal{O} = \Omega(\mathcal{G}_m^*, T), \tag{3.11}$$

where Ω is an LLM (Prompt 4) that analyzes the graph structure to produce a hierarchical outline $\mathcal{O} = \{h_1, h_2, \ldots, h_p\}$ with p sections. Each section h_i may contain subsections, forming a tree structure that reflects the conceptual organization discovered during knowledge curation.

3.3.4 Article Generation

The article generation phase transforms the outline $\mathcal{O}$ into a comprehensive, factual article $\mathcal{A}$ through an iterative process involving two collaborative agents: a Writer and a Reviewer. For each section $h_i \in \mathcal{O}$, we perform section-specific retrieval, relevance filtering, and iterative content refinement.

Section-Specific Information Retrieval

For each section h_i, we use the section heading as a query to retrieve potentially relevant snippets from our curated knowledge base. Since different sections may retrieve overlapping snippets (e.g., sections on related subtopics might retrieve similar foundational information), we employ a two-step process to ensure precision:

$$\mathcal{R}_i = \text{Retrieve}(h_i, \mathcal{K}), \tag{3.12}$$

where $\mathcal{R}_i$ contains the top-k snippets based on cosine similarity. To ensure each snippet contributes valuable information specifically for section h_i, we apply an LLM-based relevance filter:

$$\mathcal{S}_i = \{s \in \mathcal{R}_i \mid \Theta(s, h_i) = \text{relevant}\}, \tag{3.13}$$

where Θ is an LLM (Prompt 5) that validates whether snippet s provides substantive information for writing about section h_i. This filtered set $\mathcal{S}_i$ forms the supporting evidence for generating section content.

Iterative Content Generation

We employ two specialized agents that collaborate to produce high-quality, well-referenced content. For each section h_i, let $a_i^{(r)}$ denote the article section generated at revision r.

Agent 3 – *Factual Writer.* The Writer agent generates content for section h_i based on the supporting evidence:

$$a_i^{(0)} = \Gamma(h_i, \mathcal{S}_i), \tag{3.14}$$

where Γ is an LLM (Prompt 6) that: (i) synthesizes information from $\mathcal{S}_i$ into well-structured text, (ii) maintains a neutral, factual tone without expressing opinions, (iii) includes inline citations linking claims to their source URLs from the snippet metadata.

For subsequent revisions, the Writer (Prompt 7) revises the content based on feedback:

$$a_i^{(r+1)} = \Gamma^{\text{revise}}(a_i^{(r)}, \mathbb{F}_i^{(r)}, \mathcal{S}_i), \tag{3.15}$$

where $\mathbb{F}_i^{(r)}$ is the structured feedback generated by the Reviewer at revision r.

Agent 4 – *Critical Reviewer.* The Reviewer agent evaluates the generated content and maintains a feedback memory $\mathcal{M}_F$ to track revision items across iterations. The review process is formalized as:

$$\mathbb{F}_i^{(r)} = \pi(a_i^{(r)}, \mathcal{S}_i, \mathcal{M}_F), \tag{3.16}$$

where π is an LLM (Prompt 8) that: (i) verifies that all claims present in the generated section are properly supported by the cited sources, (ii) identifies gaps, unsupported statements, or logical inconsistencies, (iii) provides thorough feedback guiding the Writer agent to fix, add citations, or rewrite content, and (iv) produces a structured feedback list $\mathbb{F}_i^{(r)} = \{f_1, f_2, \ldots, f_q\}$ with actionable revision items.

In the first pass ($r = 0$), the Reviewer generates comprehensive feedback. In subsequent iterations, it consults $\mathcal{M}_F$ to remove addressed items, re-emphasize unresolved issues, or recommend additional citations where needed.

Revision Loop. The revision process continues until either: (i) the Reviewer determines all feedback items have been satisfactorily addressed, i.e., $\mathbb{F}_i^{(r)} = \emptyset$, or (ii) a maximum number of revisions $r_{\max} = 3$ is reached.

Article Assembly. The final article is constructed by concatenating all refined sections while preserving the hierarchical structure from the outline:

$$\mathcal{A} = \bigoplus_{i=1}^{p} a_i^{(r_i^*)}, \tag{3.17}$$

where r_i^* denotes the final revision number for section h_i, and $\bigoplus$ represents the concatenation operator that respects the outline hierarchy. The resulting article $\mathcal{A}$ provides a comprehensive, factual treatment of topic T within domain D, with all claims supported by evidence from the curated knowledge base $\mathcal{K}$.

Pseudo-code. The complete algorithmic implementation of the Apollo framework is detailed in Appendix D, which provides the step-by-step procedure for all three phases described above.

Chapter 4

Experiments

4.1 Dataset

To evaluate the articles generated by our method, we construct SciWiki-100, a benchmark dataset sampled from our larger SciWiki collection. Following prior work [11, 64], we randomly select 5 topics from each of the 20 scientific domains, resulting in 100 diverse topics spanning across all the domains from ScienceDirect. We use this subset to keep evaluation time manageable and low LLM API costs. For each topic in SciWiki-100, we generate Topic Pages using `apollo` and all baseline methods, then evaluate these generated articles against their corresponding human-written Wikipedia pages and a set of automatic metrics as described in our evaluation setup (Section 4.5).

Statistic	**SciWiki-100**	**SciWiki-2k**
Avg. Number of Sections	7.4	7.8
Avg. Number of All-level Headings	20.2	19.9
Avg. Length of a Section (words)	442.3	483.3
Avg. Length of Article (words)	3425.0	3672.7
Avg. Number of References	64.1	71.5

Table 4.1: Comparison of average statistics between the SciWiki-100 and SciWiki-2k datasets.

4.2 Baselines

To assess the effectiveness of our method, we compare against the following approaches:

1. **Outline-driven RAG (oRAG)** [11]: a retrieval-augmented generation baseline that follows a two-stage approach. First, given a topic, oRAG generates using an LLM a structured outline to guide the content generation of the article. Then, it processes each section independently by searching for relevant information specific to that section's title. Lastly, with the retrieved snippets, the article is created in a section-by-section manner, writing each section using the relevant retrieved content.

2. **STORM** [11]: a system that generates Wikipedia-like articles through simulated conversations. Given a topic, STORM first identifies different perspectives and simulates conversations between LLMs assigned with specific perspectives. One agent asks perspective-guided questions while another provides answers grounded on retrieved sources. The information from these conversations is organized into an outline, which then guides the section-by-section generation of the final article.

3. **OmniThink** [28]: a framework that employs a hierarchical tree structure to explore a given topic. The system iteratively builds a tree by identifying sub-topics and collecting relevant information for each branch. The framework then uses the information accumulated throughout the tree structure to generate an outline, which guides the section-by-section writing of the final article as the previous two baselines.

4.3 Models

We use GPT-4o-mini [65] as the primary LLM backbone for all methods in our experiments. This model features a context window of 128K input tokens and a maximum output length of 4,096 tokens [66]. We access this model through Azure API calls using the deployment version `2024-02-15-preview` [67]. For evaluation purposes, we employ the following models:

Claude-3.7-Sonnet [68]: We employ this model from Anthropic's Claude series, which features a 200K token context window with 8,192 maximum output tokens, for outline and article quality evaluation. We access the model through Amazon Bedrock using the model version `claude-3-7-sonnet-20250219-v1:0` [69].

Llama-3.3-70B [70]: We utilize this 70-billion parameter model from Meta, which provides strong performance on complex reasoning tasks, alongside Claude-3.7-Sonnet for knowledge graph evaluation. We access the model through Amazon Bedrock using the model version `llama3-3-70b-instruct-v1:0` [71].

M-Prometheus-7B [72]: A specialized evaluation model fine-tuned from the Qwen2.5-Instruct model [73] for assessing text quality across multiple dimensions. We use this model as our primary LLM-as-judge for article and outline quality evaluation. The model is downloaded from HuggingFace[1] and run locally.

GPT-4o [65]: OpenAI's larger model used for hallucination detection. Since all content is generated by GPT-4o-mini, we use a different model to avoid self-preference bias where LLMs tend to favor their own outputs [52, 15]. This model has a 128K context window and is accessed through Azure API using the model version `2024-12-01-preview` [67].

4.4 Implementation Details

Component	**Configuration**
LLM Backbone	gpt-4o-mini, temperature 1.0, top_p = 0.9
Retrieval	Qdrant (HNSW, $M = 16$, $ef = 128$); cosine similarity over snowflake-arctic-embed-m-v2.0 embeddings
Knowledge-curation loop	Depth $m = 4$; 5 depth-oriented + 5 breadth-oriented questions per depth, max 10 new query terms
Writer/Reviewer loop	Max 3 revisions; reviewer memory resets per section
Hardware	AWS `g5.2xlarge` instance (24GiB GPU, 8 vCPUs)

Table 4.2: Apollo implementation configuration details.

[1] https://huggingface.co/Unbabel/M-Prometheus-7B

Table 4.2 summarizes the key configuration parameters for the framework implementation. Given the need for extensive experimentation, we selected GPT-4o-mini as our backbone LLM due to its low cost per API call [74]. Following the configuration used in previous baselines [11], we set the temperature to 1.0 to encourage diverse content generation and top-p to 0.9 to control output randomness.

For retrieval, we employed Qdrant's HNSW index with $M = 16$ and $ef = 128$ to retrieve top-k snippets. We set $k = 3$ unless otherwise specified. We empirically tested various ef values (64, 128, 256, 512) and found that the selected parameters with scalar quantization maintained 98.67% recall accuracy while improving retrieval speed by 12% [75]. For embeddings, we selected snowflake-arctic-embed-m-v2.0 based on their performance across retrieval benchmarks: 55.4 BEIR, 55.2 MIRACL, and 53.9 CLEF (Table 3.2).

The knowledge curation process was limited to $m = 4$ iterations to ensure comparable retrieved snippets to the baseline methods. Specifically, we have 5 initial retrieved snippets then followed by 4 iterations of the expansion stage described in Section 3.3.2, which in total adds up to a maximum of 125 retrieved snippets ($5 + 10 \times 3 \times 4$). Following recent work [76, 77], we configured 5 depth-oriented and 5 breadth-oriented questions per iteration. The reviewer's memory resets per section to maintain independent content evaluation, and the reviewer-writer feedback loop runs for 3 steps.

4.5 Evaluation Setup

As is shown in Figure 3.3, composing a coherent Topic Page involves a multistage process. To evaluate how each aspect of our framework contributes to the generation of a complete Topic Page, we examine the following elements and describe the corresponding metrics in Section 4.6:

Knowledge Curation Quality: We assess the effectiveness of our knowledge curation module by measuring information diversity, number of unique sources retrieved, and the quality of the constructed knowledge graph.

Outline Quality: In accordance to the evaluation criteria utilized by our baseline methods we evaluate the quality of the outline generated through automatic metrics (soft recall, entity recall) and LLM-as-a-judge assessments [72] across different dimensions (content guidance, hierarchical clarity, and logical coherence). Additionally, we further evaluate the quality of the generated outline using human evaluators using the same metrics as the LLM-judges.

Article Quality: We employ both automatic metrics (ROUGE scores, entity recall) [78, 79] and an LLM-as-a-judge across different dimensions (interest, organization, relevance, depth) to evaluate the generated articles against human-written articles found in SciWiki-100. Human evaluators also assess article quality using the same dimensions employed by our LLM-judges.

Citation Quality: We evaluate whether the content generated by our method remains grounded in the provided scientific snippets. This assessment involves measuring factual accuracy by examining which claims are supported by in-line citations (hallucination rate) and determining how many sections present in the outline are covered by factual written content (coverage). Human evaluators further validate the factuality of in-line citations using identical metrics to our automated evaluations.

The details of the evaluation metrics discussed in this section are described below.

4.6 Metrics

Information Diversity. Let $\mathcal{I} = \{s_1, \dots, s_n\}$ be the set of snippets retrieved during knowledge curation and let $\mathbf{e}_i$ be the `snowflake-arctic-embed-m-v2.0` embedding of snippet s_i. We compute:

$$\mathrm{Div}(\mathcal{I}) \;=\; 1 - \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \cos\big(\mathbf{e}_i, \mathbf{e}_j\big), \qquad n = |\mathcal{I}|.$$

where $\cos(\mathbf{e}_i, \mathbf{e}_j)$ denotes the cosine similarity between embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$. The score lies in the interval $[0, 1]$ for normalized text embeddings. A higher value indicates the retrieved snippets are more diverse, suggesting the system explored the topic from a wider range of perspectives; a lower value signals redundancy and limited exploration.

Knowledge Graph Quality. To evaluate whether the extracted knowledge graphs effectively capture information from retrieved snippets, we adopt the Measure of Information in Nodes and Edges (MINE) benchmark[2]. For every document in the MINE benchmark, we extract knowledge graphs using KG-Gen [63], LightRAG [80], and our method, then query each graph with the benchmark's reference facts.

Given a query fact f, we compute semantic similarity between f and all graph nodes using the same embedding model described in Section 4.4, retrieve the top-$k = 10$ most similar nodes, and collect all relationship triples involving these nodes and their neighbors within 2-hops. Following the benchmark protocol, we use LLM evaluation to determine whether f can be inferred from the collected triples, as this requires semantic reasoning beyond exact string matching. To ensure reliability, we employ three independent evaluators (Claude-3.7-Sonnet, Llama-3.3-70B-Instruct, GPT-4o-mini) and report results in Figure 5.2.

Soft Recall. To assess outline quality, we compare generated section headings against reference headings from SciWiki-100 articles. Given reference headings $\mathcal{H}_{\text{ref}} = \{h_1, \dots, h_m\}$ and generated headings $\mathcal{H}_{\text{gen}} = \{h'_1, \dots, h'_n\}$, we compute:

$$\mathrm{SoftRecall} = \frac{1}{m} \sum_{i=1}^{m} \max_{j \in \{1, \dots, n\}} \cos(\mathbf{e}_{h_i}, \mathbf{e}_{h'_j}),$$

where $\mathbf{e}_h$ denotes the Sentence-BERT embedding of heading h.

Entity Recall. We measure entity coverage at both outline and article levels. For outlines, let $\mathcal{E}_{\text{ref}}$ be the set of named entities extracted from reference headings using FLAIR NER [81], and $\mathcal{E}_{\text{gen}}$ be the entities from generated headings. For articles, these sets represent entities from the full reference and generated content respectively. Entity recall is computed as:

$$\mathrm{EntityRecall} = \frac{|\mathcal{E}_{\text{ref}} \cap \mathcal{E}_{\text{gen}}|}{|\mathcal{E}_{\text{ref}}|}.$$

ROUGE Scores. We evaluate article quality using ROUGE-1 (R_1) and ROUGE-L (R_L) F1 scores [78] against SciWiki-100 reference articles. R_1 measures unigram overlap, while R_L captures the longest common subsequence between generated and reference content.

[2] https://github.com/stair-lab/kg-gen

AB Preference Test. To evaluate outline quality through direct comparison, we conduct pairwise preference tests between Apollo and the best baseline method. For each topic, two outlines are presented to three evaluator models (Llama-3.3-70B-Instruct, GPT-4o-mini, Claude-3.7-Sonnet) in randomized order to mitigate position bias. Each evaluator assigns scores (1-5) to both outlines based on clarity, relevance, completeness, and structure, then indicates their preferred outline.

LLM-as-a-Judge Assessment. We employ M-Prometheus-7B [82] to conduct qualitative evaluations for outlines and articles using structured rubrics. For outline assessment, we evaluate three dimensions: Content Guidance, Hierarchical Clarity, and Logical Coherence. For article assessment, we evaluate four dimensions: Interest, Organization, Relevance, and Depth. Each criterion uses a 5-point scale (1-5) where each score level has specific descriptions that guide the model's evaluation. For details on what each criterion measures, see Appendix F.

Hallucination Assessment. Following FActScore's approach [50], we evaluate factual accuracy through atomic claim verification. For each article, we extract atomic claims using an LLM. Each claim is then verified against the snippet content s_i corresponding to the article's in-line citations through entailment checking. The hallucination rate represents the proportion of unverified claims:

$$\text{Hallucination} = 1 - \frac{|C_v|}{|C|}$$

where C_v represents verified claims and C the total number of claims.

Coverage Assessment. Following ICAT's coverage methodology [51], we assess whether content across article sections is grounded in retrieved snippets. We measure the proportion of sections containing at least one verified claim, where sections without any verified claims indicate the LLM generated content without evidence:

$$\text{Coverage} = \frac{|S_v|}{|S|}$$

where S_v represents sections with at least one verified claim and S the total number of sections.

Human Assessment. Additionally to further compare the results obtained by the qualitative assessments from our LLM-judges we carry out an study to see how humans would grade the generated topic pages using the same set of evaluation rubrics as shown in Appendix F. To this end, we contacted the Data Discovery and Enrichment department at Elsevier to help us grade the generated topic pages by our framework and the second best scoring baseline method. To guide the Knowledge Representation Specialists, or named herein Subject human experts (SMEs) how to grade our topic pages, we prepare the platform Genex (Appendix G) which contains the the necessary information to follow the evaluation. Specifically, we ask each evaluator to read the topic page generated by each method and provide detailed feedback on what they observed while grading the topic page. The comment box interface, as shown in the evaluation platform (Figure 3–4), is then used to analyse the alignment between the LLM-judges and human evaluators. The results of this assessment are discussed in Section 5.5

Chapter 5

Results and Analysis

In this Chapter, we evaluate the performance of our proposed framework against the baseline methods across multiple dimensions to assess the quality of the generated topic pages. We first evaluate the effectiveness of our knowledge curation approach in Section 5.1, examining information diversity, unique source coverage, and knowledge graph quality. Subsequently, we analyse the quality of the generated outlines in Section 5.2, evaluating both automatic metrics (soft recall, entity recall) and qualitative assessments through LLM-as-judge evaluations. We then present a comprehensive evaluation of article quality in Section 5.3, comparing Apollo's generated content against human-written Wikipedia articles using ROUGE scores, entity coverage, and multi-dimensional quality assessments across interest, organization, relevance, and depth. Furthermore, Section 5.4 examines citation quality and factual grounding, analyzing hallucination rate and content coverage to determine how well our framework maintains proper source attribution. Finally, Section 5.5 presents human evaluation results where subject matter experts assess the generated articles using the same evaluation rubrics as the LLM-as-judge assessments shown in Appendix F.

5.1 Knowledge Curation

Knowledge curation forms the first step of our topic page generation pipeline. Here, the different frameworks outlined in Section 4.2 gather diverse and relevant information which is later used to write the different sections of the article. The performance of this phase directly influences the quality of all subsequent components, as insufficient or redundant information retrieval can lead to articles with limited depth and interest (Table 5.5) and poor factual grounding (Table 5.6). We are therefore interested in comparing our iterative graph-based expansion method against the baseline approaches.

Method	APOLLO	OmniThink	STORM	oRAG
Num Unique URLs ↑	105.712	83.27	60.12	45.45
Info Diversity (%) ↑	60.81	54.74	42.23	33.02

Table 5.1: Average number of unique URLs retrieved by each method.

Table 5.1 presents a quantitative comparison of knowledge curation performance across all methods. We can see from the table that Apollo consistently scores highly across both metrics, retrieving an average of 105.71 unique URLs compared to the best baseline (OmniThink at 83.27) and achieving the highest information diversity score of 60.81% compared to 54.74% for OmniThink.

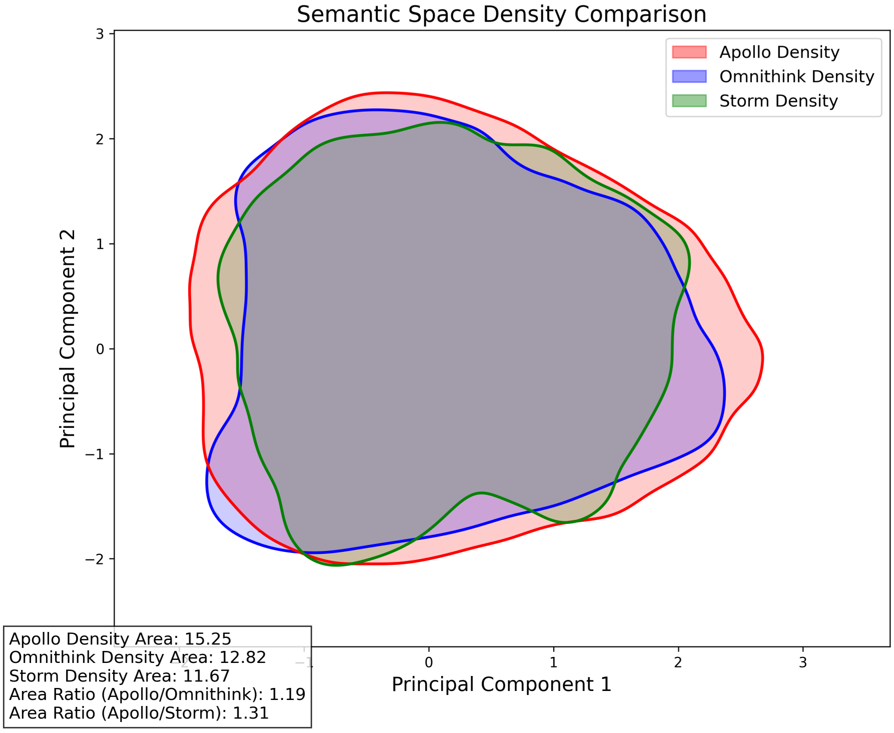


Figure 5.1: Information Diversity shown in a 2-dimensional PCA space where snippets gather from the Knowledge Curation phase are displayed to show the Density Area covered by the different frameworks.

To analyse the semantic variety of retrieved snippets, we computed embeddings for all snippets collected by each framework and projected them into a 2-dimensional PCA space, as shown in Figure 5.1. In this reduced dimensional space, each snippet corresponds to a point where proximity indicates semantic similarity. To visualize the semantic coverage of each method, we computed hulls that encircle the outermost data points for each framework and displayed these as filled regions representing the total semantic area covered. As shown by the boundaries of each method, Apollo achieves the largest density area (15.25), compared to OmniThink (12.82) and STORM (11.67). This broader semantic coverage demonstrates that our method explores more diverse areas, increasing the breadth of available knowledge for article creation. This broader semantic coverage is critical for generating high-quality articles. As demonstrated in Tables 5.4–5.5, systems with limited exploration produce articles with reduced quality scores. Additionally, when these methods lack the sufficient retrieved information to support their writing, they tend to rely on the internal LLM's knowledge rather than grounding their content generation in the snippets retrieved. As our experiments show, this results in higher hallucination rates (Figure 5.4).

5.1.1 KG Quality

Central to our iterative expansion method is the knowledge graph construction process. These graphs serve as the structured representation from which we identify information gaps and generate new queries (see Section 3.3.2). If our framework fails to effectively capture relevant information from retrieved snippets into knowledge graphs, subsequent exploration will be limited, resulting in poor snippet diversity (Table 5.1) and reduced article quality (Table 5.5). We therefore evaluate whether our method successfully synthesizes textual information from scientific snippets into meaningful knowledge graph representations. Since our KG construction approach mirrors GraphRAG methods in transforming documents into structured knowledge

[40, 77, 83], we employ the MINE benchmark [84], which specifically measures document-to-KG synthesis quality.

Backbone	**Methods**	**MINE Scores**	
		Normalized	Non-Normalized
`Claude-3.7-Sonnet`	Ours	0.714	0.701
	KG-Gen	**0.725**	0.680
	LightRAG	0.709	**0.705**
`Llama-3.3-70B`	Ours	**0.620**	**0.610**
	KG-Gen	0.580	0.550
	LightRAG	0.535	0.542
`GPT-4o-mini`	Ours	**0.501**	**0.486**
	KG-Gen	0.392	0.388
	LightRAG	0.432	0.428

Table 5.2: Comparison of MINE scores across different LLM backbones and methods. **Bold** indicates the best performance and underlined indicates the second-best performance for each LLM backbone. Gray cells highlight our proposed method.

Table 5.2 and Figure 5.2 present the performance comparison of our method against state-of-the-art approaches across three different LLM backbones [65, 68]. As shown in Table 5.2, our method performs differently depending on the underlying model strength. With Claude-3.7-Sonnet, our approach achieves a MINE score of 0.714 (normalized), placing second behind KG-Gen's 0.725 while surpassing LightRAG's 0.709. The performance pattern shifts considerably with less capable models, where our method consistently outperforms both baselines. Specifically, with Llama-3.3-70B, our approach leads with 0.620, and continues with GPT-4o-mini, where our method achieves 0.501, significantly outperforming both KG-Gen and LightRAG.

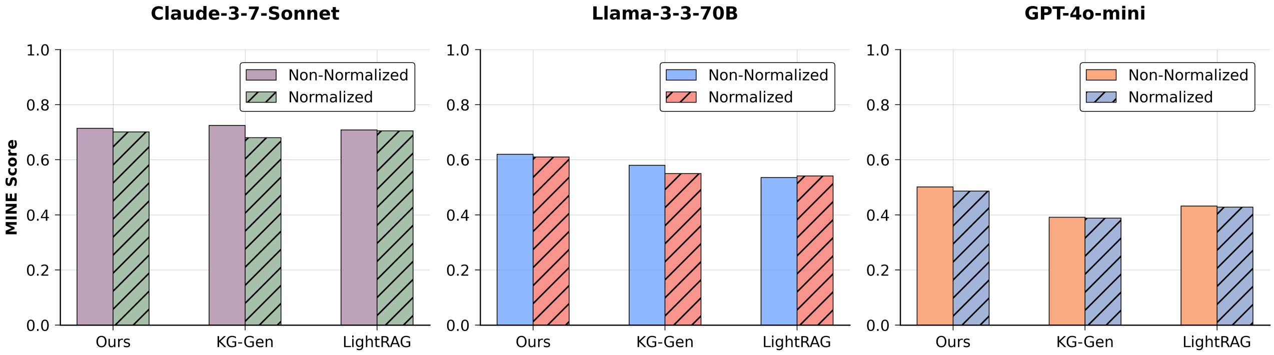


Figure 5.2: Assessment of the quality of the Knowledge Graph generated across different models using the `MINE` Score.

Given the outcomes observed in Figure 5.2, we can see that stronger models consistently yield better results across all methods. Since our retrieved snippets lack ground truth annotations, we use the MINE benchmark on scientific articles to validate our knowledge graph quality. The strong performance on MINE indicates that our method can effectively extract and organize information from documents. This improved information extraction translates to broader source coverage, as evidenced by the higher unique-URL counts and diversity scores in Table 5.1.

5.2 Outline Quality

We now turn our attention to analyse how the outlines generated by our framework compare against baseline methods. The outline serves as the structural blueprint that guides the writing process [85], organizing the curated knowledge into a logical hierarchy of sections and subsections. A well-structured outline is essential for producing coherent articles, as it determines how information is synthesized throughout the document [86, 87]. We therefore evaluate outline quality using both automatic metrics and qualitative LLM-as-a-Judge assessments.

Backbone	**Methods**	**Automatic Metrics**		**LLM-as-Judge**		
		Soft Recall	Entity Recall	Content Guidance	Hierarchical Clarity	Logical Coherence
`GPT-4o-mini`	oRAG	86.42	37.44	3.22	3.97	3.86
	STORM	87.63	37.10	3.39	3.95	3.87
	OmniThink	88.31	37.74	4.03	3.99	3.98
	APOLLO	**91.82**†	**38.52**	**4.16**†	**4.00**	**4.01**†
	w/o Reflection	80.75	36.14	3.36	3.93	3.82

Table 5.3: Outline quality comparison between different frameworks. Soft Recall and Entity Recall are measured against `SciWiki-100` outlines. The LLM-as-judge used to run these evaluations is `M-Prometheus-7B`. † denotes significant improvements ($p < 0.05$) from paired t-tests.

Table 5.3 shows the quantitative results for outline evaluation. Apollo achieves the highest scores in both automatic metrics, with Soft Recall reaching 91.82 compared to OmniThink's 88.31, and Entity Recall of 38.52 versus 37.74. Notably, while Entity Recall shows no statistical significance due to exact word matching, this indicates our generated outlines are not direct copies of Wikipedia content. In contrast, Soft Recall demonstrates statistical significance, indicating that our knowledge graph-driven generated outline shows better semantic alignment with human-written articles as compared to the baseline methods.

For the qualitative assessment (right side of Table 5.3), Apollo outperforms all baselines across all dimensions, with statistically significant improvements in Content Guidance (4.16 vs 4.03) and Logical Coherence (4.01 vs 3.98) as compared to the OmniThink framework. These results show that the structured information encoded in the knowledge graph aids in generating outlines with better hierarchical organization, as is reflected in the favourable evaluation by M-Prometheus-7B across these three dimensions (Appendix F).

To go one step further, we carry out an AB preference test where we provide the outline generated by Apollo and compare against the outlines generated by the second-best baseline: OmniThink. We report the findings of this test in the following bar plot:

Figure 5.3 shows the results from this preference test. Across all evaluator models, Apollo is consistently preferred over Omnithink. Claude-3-7-sonnet demonstrates the strongest preference for Apollo (78.4% win rate), while llama-3-3-70B and gpt-4o-mini show more moderate but still clear preferences (64% and 64.8% respectively).

Ablation Analysis : To better understand the contribution of our iterative expansion methodology, we conducted an ablation study comparing Apollo against a version without the reflective component (w/o Reflection). As described in Section 3.3.2, Agent 1 generates research questions from the knowledge graph while Agent 2 reflects on these questions, filtering and tailoring them for the retrieval process. Removing this reflective component causes substantial performance drops across all metrics (Table 5.3). Soft Recall declines from 91.82 to 80.75, Entity Recall from 38.52 to 36.14, and LLM-as-judge scores decrease significantly (Content Guidance: 4.16 to 3.36; Logical Coherence: 4.01 to 3.82). These results demonstrate that

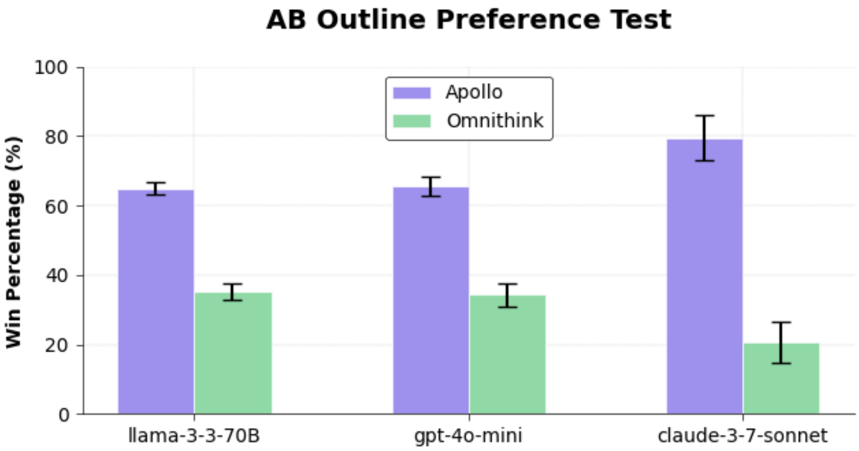


Figure 5.3: AB Preference test between best performing generated outlines among all evaluator models prefer `Apollo`, with `claude-3-7-sonnet` showing strongest preference (79.5% win rate). Error bars show standard deviation across 5 runs.

Agent 2's reflection mechanism is crucial for preventing redundant exploration and ensuring each expansion iteration targets new information gaps. As described in Section 3.3.2, Agent 2 employs memory sets to filter out previously explored query terms, which directly translates to the higher information diversity (60.81%) and unique URL counts (105.71) shown in Table 5.1. Since outlines are generated from the final knowledge graph (Section 3.3.3), this more comprehensive and diverse knowledge base enables the creation of outlines with better content coverage and logical structure, as demonstrated by the performance of our method when the reflective agent is active.

5.3 Article Quality

Having observed that Apollo generates well-structured outlines, we now examine the quality of the final articles produced by our framework. Building upon the coherent structural outlines, we evaluate how effectively Apollo transforms the curated knowledge (Section 5.1) into comprehensive articles. This evaluation follows a similar methodology to the outline assessment, employing both automatic metrics that measure lexical similarity to human-written Wikipedia articles and qualitative LLM-based assessments that examine multiple dimensions of article quality.

Backbone	**Method**	**ROUGE ↑**		**Entity ↑**
		R_1	R_L	**Recall**
`GPT-4o-mini`	oRAG	41.84	14.03	5.92
	STORM	<u>42.11</u>	<u>14.44</u>	<u>6.51</u>
	OmniThink	41.76	13.94	5.53
	APOLLO	**52.10**†	**15.81**†	**9.17**†
	w/o Reviewer	49.17	15.51	7.35

Table 5.4: **Automatic Evaluation**. Results of automatic evaluation against human-written articles. † denotes significant differences ($p < 0.05$) from a paired t-test between methods and the best baseline.

Starting from the automatic metrics, Table 5.4 shows statistically significant improvements across all measures when comparing Apollo against the baselines. Apollo achieves a ROUGE-1 score of 52.10, over the best baseline (STORM at 42.11). Similarly, ROUGE-L increases from 14.44 from to 15.81, while Entity Recall shows the most significant gain, improving from 6.51 to 9.17. These improvements suggest that Apollo generates articles with better lexical overlap with human-written Wikipedia articles.

Backbone	**Method**	**LLM-as-Judge**				
		Interest	Organization	Relevance	Depth	Average
GPT-4o-mini	oRAG	<u>2.34</u>	4.32	3.92	3.88	3.62
	STORM	1.61	<u>4.85</u>	<u>4.10</u>	<u>4.54</u>	<u>3.77</u>
	OmniThink	1.37	4.28	4.12	4.27	3.51
	APOLLO	**3.29**†	**4.92**†	**4.90**†	**4.94**†	**4.51**
	w/o Filter	1.99	4.74	4.57	4.77	4.02

Table 5.5: **LLM-as-a-judge Evaluation.** Rubric-based evaluation (1–5 scale) across four methods. Each row reports average scores across five qualitative dimensions. The LLM-as-judge used to run these evaluations is `M-Prometheus-7B`. † denotes significant differences ($p < 0.05$) compared to the best baseline.

Table 5.5 presents the qualitative assessment using M-Prometheus-7B across four key dimensions. Our framework shows consistent superiority across the evaluated criteria, achieving an average score of 4.51 compared to 3.77 for the best baseline (STORM). Most notably, Apollo excels in the Interest dimension (3.29 vs. 2.34 for oRAG), suggesting that the comprehensive knowledge exploration during the curation phase enables the generation of more engaging and diverse content. The high scores in Organization (4.92), Relevance (4.90), and Depth (4.94) indicate that Apollo's knowledge curation and refinement process produces well-organized articles that stay on topic.

Ablation Analysis : To assess the contribution of our section-specific relevance filtering mechanism, we conducted an ablation study comparing Apollo against a version without the relevance filter (w/o Filter). As described in Section 3.3.4, our approach applies an LLM-based filter Θ to ensure each retrieved snippet provides substantive information for the specific section being written. Removing this filtering mechanism leads to notable performance degradation across both automatic and qualitative metrics (Tables 5.4–5.5). The most significant impact appears in Interest scores (3.29 to 1.99), where unfiltered retrieval populates sections with generic snippets that fail to provide the "compelling narrative" and "noteworthy points" required for high engagement (Score 4; Appendix F). Relevance also suffers (4.90 to 4.57) as sections incorporate loosely related snippets that do not "contribute to a comprehensive understanding of the topic" (Score 5) for the specific context. Similarly, Depth scores decline (4.94 to 4.77) because irrelevant snippets displace substantive, section-specific information needed for "broad coverage" (Score 4). These findings confirm that the filtering step is essential for producing high-quality articles as demonstrated by the performance of our method when the filtering mechanism is active.

5.4 Citation Quality

Following up with citation quality, we analyse the percentage of claims that are properly supported by inline citations (hallucination rate), and the percentage of sections containing at least one verified claim (coverage). Table 5.6 presents the comparative results across all frameworks. The results demonstrate positive outcomes with Apollo outperforming the baselines on both metrics. Specifically, we achieve a hallucination rate of 5.70% compared to STORM's 34.34%, indicating that most claims in Apollo's generated sections are properly grounded by the snippets cited. Furthermore, analysing coverage, we reach 89.05% compared to STORM's 60.67%, demonstrating that nearly all sections written by our framework contain verifiable information with appropriate citations.

Backbone	**Method**	**References**	
		Hallucination ↓	Coverage ↑
`GPT-4o-mini`	oRAG	53.30	44.50
	STORM	34.34	60.67
	OmniThink	62.29	49.03
	APOLLO	**5.70**†	**89.05**†
	w/o Reviewer	8.63	85.85

Table 5.6: **LLM-as-a-judge Evaluation.** Results from checking whether the generated text per each section remains factual (Hallucination) and supports each section of the article (Coverage). † denotes significant differences ($p < 0.05$) compared to the best baseline.

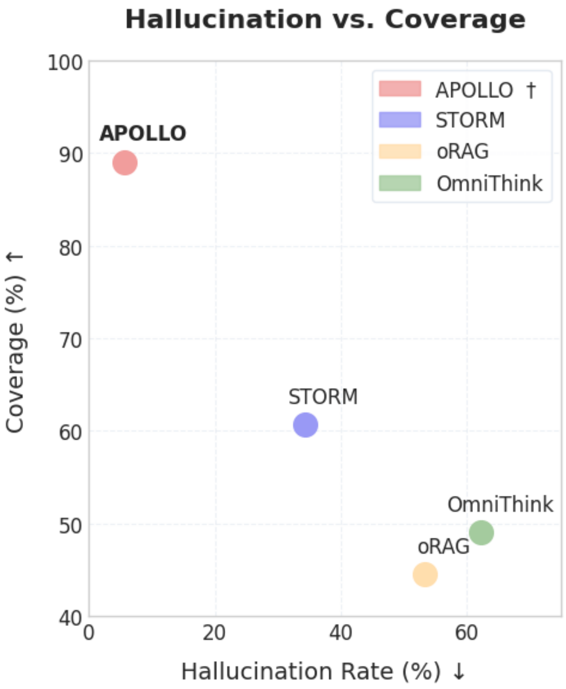


Figure 5.4: **LLM-as-a-judge Evaluation.** Results from checking whether the generated text per each section remains factual (Hallucination) and supports each section of the article (Coverage). † denotes significant differences ($p < 0.05$) compared to the best baseline.

Figure 5.4 visualizes the relationship between both metrics across all evaluated methods. The plot reveals distinct performance across methods: oRAG and OmniThink exhibit higher hallucination rates (62.29% and 53.30% respectively) while achieving lower coverage. STORM shows better hallucination control (34.34%) but with reduced coverage (60.67%). In contrast, our method (upper-left region) shows both low hallucination and high coverage, indicating that our framework achieves better performance on both dimensions relative to the existing approaches.

Ablation Analysis : To better understand the contribution of our iterative refinement process, we conducted an ablation study comparing Apollo against a version without the reviewer component (w/o Reviewer). As described in Section 3.3.4, Agent 4 (Critical Reviewer) evaluates the factuality of generated sections by checking whether claims are properly supported by the cited snippets. When unsupported statements are identified, the reviewer creates structured feedback in the form of bullet points, specifying which claims lack proper grounding and providing actionable guidance on how to address these issues. As shown in Table 5.6, this iterative interaction between the reviewer's feedback and the writer's revisions yields substantial improvements in citation quality. The inclusion of the reviewer agent reduces the hallucination rate by approximately 3 percentage points (from 8.63% to 5.70%) and increases section coverage by over 3 percentage points (from 85.85% to 89.05%).

5.5 Human Evaluation

To complement our automated evaluations, we conduct human assessments across the outline, article, citation and overall quality of a topic pages judged by subject matter experts (SMEs). SMEs are domain specialists with advanced education and professional experience in the scientific fields covered by our evaluation topics. To attain this goal, we randomly select one run from our five experimental runs and sample 20 topics from our dataset to evaluate articles generated by Apollo and STORM, the best-performing baseline according to Table 5.5. Each pair of articles is evaluated by independent SMEs who are presented with both frameworks' outputs. Prior to evaluation, SMEs complete a demographic questionnaire covering their age, education level, and familiarity with the given topic. We present these statistics in Table 5.7. For the evaluation criteria, we ask SMEs to assess articles using the same rubrics employed in our LLM-as-a-Judge evaluations, as detailed in Tables 5–6.

Characteristic	**Category**	**Count**	**%**
Age	Under 25	0	0.0
	25–35	3	15.0
	35–45	14	70.0
	Above 45	3	15.0
Education Level	Bachelor	0	0.0
	Master	17	85.0
	PhD	3	15.0
	Others	0	0.0
Topic Familiarity	Not familiar	4	20.0
	A bit familiar	10	50.0
	Very familiar	6	30.0
Total		**20**	**100.0**

Table 5.7: Demographics and topic familiarity of Subject Matter Experts (SMEs) who evaluated the generated articles. Each pair of articles was evaluated by SMEs across 20 concepts, with evaluators providing demographic information and self-assessed topic familiarity.

Table 5.8 presents the quantitative results from our human evaluation study. The results demonstrate a strong alignment with our automated LLM-as-judge assessments, providing empirical evidence for RQ3 regarding the correlation between automated and human expert evaluations. Apollo achieves superior performance on 8 out of 9 evaluated metrics, with an average advantage of +0.28 points across all dimensions. The largest improvements appear in Content Guidance and Coverage (both +0.55), followed by Depth (+0.45) and Global Assessment (+0.40). Notably, while Apollo demonstrates consistent advantages across most quality dimensions, Storm outperforms in Relevance Focus (+0.25 for Storm), creating an intriguing pattern

Category	Metric	Apollo	Storm	Difference	p-value
Outline Quality	Content Guidance	**3.85**	3.30	**+0.55**	0.10
	Hierarchical Clarity	**3.55**	3.30	**+0.25**	0.40
	Logical Coherence	**3.25**	3.15	**+0.10**	0.40
Article Quality	Interest	**3.50**	3.30	**+0.20**	0.40
	Coverage	**3.90**	3.35	**+0.55**	0.10
	Depth	**3.85**	3.40	**+0.45**	0.10
	Relevance	3.90	**4.15**	-0.25	0.40
	Verifiability	**3.85**	3.55	**+0.30**	0.20
Overall Quality	Global Assessment	**3.90**	3.50	**+0.40**	0.10

Table 5.8: Quantitative evaluation comparing Apollo and Storm across nine quality metrics organized by evaluation category. Each score represents the mean rating from 20 subject matter expert evaluations on a 1-5 scale. Bold values indicate the superior method for each metric. p-values are from two-tailed t-tests comparing score distributions between methods.

that mirrors our automated evaluations. The p-values indicate marginal significance (p = 0.10) for Apollo's strongest advantages in Content Guidance, Coverage, and Depth, suggesting meaningful practical differences despite the limited sample size of n=20 concepts.

5.5.1 Qualitative Insights from Expert Feedback

The quantitative scores are substantiated by rich qualitative feedback from our SMEs, which reveals the underlying reasons for the performance of both methods:

Apollo's Comprehensive Excellence. SME feedback consistently highlighted Apollo's superior organization and comprehensive coverage. Evaluators praised Apollo's structural advantages: *"A's sections are better organized and the flow is more natural"* and *"A is better organized, better logical flow and hierarchy. Goes from basics to advanced topics seamlessly"*. This organizational strength directly correlates with the +0.25 advantage in Hierarchical Clarity observed in Table 5.8. The substantial advantages in Coverage (+0.55) and Depth (+0.45) are explained by SME observations about Apollo's comprehensive approach: *"A covers more range of topics like economic application, environmental etc."* and *"Apollo explored the topic more extensively with relevant and essential coverage"*. Multiple evaluators noted that *"Apollo model provided explicit information, whereas Storm model was very superficial"*, directly supporting the depth advantage demonstrated in our quantitative results.

Superior Factual Grounding. The +0.30 advantage in Verifiability aligns with consistent SME praise for Apollo's citation quality. Evaluators specifically noted: *"All claims are substantiated, unlike Storm model"* and *"The citations were appropriate and sufficient"*. This feedback corroborates our automated citation quality analysis (Table 5.6), where Apollo achieved a 5.70% hallucination rate compared to Storm's 34.34%.

The Relevance Paradox. The most intriguing finding is Storm's advantage in Relevance (-0.25 for Apollo), which initially appears contradictory to Apollo's comprehensive coverage strengths. However, SME feedback revealed a fundamental trade-off between information comprehensiveness and cognitive digestibility. Evaluators who preferred Storm emphasized conciseness: *"less redundancy, more to the point"*, *"Storm was more concise"*, and crucially, *"Apollo provides extensive information, which results in an overcrowded page"*.

This apparent contradiction reflects the inherent tension between providing comprehensive coverage and maintaining focused relevance. Our evaluation rubric defines high relevance as content where *"every piece of information contributes to a comprehensive understanding of the topic"* (Score 5; Appendix F). However, human evaluators may experience cognitive overload [5, 88] when presented with Apollo's extensive information, making them perceive the content as less focused despite the content actually staying on topic.

Importantly, the same SMEs who noted information overload also acknowledged Apollo's superior content quality: *"Even though Apollo article was longer and contained more information, it was all quite relevant"* and *"It provided interesting background information and provided a lot of scientific detail"*. This suggests that Apollo's perceived relevance limitation stems from presentation density rather than off-topic content, supporting the +0.20 advantage in Interest scores where evaluators appreciated the comprehensive yet engaging coverage.

Chapter 6

Conclusion

In this work, we have introduced Apollo, a novel iterative graph-based framework designed for automatic generation of high-quality scientific topic pages. We evaluated each component of the topic page generation pipeline using the SciWiki-100 dataset, comparing Apollo against several baseline methods including STORM, OmniThink, and oRAG.

Through extensive experiments addressing RQ1, we explored how different components of our iterative knowledge curation process and collaborative content generation approach contribute to improvements in automatic metrics and content quality assessments compared to existing baseline methods. Our novel graph-based approach uniquely leverages knowledge graphs to systematically extract and organize relevant information from scientific snippets, subsequently guiding further exploration through collaborative agents. As evidenced by our results (Table 5.1 and Figure 5.1), using this knowledge graph method allowed Apollo to retrieve significantly more unique snippets and achieve greater information diversity, indicated by the broader hull in the semantic embedding space. Furthermore, using the MINE benchmark, we validated that our knowledge graph construction is competitive with state-of-the-art document-to-KG approaches, confirming that our framework effectively captures semantic relationships from scientific content.

Through ablation studies, we showed the critical role of the reflective mechanism in avoiding redundant queries and maximizing semantic coverage. Building upon these findings, we then analyzed how effectively these richer knowledge graphs helped in creating outlines. Our results (Table 5.3) showed clear advantages in outline coherence, logical organization, and semantic alignment, corroborated through an AB preference test demonstrating consistent superiority of Apollo generated outlines over the best-performing baseline across multiple LLM evaluators.

For the second part of RQ1, examining article quality, we found through ablation studies that incorporating a relevance filter was critical. This filter ensures each retrieved snippet provides sufficient depth for specific sections, significantly enhancing article quality, particularly in metrics such as interest, depth, and relevance. Additionally, automatic evaluations using ROUGE and entity recall metrics further affirmed that Apollo generated content closely resembles human-written articles.

In response to RQ2, our evaluations revealed that Apollo's collaborative agent system significantly improves factual grounding and citation quality. Specifically, the critical reviewer agent substantially reduced hallucination rates and enhanced citation coverage (Table 5.6), demonstrating its vital role in maintaining verifiable content. The iterative refinement loop, where generated content undergoes systematic critical feedback is indispensable for reinforcing factual accuracy. Our ablation analysis further confirmed that removing the critical reviewer component significantly impacted content reliability, highlighting the necessity of the iterative critical assessment to uphold high-quality, trustworthy articles.

Finally, addressing RQ3, our human evaluation showed strong alignment with automated

LLM-based evaluations when both applied the same set of rubrics. Human experts rated Apollo consistently higher across key metrics such as interest, depth, and factual verifiability, closely mirroring automated evaluation outcomes (Table 5.8). Although some variations existed due to inherent subjectivity and sample size constraints, the overall agreement suggests automated LLM assessments can serve as valid proxies for human evaluation.

6.1 Limitations & Future Research

While this research has shown positive outcome towards the automation of topic pages, our method could be improved further, namely:

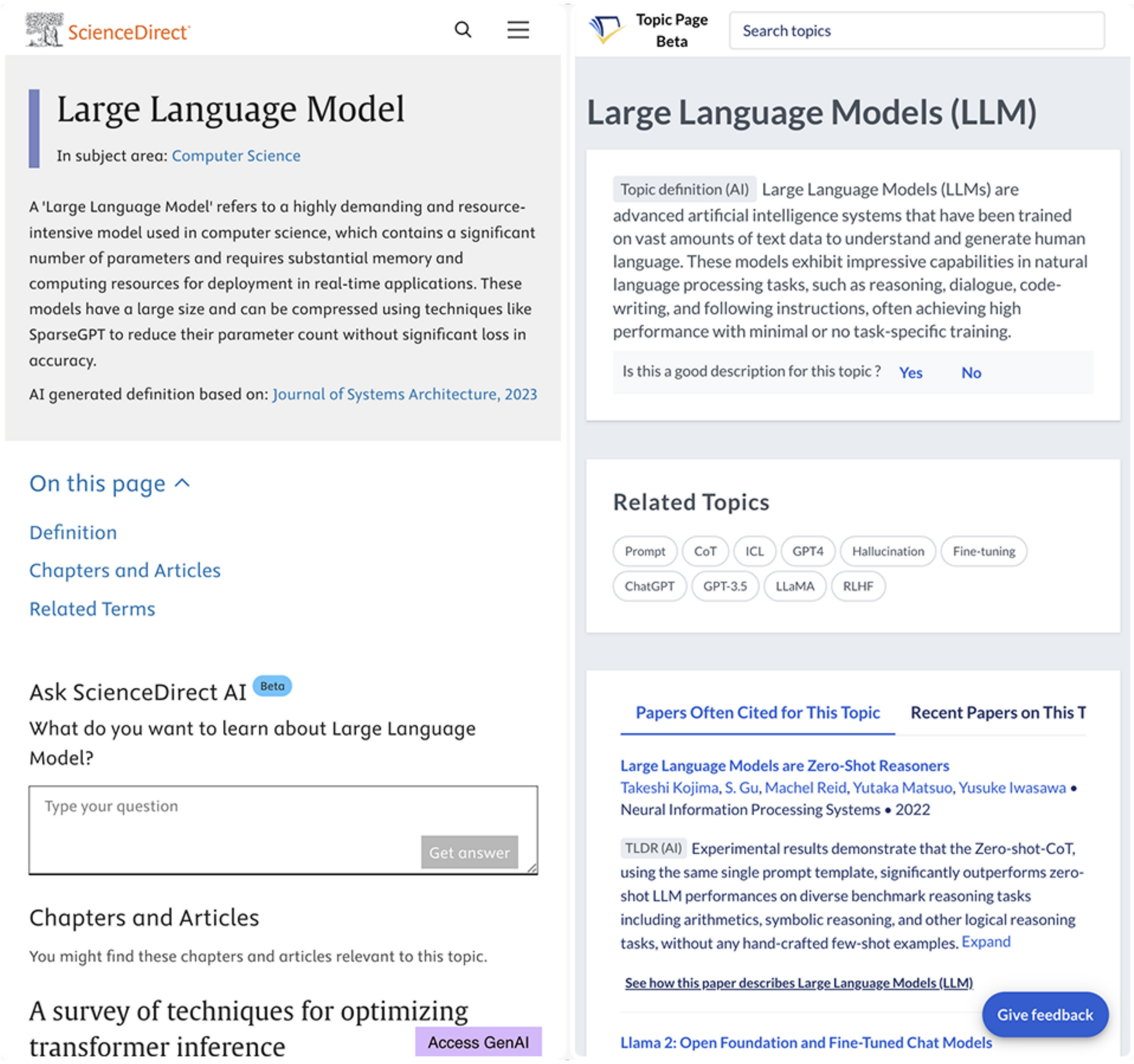
Context Window Saturation & Stopping Criterion Our iterative knowledge expansion faces scalability limitations due to LLM context window constraints [89]. As the amount of depth increases the accumulated knowledge graphs and retrieved snippets may exceed available context windows. In our experiments we are able to synthesize 125 snippets of 512 max tokens each. While this number is substantially lower compared to the context window of gpt-4o-mini [90] the generalization to other models with smaller windows may limit its usability. Additionally, the current exploration of a topic lacks of a intelligent stopping criteria, potentially leading to unnecessary API calls when sufficient information has already been gathered. Future research should implement adaptive exploration strategies that balance information discovery against computational costs. For instance, an extension to our work could be the adaptation of the generated content tailored to specific audience needs.

Evaluation Methodology & Gold Standard Dataset While our evaluation provides meaningful insights into Apollo's performance, several aspects could be strengthened in future work. Our human evaluation, conducted on 20 topic pages with subject matter experts, may require more topic assessments for effectively establishing statistical significance regarding human-LLM alignment across diverse scientific domains [14]. Additionally, we lack access to gold standard Wikipedia-like pages with their corresponding source materials. This constraint necessitated using SciWiki as a proxy for comparing generated content against human-written articles, potentially introducing evaluation biases [91]. The absence of ground truth source-to-content mappings makes it difficult to assess optimal knowledge synthesis strategies. Future work should establish larger-scale evaluation protocols with diverse expert populations and develop a benchmark dataset that include explicit source-to-content relationships for more rigorous assessment.

Multi-modal Content Generation Our framework produces exclusively textual content, ignoring essential scientific communication elements such as equations, figures, tables, and diagrams. These are elements that constitute fundamental building blocks on the scientific understanding. Real scientific articles rely heavily on visual representations, mathematical formulations, and structured data presentations to convey complex concepts effectively [92]. This limitation significantly constrains practical applicability, as users expect comprehensive topic pages to integrate multiple content modalities. Future work should investigate multi-modal content generation capabilities, including automatic figure selection, equation extraction, and table generation from structured data. For instance, an extension to our work could be the use of Visual Language Models (VLMs) [93] to automatically select relevant figures and generate descriptive captions to align with the textual content.

Appendix

A Topic Pages



ScienceDirect

Large Language Model

In subject area: Computer Science

A 'Large Language Model' refers to a highly demanding and resource-intensive model used in computer science, which contains a significant number of parameters and requires substantial memory and computing resources for deployment in real-time applications. These models have a large size and can be compressed using techniques like SparseGPT to reduce their parameter count without significant loss in accuracy.

AI generated definition based on: Journal of Systems Architecture, 2023

On this page

Definition

Chapters and Articles

Related Terms

Ask ScienceDirect AI Beta

What do you want to learn about Large Language Model?

Type your question

Get answer

Chapters and Articles

You might find these chapters and articles relevant to this topic.

A survey of techniques for optimizing transformer inference

Access GenAI

Topic Page Beta

Search topics

Large Language Models (LLM)

Topic definition (AI) Large Language Models (LLMs) are advanced artificial intelligence systems that have been trained on vast amounts of text data to understand and generate human language. These models exhibit impressive capabilities in natural language processing tasks, such as reasoning, dialogue, code-writing, and following instructions, often achieving high performance with minimal or no task-specific training.

Is this a good description for this topic ? Yes No

Related Topics

Prompt CoT ICL GPT4 Hallucination Fine-tuning ChatGPT GPT-3.5 LLaMA RLHF

Papers Often Cited for This Topic Recent Papers on This T

Large Language Models are Zero-Shot Reasoners
Takeshi Kojima, S. Gu, Machel Reid, Yutaka Matsuo, Yusuke Iwasawa •
Neural Information Processing Systems • 2022

TLDR (AI) Experimental results demonstrate that the Zero-shot-CoT, using the same single prompt template, significantly outperforms zero-shot LLM performances on diverse benchmark reasoning tasks including arithmetics, symbolic reasoning, and other logical reasoning tasks, without any hand-crafted few-shot examples. Expand

See how this paper describes Large Language Models (LLM)

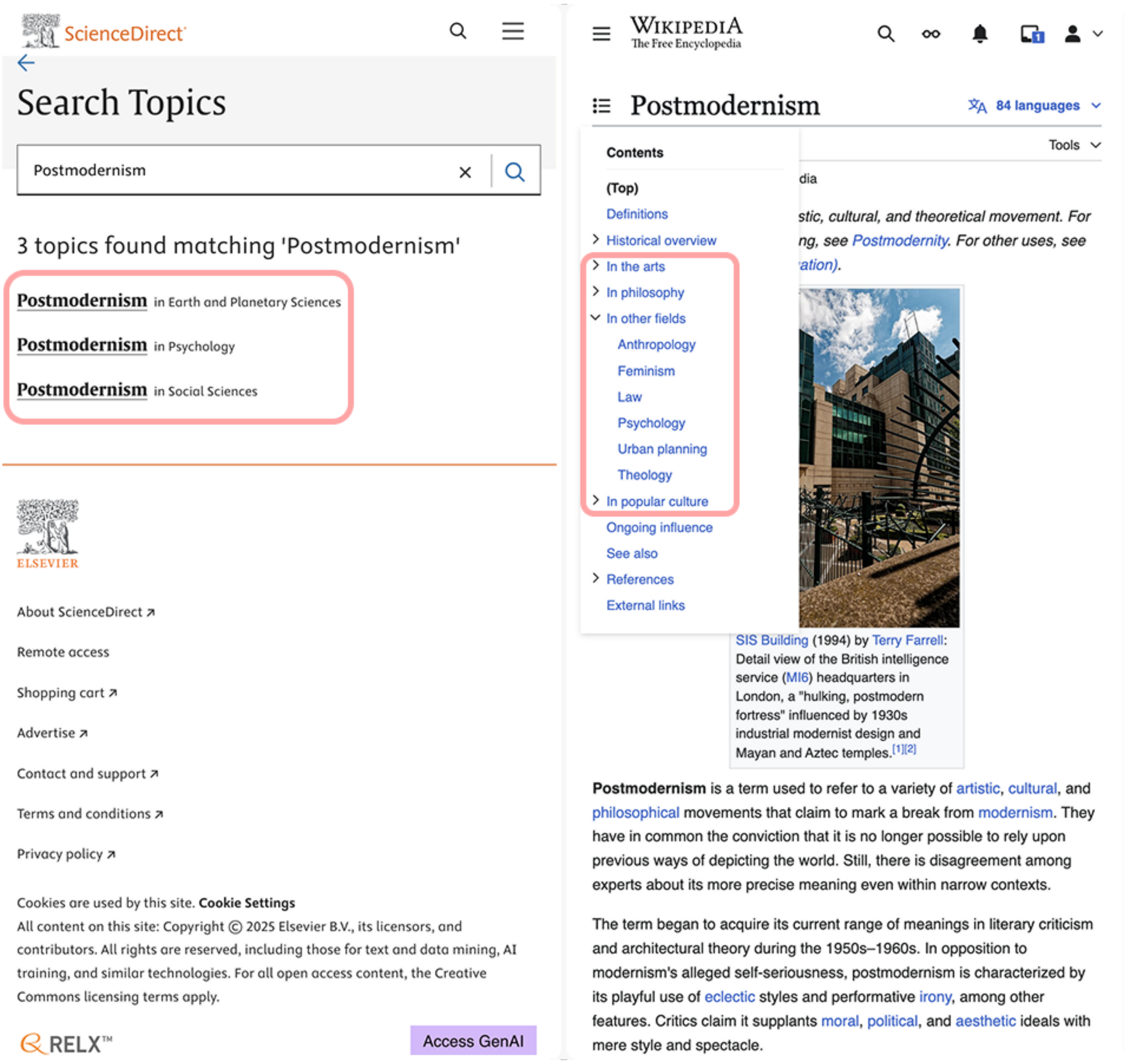
Llama 2: Open Foundation and Fine-Tuned Chat Models

Give feedback

Figure 1: Comparison of current topic pages from different providers: ScienceDirect (left) and Semantic Scholar (right). Both follow a similar layout, displaying the title, definition, related papers, and related topics.

B Dataset Details

B.1 Design Choices



Search Topics

Postmodernism

3 topics found matching 'Postmodernism'

Postmodernism in Earth and Planetary Sciences

Postmodernism in Psychology

Postmodernism in Social Sciences

Cookies are used by this site. **Cookie Settings**
All content on this site: Copyright © 2025 Elsevier B.V., its licensors, and contributors. All rights are reserved, including those for text and data mining, AI training, and similar technologies. For all open access content, the Creative Commons licensing terms apply.

Postmodernism

dia

stic, cultural, and theoretical movement. For
ng, see Postmodernity. For other uses, see
ation).

SIS Building (1994) by Terry Farrell: Detail view of the British intelligence service (MI6) headquarters in London, a "hulking, postmodern fortress" influenced by 1930s industrial modernist design and Mayan and Aztec temples.[1][2]

Postmodernism is a term used to refer to a variety of artistic, cultural, and philosophical movements that claim to mark a break from modernism. They have in common the conviction that it is no longer possible to rely upon previous ways of depicting the world. Still, there is disagreement among experts about its more precise meaning even within narrow contexts.

The term began to acquire its current range of meanings in literary criticism and architectural theory during the 1950s–1960s. In opposition to modernism's alleged self-seriousness, postmodernism is characterized by its playful use of eclectic styles and performative irony, among other features. Critics claim it supplants moral, political, and aesthetic ideals with mere style and spectacle.

Figure 2: Comparing the topic: `Postmodernism`. Results from Science Direct (SD) (left) and the corresponding Wikipedia page (right). The current knowledge base at Science Direct treats `Postmodernism` from different perspective while its counterpart covers a wide range of categories.

B.2 Topic Concepts

No.	Domain	Concept	Corpus Size
1	Economics	Division of labour	3816
		Economic globalization	2105
		Mercantilism	144
		Oligopoly	1790
		Quantity theory of money	329
2	AgriBio	Carica papaya	29385
		Conventional farming	7406
		Moringa oleifera	8023
		Plant morphology	6523
		Rhizosphere	140535
3	ComputerScience	Cyclic redundancy check	2630
		Ensemble learning	9476
		Linear discriminant analysis	10296
		Network time protocol	1596
		Trunked radio system	80
4	Immunology_Microbiology	Bacterial taxonomy	584
		Cestoda	8825
		Diagnostic microbiology	856
		Germ theory of disease	647
		Skin flora	3497
5	Mathematics	Bayesian network	3503
		Brahmagupta	184
		Discrete wavelet transform	938
		Multivariate normal distribution	2026
		Quadratic equation	569
6	FoodScience	Breakfast cereal	12365
		Food colorant	616
		Food-borne disease	50150
		Iodized salt	1358
		Sourdough bread	3280
7	Neuro	Dopamine hypothesis of schizophrenia	1675
		Laudanum	145
		Microglia	146
		Nanopore sequencing	372
		Psychological testing	2090
8	Biochem_Genetics_MolBio	Allosteric regulation	43584
		Genome annotation	10656
		Molecular cloning	26835
		Somatic cell nuclear transfer	7142
		Xyy syndrome	668

Table 1: SciWiki-100 Concepts: Domains 1–8 (Economics–Biochem_Genetics_MolBio)

No.	Domain	Concept	Corpus Size
9	Chemistry	Alpha-helix	32939
		Azeotropic mixture	8242
		Metrology	7433
		Second-harmonic generation	18362
		Thermal runaway	3680
10	Engineering	Bioplastics	6183
		Cellphone	17851
		Multi-criteria decision-making	9497
		Solar thermal energy	21415
		Wireless sensor network	30050
11	Psychology	Agreeableness	16619
		Hedonic adaptation	272
		Mind wandering	6802
		Openness to experience	9378
		Reconstructive memory	298
12	SocialSciences	Ecofeminism	290
		Feudalism	1474
		Group cohesion	2031
		Positivism	2102
		Posthumanism	352
13	Physics	Cherenkov radiation	3732
		Eyepiece	2476
		Fractional calculus	5991
		Quantum chromodynamics	21966
		Topological insulator	11749
14	Nursing_HealthProf	Drug dependence	8994
		Heart lung machine	386
		Lie detection	72
		Nuclear magnetic resonance imaging	77588
		Tourniquet	6079
15	VeterinaryMedicine	Cushing reflex	171
		Epinephrine	12070
		Parthenogenesis	766
		Progressive systemic sclerosis	35
		Xylazine	14026
16	EarthPlanetaryScience	Basalt	133082
		Global warming potential	33685
		Gobi desert	3158
		Species concept	1334
		Younger dryas	13756
17	Med_Dentistry	Bloodstain pattern analysis	379
		Mycobacterium tuberculosis	24239
		Ovarian cyst	21686
		Prefrontal cortex	21556
		Wilcoxon signed ranks test	1706

Table 2: SciWiki-100 Concepts: Domains 9–17 (Chemistry–Med_Dentistry)

No.	Domain	Concept	Corpus Size
18	ChemicalEngineering	Kaolinite	54559
		Metal foam	12206
		Polysilicon	8951
		Sodium bicarbonate	30239
		Zeolitic imidazolate framework	67765
19	Pharma_Tox	Dimethyl sulfoxide	82582
		Ivermectin	18999
		Nerium oleander	4858
		Phytohormone	35319
		Semaglutide	8141
20	MaterialScience	Fatigue of materials	35827
		Hydrogen bonding	65127
		Photoelectrochemical cell	11712
		Pitting corrosion	43214
		Prestressed concrete	7423

Table 3: SciWiki-100 Concepts: Domains 18–20

C Terminology

symbol	meaning	symbol	meaning
T	topic	$\mathcal{G}_m^*$	normalized KG at depth m
D	domain	V_i, E_i	vertices and edges of subgraph G_i
$\mathcal{C}_D$	domain-specific vector collection	$\mathbb{Q}_m$	set of research questions at depth m
s_i	snippet (entry point)	q_j, ρ_j	question and rationale pair
c_i	raw text content	$\mathbb{L}_m$	set of query terms at depth m
$\mathbf{e}_i$	embedding vector	ℓ_j	individual query term
m_i	metadata (domain, topic, URL)	$\mathcal{M}_Q, \mathcal{M}_L$	question and query memory sets
$\mathcal{I}_m$	retrieved information at depth m	$\mathcal{K}$	curated knowledge base
G_i	snippet-level subgraph	$\mathcal{O}$	article outline
$\mathcal{G}_m$	knowledge graph at depth m	h_i	section heading
Φ	triplet extraction operator	$\mathcal{R}_i$	retrieved snippets for section i
η	graph normalization function	$\mathcal{S}_i$	filtered snippets for section i
Ψ	question generation operator	$a_i^{(r)}$	article section at revision r
Λ	query synthesis operator	$\mathbb{F}_i^{(r)}$	feedback at revision r
Ω	outline generation operator	$\mathcal{M}_F$	feedback memory
Γ	writer operator	$\mathcal{A}$	final article
π	reviewer operator	$r_{\max}$	maximum revision count
Θ	relevance filter	m	maximum expansion depth
(h, r, t)	knowledge triplet	k	top-k retrieval parameter

Table 4: Table of symbols and meanings for the Apollo framework.

D Pseudo Code

Algorithm 1 APOLLO Framework

```
Input: Topic $T$, Domain $D$, Vector collection $\mathcal{C}_D$, Max depth $m$
Output: Final topic page article $\mathcal{A}$
Phase 1: Knowledge Curation
Initialization:
$\mathcal{I}_0 \leftarrow \text{Retrieve}(T, \mathcal{C}_D)$    ▷ Initial query retrieval
$\{G_i\}_{s_i \in \mathcal{I}_0} \leftarrow \Phi(\mathcal{I}_0)$    ▷ Extract triplets from snippets
$\mathcal{G}_0 \leftarrow \bigcup_i G_i$    ▷ Aggregate subgraphs
$\mathcal{G}_0^* \leftarrow \eta(\mathcal{G}_0)$    ▷ Normalize knowledge graph
Initialize $\mathcal{M}_Q \leftarrow \emptyset$, $\mathcal{M}_L \leftarrow \emptyset$    ▷ Memory sets
Iterative Expansion:
for depth $j = 0$ to $m - 1$ do
    Agent 1 - Post-doc Researcher:
    $\mathbb{Q}_j \leftarrow \Psi(\mathcal{G}_j^*, \mathcal{M}_Q)$    ▷ Generate research questions
    $\mathcal{M}_Q \leftarrow \mathcal{M}_Q \cup \{q \mid (q, \rho) \in \mathbb{Q}_j\}$
    Agent 2 - Query Synthesizer:
    $\mathbb{L}_j \leftarrow \Lambda(\mathbb{Q}_j, \mathcal{M}_L)$    ▷ Synthesize query terms
    $\mathcal{M}_L \leftarrow \mathcal{M}_L \cup \mathbb{L}_j$
    Retrieval and Graph Update:
    $\mathcal{I}_{j+1} \leftarrow \text{Retrieve}(\mathbb{L}_j, \mathcal{C}_D) \setminus \bigcup_{i=0}^{j} \mathcal{I}_i$
    $\{G_i\}_{s_i \in \mathcal{I}_{j+1}} \leftarrow \Phi(\mathcal{I}_{j+1})$
    $\mathcal{G}_{j+1}^* \leftarrow \eta(\mathcal{G}_j^* \cup \bigcup_{s_i \in \mathcal{I}_{j+1}} G_i)$
end for
$\mathcal{K} \leftarrow \bigcup_{j=0}^{m} \mathcal{I}_j$    ▷ Curated knowledge base
Phase 2: Outline Generation
$\mathcal{O} \leftarrow \Omega(\mathcal{G}_m^*, T)$    ▷ Generate hierarchical outline
Phase 3: Article Generation
for each section $h_i \in \mathcal{O}$ do
    $\mathcal{R}_i \leftarrow \text{Retrieve}(h_i, \mathcal{K})$    ▷ Section-specific retrieval
    $\mathcal{S}_i \leftarrow \{s \in \mathcal{R}_i \mid \Theta(s, h_i) = \text{relevant}\}$    ▷ Filter relevance
    Writer-Reviewer Collaboration:
    $a_i^{(0)} \leftarrow \Gamma(h_i, \mathcal{S}_i)$    ▷ Initial section draft
    $r \leftarrow 0$, $\mathcal{M}_F \leftarrow \emptyset$
    repeat
        Agent 4 - Reviewer:
        $\mathbb{F}_i^{(r)} \leftarrow \pi(a_i^{(r)}, \mathcal{S}_i, \mathcal{M}_F)$    ▷ Generate feedback
        $\mathcal{M}_F \leftarrow \mathcal{M}_F \cup \mathbb{F}_i^{(r)}$
        if $\mathbb{F}_i^{(r)} \neq \emptyset$ and $r < r_{\max}$ then
            Agent 3 - Writer:
            $a_i^{(r+1)} \leftarrow \Gamma^{\text{revise}}(a_i^{(r)}, \mathbb{F}_i^{(r)}, \mathcal{S}_i)$    ▷ Revise section
            $r \leftarrow r + 1$
        end if
    until $\mathbb{F}_i^{(r)} = \emptyset$ or $r = r_{\max}$
end for
Article Assembly:
$\mathcal{A} \leftarrow \bigoplus_{i=1}^{|\mathcal{O}|} a_i^{(r_i^*)}$    ▷ Concatenate final sections
Return: Complete topic page $\mathcal{A}$
```

E Prompts

Knowledge Extraction Agent Prompt

```
You are a top-tier algorithm designed for extracting information in structured formats to build a knowledge graph
     about {topic}.

# Instructions
Adhere to the following steps:

// Question Exploration
0. Based on the snippet provided formulate a set of questions that can expand our knowledge about the topic.

// Entity Extraction
1. Identify all relevant entities to fully understand the provided snippet. For each identified entity, extract the
     following information:
- entity_name: Full name of the entity. An entity in a knowledge graph is a node that represents a real-world object
     , concept, or abstract idea, which can be uniquely identified.
- entity_description: short description of the entity and why is important to analyze it in this context to
     understand the snippet.

// Relationship Extraction
2. From the entities identified in step 1, identify all pairs of (source_entity, target_entity) that are *clearly
     related* to each other.
For each pair of related entities, extract the following information:
- source_node_id: the node_id of the source entity, as identified in step 1
- target_node_id: the node_id of the target entity, as identified in step 1
- relationship_type: a short description (lower-case with underscores & between 1-4 words) why the two entities
     related to each other.
- relationship_description: explanation as to why you think the source entity and the target entity are related to
     each other
- To construct the entity related pairs use the description that we found in step 1.

// Key Words
3. Identify high-level key words that summarize the main concepts, themes, or topics of the entire text. These
     should capture the overarching ideas present in the document.

// Consideration:
- Do not invent entities or relations not directly stated.

# Output:
- Do not return any additional information other than the JSON format below:
{
  "nodes": [
    {
      "id": "entity_name",
      "label": "Human readable name",
      "description": "entity_description"
    },
    {
      "id": "project_apollo",
      "label": "Project Apollo",
      "description": "entity_description"
    },
    {
      "id": "nasa",
      "label": "NASA",
      "description": "entity_description"
    },
    ...
  ],

  "edges": [
    {
      "from": "source_node_id",
      "to": "target_node_id",
      "relationship": "relationship_type",
      "relationship_description": "relationship_description"
    },
    {
      "from": "project_apollo",
      "to": "nasa",
      "relationship": "lead_by",
      "relationship_description": "relationship_description"
    },
    ...
  ]
}
```

Agent 1: Post-doc Researcher Prompt

```
You are a helpful assistant working for a research team on expanding the provided knowledge graph.
Your task is to come up with queries to deepen (depth) and expand (breadth) our understanding of the topic: {topic}.

Instructions:
1. Analyze the knowledge graph carefully to identify gaps in our understanding
2. Look for areas that are under-explored or completely missing
3. Consider aspects that would provide new perspectives not currently represented

Considerations:
- IMPORTANT: The knowledge graph represents our accumulated knowledge so far - focus on what’s MISSING, not what’s
     already there
- Examine both nodes and relationships to find areas needing exploration
- Consider formulating queries that explore:
  - General queries: Areas entirely absent from the graph that could broaden understanding
  - In-depth queries: Existing areas that need deeper investigation
- Analyze which of these two types (general vs in-depth) would be more valuable at this stage
- For each query, provide one sentence explaining why this information would specifically fill a gap in the current
     knowledge graph

Constraints:
- Do NOT repeat any of the questions below as we have already explored them:
{questions_seen}

Output:
Adhere to the following format and do not output anything else:
{
  ”general_queries”: [
    {
      ”query1”: ”General Query 1”,
      ”explanation1”: ”This addresses [specific gap] not currently represented in the graph”
    },
    ...
  ],
  ”in_depth_queries”: [
    {
      ”query1”: ”In-depth Query 1”,
      ”explanation1”: ”This expands on [specific node/relationship] which is currently superficial”
    },
    ...
  ]
}
```

Agent 2: Reflective Query Synthesiser Prompt

```
You are an expert researcher on the topic: {topic}. Your task is to transform the provided questions
into effective search queries to expand our breath and depth of the topic treated.

Instructions:
- Think from the perspective of {audience}.
- Convert the ‘general_queries‘ and ‘in_depth_queries‘ into precise search terms for a search engine
- Create search queries that will provide the most valuable NEW information
- Focus on generating DIVERSE queries that cover different aspects of the topic
- Format each query as concise keywords (2-5 words) suitable for a search engine
- Rank the queries by importance, with the most critical knowledge gaps first

Constraints:
- Do not add include the ’{topic}‘ in your search queries.
- Create search queries that MUST NOT repeat or paraphrase any query from the ‘queries_seen‘ provided below.
- Avoid any two queries that share more than one keyword in common.
- Avoid creating queries that are likely to retrieve the same information.
- CRITICAL: Do NOT generate queries similar to any of the queries below:
{queries_seen}

Output:
- Adhere to the following format and do not add any other text:
{
    ”combined_queries”: [
        ”query_1”,
        ”query_2”,
        ...
        ”query_N”
    ]
}
```

Outline Construction Agent Prompt

```
Based on this knowledge graph construct an outline for a Wikipedia Article that covers all the nodes and edges in
     the graph. The outline should be structured with headings and subheadings, and should build a comprehensive
     overview of the topic.

Here is the format of your writing:
1. Use "#" Title" to indicate section title, "##" Title" to indicate subsection title, "###" Title" to indicate
     subsubsection title, and so on.
2. Do not include other information.
3. Do not include topic name itself in the outline.
```

Section-Specific Relevance Filter Prompt

```
You are a thorough Wikipedia reviewer that needs to check whether the provided snippet is relevant to explain the
     section provided about the topic.

The snippet must meet BOTH criteria:
1. Be relevant to the section theme
2. Actually mention or discuss the main topic

Example 1:
- Topic: "Neural Networks"
- Section: "Backpropagation"
- Snippet: "Backpropagation is an algorithm used to train neural networks by adjusting weights based on the error
     gradient."
- Answer: "yes"

Example 2:
- Topic: "Neural Networks"
- Section: "Backpropagation"
- Snippet: "Neural networks are computational models inspired by the human brain."
- Answer: "no" (this is about neural networks generally, not specifically about backpropagation)

Output:
- Reply ONLY with 'yes' or 'no' to indicate whether the snippet is relevant to the section. Do not provide any other
      information or explanation.
```

Agent 3: Factual Writer Prompt

```
You are an expert Wikipedia writer tasked with creating FACTUAL, clear, and thoroughly cited sections based on
     provided reference materials.

# CORE GUIDELINES

## 1. Reference Selection and Analysis
- Thoroughly analyze all provided ‘Ref: [digit]‘ snippets before writing.
- Map each snippet to relevant parts of the outline section.
- Identify overlapping information across multiple references to strengthen claims.
- Never write anything that cannot be directly supported by the provided references.

## 2. Citation Requirements
CRITICAL: Every factual statement MUST have a citation.

### Mandatory Citation Rules:
- EVERY sentence containing factual information must end with a citation [1] or multiple citations [1][2].
- ALL opening sentences of sections and subsections MUST have citations.
- ALL definitional statements (using "is”, "are”, "refers to”, "encompasses”, ...) MUST be cited.
- ALL claims about effectiveness (using "enhances”, "improves”, "reduces”, ...) MUST be cited.
- Descriptive or analytical statements must be cited if they interpret or synthesize information.
- Only pure transitional phrases like "This section discusses...” may omit citations.
- When in doubt, cite - over-citation is preferable to under-citation.
- NEVER end mid-sentence - ensure all sentences are complete with proper punctuation and citations.

### Citation Placement:
- Place citations immediately after the claim they support.
- For compound sentences, place citations after each distinct claim.
  Example: "The process involves three steps [1], which were first documented in 2020 [2].”

## 3. Content Requirements

### Neutrality and Accuracy:
- Maintain Wikipedia’s neutral point of view (NPOV).
- Present facts without bias or opinion.
- Use precise, encyclopedic language.
- AVOID weasel words or unsupported generalizations.

### Comprehensiveness:
- Include all relevant information from provided references.
- Synthesize information when multiple sources discuss the same topic.
- Ensure logical flow between paragraphs.

## 4. Specific Constraints

### DO NOT:
- Include information not present in the provided references.
- Make logical leaps or assumptions beyond the source material.
- Use author names from the references (use citation numbers instead).
- Create a separate references section.
- Leave any factual claim without a citation.

### DO:
- Start with "# section name” for the main section.
- Use "## subsection name” and "### sub-subsection name” as needed.
- Cite every piece of information that comes from a reference.
- Use multiple citations [1][2][3] when a claim is supported by multiple sources.
- Write in clear, accessible language while maintaining accuracy.
```

Factual Editor Prompt

```
You are a Wikipedia editor fixing a specific section based on reviewer feedback. Your primary goal is to ensure
     every claim is factually accurate and properly supported by the provided references.

Your Task:
- Fix ONLY the issues mentioned in the feedback
- Ensure all claims are supported by the cited references
- Maintain the overall structure and flow of the section
- If you cannot cite it verbatim in the given reference, DELETE the statement. Do not paraphrase or hedge.

Editing Guidelines:
1. Addressing Feedback:
   - For each feedback item, locate the mentioned claim in the section
   - Fix the issue by either:
     a) Correcting the citation to match a supporting reference
     b) Rewriting the claim to accurately reflect what's in the references
     c) IMPORTANT: DELETE the claim ONLY if no reference supports it

2. Citation Rules:
   - Use only citation numbers like [1], [2], etc.
   - Only cite references that actually support the claim
   - Every factual claim must have a citation

3. Preserving Content:
   - Keep all correctly cited content unchanged
   - Maintain the section's structure (headings, paragraph breaks)
   - Preserve writing style and tone
   - Only modify sentences explicitly mentioned in feedback

4. When References Don't Support a Claim:
   - First try to rewrite the claim to match what the references actually say
   - Only remove the claim if no reference supports any version of it
   - If removing, ensure the text still flows naturally

5. CRITICAL: Only make claims that are DIRECTLY stated in the references.

Output: The revised section with only the necessary changes made.
```

Agent 4: Critical Reviewer Prompt

```
You are a strict Wikipedia fact-checker collaborating with an editor. Your job is to review this specific section
     and ensure that every atomic claim (i.e., each coherent statement or set of sentences followed by a citation)
     is properly supported by the cited reference.

Review Mode:
- If ‘previous_feedback‘ is empty:
  - Perform a complete review of all atomic claims in this section.
- If ‘previous_feedback‘ is NOT empty:
  - ONLY review the issues listed in ‘previous_feedback‘ for this section.
  - Do NOT re-raise the same issue if it was addressed by removal
  - IMPORTANT:
    a) First check if the quoted text from each feedback item still exists in the current section content. If the
         exact quoted text cannot be found, that issue is resolved (the claim was likely rewritten or removed).
    b) Citation numbers in ‘previous_feedback‘ may no longer be valid. Focus on the quoted text content, not citation
          numbers. If the quoted text no longer exists in the current content, that issue is resolved.
    c) If a problematic sentence was deleted entirely, consider that issue RESOLVED

Review Process:
1. Read through the section content, identifying each atomic claim.
2. For each atomic claim with a citation [X]:
  - Check if Ref: [X] in the provided references fully supports the claim.
  - Accept semantic equivalence (e.g., "distributed in" = "found in")
  - If supported, no feedback is needed.
  - If not supported, specify exactly what is unsupported.
3. For each atomic claim without a citation:
  - Determine if it contains a factual claim that requires a citation.
  - If so, specify which Ref: [digit] should be added.

Approval Criteria:
- Verdict is "approved" if every atomic claim is correctly cited or does not require a citation.
- If any atomic claim is not properly supported, verdict is "needs revision".

Output Format:
- Verdict: "approved" or "needs revision"
- Feedback: List of specific issues to fix
```

F Rubric Grading

Here, we provide the detailed rubric grading criteria described in Section 4, for the Article and Outline evaluation by M-Prometheus-7B.

F.1 Outline Evaluation

Content Generation Guidance: Does the outline effectively guide content generation?	
Score 1	The outline fails to guide content generation, omitting significant aspects of the topic or providing insufficient direction.
Score 2	The outline provides limited guidance, covering some key areas but lacking depth or completeness in addressing the topic.
Score 3	The outline provides moderate guidance for content generation, addressing most key areas but leaving some gaps or ambiguities.
Score 4	The outline effectively guides content generation, covering all significant aspects with clear direction, though minor refinements could enhance comprehensiveness.
Score 5	The outline is exemplary in guiding content generation, thoroughly addressing all aspects of the topic with clear, detailed direction and no significant gaps.
Hierarchical Clarity: Does the outline clearly define a hierarchy of topics and subtopics?	
Score 1	The outline exhibits no discernible hierarchical structure. Topics and subtopics are jumbled together without logical separation or clear levels.
Score 2	The outline attempts to establish a hierarchy but fails to maintain logical consistency. Main topics and subtopics are frequently misclassified, and the structure is overly rigid or disjointed.
Score 3	The outline has a recognizable hierarchical structure but lacks diversity in organization style. While main topics are somewhat clear, subtopics occasionally overlap or are misaligned.
Score 4	The outline displays a clear, logical, and diverse hierarchical structure. Main topics are distinct, and subtopics are properly nested. While most elements are well-placed, there may be minor redundancies.
Score 5	The outline showcases an exceptional, flawless hierarchical structure. Each main topic is distinct, and subtopics are logically nested with absolute clarity and stylistic diversity.
Logical Coherence: Does the outline logically organize topics ensuring smooth flow of ideas?	
Score 1	The outline is highly disjointed and incoherent. Topics and subtopics appear in a random, unordered manner, with no logical flow or sense of progression.
Score 2	The outline shows some attempt at logical organization, but it contains frequent inconsistencies, abrupt shifts, or logical missteps. Topics and subtopics are misaligned or lack proper transitions.
Score 3	The outline demonstrates a basic level of logical coherence. Most topics follow a general sequence, but some sections feel forced, with weak or unclear transitions.
Score 4	The outline exhibits a strong sense of logical flow, with ideas presented in a mostly smooth and connected manner. Transitions between topics and subtopics are clear, but a few minor adjustments could make the flow more seamless.
Score 5	The outline achieves exceptional logical coherence. Each topic and subtopic follows a deliberate, thoughtful progression, with clear, natural, and intuitive transitions.

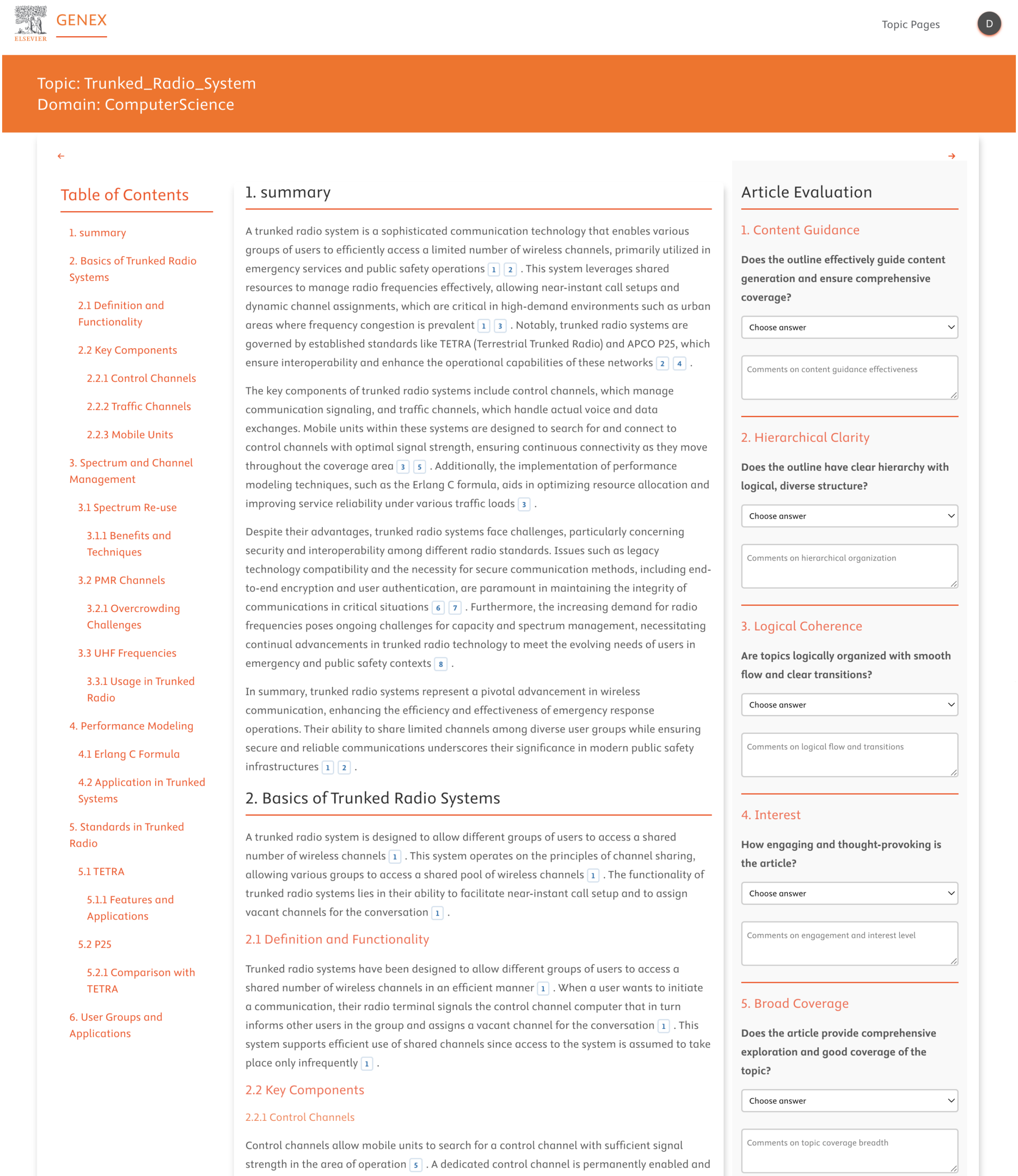
Table 5: Scoring rubrics on a 1–5 scale for outline quality evaluation by M-Prometheus-7B.

F.2 Article Evaluation

Interest Level:	How engaging and thought-provoking is the article?
Score 1	Not engaging at all; no attempt to capture the reader’s attention.
Score 2	Fairly engaging with a basic narrative but lacking depth.
Score 3	Moderately engaging with several interesting points.
Score 4	Quite engaging with a well-structured narrative and noteworthy points that frequently capture and retain attention.
Score 5	Exceptionally engaging throughout, with a compelling narrative that consistently stimulates interest.
Coherence and Organization:	Is the article well-organized and logically structured?
Score 1	Disorganized; lacks logical structure and coherence.
Score 2	Fairly organized; a basic structure is present but not consistently followed.
Score 3	Organized; a clear structure is mostly followed with some lapses in coherence.
Score 4	Good organization; a clear structure with minor lapses in coherence.
Score 5	Excellently organized; the article is logically structured with seamless transitions and a clear argument.
Relevance and Focus:	Does the article stay on topic and maintain a clear focus?
Score 1	Off-topic; the content does not align with the headline or core subject.
Score 2	Somewhat on topic but with several digressions; the core subject is evident but not consistently adhered to.
Score 3	Generally on topic, despite a few unrelated details.
Score 4	Mostly on topic and focused; the narrative has a consistent relevance to the core subject with infrequent digressions.
Score 5	Exceptionally focused and entirely on topic; the article is tightly centered on the subject, with every piece of information contributing to a comprehensive understanding of the topic.
Depth of Exploration:	How thoroughly does the report explore the initial topic and its related areas?
Score 1	Very superficial; provides only a basic overview with significant gaps in exploration.
Score 2	Superficial; offers some detail but leaves many important aspects unexplored.
Score 3	Moderate depth; covers key aspects but may lack detailed exploration in some areas.
Score 4	Good depth; explores most aspects in detail with minor gaps.
Score 5	Excellent depth; thoroughly explores all relevant aspects with comprehensive detail, reflecting a deep and dynamic discourse.

Table 6: Scoring rubrics on a 1–5 scale for article quality evaluation by M-Prometheus-7B.

G Human Evaluation Details



Topic: Trunked_Radio_System
Domain: ComputerScience

Table of Contents

1. summary

A trunked radio system is a sophisticated communication technology that enables various groups of users to efficiently access a limited number of wireless channels, primarily utilized in emergency services and public safety operations [1] [2] . This system leverages shared resources to manage radio frequencies effectively, allowing near-instant call setups and dynamic channel assignments, which are critical in high-demand environments such as urban areas where frequency congestion is prevalent [1] [3] . Notably, trunked radio systems are governed by established standards like TETRA (Terrestrial Trunked Radio) and APCO P25, which ensure interoperability and enhance the operational capabilities of these networks [2] [4] .

The key components of trunked radio systems include control channels, which manage communication signaling, and traffic channels, which handle actual voice and data exchanges. Mobile units within these systems are designed to search for and connect to control channels with optimal signal strength, ensuring continuous connectivity as they move throughout the coverage area [3] [5] . Additionally, the implementation of performance modeling techniques, such as the Erlang C formula, aids in optimizing resource allocation and improving service reliability under various traffic loads [3] .

Despite their advantages, trunked radio systems face challenges, particularly concerning security and interoperability among different radio standards. Issues such as legacy technology compatibility and the necessity for secure communication methods, including end-to-end encryption and user authentication, are paramount in maintaining the integrity of communications in critical situations [6] [7] . Furthermore, the increasing demand for radio frequencies poses ongoing challenges for capacity and spectrum management, necessitating continual advancements in trunked radio technology to meet the evolving needs of users in emergency and public safety contexts [8] .

In summary, trunked radio systems represent a pivotal advancement in wireless communication, enhancing the efficiency and effectiveness of emergency response operations. Their ability to share limited channels among diverse user groups while ensuring secure and reliable communications underscores their significance in modern public safety infrastructures [1] [2] .

2. Basics of Trunked Radio Systems

A trunked radio system is designed to allow different groups of users to access a shared number of wireless channels [1] . This system operates on the principles of channel sharing, allowing various groups to access a shared pool of wireless channels [1] . The functionality of trunked radio systems lies in their ability to facilitate near-instant call setup and to assign vacant channels for the conversation [1] .

2.1 Definition and Functionality

Trunked radio systems have been designed to allow different groups of users to access a shared number of wireless channels in an efficient manner [1] . When a user wants to initiate a communication, their radio terminal signals the control channel computer that in turn informs other users in the group and assigns a vacant channel for the conversation [1] . This system supports efficient use of shared channels since access to the system is assumed to take place only infrequently [1] .

2.2 Key Components

2.2.1 Control Channels

Control channels allow mobile units to search for a control channel with sufficient signal strength in the area of operation [5] . A dedicated control channel is permanently enabled and

Article Evaluation

1. Content Guidance

Does the outline effectively guide content generation and ensure comprehensive coverage?

Choose answer ⌄

Comments on content guidance effectiveness

2. Hierarchical Clarity

Does the outline have clear hierarchy with logical, diverse structure?

Choose answer ⌄

Comments on hierarchical organization

3. Logical Coherence

Are topics logically organized with smooth flow and clear transitions?

Choose answer ⌄

Comments on logical flow and transitions

4. Interest

How engaging and thought-provoking is the article?

Choose answer ⌄

Comments on engagement and interest level

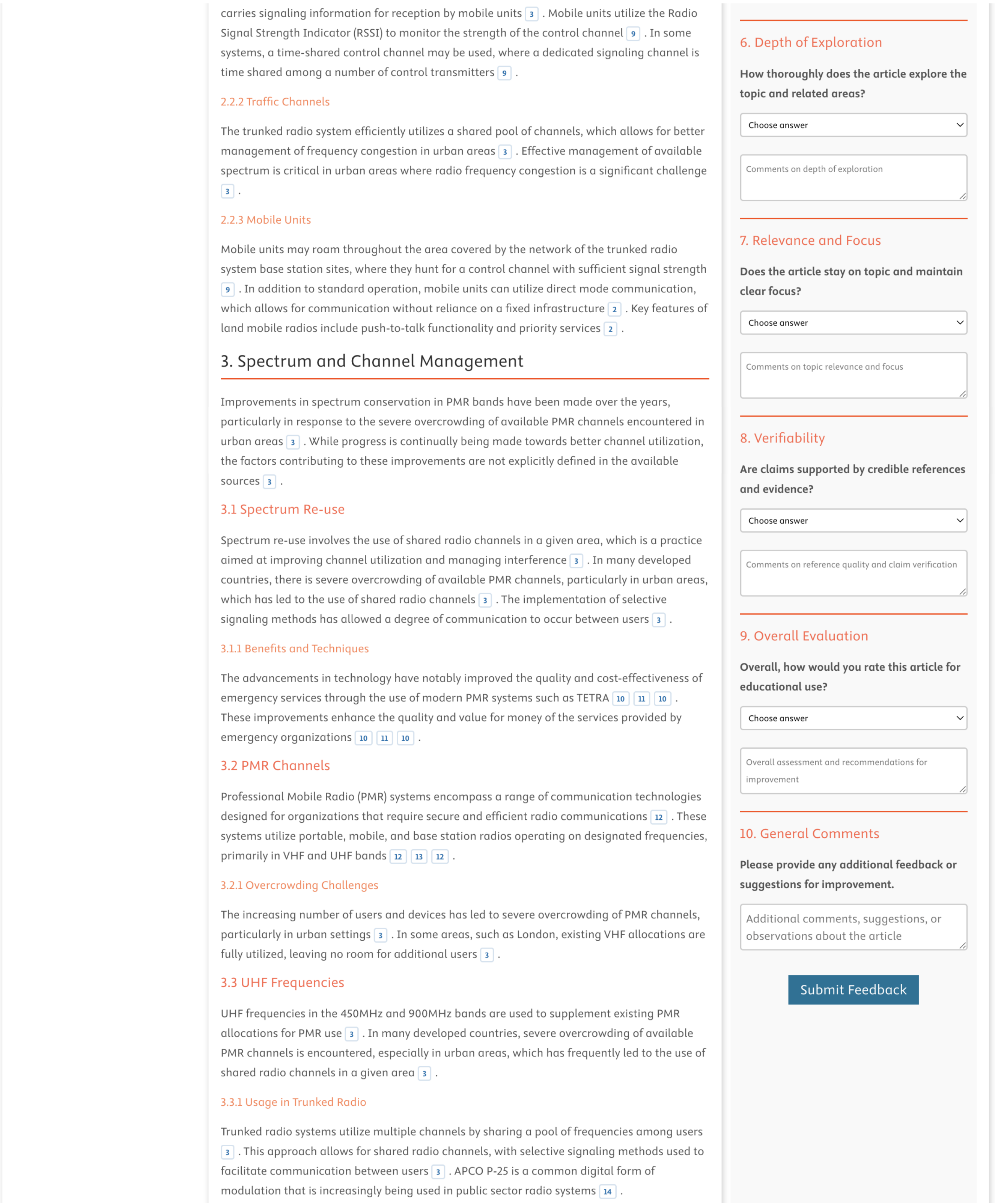
5. Broad Coverage

Does the article provide comprehensive exploration and good coverage of the topic?

Choose answer ⌄

Comments on topic coverage breadth

Figure 3: **Human Evaluation Interface:** Overview of the three-panel layout used for assessing generated topic pages (Part 1). The interface displays a topic from a specific domain with the structured outline (left), generated article content with inline citations (center), and qualitative evaluation metrics (right) using the rubric scores detailed in Appendix F.



carries signaling information for reception by mobile units [3] . Mobile units utilize the Radio Signal Strength Indicator (RSSI) to monitor the strength of the control channel [9] . In some systems, a time-shared control channel may be used, where a dedicated signaling channel is time shared among a number of control transmitters [9] .

2.2.2 Traffic Channels

The trunked radio system efficiently utilizes a shared pool of channels, which allows for better management of frequency congestion in urban areas [3] . Effective management of available spectrum is critical in urban areas where radio frequency congestion is a significant challenge [3] .

2.2.3 Mobile Units

Mobile units may roam throughout the area covered by the network of the trunked radio system base station sites, where they hunt for a control channel with sufficient signal strength [9] . In addition to standard operation, mobile units can utilize direct mode communication, which allows for communication without reliance on a fixed infrastructure [2] . Key features of land mobile radios include push-to-talk functionality and priority services [2] .

3. Spectrum and Channel Management

Improvements in spectrum conservation in PMR bands have been made over the years, particularly in response to the severe overcrowding of available PMR channels encountered in urban areas [3] . While progress is continually being made towards better channel utilization, the factors contributing to these improvements are not explicitly defined in the available sources [3] .

3.1 Spectrum Re-use

Spectrum re-use involves the use of shared radio channels in a given area, which is a practice aimed at improving channel utilization and managing interference [3] . In many developed countries, there is severe overcrowding of available PMR channels, particularly in urban areas, which has led to the use of shared radio channels [3] . The implementation of selective signaling methods has allowed a degree of communication to occur between users [3] .

3.1.1 Benefits and Techniques

The advancements in technology have notably improved the quality and cost-effectiveness of emergency services through the use of modern PMR systems such as TETRA [10] [11] [10] . These improvements enhance the quality and value for money of the services provided by emergency organizations [10] [11] [10] .

3.2 PMR Channels

Professional Mobile Radio (PMR) systems encompass a range of communication technologies designed for organizations that require secure and efficient radio communications [12] . These systems utilize portable, mobile, and base station radios operating on designated frequencies, primarily in VHF and UHF bands [12] [13] [12] .

3.2.1 Overcrowding Challenges

The increasing number of users and devices has led to severe overcrowding of PMR channels, particularly in urban settings [3] . In some areas, such as London, existing VHF allocations are fully utilized, leaving no room for additional users [3] .

3.3 UHF Frequencies

UHF frequencies in the 450MHz and 900MHz bands are used to supplement existing PMR allocations for PMR use [3] . In many developed countries, severe overcrowding of available PMR channels is encountered, especially in urban areas, which has frequently led to the use of shared radio channels in a given area [3] .

3.3.1 Usage in Trunked Radio

Trunked radio systems utilize multiple channels by sharing a pool of frequencies among users [3] . This approach allows for shared radio channels, with selective signaling methods used to facilitate communication between users [3] . APCO P-25 is a common digital form of modulation that is increasingly being used in public sector radio systems [14] .

6. Depth of Exploration

How thoroughly does the article explore the topic and related areas?

Choose answer

Comments on depth of exploration

7. Relevance and Focus

Does the article stay on topic and maintain clear focus?

Choose answer

Comments on topic relevance and focus

8. Verifiability

Are claims supported by credible references and evidence?

Choose answer

Comments on reference quality and claim verification

9. Overall Evaluation

Overall, how would you rate this article for educational use?

Choose answer

Overall assessment and recommendations for improvement

10. General Comments

Please provide any additional feedback or suggestions for improvement.

Additional comments, suggestions, or observations about the article

Figure 4: **Human Evaluation Interface:** Overview of the three-panel layout used for assessing generated topic pages (Part 2). This page has been cropped; the actual topic page continues, showing a list of references at the bottom of the page similar to a Wikipedia style article. Consistent with our factuality evaluation metric (Section 4.6), each in-line citation is clickable, allowing human evaluators to access and review the actual snippet content that was used to support each referenced claim in the topic page.

Bibliography

[1] Lutz Bornmann and Rüdiger Mutz. "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references". In: *Journal of the Association for Information Science and Technology* 66.11 (2015), pp. 2215–2222.

[2] Peder Olesen Larsen and Markus von Ins. "The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index". In: *Scientometrics* 84.3 (2010), pp. 575–603. DOI: 10.1007/s11192-010-0202-z.

[3] Artemis Capari et al. "ScienceDirect Topic Pages: A Knowledge Base of Scientific Concepts Across Various Science Domains". en. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Washington DC USA: ACM, July 2024, pp. 2976–2980. ISBN: 979-8-4007-0431-4. DOI: 10.1145/3626772.3661353. URL: https://dl.acm.org/doi/10.1145/3626772.3661353 (visited on 02/07/2025).

[4] *ScienceDirect Topics pages — Elsevier*. en-US. URL: https://www.elsevier.com/products/sciencedirect/topics (visited on 06/22/2025).

[5] John Sweller. "Cognitive load during problem solving: Effects on learning". In: *Cognitive science* 12.2 (1988), pp. 257–285.

[6] Hiller A Spires et al. "Exploring the collaborative synthesis of information during online reading". In: *Computers & Education* 137 (2019), pp. 146–157.

[7] Mingyang Cao et al. "Attention switching through text dissimilarity: a cognition research on fragmented reading behavior". In: *Frontiers in Human Neuroscience* 18 (2024), p. 1402746.

[8] Aaron Halfaker et al. "The rise and decline of an open collaboration system: How Wikipedia's reaction to popularity is causing its decline". In: *American behavioral scientist* 57.5 (2013), pp. 664–688.

[9] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[10] Iz Beltagy, Kyle Lo, and Arman Cohan. "SciBERT: A pretrained language model for scientific text". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 3615–3620. DOI: 10.18653/v1/D19-1371.

[11] Yijia Shao et al. *Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models*. Apr. 8, 2024. DOI: 10.48550/arXiv.2402.14207. arXiv: 2402.14207[cs]. URL: http://arxiv.org/abs/2402.14207 (visited on 12/20/2024).

[12] Ziwei Ji et al. "Survey of hallucination in natural language generation". In: *ACM Computing Surveys* 55.12 (2023), pp. 1–38.

[13] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive NLP tasks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474.

[14] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. “Why we need new evaluation metrics for NLG”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017), pp. 2241–2252.

[15] Lianmin Zheng et al. “Judging LLM-as-a-judge with MT-bench and chatbot arena”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[16] Yinheng Liu, Yixuan Zhou, Shengchao Hu, et al. “LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods”. In: *arXiv preprint arXiv:2412.05579* (2024).

[17] Dawei Wang, Yifan Dong, Yufeng Li, et al. “From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge”. In: *arXiv preprint arXiv:2411.16594* (2024).

[18] Yupeng Chang et al. “A survey on evaluation of large language models”. In: *ACM Transactions on Intelligent Systems and Technology* (2023).

[19] Mingmeng Geng et al. “The Impact of Large Language Models in Academia: from Writing to Speaking”. In: *arXiv preprint arXiv:2409.13686* (2024). Available at: https://arxiv.org/abs/2409.13686.

[20] Natalie Cooper et al. “Harnessing large language models for coding, teaching and inclusion to empower research in ecology and evolution”. In: *Methods in Ecology and Evolution* (2024). Available at: https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.14325.

[21] Various Liu. “Understanding Context window and Retrieval-Augmented Generation (RAG) in Large Language Models”. In: *Understanding AI* (2024). Available at: https://www.understandingai.org/p/why-large-language-models-struggle.

[22] IBM Research. “What is a context window?” In: *IBM Think Topics* (2025). Available at: https://www.ibm.com/think/topics/context-window.

[23] Lei Huang et al. “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions”. In: *arXiv preprint arXiv:2311.05232* (2023). Available at: https://arxiv.org/abs/2311.05232.

[24] Taicheng Guo et al. “Large Language Model based Multi-Agents: A Survey of Progress and Challenges”. In: *arXiv preprint arXiv:2402.01680* (2024). Available at: https://arxiv.org/abs/2402.01680.

[25] Rosa Munoz-Luna. “Main Ingredients for Success in L2 Academic Writing: Outlining, Drafting and Proofreading”. en. In: *PLOS ONE* 10.6 (June 2015). Publisher: Public Library of Science, e0128309. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0128309. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0128309 (visited on 06/30/2025).

[26] Various Chen. “Chain of Agents: Large language models collaborating on long-context tasks”. In: *Google Research Blog* (2024). Available at: https://research.google/blog/chain-of-agents-large-language-models-collaborating-on-long-context-tasks/.

[27] Various Hong. “Large language models empowered agent-based modeling and simulation: a survey and perspectives”. In: *Humanities and Social Sciences Communications* (2024). Available at: https://www.nature.com/articles/s41599-024-03611-3.

[28] Zekun Xi et al. *OmniThink: Expanding Knowledge Boundaries in Machine Writing through Thinking*. arXiv:2501.09751 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2501.09751. URL: http://arxiv.org/abs/2501.09751 (visited on 05/03/2025).

[29] Markus J. Buehler. *Accelerating Scientific Discovery with Generative Knowledge Extraction, Graph-Based Representation, and Multimodal Intelligent Graph Reasoning*. arXiv:2403.11996 [cs]. June 2024. DOI: 10.48550/arXiv.2403.11996. URL: http://arxiv.org/abs/2403.11996 (visited on 04/04/2025).

[30] NVIDIA Technical Blog. “Insights, Techniques, and Evaluation for LLM-Driven Knowledge Graphs”. In: (2024). Available at: https://developer.nvidia.com/blog/insights-techniques-and-evaluation-for-llm-driven-knowledge-graphs/.

[31] DataCamp. “Enhancing Large Language Models with Knowledge Graphs”. In: (2025). Available at: https://www.datacamp.com/blog/knowledge-graphs-and-llms.

[32] Data Science Dojo. “Applications of Knowledge Graphs in LLM Applications”. In: (2025). Available at: https://datasciencedojo.com/blog/knowledge-graphs/.

[33] Xinbang Dai et al. *Large Language Models Can Better Understand Knowledge Graphs Than We Thought*. arXiv:2402.11541 [cs] version: 3. June 2024. DOI: 10.48550/arXiv.2402.11541. URL: http://arxiv.org/abs/2402.11541 (visited on 04/21/2025).

[34] Elan Markowitz et al. *KG-LLM-Bench: A Scalable Benchmark for Evaluating LLM Reasoning on Textualized Knowledge Graphs*. arXiv:2504.07087 [cs] version: 1. Apr. 2025. DOI: 10.48550/arXiv.2504.07087. URL: http://arxiv.org/abs/2504.07087 (visited on 04/17/2025).

[35] arXiv. “Knowledge in Triples for LLMs: Enhancing Table QA Accuracy with Semantic Extraction”. In: *arXiv preprint arXiv:2409.14192* (2024).

[36] ScienceDirect. “Knowledge Graphs, Large Language Models, and Hallucinations: An NLP Perspective”. In: (2025). Available at: https://www.sciencedirect.com/science/article/pii/S157082682

[37] Haoyu Han et al. *RAG vs. GraphRAG: A Systematic Evaluation and Key Insights*. arXiv:2502.11371 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2502.11371. URL: http://arxiv.org/abs/2502.11371 (visited on 04/07/2025).

[38] arXiv. “LLM Inference Enhanced by External Knowledge: A Survey”. In: *arXiv preprint arXiv:2505.24377* (2024).

[39] Hao Liu et al. *HopRAG: Multi-Hop Reasoning for Logic-Aware Retrieval-Augmented Generation*. arXiv:2502.12442 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2502.12442. URL: http://arxiv.org/abs/2502.12442 (visited on 04/08/2025).

[40] Xiangrong Zhu et al. *Knowledge Graph-Guided Retrieval Augmented Generation*. arXiv:2502.06864 [cs] version: 1. Feb. 2025. DOI: 10.48550/arXiv.2502.06864. URL: http://arxiv.org/abs/2502.06864 (visited on 03/31/2025).

[41] Yuqi Zhu et al. *LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities*. arXiv:2305.13168 [cs]. Dec. 2024. DOI: 10.48550/arXiv.2305.13168. URL: http://arxiv.org/abs/2305.13168 (visited on 04/25/2025).

[42] Lang Cao, Jimeng Sun, and Adam Cross. *AutoRD: An Automatic and End-to-End System for Rare Disease Knowledge Graph Construction Based on Ontologies-enhanced Large Language Models*. arXiv:2403.00953 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2403.00953. URL: http://arxiv.org/abs/2403.00953 (visited on 04/16/2025).

[43] Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. *Enhancing Knowledge Graph Construction Using Large Language Models*. arXiv:2305.04676 [cs]. May 2023. DOI: 10.48550/arXiv.2305.04676. URL: http://arxiv.org/abs/2305.04676 (visited on 04/17/2025).

[44] Cameron R. Wolfe. “Using LLMs for Evaluation”. In: *Technical Blog* (2024). Available at: https://cameronrwolfe.substack.com/p/llm-as-a-judge.

[45] Seonghyeon Ye, Doyoung Kim, Sungdong Kim, et al. "FLASK: Fine-grained Language Model Evaluation based on Alignment Skill Sets". In: *arXiv preprint arXiv:2307.10928* (2023).

[46] Seungone Kim, Juyoung Suk, Shayne Longpre, et al. "Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 4334–4353.

[47] Seungone Kim et al. *The BiGGen Bench: A Principled Benchmark for Fine-grained Evaluation of Language Models with Language Models*. arXiv:2406.05761 [cs]. June 2024. DOI: 10.48550/arXiv.2406.05761. URL: http://arxiv.org/abs/2406.05761 (visited on 03/23/2025).

[48] Seonghyeon Ye et al. "FLASK: FINE-GRAINED LANGUAGE MODEL EVALUATION BASED ON ALIGNMENT SKILL SETS". en. In: (2024).

[49] Kuang-Huei Lee et al. *A Human-Inspired Reading Agent with Gist Memory of Very Long Contexts*. arXiv:2402.09727 [cs]. July 2024. DOI: 10.48550/arXiv.2402.09727. URL: http://arxiv.org/abs/2402.09727 (visited on 04/05/2025).

[50] Sewon Min et al. "FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation". In: *arXiv preprint arXiv:2305.14251* (2023).

[51] Chris Samarinas et al. "Beyond Factual Accuracy: Evaluating Coverage of Diverse Factual Information in Long-form Text Generation". In: *arXiv preprint arXiv:2501.03545* (2025).

[52] Arjun Panickssery, Samuel R Bowman, and Shi Feng. "LLM evaluators recognize and favor their own generations". In: *arXiv preprint arXiv:2404.13076* (2024).

[53] Fan Gao et al. *Large Language Models on Wikipedia-Style Survey Generation: an Evaluation in NLP Concepts*. arXiv:2308.10410 [cs]. May 2024. DOI: 10.48550/arXiv.2308.10410. URL: http://arxiv.org/abs/2308.10410 (visited on 02/14/2025).

[54] *ORES*. en. URL: https://www.mediawiki.org/wiki/ORES (visited on 06/22/2025).

[55] *Wikipedia:Content assessment*. en. Page Version ID: 1276507661. Feb. 2025. URL: https://en.wikipedia.org/w/index.php?title=Wikipedia:Content_assessment&oldid=1276507661 (visited on 03/07/2025).

[56] *Postmodernism - an overview — ScienceDirect Topics*. URL: https://www.sciencedirect.com/topics/psychology/postmodernism (visited on 06/22/2025).

[57] *Snowflake/snowflake-arctic-embed-m-v2.0 · Hugging Face*. Dec. 2024. URL: https://huggingface.co/Snowflake/snowflake-arctic-embed-m-v2.0 (visited on 03/27/2025).

[58] Sabrina Aquino Myriel David. *A Complete Guide to Filtering in Vector Search - Qdrant*. en. URL: https://qdrant.tech/articles/vector-search-filtering/ (visited on 03/12/2025).

[59] Mark Petticrew and Helen Roberts. *Systematic Reviews in the Social Sciences: A Practical Guide*. Oxford: Blackwell Publishing, 2006, p. 352. ISBN: 1405121106.

[60] David Moher et al. "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement". In: *PLOS Medicine* 6.6 (2009), e1000097. DOI: 10.1371/journal.pmed.1000097.

[61] Reijo Savolainen. "Berrypicking and information foraging: Comparison of two theoretical frameworks for studying exploratory search". In: *Journal of Information Science* 44.5 (2018), pp. 580–593.

[62] Gary Marchionini. “Exploratory Search: From Finding to Understanding”. In: *Communications of the ACM* 49.4 (2006), pp. 41–46.

[63] Samuel Schmidgall et al. *Agent Laboratory: Using LLM Agents as Research Assistants*. arXiv:2501.04227 [cs]. Jan. 2025. DOI: 10.48550/arXiv.2501.04227. URL: http://arxiv.org/abs/2501.04227 (visited on 03/13/2025).

[64] Yucheng Jiang et al. *Into the Unknown Unknowns: Engaged Human Learning through Participation in Language Model Agent Conversations*. Oct. 17, 2024. DOI: 10.48550/arXiv.2408.15232. arXiv: 2408.15232[cs]. URL: http://arxiv.org/abs/2408.15232 (visited on 01/07/2025).

[65] OpenAI. “GPT-4: OpenAI’s Multimodal Model for Complex Reasoning Tasks”. In: *arXiv* (2023). URL: https://arxiv.org/abs/2303.08774.

[66] A. Author and B. Author. “State-of-the-Art Language Models: A Review of Recent Advances”. In: *Journal of Artificial Intelligence* 12 (2024), pp. 1–34. DOI: 10.1234/jaai.2024.0123456.

[67] mrbullwinkle. *Azure OpenAI in Azure AI Foundry Models - Azure OpenAI*. en-us. URL: https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models (visited on 06/23/2025).

[68] Anthropic. *Claude 3.5 Sonnet*. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2024. 2024.

[69] *Anthropic’s Claude in Amazon Bedrock*. en-US. URL: https://aws.amazon.com/bedrock/anthropic/ (visited on 06/22/2025).

[70] Meta AI. *Introducing Llama 3.3*. https://ai.meta.com/blog/llama-3-3/. Accessed: 2024. 2024.

[71] *Meta Llama - Models in Amazon Bedrock - AWS*. en-US. URL: https://aws.amazon.com/bedrock/meta/ (visited on 06/23/2025).

[72] Seungone Kim et al. *Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models*. 2024. arXiv: 2405.01535 [cs.CL].

[73] An Yang et al. *Qwen2.5 Technical Report*. https://arxiv.org/abs/2412.15115. 2024.

[74] OpenAI. *Pricing*. Accessed: 2024. 2024. URL: https://openai.com/pricing.

[75] Qdrant. *Indexing*. Accessed: 2024. 2024. URL: https://qdrant.tech/documentation/concepts/indexing/.

[76] Shilong Li et al. *GraphReader: Building Graph-based Agent to Enhance Long-Context Abilities of Large Language Models*. arXiv:2406.14550 [cs]. Nov. 2024. DOI: 10.48550/arXiv.2406.14550. URL: http://arxiv.org/abs/2406.14550 (visited on 04/06/2025).

[77] Linhao Luo et al. *GFM-RAG: Graph Foundation Model for Retrieval Augmented Generation*. arXiv:2502.01113 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2502.01113. URL: http://arxiv.org/abs/2502.01113 (visited on 03/31/2025).

[78] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004. URL: https://aclanthology.org/W04-1013/.

[79] Feng Nan et al. “Entity-level Factual Consistency of Abstractive Text Summarization”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2727–2733. DOI: 10.18653/v1/2021.eacl-main.235. URL: https://aclanthology.org/2021.eacl-main.235/.

[80] Zirui Guo et al. “LightRAG: Simple and Fast Retrieval-Augmented Generation”. In: *arXiv preprint arXiv:2410.05779* (2024). URL: https://arxiv.org/abs/2410.05779.

[81] Alan Akbik et al. “FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 54–59. URL: https://aclanthology.org/N19-4010.

[82] Seungone Kim et al. *Prometheus 2: An Open-Source Language Model Specialised in Evaluating Other LLMs*. https://github.com/prometheus-eval/prometheus-eval. 2024.

[83] Darren Edge et al. *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. arXiv:2404.16130 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2404.16130. URL: http://arxiv.org/abs/2404.16130 (visited on 04/04/2025).

[84] Belinda Mo et al. *KGGen: Extracting Knowledge Graphs from Plain Text with Language Models*. arXiv:2502.09956 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2502.09956. URL: http://arxiv.org/abs/2502.09956 (visited on 04/21/2025).

[85] Laura Dietz and John Foley. “TREC CAR Y3: Complex Answer Retrieval Overview”. In: *Proceedings of the Text REtrieval Conference (TREC)*. 2019. URL: https://trec-car.cs.unh.edu/trec-car-overview-2019.pdf.

[86] Angela Fan and Claire Gardent. “Generating biographies on Wikipedia: The impact of gender bias on the retrieval-based generation of women biographies”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 8561–8576.

[87] Sukarta Kartawijaya. “Improving Students’ Writing Skill in Writing Paragraph through an Outline Technique”. In: *Curricula: Journal of Teaching and Learning* 3.3 (2018), p. 155. DOI: 10.22216/jcc.2018.v3i3.3429.

[88] Martin J Eppler and Jeanne Mengis. “The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines”. In: *The information society* 20.5 (2004), pp. 325–344.

[89] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint arXiv:1810.04805* (2019).

[90] OpenAI. *GPT-4o mini: Advancing cost-efficient intelligence*. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2024. 2024.

[91] Ehud Reiter. “A structured review of the validity of BLEU”. In: *Computational linguistics* 44.3 (2018), pp. 393–401.

[92] Bruno Latour. *Drawing things together*. MIT Press, 1990, pp. 19–68.

[93] Junnan Li et al. “BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models”. In: *arXiv preprint arXiv:2301.12597* (2023).