PaperMatch: AI-Enhanced Insights for Academic & Media Discovery via Telegram

Alvin Wong Ann Ying
Home Team Science and Technology Agency
Singapore
Alvin_WONG@htx.gov.sg

Bertrand Tan Yu-Jin
Ministry of Digital Development and Institution
Singapore
Bertrand_Tan@mddi.gov.sg

ABSTRACT

In the era of information overload, professionals and researchers often struggle to efficiently access and synthesize relevant insights from both academic literature and current news sources. This paper presents a novel end-to-end solution that bridges academic research and real-world reporting through an intelligent, query-driven knowledge retrieval system. At its core, the platform integrates a centralized knowledge database comprising curated academic papers and news articles from reputable platforms such as Papers with Code, ACM Digital Library (ACM DL), BBC, CNA, CNN, and *The Straits Times*. Users interact with the system via a Telegram chatbot interface, allowing seamless natural language queries. In response, the system retrieves the most relevant documents, applies advanced summarization techniques, and compiles a downloadable PDF report. Each report features structured summaries of the top three academic papers and news articles most closely aligned with the query. While highly effective, the current implementation supports a focused set of five topics: *Cybercrime and Digital Fraud*, *Forensic Science and Criminal Investigation*, *Misinformation and Fake News*, *Organised Crime and Drug Trafficking*, and *Medical Fraud and Malpractice*. This paper demonstrates the platform's value in these domains, highlighting its potential to empower users with concise, actionable knowledge for decision-making and continuous learning. The complete solution is available at: https://github.com/MRAWAY77/PaperMatch

KEYWORDS

AI-powered summarization, Natural Language Processing, Large Language Models, Semantic search, Telegram chatbot, Academic discovery, News summarization, Topic modeling, Vector databases, LLM-as-a-Judge

1 INTRODUCTION

The exponential growth of academic literature, coupled with an increasingly complex and fast-paced media landscape, poses significant challenges for researchers seeking to remain informed and contextually aware. With new findings published daily across diverse disciplines and platforms, traditional methods of literature review and media monitoring have become inadequate. Researchers are frequently overwhelmed not only by the volume of academic content but also by its fragmented distribution and the disconnect between scholarly work and its public interpretation.

This paper introduces **PaperMatch**, an AI-powered platform designed to address these challenges by seamlessly integrating academic research discovery with contextual media analysis. Leveraging state-of-the-art natural language processing (NLP) techniques, large language models (LLMs), and intelligent clustering algorithms, PaperMatch provides a unified environment for users to explore scholarly and media content relevant to their field of interest.

Unlike conventional literature discovery tools that focus solely on academic sources, PaperMatch bridges the gap between peer-reviewed research and reputable news reporting. The platform aggregates publications from leading academic repositories—such as the ACM Digital Library, IEEE Xplore, arXiv, and Papers with Code—and matches them with related news articles from trusted media outlets including BBC, CNA, CNN, and *The Straits Times*. Through a Telegram chatbot interface, users can submit natural language queries and receive a downloadable PDF report summarizing the top three most relevant academic papers and news articles. These summaries, generated using advanced summarization models, are thematically clustered into one of five supported domains: **Cybercrime and Digital Fraud**, **Forensic Science and Criminal Investigation**, **Misinformation and Fake News**, **Organised Crime and Drug Trafficking**, and **Medical Fraud and Malpractice**.

By enabling this dual-source, query-driven exploration, PaperMatch enhances the efficiency of literature review and empowers users to understand how academic topics are represented and interpreted in the broader societal discourse. This integration is particularly valuable in domains where public narratives and misinformation can influence both policy responses and research trajectories.

The following sections detail the system architecture, data pipelines, summarization techniques, and evaluation metrics employed in PaperMatch. We also discuss current limitations and future development plans, including expansion to additional topics and multilingual support.

2 DATASETS

A total of 2,939 samples were scrapped, with 1,639 academic papers and 1,322 news articles. These samples are sourced primarily from reputable platforms such as PaperWithCode, BBC, CNA, CNN, and *The Straits Times*. This dataset is specifically curated to build a robust knowledge database, ensuring the quality and relevance of the content.

To maintain the integrity of the dataset and prevent the inclusion of potentially misleading or biased information, we deliberately excluded sources such as social media posts and blogs. These types of content, which can often spread fake news or personal opinions, are not suitable for a reliable dataset. By focusing exclusively on peer-reviewed academic papers and news articles from trusted platforms, we eliminate the need for extensive content screening and judgment calls, ensuring the accuracy and credibility of the results.

3 SYSTEM ARCHITECTURE

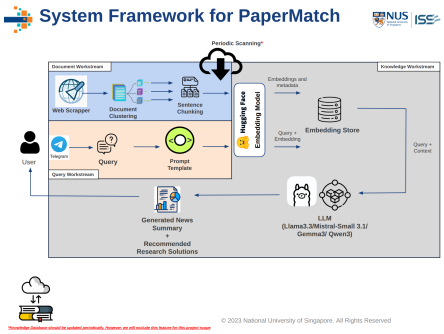

Figure 1: System Framework for PaperMatch

The PaperMatch system operates through a seamless integration of its three core components, as illustrated in Figure 1. The **Document Workstream** functions in the background, continuously scraping, processing, and embedding documents into a central knowledge store. When a user initiates a query through Telegram, the **Query Workstream** formats and vectorises the question, retrieves relevant documents from the **Embedding Store**, and passes this information to the **Knowledge Workstream**. There, a powerful LLM generates a rich, informed response comprising a news summary and potential research recommendations. This response is delivered back to the user, closing the loop. Although the architecture is designed to support **periodic updates** of the knowledge database for continuous learning, it is noted that this feature is **excluded from the current project scope** and could be denoted as future work.

4 DOCUMENT WORKSTREAM

The Document Workstream is designed to continuously ingest and structure external textual content, forming the foundation of the system's knowledge base. It begins with a **Web Scraper**, which periodically scans open-access platforms to retrieve relevant documents such as news articles, academic publications, and blog posts. These documents are then organised using a **Document Clustering** module that groups them based on semantic similarity, enabling coherent thematic organisation. Following this, the **Sentence Chunking** component segments the clustered documents into smaller units—typically sentences or paragraphs—to facilitate efficient processing. Each chunk is then transformed into a dense vector representation using a **Hugging Face Embedding Model**. These embeddings, along with their corresponding metadata, are stored in an **Embedding Store**, a vector database that enables fast and context-aware information retrieval. The workstream is tailored for both academic and news content, with each sub-stream employing a three-stage pipeline: document collection, semantic analysis through topic modelling, and content clustering. This structured approach enables the system to maintain an up-to-date, searchable, and semantically rich knowledge repository.

4.1 Web Scraper

To automate the collection of research papers on various machine learning security topics, a custom web scraper was developed using Python. The scraper targets the website Papers with Code, which categorizes papers by task. It systematically extracts the latest publications across 17 selected tasks, including *adversarial attack*, *backdoor defense*, and *data poisoning*. The solution is compatible with both Firefox and Chrome browsers, offering flexibility and ease of use across different environments.

Key Components:

(1) **Selenium Automation**:
Selenium was employed with a headless Chrome browser to navigate dynamic content and handle JavaScript-rendered elements. The scraper simulates user interaction, such as clicking the "Paper" and "PDF" buttons to access full-text documents.

(2) **CAPTCHA Handling**:
Due to anti-bot protections on the site, the script pauses and prompts the user to manually solve CAPTCHAs whenever detected. This ensures continuity without violating site terms or triggering access blocks.

(3) **PDF Downloading**:
After identifying the final URL of each paper's PDF (typically hosted on arXiv), the script downloads the file directly using `requests`. Filenames are sanitized and saved in a local `scraped_papers` directory.

(4) **Metadata Logging**:
For each downloaded paper, metadata including the title, PDF URL, source page, and local filename is saved into a structured JSON file. This facilitates downstream processing or referencing.

(5) **Pagination and Deduplication**:
The scraper supports pagination to navigate through multiple pages of papers. It tracks visited paper URLs to avoid redundant downloads and halts if too many consecutive empty pages are encountered.

(6) **Task Slug Iteration**:
The script iterates over a list of predefined "task slugs" (e.g., `"adversarial-defense"`, `"benchmarking"`) and collects up to 100 papers per task, providing flexibility in scope and scale.

4.2 Topic Modeling

Once the datasets were scraped, we applied topic modeling to analyze a collection of research papers and news articles related to five key topics of interest. The goal was to automatically categorize the documents into topics based on the bag of words defined in Annex 9, where the keyword classifier includes multiple categories relevant to the analysis.

Out of a total of 2,961 scraped documents, only 396 academic papers and 533 news articles were considered relevant to this process. The remaining documents were excluded due to irrelevance or insufficient content.

We began by extracting text from PDF documents using `pdfplumber` for standard PDFs and `easyocr` for image-based PDFs. The extracted text was cleaned using `spaCy`, which involved removing

Topic	T1	T2	T3	T4	T5
Academic Papers	163	28	15	153	37
News Articles	158	75	36	92	172

Table 1: Distribution of Documents Across Topics

Note: T1 = Cybercrime and Digital Fraud, T2 = Forensic Science and Criminal Investigation, T3 = Medical Fraud and Malpractice, T4 = Misinformation and Fake News, T5 = Organised Crime and Drug Trafficking

Topic	T1	T2	T3	T4	T5
Academic Papers	22	4	3	19	6
News Articles	20	14	7	17	20

Table 2: Number of Clusters within Topics

Note: T1 = Cybercrime and Digital Fraud, T2 = Forensic Science and Criminal Investigation, T3 = Medical Fraud and Malpractice, T4 = Misinformation and Fake News, T5 = Organised Crime and Drug Trafficking

stop words, punctuation, and non-English words, as well as lemmatizing the text to standardize its format.

For topic modeling, we used **BERTopic**, which combines **UMAP** for dimensionality reduction and **CountVectorizer** for text vectorization. The model was configured to generate topics from the processed text, ensuring that each topic contained at least two documents.

After fitting the model, we generated a summary of the documents that fell under each topic, based on the categories defined in Annex 9. Each topic's top keywords were extracted to help interpret its theme, and the relevant documents were categorized accordingly. This approach enabled us to identify key trends in both academic research and news reporting on the selected topics. The distribution of documents across the five topics is presented in **Table 1**. This approach enabled us to identify key trends in both academic research and news reporting on the selected topics.

4.3 Cluster Embeddings

Building on the pre-processing and topic modeling done previously, the next step involves generating embeddings for the research papers and grouping them into meaningful clusters. This process provides a deeper layer of analysis by converting textual content into high-dimensional vectors, making it possible to evaluate similarities and discover hidden patterns across documents.

For the clustering task, we first focus on processing research papers stored in PDF and TXT formats. The text from each document is extracted, cleaned, and lemmatized using the previously discussed text-cleaning function, which helps standardize the language and remove irrelevant words. Unlike in the topic modeling approach, where raw text was used directly, the focus here is on representing each document by the most frequent and meaningful words, forming a compressed summary of the paper's content.

Once the text has been pre-processed, the **SentenceTransformer** model (`paraphrase-MiniLM-L3-v2`) is used to generate dense embeddings for each document. These embeddings effectively capture the semantic meaning of the paper, transforming the raw content into a vector representation that can be used to assess similarity between documents. This process allows the system to group and compare documents based on their underlying concepts rather than just surface-level keywords. To enhance processing efficiency, the embeddings are generated in batches, enabling the system to handle large datasets quickly and without sacrificing accuracy.

Alongside embedding generation, the system performs a word frequency analysis on the cleaned text. This step identifies the most common words within each document, helping to highlight the key terms that dominate the content. The system then extracts the top 100 most frequent words, providing a snapshot of the most significant topics or concepts within the document. These frequent terms complement the semantic embeddings by offering additional insights into the document's core themes, further aiding in the process of grouping and retrieving similar documents based on their content.

The generated embeddings are then saved, along with the corresponding paper names, in a structured format. This allows for easy retrieval and future analysis, such as clustering, to group documents by their content similarity. The embeddings for each cluster of documents are saved into `.pt` files, ensuring that the resulting data is easy to load and work with. The total number of clusters can be seen in **Table 2**.

4.4 Graph Network

We also performed a cluster-to-cluster matching, focusing on converting similarity scores between academic clusters and news segments into clear, visual graphs. By selecting only the top 20 most similar pairs for each topic, we ensure that the graphs emphasize the most significant connections, eliminating less relevant relationships to avoid unnecessary clutter. Illustrated in Figure 2 , it

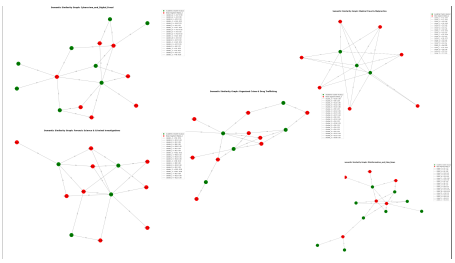

Figure 2: Graph network for top 20 clusters pair

showcases the semantic similarity between academic papers and news segments for various topics, as visualized in a graph network. Among the clusters, the highest similarity scores are observed between **misinformation and fake news** (0.72) and **cyber and digital fraud** (0.58), suggesting a significant overlap in the focus of both academic research and news coverage. This indicates that these issues are not only highly relevant to the current societal concerns but are also drawing substantial attention in both research and media. The relatively high scores for these topics suggest that researchers are actively investigating the growing concerns around

misinformation and cybercrime, which are becoming increasingly intertwined in the digital age.

However, it's important to note that the nature of the academic papers scraped often differs from the issues raised in news reports, even though they share the same general topic. For example, **organised crime and drug trafficking** (0.38), **forensic science and criminal investigation** (0.52), and **medical fraud and malpractice** (0.35) exhibit lower similarity scores, highlighting that while the academic research may cover related topics, the specific issues addressed in the news articles may not always align directly with the research content. This reflects the nuanced nature of academic work, which often investigates broader or different aspects of a problem, whereas news articles tend to focus on more immediate or specific issues.

5 QUERY WORKSTREAM

The Query Workstream handles real-time user interactions, serving as the entry point through which users can pose queries and receive meaningful responses. Users interact with the system via a **Telegram Interface**, which captures natural language questions or prompts. These queries are parsed and structured using a predefined **Prompt Template** to maintain consistency and to optimise compatibility with the downstream language model. The system then transforms the user query into a vector embedding using the same Hugging Face embedding model used in the Document Workstream. This embedding is submitted to the **Embedding Store**, where it is compared against the stored document embeddings to retrieve the most semantically relevant chunks. This retrieval step ensures that the user's query is matched with highly related information, setting the stage for contextual response generation. By leveraging a lightweight and familiar chat interface (Telegram), this workstream facilitates accessible and intuitive interactions for end users while tightly integrating with the system's backend retrieval and summarisation pipeline.

5.1 Telegram API

The system leverages the Telegram API through the `Telethon` library to provide a conversational interface that allows users to submit queries and receive responses directly within a Telegram chat. Once a user sends a message beginning with the `/ask` command, the bot parses the text to extract the query. If no query is included after the command, the system immediately replies with a prompt—"`Please include a question after the /ask command.`"—as illustrated in Figure 3, ensuring early validation of user input.

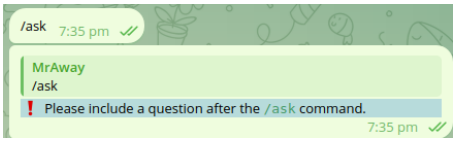

Figure 3: No Query Inserted after the /ask command

When a valid query is detected, the bot captures essential metadata such as the user's ID, chat context, and username for logging and traceability. It then sends an acknowledgment message to confirm receipt of the query and initiates background processing. Asynchronous callbacks are used to send intermediate responses and deliver the final output. A `result_callback` sends textual summaries or updates to the Telegram channel, while a `report_callback` handles the delivery of a compiled PDF report. All messages are routed to a predefined channel specified in the configuration file (`config.TARGET_CHANNEL`). This integration ensures that users can receive both immediate responses and comprehensive outputs in a familiar and easily accessible messaging environment.

5.2 Prompt Engineering

Initially, the raw query will undergo a template step. This is an important preprocessing stage that standardizes the raw user query before it is passed to the embedding model. During this phase, the query is restructured into a more neutral and consistent format that aligns with the expectations of the embedding model. This transformation often involves removing any extraneous details, simplifying complex phrasing, or rewording the query in a way that focuses on its core intent. For example, specific user jargon or colloquial language might be replaced with more formal or general terms. The goal of this step is to reduce ambiguity and increase the likelihood that the embedding model will interpret the query in a way that accurately reflects its intended meaning. By ensuring that the query follows a predictable pattern, the model can better understand and process a wide variety of inputs, making it more effective in classifying and retrieving relevant information.

5.3 Embeddings

The embedding process is a critical component in transforming user queries into fixed-size semantic vectors, making them suitable for various machine learning tasks. Once the query is preprocessed, it is vectorized using the **`all-MiniLM-L6-v2`** model from Sentence-Transformers. This model encodes the query into a dense tensor representation that captures its semantic meaning. This vector representation plays a key role in two primary tasks: semantic topic classification and similarity search. In the case of topic classification, the query embedding is used as a fallback mechanism when keyword-based matching fails, ensuring that the query is appropriately categorized even if exact keywords are not found. Additionally, query embedding is used in similarity search operations, where it is compared to precomputed embeddings of academic and news documents to retrieve the most relevant content.

If the query is determined to have no relevance to the intended topics, the system will classify it as "Unknown." This classification marks the end of the process for that particular query, meaning no further action is taken. The "Unknown" category acts as a safeguard, ensuring that irrelevant queries do not proceed further in the system, as illustrated in Figure 4.

On the other hand, if the query is deemed relevant to one of the five topics managed by **PaperMatch**, the process continues. The system will classify the query into one of these topics, using the semantic meaning captured by the embedding to make an accurate determination. Once classified, the query triggers a similarity search to retrieve the top three most relevant academic papers and news articles related to the identified topic. These selected documents

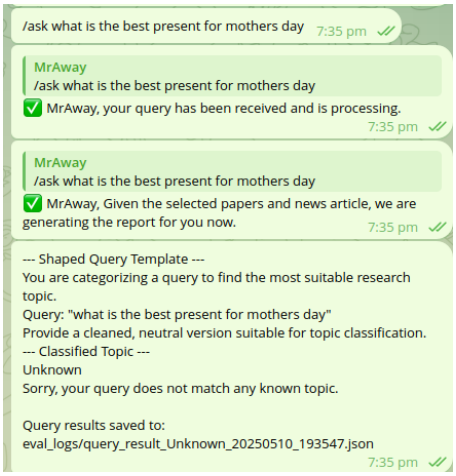

Figure 4: Invalid Query Given by User Input

are then prepared for further processing, ensuring that the system responds with the most pertinent information to the user's query as illustrated in Figure 5.

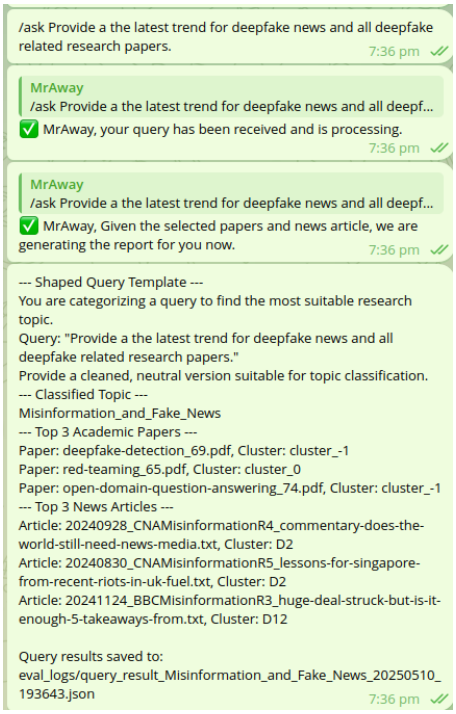

Figure 5: Valid Query Given by User Input

6 KNOWLEDGE WORKSTREAM

The Knowledge Workstream plays a central role in synthesising meaningful responses from retrieved information. At its core is the **Embedding Store**, a specialised database that maintains both the document and query embeddings alongside their metadata. When a user submits a query, the system retrieves the top-matching document chunks from the embedding store and feeds them, along with the original query, into a **Large Language Model (LLM)**. The system supports multiple state-of-the-art LLMs via Ollama framework, such as Llama 3.3, Mistral, and Gemma 3 which are capable of understanding and summarising complex textual data. These models take the query and the contextual documents as input and generate a coherent, concise response tailored to the user's informational needs. The output typically includes a summarised version of the most relevant news or research content, as well as a list of suggested research directions or solutions. This response is then returned to the user via the Telegram interface. This workstream ensures that users receive high-quality, context-aware, and actionable information in near real time.

6.1 Embedding Store

The embedding store is designed with a **domain-specific structure** to support **query-driven document retrieval**, serving as a crucial intermediary between raw document ingestion and large language model (LLM) inference. Each entry in the store includes not only the processed textual content but also **rich metadata** such as the document type (academic or news), cluster classification, title, file path, and original source URL. This metadata enables fine-grained filtering and organization of documents, allowing for intelligent retrieval based on both **semantic content** and **contextual relevance**.

When a user submits a query—such as a request for a summary, a synthesis across articles, or an insight on a specific topic—the system performs a **retrieval operation** over the embedded content. These are bundled with all associated metadata, preserving the provenance and context of the information being passed forward.

The **"content" field** - which comprises cleaned and lemmatized text extracted from scraped PDF or TXT files - is a key component of the retrieval process. It serves as the **contextual input to the LLM**, ensuring that the model's output is grounded in authentic domain-relevant data rather than generated isolated. Leveraging this content, the LLM—be it Llama3.3, Mistral, or Gemma3 running within the **Ollama framework**.

6.2 LLMs Summarization

The summarization workflow is designed to generate concise and academically appropriate output from lengthy and often repetitive source documents. Once the relevant document content is retrieved from the embedding store, it is passed into the LLM using a **strictly defined prompt template**. This prompt imposes formatting rules to ensure that generated summaries exclude rhetorical questions, conversational phrases, or informal recommendations - elements that are inappropriate in academic or news-based contexts. The length of the initial output of the LLM is then evaluated, specifically compared to a limit **of 100 words**. This step is crucial because many of the input documents span multiple pages and naturally produce verbose summaries with redundant or overlapping points. To address this, if the output exceeds the word limit, the model is prompted again, this time using the initial summary as the input. This **iterative refinement loop** allows the system to further distill the content, eliminating duplicate information, and ensuring that

the final summary remains clear, focused and within the desired constraints. This layered approach enhances both the **precision** and **readability** of the generated outputs.

6.3 Report Generation

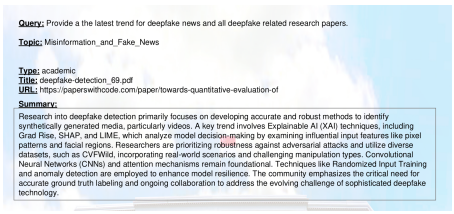

Figure 6: Sample content In PDF Report

After all summaries are generated, they are assembled into a well-structured PDF report that includes the original query, the assigned topic, and concise summaries for each document. Each section of the report displays the document type, title, source URL, and its corresponding summary. To enhance readability, the summaries are placed within bordered text boxes, allowing for easy review of key information. We illustrated a sample section of the report in Figure 6. A custom background image is applied to every page, providing a consistent and polished visual design. The report layout maintains clean formatting and proper spacing for clarity. Once completed, the PDF is saved locally with a timestamped filename and delivered to the user via a Telegram channel through a callback function, ensuring timely and professional dissemination of insights in Figure 7.

Figure 7: Sample content In PDF Report

7 EVALUATION

The project developed a comprehensive pipeline for automated document summarization using Llama3.3, Mistral and Gemma3, processing both academic papers and news articles. Claude 3.7 Sonnet was subsequently deployed for "LLM-as-Judge" purposes [8]. A benchmark set of 100 challenge questions spanning diverse themes—cybercrime, digital fraud, organized crime, forensic science, medical fraud, and misinformation—was designed to stress-test the system's capability to accurately select and summarize relevant documents from the knowledge base while preserving semantic fidelity.

7.1 Implementation

The implementation followed a five-phase methodology:

(1) **Phase I:** Extracted existing summaries from PDFs using pattern matching and structured format recognition.
(2) **Phase II:** Applied fuzzy matching algorithms to associate extracted titles with source filenames, achieving high document identification accuracy.
(3) **Phase III:** Automated file organization into structured directories (`Raw_AP`, `Raw_NA`) for academic papers and news articles.
(4) **Phase IV:** Deployed Claude API with rate limiting and token management for scalable summary generation, incorporating text preprocessing and metadata extraction.
(5) **Phase V:** Performed a comprehensive evaluation using metrics including ROUGE, BERTScore, METEOR, keyword overlap, and content coverage.

Due to constraints in human evaluators, Claude 3.7 Sonnet was used as an automated judge to evaluate summary quality. This "LLM-as-Judge" method is increasingly adopted in AI evaluation literature, offering an objective, scalable, and consistent alternative to manual review [8]. Key benefits include:

- Reduced subjectivity in grading open-ended outputs,
- Consistent scoring across large datasets, and
- Scalability in time-limited scenarios.

Potential bias was mitigated by validating LLM judgments against automated metrics through reference-guided grading. Technical safeguards included incremental saving to prevent data loss, robust error handling, and efficient API usage via intelligent text truncation and batching.

7.2 Evaluation for Topic Clusters

The theme classification system performs well overall with **89%** accuracy, correctly classifying themes for 89 out of 100 queries. The performance can be found in both `fig` (8) and table (3).

The system made 11 errors, primarily in Misinformation and Fake News (7 errors) and Cybercrime and Digital Fraud (high false positives). For the remaining topics, such as **Forensic Science & Criminal Investigation**, **Medical Fraud & Malpractice**, and **Organized Crime & Drug Trafficking**, the model generally performed well.

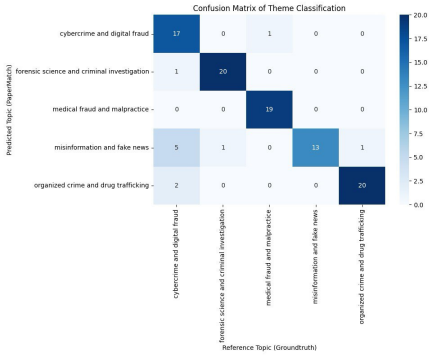

Figure 8: Confusion Matrix for Topic Clusters

Based on the analysis of the classification system, it's clear that the Misinformation and Fake News and Cybercrime and Digital

Theme	Precision	Recall	F1 Score
Cybercrime and Digital Fraud	68.00	94.44	79.07
Forensic Science and Criminal Investigation	95.24	95.24	95.24
Medical Fraud and Malpractice	95.00	100.00	97.44
Misinformation and Fake News	100.00	65.00	78.79
Organized Crime and Drug Trafficking	95.24	90.91	93.02

Table 3: Performance For Topic Clusters

Metric	Mistral (AP)	Gemma3 (AP)	Llama3.3 (AP)	Mistral (NA)	Gemma3 (NA)	Llama3.3 (NA)
ROUGE-1	**0.2195**	0.2017	0.1928	**0.2788**	0.2517	0.2774
ROUGE-2	**0.0309**	0.0260	0.0241	0.0600	0.0525	**0.0605**
ROUGE-L	**0.1974**	0.1822	0.1774	**0.2493**	0.2237	0.2470
BERTScore	**0.8480**	0.8472	0.8302	**0.8745**	0.8725	0.8658
METEOR	**0.2329**	0.2081	0.2059	**0.3165**	0.2654	0.3149
Keyword Overlap	**0.0967**	0.0876	0.0798	**0.1477**	0.1230	0.1287
Content Coverage	**0.2280**	0.2104	0.2068	0.3756	0.2925	**0.3844**
Overall Score	**0.3031**	0.2912	0.2834	**0.3562**	0.3323	0.3522

Table 4: Evaluation Metrics for Summarization

Note: AP = Academic Papers, NA = News Articles

Fraud categories share a high degree of correlation in terms of query content. The difficulty arises from the fact that both topics often involve elements related to manipulation, deception, and digital threats, which can make distinguishing between them a challenge. Let's break down how the system struggles to classify certain queries into the correct theme, as illustrated in the examples below:

Example 1: Misinformation vs. Cybercrime
Query: What research addresses the long-term societal impacts of persistent exposure to misinformation?
Expected Theme: Misinformation and Fake News
Predicted Theme: Cybercrime and Digital Fraud

In this case, the system incorrectly classifies the query as related to Cybercrime and Digital Fraud. However, the topic revolves around the societal impacts of misinformation, which primarily falls under Misinformation and Fake News. The misclassification occurs because misinformation can sometimes overlap with cybercrime when it concerns the spread of false information through digital platforms for malicious purposes.

Example 2: Misinformation and Fake News vs. Cybercrime
Query: What are the latest techniques for source verification in digital journalism?
Expected Theme: Misinformation and Fake News
Predicted Theme: Cybercrime and Digital Fraud

In this case, the query focuses on source verification techniques in digital journalism, which is most closely related to Misinformation and Fake News. However, the system misclassifies it as related to Cybercrime and Digital Fraud. The confusion likely arises because source verification is crucial in combating misinformation and fake news, but the reference to digital journalism and techniques can be associated with cybercrime activities like digital fraud prevention. The model struggles due to the overlap between verifying information in the context of digital platforms and combating fraudulent digital activities.

7.3 Evaluation on Summaries

Table 4 showcases the performance of three summarization models—Mistral, Gemma3, and Llama3.3—on academic papers (AP) and news articles (NA). Mistral consistently achieves the highest scores across nearly all metrics, outperforming both Gemma3 and Llama3.3 in 6 out of 7 metrics for academic papers and 5 out of 7 for news articles.

In academic summarization, Mistral leads in every metric, including ROUGE (1, 2, L), BERTScore, METEOR, Keyword Overlap, and Content Coverage. This results in an overall score of 0.3031, which is 7.0% higher than Llama3.3 and 4.1% higher than Gemma3.

In the news domain, Mistral maintains strong performance, achieving the highest scores in ROUGE-1, ROUGE-L, BERTScore, METEOR, and Keyword Overlap. Although Llama3.3 narrowly surpasses Mistral in ROUGE-2 and Content Coverage, Mistral secures the highest overall score of 0.3562, outperforming Gemma3 by 7.2% and Llama3.3 by 1.1%.

Across both domains, Mistral also attains the best BERTScore indicating superior semantic similarity with the reference summaries (0.8480 for AP, 0.8745 for NA). All models show better performance on news articles than academic papers, reflecting the simpler and more structured language in news content.

7.4 Discussion of Results

The evaluation metrics used in this study, leveraging the "**LLM-as-Judge**" concept, produced relatively low scores across all models. It is important to clarify that these low scores do not necessarily indicate poor performance. Rather, they reflect the inherent nature of the evaluation method, which maps the similarity across all LLM-generated output summaries. Thus, the lower scores imply that the outputs were different from one another, rather than one being inferior to another. In other words, the evaluation focuses on the degree of alignment between the candidate summaries and the reference summaries, rather than directly judging their quality in an absolute sense.

To illustrate, we compare two examples of reference and candidate summaries.

7.4.1 Example 1: Reference Summary (Anthropic Claude API). **Reference Summary:**

> "The Justice Department is intensifying its efforts to combat cybercrime, primarily through proactive measures to mitigate harm to victims and disrupt criminal activity. This strategy involves a shift toward preventing damaging hacking incidents and ransomware attacks. Key components include utilizing investigative techniques, such as tracking digital currency via Chainalysis, establishing a Virtual Asset Exploitation Unit with cryptocurrency experts, and coordinating with the FBI. The department's actions encompass seizing cryptocurrency, arresting cybercriminals, and prosecuting individuals involved in ransomware attacks, like those targeting Colonial Pipeline. These operations aim to dismantle lucrative criminal models and address the significant financial impact of cybercrime."

This summary, generated by the Anthropic Claude API, contains 81 words and is clear, concise, and informative.

Metric	Gemma3 (NA)	Anthropic Claude API (Reference)
ROUGE-1	0.2692	0.2436
ROUGE-2	0.0787	0.2576
ROUGE-L	0.1765	0.3000
BERTScore	0.8600	0.8771
METEOR	0.2576	0.2576
Keyword Overlap	0.1765	0.3000
Content Coverage	0.3000	0.3844

Table 5: Individual Comparison of Summaries

Note: NA = News Articles

7.4.2 Example 2: Candidate Summary (Gemma3 via Ollama). **Candidate Summary:**

> "The Justice Department is shifting its cybercrime approach to prioritize preventing harm to victims, even if it means potentially compromising arrests. Deputy Attorney General Lisa Monaco announced that officials will consider disruptive actions like providing decryption keys to victims and seizing servers used by cybercriminals. The FBI is forming a new cryptocurrency expert team for blockchain analysis and asset seizure. This follows criticism that the FBI previously withheld a decryption key that could have helped hundreds of businesses affected by ransomware."

This candidate summary, generated by `Gemma3` using `Ollama`, contains 99 words. Although it differs in structure and wording, it conveys similar information to the reference summary, achieving over 86% semantic similarity as indicated by the BERTScore.

7.4.3 Evaluation Metrics: Based on the two examples, the evaluation score is shown below in **Table 5**.

8 LIMITATIONS

While PaperMatch demonstrates strong potential in facilitating summarization and knowledge extraction, several limitations remain:

- **Lack of Real-Time Updates**: PaperMatch does not support constant live updates or real-time scraping from various platforms. This restricts its ability to ensure the most up-to-date information is always reflected in the knowledge workstream.
- **Human Annotation Challenges**: The process of manually labeling summaries is labor-intensive and inherently subjective. Different annotators tend to focus on varying aspects of a document, particularly when tasked with compressing a 10-page report into a concise 100-word summary. Similarly, LLMs may also generate inconsistent summaries due to differences in interpretation and emphasis.
- **Limited Source Coverage**: Currently, PaperMatch does not include data scraping from live blogs, social media, or informal user-generated content. These sources often contain subjective or unverified information. Moreover, there is no standardized methodology in place to assign appropriate weight to published, peer-reviewed material versus non-published content (e.g., comments, posts, or Medium articles), which lack formal editorial oversight.
- **Text-Only Query Support**: The solution presently supports only textual queries and does not handle visual inputs. Extending support to pictorial or multimodal queries would require the integration of Vision-Language Models (VLMs) such as LLaVA or GPT-4V, which is beyond the current scope of this work.

9 FUTURE WORK

In light of the limitations identified, several avenues for future development of the PaperMatch system are proposed. Firstly, one major constraint is the system's inability to handle continuous live updates and data scraping from various platforms. To address this, future iterations could incorporate automated web crawlers and scheduled scrapers that periodically collect data from academic databases, news outlets, and authoritative government sources. These could be integrated with APIs (such as those provided by Google Scholar or PubMed) and scheduled through tools like Azure or AWS Lambda to ensure up-to-date knowledge ingestion [11], [12]. Secondly, the current reliance on human-generated summaries introduces inconsistencies and inefficiencies. Summarizing large documents, especially those exceeding ten pages, into concise 100-word summaries is a highly subjective and laborious task. Future work can explore scalable solutions such as active learning loops, where the model identifies ambiguous or high-uncertainty summaries for human verification [15], or weak supervision frameworks like Snorkel to generate approximate labels using heuristic rules [13]. Additionally, introducing multi-annotator pipelines with inter-annotator agreement metrics can improve labeling consistency across summarization tasks [16].

Another limitation is the exclusion of content from live blogs, social media, and informal online articles. These sources, while potentially rich in real-time insights, often lack verifiable references and editorial oversight. To mitigate the risk of misinformation while enhancing data richness, future implementations could incorporate a credibility scoring mechanism that evaluates source reliability based on author identity, publication domain, engagement signals, and alignment with verified content [17]. Natural Language Inference (NLI) models could also be employed to assess the factual consistency between informal and formal sources before inclusion [17]. Furthermore, PaperMatch currently focuses exclusively on text-based queries and cannot handle pictorial or multimodal inputs. To bridge this gap, future versions may integrate vision-language models (VLMs) such as GPT-4V, BLIP-2, or LLaVA, which can interpret and summarize visual content such as graphs, tables, or scanned documents [18]. These models, coupled with OCR tools and layout parsers, can extract and embed information from images to enable meaningful cross-modal retrieval and summarization.

Lastly, to maintain an up-to-date and structured internal representation of knowledge, the development of a dynamic knowledge graph is recommended. This graph can be continuously enriched and updated through entity-relation extraction from newly ingested documents, ensuring better traceability and understanding of semantic relationships. Background syncing pipelines, possibly orchestrated using Apache Airflow, can be used to monitor changes, update embeddings, and refresh graph nodes efficiently [14]. By combining these proposed advancements, PaperMatch can evolve

into a more intelligent, multimodal, and context-aware knowledge retrieval and summarization system capable of meeting the demands of real-world applications.

REFERENCES

[1] Bawden, D., & Robinson, L. (2009). The dark side of information: Overload, anxiety and other paradoxes and pathologies. *Journal of Information Science*, 35(2), 180–191.

[2] Entman, R. M. (1993). Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, 43(4), 51–58.

[3] Aggarwal, C. C., & Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. CRC Press.

[4] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[5] Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060), 1226–1227.

[6] See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1073–1083).

[7] Tenopir, C., Dalton, E., Fish, A., & Dorsett, K. (2012). Scholarly article seeking, reading, and use: A continuing evolution from print to electronic. *Journal of the Association for Information Science and Technology*, 63(11), 2236–2247.

[8] Kannappan, G. (2024, December 2). LLM-as-a-Judge: Unveiling its potential and applications. *Medium*. https://medium.com/@ganeshkannappan/llm-as-a-judge-unveiling-its-potential-and-applications-cbfb3db14e26

[9] Ollama. (n.d.). *Ollama search*. Retrieved May 11, 2025, from https://ollama.com/search

[10] Papers with Code. (n.d.). *Papers with code*. Retrieved May 11, 2025, from https://paperswithcode.com/

[11] Amazon Web Services. (n.d.). *AWS Whitepapers & Guides*. Retrieved May 11, 2025, from https://aws.amazon.com/whitepapers/?whitepapers-main.sort-by=item.additionalFields.sortDate&whitepapers-main.sort-order=desc&awsf.whitepapers-content-type=*all&awsf.whitepapers-global-methodology=*all&awsf.whitepapers-tech-category=*all&awsf.whitepapers-industries=*all&awsf.whitepapers-business-category=*all

[12] Microsoft Azure. (n.d.). *Research & Insights*. Retrieved May 11, 2025, from https://azure.microsoft.com/en-us/resources/research

[13] Snorkel AI. (n.d.). *Snorkel: Programmatically Build and Manage Training Data*. Retrieved May 11, 2025, from https://snorkel.ai/

[14] The Apache Software Foundation. (n.d.). *Apache Airflow*. Retrieved May 11, 2025, from https://airflow.apache.org/

[15] Scaling of Search and Learning: A Roadmap to Reproduce o1 from Reinforcement Learning Perspective. (2024). *arXiv*. Retrieved May 11, 2025, from https://arxiv.org/html/2412.14135v1

[16] Measuring Annotator Agreement Generally across Complex Structured, Multi-object, and Free-text Annotation Tasks. (2022, December 15). *arXiv*. Retrieved May 11, 2025, from https://arxiv.org/pdf/2212.09503

[17] Thibault, C., Peloquin-Skulski, G., Tian, J.-J., Laflamme, F., Guan, Y., Rabbany, R., Godbout, J.-F., & Pelrine, K. (2024, November 7). A Guide to Misinformation Detection Datasets. *arXiv*. Retrieved May 11, 2025, from https://arxiv.org/html/2411.05060v1

[18] Masry, A., & Rodriguez, J. A. (2025, February 2). AlignVLM: Bridging Vision and Language Latent Spaces for Multimodal Understanding. *arXiv*. Retrieved May 11, 2025, from https://arxiv.org/html/2502.01341v1

ANNEX: KEYWORD CLASSIFIER CATEGORIES

Topic Category	Associated Keywords
Medical Fraud and Malpractice	medical fraud, healthcare fraud, insurance fraud, billing fraud, medicare fraud, medicaid fraud, upcoding, phantom billing, patient brokering, kickback scheme, medical identity theft, prescription fraud, fraudulent diagnosis, unnecessary procedure, health insurance fraud, pharmaceutical fraud, malpractice, unethical treatment, experimental treatment, informed consent violation, patient exploitation, medical negligence, falsified credentials, medical license fraud, counterfeit medicine, unlicensed practice, medical data manipulation, clinical trial fraud
Misinformation and Fake News	fake news, disinformation, misinformation, propaganda, conspiracy theory, information warfare, social media manipulation, deepfake, fact-checking, media literacy, information disorder, filter bubble, echo chamber, algorithmic bias, synthetic media, information operations, coordinated inauthentic behavior, influence operation, astroturfing, computational propaganda, source verification, media bias, journalistic integrity, information source, primary source, secondary source, citation needed, unverified claim, anonymous source, information provenance, source attribution, fact versus opinion, source criticism
Organised Crime and Drug Trafficking	organized crime, organised crime, criminal syndicate, mafia, crime network, criminal organization, criminal organisation, mob, racketeering, criminal enterprise, illegal operation, crime family, criminal group, underworld, yakuza, triads, criminal clan, criminal conspiracy, crime syndicate, drug trafficking, drug trade, narcotics trade, illegal drug, drug smuggling, cocaine trade, heroin distribution, methamphetamine, drug cartel, drug syndicate, drug network, illicit drug, drug smuggler, controlled substance, drug ring, international drug trade, drug bust, narcotics trafficking, illegal drug trade, narcotic distribution, heroin trafficking, methamphetamine trade, drug distribution, illegal drug network, drug interdiction, drug supply chain
Cybercrime and Digital Fraud	cyber attack, malware, ransomware, phishing, data breach, hacking, cybercrime, cyber security, cyber criminal, dark web, cyber fraud, identity theft, cyber espionage, botnet, DDoS attack, cyber warfare, computer virus, data theft, online fraud, cryptocurrency crime, zero-day exploit, social engineering, encryption, keylogger, backdoor, brute force attack, SQL injection, man-in-the-middle, password cracking, spyware, trojan horse, rootkit, cryptojacking, extortion
Forensic Science and Criminal Investigation	DNA analysis, fingerprint analysis, ballistics, toxicology, forensic pathology, crime scene investigation, digital forensics, blood pattern analysis, forensic anthropology, trace evidence, forensic entomology, forensic psychology, autopsy, serology, chain of custody, forensic odontology, chromatography, spectroscopy, PCR amplification, mass spectrometry, microscopy, luminol test, substance identification, comparative analysis, facial reconstruction, voice analysis, handwriting analysis, geographic profiling