El Gemayel, C., El Gemayel, J., & Constantin, J. (2026). A distributed RAG-based framework for automated extraction of information from multiple types of resources. In 2025 IEEE/ACS 22nd International Conference on Computer Systems and Applications (AICCSA) (2025 IEEE/ACS 22nd International Conference on Computer Systems and Applications (AICCSA)). IEEE. For the purpose of open access, a CC BY 4.0 licence has been applied.

A Distributed RAG-based Framework for Automated Extraction of Information from Multiple Types of Resources

1st Charbel El Gemayel
TICKET Lab., Technology in Computer Sciences department, Antonine University
Hadat-Baabda, Lebanon
charbel.gemayel@ua.edu.lb

2nd Joseph El Gemayel
Computer & Information Sciences University of Strathclyde
Glasgow, United Kingdom
joseph.el-gemayel@strath.ac.uk

3rd Joseph Constantin
LaRRIS Laboratory Lebanese University
Fanar, Lebanon
cjoseph@ul.edu.lb

***Abstract*—Accessing authoritative information in areas such as healthcare, cybersecurity, and artificial intelligence remains a challenge due to the heterogeneity of data sources and the varying credibility of content. With the increasing integration of advanced technologies into daily life, there is an urgent need for systems that can streamline the retrieval of information and extraction of knowledge from different formats. In this paper, we present a distributed, Retrieval-Augmented Generation (RAG) based framework that aims to automate the extraction and structuring of information from multimodal resources, such as websites, PDFs, images, audio, and video. The framework supports real-time data processing and is optimized for the creation of open data sets in any subject area. To validate our approach, we applied it to cigars and beverages, using content from online articles, reviews, and posts. Our results show the framework's potential to simplify data integration, improve usability and enable scalable, contextual knowledge generation.**

***Index Terms*—Large Language Models, RAG Model, Artificial Intelligence, Multi-modal Data Integration, Distributed Systems.**

I. Introduction

In today's world, technology seems to have found its way into every industry. Its impact in healthcare, retail, logistics, manufacturing, and every other area where it has made inroads is nothing short of astounding. With the immense success of integrating advanced processes in other industries, it's important that we start thinking of innovative ways to use the technologies at our disposal to enhance people's experiences in social, leisure, and entertainment settings. Retrieval Augmented Generation (RAG) [4] is the process of giving an existing Large Language Model (LLM) [1]–[3] the ability to generate responses based on information that is not part of its training data. The ability to customize an LLM to your needs can be beneficial for many reasons:

- Generate accurate and credible domain-specific responses: Because RAG uses an LLM to retrieve data from a specific dataset, you are guaranteed to get an accurate response for each input that relates to the information in the data sources. RAG also lists the specific sources that inspired its response.
- Improve usability: RAG allows users to interact with data in a natural way, using voice-based text, just as you would talk to an expert.
- Context-specific generation: Since the LLM retrieves the most important documents by relevance, its responses are guaranteed to be context-specific, improving the accuracy of the response.

A RAG system consists of three main components:

- Embedding Model: used to create vector embeddings.
- LLM: used to understand the user's input and translate the embeddings into meaningful text responses
- Vector store: is used to store and query the embeddings. With a vector store, a retriever is used to query the embeddings.

In this context, the current study presents a comprehensive RAG-based framework designed to facilitate the integration and use of data from different sources. A user-friendly interface is provided, that offers intuitive controls for data input, processing, and output generation. To ensure optimal performance, the platform supports real-time data processing and seamless interaction with structured and unstructured data sources. Although our proposed framework shares fundamental features with recent models such as MMKB-RAG [5], VisRAG [6], HM-RAG [7], AlzheimerRAG [8], and RAMQA [9], it introduces a distinctive and highly practical direction that clearly distinguishes it from them. Similar to these systems, our framework supports multimodality by integrating different types of data - including websites, PDFs, images, audio, and video — and utilizes Retrieval Augmented Generation (RAG) techniques to improve the precision and relevance of retrieved information. It also meets the overarching goal of standardizing and simplifying the management of heterogeneous resources through a cohesive and efficient process. However, unlike existing approaches that primarily target specific tasks such as question answering, biomedical search or the creation of structured knowledge graphs, our framework

This project has been funded with support from the TICKET Laboratory at the Antonine University.

is uniquely designed to facilitate the global, flexible creation of open-domain datasets. It is not limited to a specific sector or narrowly defined use case, but allows users to independently collect, organize, and structure information on virtually any topic. This universality, combined with the system's ability to process complex, data from multiple sources, makes our framework as a novel and valuable advance over the current state of the art. Each feature in Table I plays a crucial role in improving the performance and applicability of a framework. Multimodal data processing is fundamental to enable the integration of different types of data — text, images, audio, and video — which is essential for tackling real-world, complex problems involving multiple forms of data [10]. The use of RAG architecture is equally important, as it allows frameworks to retrieve external, relevant information and augment the generative process, improving the quality and relevance of []. Unified processing of heterogeneous sources is essential to ensure that data from different formats can be seamlessly integrated, processed, and used without major obstacles [12]. This is particularly important in applications where multiple data sources need to work together to gain a comprehensive understanding. Applicability in open domains ensures that the framework is not limited to specific domains, making it more versatile and capable of solving a wide range of problems in different industries [13]. Autonomous dataset creation is a key feature that enables the framework to automatically create valuable, domain-specific datasets, which is essential for creating large-scale datasets without manual intervention [14]. The distributed and scalable architecture is critical for processing large datasets and parallel processing, which is crucial for large-scale real-time, applications [15]. Application independence further increases the flexibility of the framework , as it can be used in different domains without the need for reconfiguration [16]. Finally, in today's connected world where information is constantly updated, gathering information from the internet and media is becoming increasingly important. The ability to autonomously collect and organize data from various online resources can provide real-time insights [17]. We present a distributed architecture for processing multimodal data using advanced techniques such as RAG. The rest of the paper is organized as follows. Section II describes the related work of recent existing solutions. It gives a general overview of successful research techniques, and algorithms that have already been used. Section III illustrates our proposed framework that helps researchers optimize their time by streamlining the processes of data integration and retrieval. In this section, the hardware and software components of our proposed solution are presented. Experiments are conducted in Section IV, where some drawbacks of the system are highlighted. We conclude our paper in Section V.

II. Background Study

Rapidly advancing language models, such as GPT-4 [18], [19], LLaMA [20], Google's PaLM [21], and Anthropic's Claude [22], Galactica [23], have revolutionized natural language processing (NLP) and artificial intelligence (AI). These models are able to generate human-like text, understand complicated linguistic nuances, and perform a wide range of tasks, from machine translation to text summarization and sentiment analysis. Their applications span numerous fields, including healthcare, legal analysis, customer service, and the creative industries, where they are increasingly being used to automate complex processes and improve decision-making [24]–[26]. The RAG framework consists of core components that are required for its operation. A RAG system relies on three key elements:

- The first element is an embedding model for converting text into embeddings [30], an OpenAI's text-embedding-3-small with 1536 dimensions due to its accuracy and cost-effectiveness. Due to its popularity and cost-effectiveness, it uses ChatGPT 3.5 – Turbo from OpenAI.
- The second element is a vector store to store and retrieve these embeddings, and a Large Language Model (LLM) to interpret user input and 'translate the embeddings into coherent and relevant responses. It is a DataStax Astra DB as our vector store, as it is a paid Cloud Database as a Service (DBaaS) with free starting credits. It is important to know that Astra is based on the Apache Cassandra [31] database, an open source NoSQL database management system.
- The third element is a Large Language Model to understand the user input and translate the embeddings into meaningful responses. To link these components together, we use Python as the programming language and utilize the LangChain library due to its relative ease of use and extensively documented methods.

Building on these findings, this section explores the theoretical background of this study, focusing on the limitations of traditional knowledge-based systems and the need for more dynamic, transparent approaches in domains where real-time information is essential. Traditional answer generation systems, such as GPT, BERT [27], BART [28] , and T5 [29], are typically trained on large static datasets and operate with a fixed knowledge base, meaning that they can only provide responses based on the data available at the time of their last update. Once these systems are trained, they can no longer take new developments or data into account unless they undergo a complete retraining process. As a result, their ability to respond accurately and reliably decreases as new information becomes available. This limitation is particularly critical in areas such as healthcare, legal analysis, customer service, and technical support, where information is rapidly evolving. In healthcare, for example, new medical research, new treatment protocols, and drug efficacy updates require systems that can continuously incorporate the latest information. Similarly, in the legal and customer service fields, changes in laws, regulations, and customer preferences require up-to-date responses. Relying on outdated or static information in these areas can lead to errors or missed opportunities for timely interventions. In addition, traditional systems often lack

TABLE I
COMPARISON OF KEY FEATURES AND UNIQUENESS OF OUR FRAMEWORK

Feature	MMKB-RAG	VisRAG	HM-RAG	AlzheimerRAG	RAMQA	Our Solution
Multi-modal data handling	X	X	X	X	X	X
Use of RAG architecture	X	X	X	X	X	X
Unified processing of heterogeneous sources	Partial	Partial	Partial	Domain-specific	Focused (QA)	X
Open-domain applicability	NA	NA	NA	NA	NA	X
Autonomous dataset construction	NA	NA	Partial	Domain-specific	Focused (QA)	X
Distributed and scalable architecture	NA	NA	X	NA	NA	X
Application independence	NA	NA	NA	Domain-specific	Limited to QA	X
Designed for data curation	NA	NA	NA	Partial	Focused (QA)	X
Globalized information gathering from web and media	NA	NA	NA	NA	NA	X

transparency, making it difficult to understand the reasons for a response, which is particularly problematic in areas where accountability, explainability, and compliance are paramount. Given these challenges, there is a growing need for systems that not only keep pace with evolving information, but also provide transparency and clarity in their decision-making processes. This study proposes a new framework to address existing gaps by enabling researchers to autonomously retrieve and integrate real-time information from external sources. This approach combines answer generation with the ability to access and integrate the most up-to-date data available, to ensure that responses are both accurate and contextually relevant. In addition, the proposed framework places a strong emphasis on comprehensibility, allowing users to understand how responses are generated and ensuring that decisions can be justified and validated, which is critical for building trust and ensuring compliance in fast-moving industries.

III. PROPOSED PLATFORM APPROACH

Our framework, presented in Figure 1, is the architecture of a high-performance distributed system designed for large-scale scraping, processing and analysis of image, text or voice memos. The system uses a service architecture to ensure scalability, resilience, and maintainability, with services communicating via message queues and fast APIs. The architecture includes comprehensive monitoring with Prometheus [32] and uses Kubernetes for orchestration across distributed nodes. The framework is designed to ensure high availability through:

- Multi-zone deployment: The framework is designed so that it can be deployed across multiple availability zones. This ensures that , the system can continue to work seamlessly from other zones, even if there are problems in one zone in order to avoid service interruptions.
- service redundancy: By providing redundant instances of critical services across different nodes or regions, the proposed solution ensures that if one instance fails, another can take over without any noticeable interruption to the user. This redundancy increases fault tolerance and ensures the continuity of services.
- Database replication: The proposed solution can use database replication, where data is mirrored in real time across multiple database servers.
- Load balancing: Load balancers distribute incoming traffic evenly across multiple servers or instances, preventing a single server from being overloaded.
- Circuit Breaker: If a service repeatedly fails, the Circuit Breaker temporarily stops requests to that service, allowing the system to recover and maintain overall stability.
- retry mechanisms: In the event of temporary failures, the proposed solution uses retry mechanisms to automatically retry failed operations. This ensures that temporary problems do not lead to a permanent failure of processes, which increases reliability.
- Graceful Degradation policies: The proposed solution is designed to degrade gracefully, i.e. it continues to operate with limited functionality rather than crashing completely. For example, if a video processing service is no longer available, the system can continue to process other types of data without a complete system failure.

The system is divided into primary microservices, each of which has different responsibilities:

A. Scraper Service

The Scraper Service is the entry point for capturing various types of media content, including images, text, and voice data, using Scrapy and custom extractors to crawl and process content from target websites. The workflow includes:

- Receives target URLs and extraction instructions from the API service via RabbitMQ.
- Distributes crawling tasks to worker nodes for scalability and parallel processing
- Identifies and retrieves available content types — such as images, text blocks, and audio snippets — along with the associated basic metadata from web pages.
- Publishes the extracted content to a processing queue for subsequent multimodal processing
- Stores initial metadata and content references in MongoDB.
- Reports the state of the service, crawl statistics, and processing metrics to Prometheus and Grafana [33], as shown in figures 2 and 3.

B. Processing Service

The processing service receives raw content (images, text, or speech) from the scraper service and performs the necessary

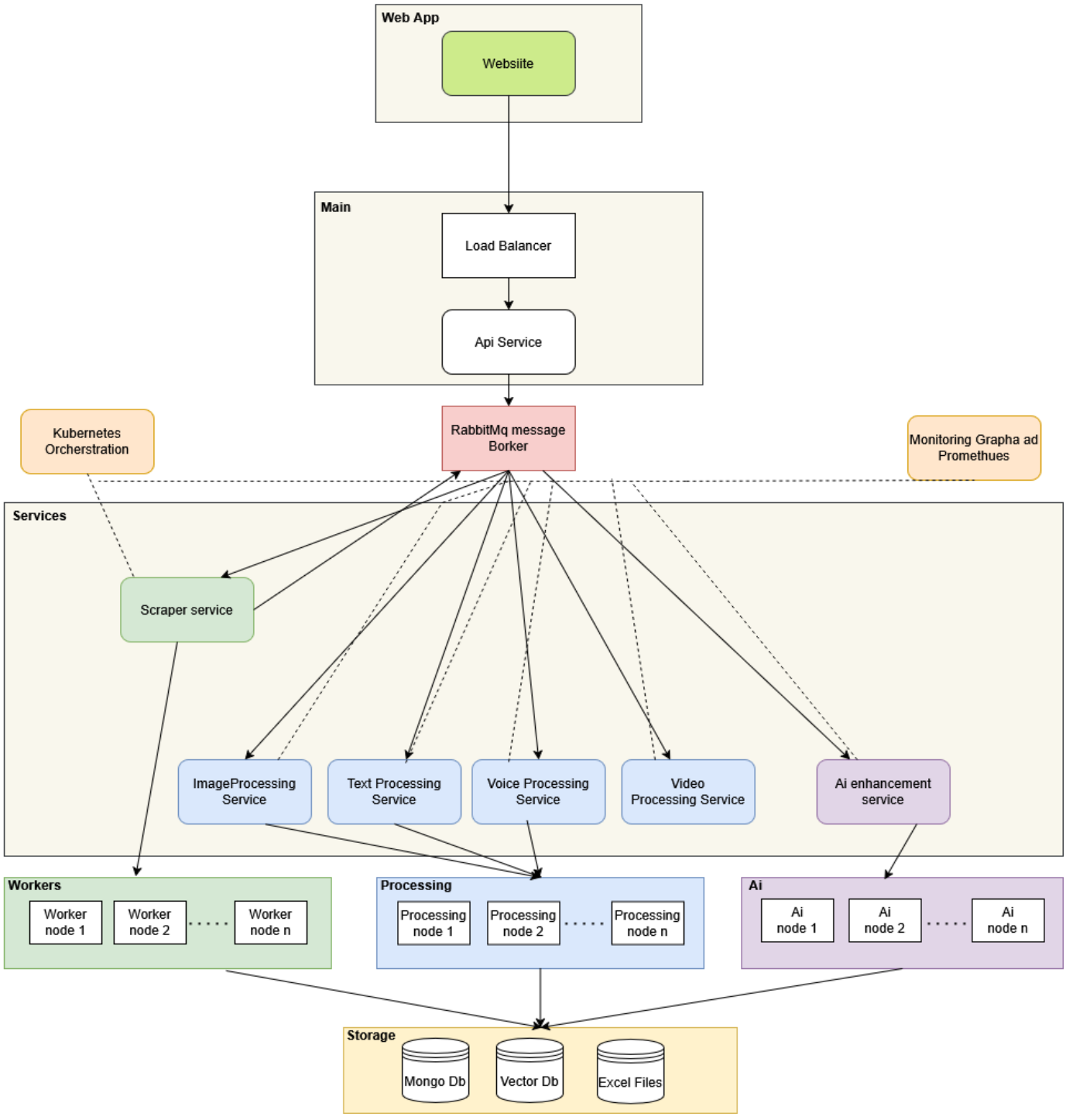

Fig. 1. A detailed architecture of the proposed framework

transformations, normalizations and feature extraction for the subsequent AI analysis. It comprises four main services:

- Image Processing Service: Performs normalization, feature extraction, and embedding generation for image data.
- Text Processing Service: Manages text normalization and embedding generation for textual content
- Speech Processing Service: Performs pre-processing, feature extraction, and embedding generation for audio data
- Video Processing Service: Performs video analysis, including frame extraction, feature detection, and embed generation for video data.

The flow of this service is as follows:

- Processes raw HTML and related content from the scraper queue.

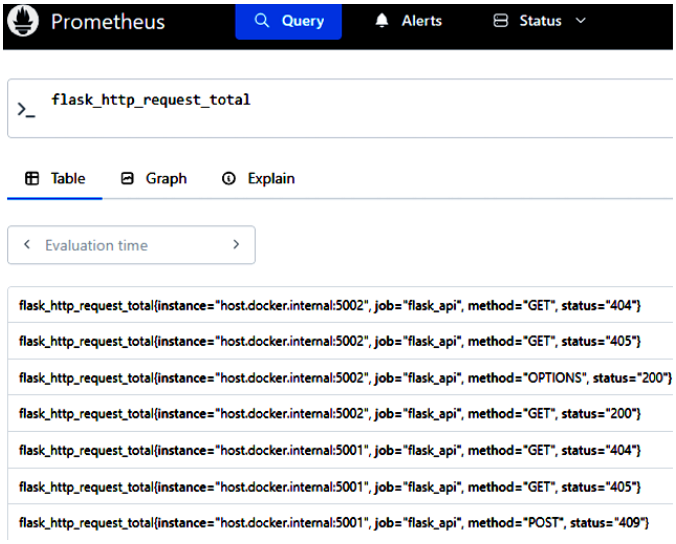

Fig. 2. Prometheus interface

- Extracts relevant media (e.g., images, text sections, audio sources) from the HTML content.
- Processes each content type (e.g., image normalization, text cleanup, audio preprocessing).
- Generates embeddings and stores them in the vector database.
- Publishes the processed content to the AI queue for further analysis
- Updates the processing status and metadata in MongoDB.

C. AI Enhancement Service

The AI Enhancement Service uses machine learning models to analyze different types of content such as images, text, and speech and generate rich, contextual metadata. It includes:

- Image AI Enhancement Service: Provides advanced tasks such as generating contextual metadata for images.
- Text AI Enhancement Service: Provides AI-supported functions such as summarizing texts and generating context-related metadata.
- Voice AI Enhancement Service: Provides features such as speech transcription and contextual metadata generation for audio content
- Video AI Enhancement Service: Provides advanced features such as video transcription, scene recognition, object recognition, and contextual metadata generation for video content.

The sequence of this service includes the following:

- Takes processed content (images, text, or audio) and from the processing queue.
- Applies special ML models.
- Creates and stores extended metadata in MongoDB.
- Provides internal API endpoints for the main API service to query or trigger extension operations.

D. API service

The API service provides endpoints for client interaction and system control. It provides the interface for client applications and takes care of authentication, routing, and request management. The flow of this service includes the following:

- Authenticates incoming requests.
- Routes incoming requests to the appropriate queues.
- Queries databases to retrieve results.
- Communicates with the AI Enhancement Service for special queries.
- Applies load balancing.

E. Additional Components

The additional supporting infrastructure includes the following:

- RabbitMQ serves as the backbone for asynchronous communication between the services.
- MongoDB is used to store metadata.
- Vector database for managing embeds across different content types.
- Prometheus & Grafana for monitoring and observability.
- Kubernetes for orchestration and scaling.
- React-based frontend for user interaction.

The system uses Kubernetes for deployment and orchestration:

- deployment strategy:
  - Each microservice is deployed as separate pod(s)
  - Horizontal pod autoscaling (HPA) based on load.
  - Resource quotas and limits.
  - Service discovery via Kubernetes Services.
  - Ingress controller for external access.
- Scaling approach:
  - Worker nodes scale based on queue depth.
  - Processing nodes scale based on CPU/memory utilization.
  - API instances scale based on request rate.
  - Database replicas are managed via StatefulSets.

IV. Experiments and Results

To evaluate the proposed framework, we selected a domain-specific use case focusing on cigars, beverages and corresponding product guides. These categories represent multi-modal data sources commonly found in the lifestyle industry and allow us to demonstrate the system's ability to process different web content. The dataset comes from the website referenced in [34], which provides a wide range of cigar and beverage offerings with descriptive content and metadata. The Cigars category is further divided into two subcategories: Cuban Cigars and New World Cigars. The latter refers to cigars produced in regions outside of Cuba, such as the Dominican Republic and Nicaragua. Using the scraper service, the system automatically extracted all product links from the specified categories on the target website. Figure 4 shows the script output after running the link extraction process. Once the URLs were collected, the scraper service performed individual extraction tasks on each product page. This ensured that each

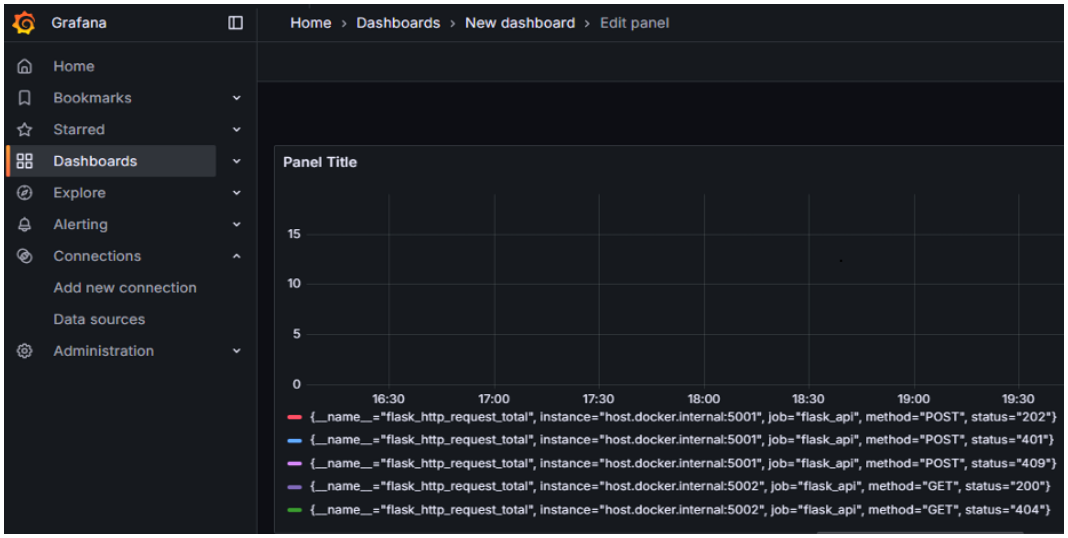

Fig. 3. Grafana interface

Fig. 4. Product Links Extraction Screenshot

item contained a unique and complete data set, including relevant text, image content, and associated metadata. The extracted data was then processed and stored in the system's structured storage pipeline. An example of the structured output can be found in Figure 5. This experiment confirms the framework's ability to autonomously crawl, extract, and organize heterogeneous web content into a coherent, queryable format, demonstrating its applicability for the creation of open domain datasets.

V. Conclusion

The framework presented in this paper provides an innovative and adaptable solution for processing multimodal data from a variety of resources. By incorporating advanced methods such as RAG and using a distributed, scalable architecture, the system can autonomously collect, organize, and structure data from different sources. Unlike existing approaches, which are often limited to specific tasks or domains, our framework's openness and focus on dataset creation makes it a highly versatile tool that can tackle a wide range of real-world challenges. This framework not only optimizes data usage, but also paves the way for new applications in areas such as research, healthcare, business intelligence, and beyond.

References

[1] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. J. Mach. Learn. Res. 2023, 24, 1–113.

[2] Brown, T.B. Language models are few-shot learners. arXiv 2020, arXiv:2005.14165.

[3] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. arXiv 2023, arXiv:2307.09288.

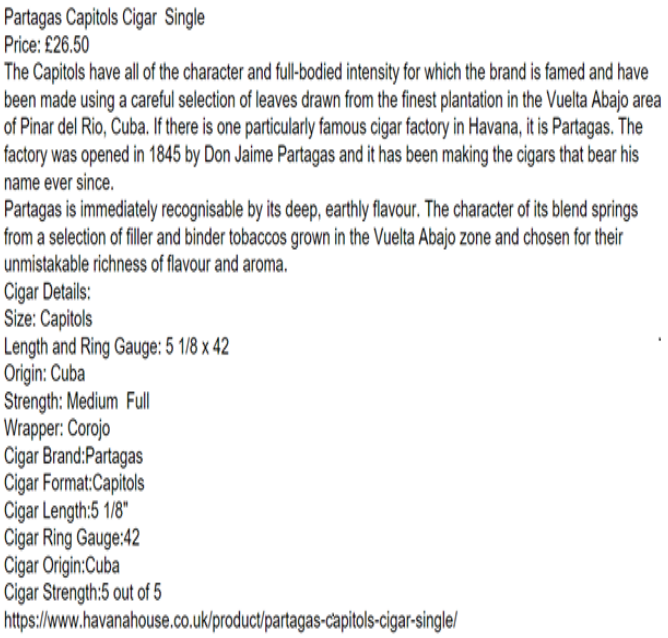

Fig. 5. Content result example

[4] Patrick Lewis et al., Retrieval-augmented generation for knowledge-intensive nlp tasks. In Advances in Neural Information Processing Systems (NeurIPS 2020), 2020.

[5] Ling, Zihan, et al. "MMKB-RAG: A Multi-Modal Knowledge-Based Retrieval-Augmented Generation Framework." arXiv preprint arXiv:2504.10074 (2025).

[6] Yu, Shi, et al. "Visrag: Vision-based retrieval-augmented generation on multi-modality documents." arXiv preprint arXiv:2410.10594 (2024).

[7] Liu, Pei, et al. "HM-RAG: Hierarchical Multi-Agent Multimodal Retrieval Augmented Generation." arXiv preprint arXiv:2504.12330 (2025).

[8] Lahiri, Aritra Kumar, and Qinmin Vivian Hu. "AlzheimerRAG: Multimodal Retrieval Augmented Generation for PubMed articles." arXiv preprint arXiv:2412.16701 (2024).

[9] Bai, Yang, Christan Earl Grant, and Daisy Zhe Wang. "RAMQA: A Unified Framework for Retrieval-Augmented Multi-Modal Question Answering." arXiv preprint arXiv:2501.13297 (2025).

[10] Lang Mei and Siyu Mo and Zhihan Yang and Chong Chen. "A Survey of Multimodal Retrieval-Augmented Generation." arXiv preprint arXiv:2504.08748 (2025).

[11] Lang Mei and Siyu Mo and Zhihan Yang and Chong Chen. "Developing Retrieval Augmented Generation (RAG) based LLM Systems from PDFs: An Experience Report." arXiv preprint arXiv:2410.15944 (2024).

[12] I Made Putrama and Péter Martinek: Heterogeneous data integration: Challenges and opportunities, Data in Brief, vol. 56, (2024).

[13] Haolong Chen et al., An overview of domain-specific foundation model: key technologies, applications and challenges. arXiv preprint arXiv:2409.04267, (2024).

[14] Minghao Liu et al., Automatic Dataset Construction (ADC): Sample Collection, Data Curation, and Beyond. arXiv preprint arXiv:2408.11338, (2024).

[15] Scalable Architecture for Machine Learning Applications. The Journal of Scientific and Engineering Research. vol. 11. pp. 111 - 117, 10.5281/zenodo.13753585, 2024.

[16] Marnfeldt K and Wilber K. The Safety-Autonomy Grid: A Flexible Framework for Navigating Protection and Independence for Older Adults. Gerontologist. 2025 Mar 17:gnaf111. doi: 10.1093/geront/gnaf111. Epub ahead of print. PMID: 40096528.

[17] Hapipah et al.,The Impact of Globalization On the Media Industry Challenges and Opportunities In the Context of the Digital Economy. International Journal of Progressive Sciences and Technologies. Vol. 45 No. 2, (2024),

[18] J. Achiam et al., "GPT-4 technical report", arXiv preprint arXiv:2303.08774, 2023.

[19] T. Brown et al., "Language models are few-shot learners", Adv. Neural Inf. Process Syst., vol. 33, pp. 1877–1901, 2020.

[20] H. Touvron et al., "LLaMA: Open and efficient foundation language models", arXiv preprint arXiv:2302.13971, 2023.

[21] A. Chowdhery et al., "PaLM: Scaling language modeling with pathways", J. Mach. Learn. Res., vol. 24, no. 240, pp. 1–113, 2023

[22] S. Zhang et al., "OPT: Open pre-trained transformer language models", arXiv preprint arXiv:2205.01068, 2022.

[23] R. Taylor et al., "Galactica: A large language model for science", arXiv preprint arXiv:2211.09085, 2022.

[24] Okonji, O. R., Yunusov, K., & Gordon, B. (2024). Applications of Generative AI in Healthcare: Algorithmic, Ethical, Legal, and Societal Considerations. IEEE Engineering Management Review. DOI: 10.1109/EMR.2024.3355973

[25] Inavolu, S. M., Exploring AI-Driven Customer Service: Evolution, Architectures, Opportunities, Challenges, and Future Directions. International Journal of Engineering and Advanced Technology, 13(3), 156-163. Available: https://www.researchgate.net/publication/381224987,2024.

[26] Anantrasirichai, N., & Bull, D., Artificial Intelligence in the Creative Industries: A Review. arXiv preprint arXiv:2007.12391. Available: https://arxiv.org/abs/2007.12391,2020.

[27] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805, 2018.

[28] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension", arXiv preprint arXiv:1910.13461, 2019.

[29] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer", J. Mach. Learn. Res., vol. 21, no. 1, pp. 5485–5551, 2020.

[30] Website, openAI, https://platform.openai.com/docs/guides/embeddings, last accessed, April 2025

[31] Website, cassandra, https://cassandra.apache.org/index.html,last accessed, April 2025

[32] Website, Prometheus, https://prometheus.io/, last accessed, April 2025

[33] Website, Grafana, https://grafana.com/, last accessed, April 2025

[34] Website, https://www.havanahouse.co.uk/, last accessed, April 2025