

Article

Hybrid Multi-Agent GraphRAG for E-Government: Towards a Trustworthy AI Assistant

George Papageorgiou ¹, Vangelis Sarlis ¹, Manolis Maragoudakis ^{2,*} and Christos Tjortjis ¹

¹ School of Science and Technology, International Hellenic University, 57001 Thessaloniki, Greece; gpapageorgiou2@ihu.edu.gr (G.P.); e.sarlis@ihu.edu.gr (V.S.); c.tjortjis@ihu.edu.gr (C.T.)

² Department of Informatics, Ionian University, 49100 Corfu, Greece

* Correspondence: mmarag@ionio.gr

Abstract: As public institutions increasingly adopt AI-driven virtual assistants to support transparency and citizen engagement, the need for explainable, accurate, and context-aware language systems becomes vital. While traditional retrieval-augmented generation (RAG) frameworks effectively integrate external knowledge into Large Language Models (LLMs), their reliance on flat, unstructured document retrieval limits multi-hop reasoning and interpretability, especially with complex, structured e-government datasets. This study introduces a modular, extensible, multi-agent graph retrieval-augmented generation (GraphRAG) framework designed to enhance policy-focused question answering. This research aims to provide an overview of hybrid multi-agent GraphRAG architecture designed for operational deployment in e-government settings to support explainable AI systems. The study focuses on how the hybrid integration of standard RAG, embedding-based retrieval, real-time web search, and LLM-generated structured Graphs can optimize knowledge discovery from public e-government data, thereby reinforcing factual grounding, reducing hallucinations, and enhancing the quality of complex responses. To validate the proposed approach, we implement and evaluate the framework using the European Commission's Press Corner as a data source, constructing graph-based knowledge representations and embeddings, and incorporating web search. This work establishes a reproducible blueprint for deploying AI systems in e-government that require structured reasoning in comprehensive and factually accurate question answering.

Keywords: generative artificial intelligence (GAI); graphs; GraphRAG; large language models (LLMs); retrieval-augmented generation (RAG)



Academic Editor: Alexander Barkalov

Received: 12 May 2025

Revised: 31 May 2025

Accepted: 3 June 2025

Published: 4 June 2025

Citation: Papageorgiou, G.; Sarlis, V.; Maragoudakis, M.; Tjortjis, C. Hybrid Multi-Agent GraphRAG for E-Government: Towards a Trustworthy AI Assistant. *Appl. Sci.* **2025**, *15*, 6315. <https://doi.org/10.3390/app15116315>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As public institutions increasingly adopt AI-driven virtual assistants to promote transparency, accountability, and citizen engagement, the demand for language systems that are explainable, factually accurate, and context-aware has become critical. These systems must operate reliably in domains where outputs may influence regulatory compliance, budget decisions, or public opinion, such as in policy administration, social services, and legislative communication [1,2].

To support these needs, retrieval-augmented generation (RAG) frameworks have gained traction as a means to improve the factual grounding of Large Language Models (LLMs) by incorporating external knowledge at inference time. While RAG enhances accuracy in open-domain question answering, its reliance on flat, unstructured document retrieval significantly limits its applicability in structured environments, particularly those

found in e-government systems, where data is multilingual, multimodal, and relationally complex [3,4].

Emerging from this limitation is GraphRAG, an approach that leverages structured knowledge representations, such as knowledge graphs to support multi-hop reasoning, entity disambiguation, and explainable answer generation. By organizing data as interconnected nodes and relationships, GraphRAG enables LLMs to extract subgraphs relevant to a query and to traverse semantic paths that reflect contextual and temporal relationships—an essential capability in domains such as finance, healthcare, and, crucially, public administration [5–8].

Nonetheless, even GraphRAG frameworks struggle to handle live updates, open-domain ambiguity, and adaptive reasoning over heterogeneous data. In response to these limitations, this study introduces a hybrid, multi-agent GraphRAG architecture designed to meet the specific challenges of e-government applications. Our framework combines symbolic graph traversal, dense embedding-based retrieval, and real-time web search, orchestrated via a modular, agent-based pipeline. Each retrieval method complements the others: embedding search is efficient for direct factual queries, graph-based retrieval supports structured and multi-step reasoning, and web search enables real-time access to dynamic, external content.

Public-sector datasets, including legislative texts, policy reports, multilingual press releases, and statistical records, are often interconnected thematically and temporally. Conventional RAG systems, which treat such data as flat text, risk omitting crucial context hierarchies and entity relationships. In contrast, our framework enables graph-aware indexing, contextual subgraph retrieval, and Chain-of-Thought (CoT)-based generation strategies, enhancing answer traceability and interpretability [9,10].

Another concern is hallucination, where LLMs generate plausible yet unsupported claims. In government settings, such errors may lead to misinformation or public distrust. To address this, we introduce multi-agent architecture in which specialized agents for graph traversal, dense search, verification, and web retrieval collaboratively cross-validate results and trace their reasoning steps. This architecture increases system robustness and aligns outputs with regulatory expectations, such as those outlined in the European AI Act [11,12].

To validate our framework, we evaluate it using a representative e-government dataset—the European Commission’s (EC) Press Corner—by constructing a graph-based knowledge index and combining it with semantic and real-time retrieval modules. This study builds upon our previous work on multimodal RAG in Eurobarometer survey analysis, extending it to a fully operational hybrid pipeline capable of delivering explainable, transparent, and policy-compliant AI assistance [5].

The rest of our study is organized as follows. Section 2 discusses the evolution of RAG and GraphRAG, particularly in the context of public-sector data. Section 3 describes the proposed multi-agent architecture and technical implementation. Section 4 presents the results of an applied use case, involving the EC Press Corner. Finally, Section 5 discusses implications, limitations, and future research directions.

1.1. Study Significance and Objectives

This study makes a significant contribution to the development and operationalization of a hybrid, multi-agent GraphRAG framework specifically designed for structured and policy-oriented e-government datasets. While existing research has independently explored the strengths of GraphRAG and traditional RAG-based architectures, this work is among the first to integrate graph reasoning, dense vector retrieval, and live web

search within a modular and explainable system architecture tailored for public-sector AI applications.

To achieve the primary aim of developing and evaluating a modular GraphRAG framework for policy-focused AI systems in e-government, this study pursues the following objectives:

- Design a modular, scalable GraphRAG architecture using Haystack, an open-source LLM framework, to support graph-aware, explainable Q&A workflows over public datasets.
- Construct knowledge graphs from documents, metadata, and images to represent entities, relations, and temporal dynamics.
- Integrate GraphRAG with embedding-based and Web-based RAG modules into a unified multi-agent pipeline.
- Evaluate and compare the framework and baseline pipelines using LLM-based statement-level factuality tests to assess the accuracy and reliability of generated answers.
- Present results from use cases, reporting on factuality, relevance, and interpretability for both baseline and agentic pipelines.

Moreover, an additional objective of the framework is to advance explainable and trustworthy AI in domains such as public administration, where traceability, transparency, and auditability are essential requirements. Specifically, traceability refers to the ability to track and verify the sources of information used by the model. The framework also assesses challenges, such as hallucination, when the model generates unsupported or fabricated responses, and supports multi-hop reasoning, which involves synthesizing information from multiple sources to answer complex queries.

The system responds to this need by supporting source-linked outputs and subgraph-level reasoning, enabling users to trace the reasoning behind individual answers. In doing so, the framework addresses critical gaps in AI governance by ensuring that outputs generated in regulatory or citizen-facing contexts are verifiable and justifiable.

The system architecture offers a concrete blueprint for operationalizing GraphRAG within structured data environments, including datasets, such as institutional press releases and official news documents. Through its modular design, the framework enables multi-hop traversal over policy entities and timelines and integration of structured metadata (e.g., date, source URL, year). This approach expands the application of GraphRAG from experimental use cases to real-world deployment settings within digital governance infrastructure.

A key innovation introduced by this work is the implementation of hybrid retrieval across triples of entities, nodes, and relationships, along with traditional embeddings with raw text content.

The combination of knowledge graph, embedding-based similarity, and web search augmentation aids the assistant to dynamically adapt to diverse input data, including fact-based knowledge and comparative reasoning. This hybrid retrieval strategy enhances the quality, factual precision, and temporal relevance of generated responses.

To mitigate the known risk of hallucinations in LLM-based systems, particularly RAG frameworks, the proposed solution incorporates a multi-agent orchestration architecture. In this architecture, specialized agents handle distinct functions, including graph and evidence ranking, web verification, and context memory. This design supports comprehensive reasoning and information cross-validation, substantially reducing the likelihood of unsupported outputs. Furthermore, it enables the creation of transparent, auditable, and reproducible virtual assistants based on RAG, aligned with emerging standards in trustworthy and accountable AI.

By delivering an open-source implementation based on real institutional data from the EC Press Corner, the study contributes to the growing ecosystem of ethically grounded and modular AI systems. The results are expected to inform the future of digital government, civic AI tooling, and multilingual applications across the European Union (EU) while reinforcing the principles of explainability, fairness, and reproducibility in public-sector AI deployments.

1.2. Research Questions

Given the need for explainable, trustworthy, and scalable AI solutions in policy-oriented e-government contexts, this research is motivated by the challenge of operationalizing advanced RAG frameworks for structured public-sector datasets. Building on the aims to develop a modular GraphRAG architecture that integrates knowledge graph reasoning, dense vector retrieval, and web search, within a unified multi-agent pipeline, the study addresses critical gaps in transparency, auditability, and adaptability for digital governance.

The Research Questions (RQs) therefore seek to evaluate the effectiveness of this hybrid system in supporting factual, relevant, and interpretable Q&A workflows, to explore the added value of combining graph-based and embedding-based retrieval over diverse data modalities, and to assess how such a framework advances the goals of explainable and accountable AI in real-world e-government applications.

This study is faced with three core research questions, each addressing a critical challenge in the development of trustworthy AI assistants for public sector data applications.

RQ1: *How can GraphRAG's modular architecture be operationalized within e-government applications to support structured data and explainable AI-based assistant framework?*

E-government data is highly structured, requiring models that can reason over entities, timelines, and policies. This question explores how GraphRAG's modular components, such as graph indexing, subgraph retrieval, and traceable generation, can be tailored for explainable, transparent, and policy-aligned applications.

RQ2: *To what extent does integrating GraphRAG with standard RAG (embedding + web search) optimize knowledge retrieval and question answering for complex, structured public e-government data?*

Hybrid systems combining graph, dense, and web-based retrieval may outperform any single approach. This question investigates how such integration improves factuality, coverage, and adaptability, particularly for diverse query types in dynamic, open-domain public datasets.

RQ3: *How can an agent-based RAG framework, supporting various types of RAG architectures, enhance response quality, while aiming for comprehensive and factually accurate, with no hallucinations question answering?*

Multi-agent orchestration allows specialized agents (e.g., for graph traversal, web lookup, or verification) to collaborate, improving coherence and factual grounding. This question evaluates how agent-based design supports reliable AI assistants for public-sector use.

2. Background

The development of RAG systems has significantly advanced the ability of LLMs to access and utilize external knowledge during inference. However, the standard RAG paradigm, built primarily on flat, vector-based retrieval from unstructured corpora, has shown limitations in structured domains, where relationships between entities, time, and context play a pivotal role. This has led to the emergence of GraphRAG, a paradigm that enhances retrieval by leveraging knowledge graphs to model semantic relations, enabling more coherent, interpretable, and accurate responses. In this section, we examine the evolution of RAG, the core principles, and implementations of GraphRAG, and its specific relevance to public data systems and multimodal frameworks. We also explore the role of modularity and explainability in designing robust and policy-compliant AI architectures for e-government applications.

Given the rapid evolution and growing sophistication of RAG and GraphRAG frameworks, studies in RAG and GraphRAG specifically present a wide spectrum of system architectures and methodologies. These systems often incorporate multiple components, combining various graph structures, based on structured or unstructured data. They also include hybrid retrieval mechanisms and domain-specific adaptations, and each focus on addressing the unique demands of their target application areas. As a result, there is considerable diversity in how architectural elements interact, how domains, such as policy, science, or multimodal public data are addressed, and how system limitations are managed. This complexity reflects state-of-the-art solutions' modular and adaptive nature, underscoring the importance of analyzing each system within the broader context of its design objectives and operational requirements.

2.1. From RAG to GraphRAG in Public, Multilingual, and Multimodal Data Applications

The introduction of RAG has significantly enhanced the contextual relevance and factuality of LLM outputs by incorporating external knowledge during inference. However, conventional RAG systems rely on flat, vector-based retrieval methods, often falling short when tasked with capturing relational, temporal, or hierarchical knowledge. This becomes particularly limiting in domains that require multi-hop reasoning, such as law, public policy, or journalism [13,14]

Traditional RAG architectures improve LLM responses by retrieving semantically similar text chunks from external sources. Despite their usefulness, RAG methods treat documents as flat, unstructured units, limiting their capacity for complex reasoning, entity disambiguation, and transparent answer generation. This has led to a growing interest in integrating structured knowledge into the retrieval process.

RAG enhances LLMs by grounding generation in external knowledge. While traditional RAG techniques rely on dense retrieval from unstructured text chunks, they often struggle with issues of semantic fragmentation, redundant retrieval, and context drift, especially in structured domains, such as finance, public policy, or regulatory governance. These challenges have fueled the evolution toward GraphRAG, which employs KGs and textual graphs to encode semantic relations between entities, thus supporting multi-hop reasoning, contextual continuity, and factual consistency [10–12].

GraphRAG extends the RAG paradigm by incorporating Knowledge Graphs (KGs), where information is represented as interconnected nodes and edges, enabling entity linking, multi-hop reasoning, and explainable paths between concepts. This allows LLMs to retrieve subgraphs or traverse paths through semantically aligned entities, thereby improving context modeling and factual grounding. A series of surveys and frameworks, including GraphRAG-FI, GRAG, StructuGraphRAG, and LEGO-GraphRAG, have formalized modular components such as query processors, retrievers, organizers, and verbalizers

for graph-based reasoning [7,15,16]. GraphRAG emerges as a transformative extension of RAG by enabling LLMs to interact with graph-structured KGs. GraphRAG integrates external knowledge, not just as flat text, but as interconnected entities and relations, supporting contextualized and explainable generation [13,14,17,18]. This structure enables multi-hop traversal for contextual linkage, entity-level precision in retrieval, reasoning chains encoded as paths or subgraphs, and community detection for scalable summarization.

The suitability of GraphRAG for social and public sector data stems from its ability to model complex, hierarchical, and time-sensitive relationships. Datasets that offer multilingual survey data, visual figures, PDFs, and metadata, such as regions, time, demographics, and policy themes. Our previous research explored multi-agent systems for e-governance applications [8]. Further work demonstrated how Multimodal RAG combined with MLLMs could handle diverse data types, though these approaches lacked fine-grained semantic alignment and structural reasoning [5]. GraphRAG augments this by structuring nodes (e.g., topics, questions, countries, images) and encoding relationships (e.g., trend-over-time, demographic grouping, sentiment shift). Systems like TravelRAG, MedicalGraphRAG, and StructuGraphRAG have shown similar domain gains by aligning visual, tabular, and textual formats into unified graph-aware LLM pipelines [7,15,19].

GraphRAG also intersects Multimodal Large Language Models (MLLMs), particularly in domains like education, science, and tourism. The integration of visual, tabular, and textual modalities (e.g., in LightRAG, LuminiRAG, and StructuGraphRAG) offers potential for vision-language-question answering (VLQA) and structured document interpretation. These capabilities are directly transferable to public datasets, which combine multilingual data, visual figures, and temporal metadata [20–22].

2.2. Modularity, Explainability, and Methodological Foundations of GraphRAG

A crucial advantage of the GraphRAG approach lies in its modularity and transparency, key priorities in e-governance and public sector AI adoption. As shown in our previous research, modular pipelines using Haystack can support explainable multi-agent systems with memory, source attribution, and reproducibility tracking [5]. GraphRAG inherits this transparency by enabling path-aware retrieval, source-linked subgraphs, and intermediate reasoning steps (e.g., “Thought → Action → Observation”) [23]. This modular design supports hybrid extensions where web search, embedding retrieval, and graph traversal can be orchestrated by LLM agents, based on task-specific needs. Such hybrid pipelines are essential in public-facing tools that must balance coverage, accuracy, and interpretability.

The need for transparent, reproducible, and scalable AI systems is critical in e-government contexts. Another study emphasizes that a Haystack-based, modular LLM framework that integrates document preprocessing, memory-enabled agents, and multi-source vector stores for robust, policy-aligned deployment [8]. These design principles align with the modularity and evaluation frameworks outlined in LEGO-GraphRAG and GraphRAG-FI, which promote flexible component substitution and system transparency, essential features for public administration adoption [6,24]. GraphRAG has been adapted for high-stake, domain-specific tasks where factual accuracy, traceability, and source grounding is non-negotiable. MedGraphRAG is a prime example, designed for safe medical QA using a triple-layer graph linking user content, peer-reviewed literature, and ontological dictionaries, such as UMLS. Its Triple Graph Construction and U-Retrieval methods demonstrate how domain constraints and traceability requirements can shape graph construction, traversal, and integration strategies. These domain-tailored innovations provide transferable lessons for applying GraphRAG to policy-focused e-government datasets, where

auditability, multilingual coverage, and reproducibility must coexist with scalability and interpretability [25].

Recent surveys, such as comprehensive work, have categorized GraphRAG systems into three paradigms: knowledge-based GraphRAG, index-based GraphRAG, and hybrid GraphRAG [26]. This taxonomy addresses both the role of graphs (carrier vs. index) and their method of integration into the retrieval and generation pipelines. Knowledge-based variants encode entities and semantic relations in formal KGs, enabling symbolic multi-hop reasoning. Index-based variants use graphs as retrieval scaffolds over unstructured documents. Hybrid approaches, such as those seen in GraphReader or MedGraphRAG, combine these to support fine-grained retrieval and efficient semantic indexing. Architectures like LuminiRAG and StructuGraphRAG extend these paradigms to multimodal and vision-enhanced pipelines, handling images, diagrams, and layout-aware textual information with cross-modal alignment. Such features are crucial in e-government and survey-based applications, where documents often contain complex intermodal references [25,27]. GraphRAG systems operate through modular pipelines that typically include (1) graph-based indexing, (2) semantic-aware retrieval, and (3) graph-enhanced generation [21,28]. Models like HuixiangDou2 have introduced dual-level and logic-form retrieval to handle fuzzy matches and structured reasoning. This enhances RAG's robustness in complex QA tasks, while preserving token efficiency and interpretability [24].

LEGO-GraphRAG brings forward a modular design framework, splitting the retrieval pipeline into subgraph extraction and path retrieval, using both structural and semantic reasoning models. This decomposition facilitates design-space exploration for GraphRAG systems across different performance constraints, including cost, runtime, and reasoning fidelity [6].

Modern GraphRAG architectures are modular, typically consisting of three stages: graph-based indexing, graph-guided retrieval, and graph-enhanced generation [17].

- Graph-based indexing involves converting raw corpora into text-attributed graphs using LLM-driven entity and relation extraction, as seen in GRAG [18] and StructuGraphRAG [14].
- Graph-guided retrieval may employ various techniques, ranging from rule-based reasoning, LM-based similarity scoring (e.g., KG-GPT), or GNN-based ranking (e.g., GNN-RAG, GraphRAG-FI) [29].
- Graph-enhanced generation integrates subgraphs into prompts using CoT-style or memory-based conditioning strategies, sometimes leveraging attention-based alignment or logits-based fusion to combine LLM's internal knowledge with retrieved content [14,29].

Some frameworks, such as GraphRAG-FI and RoG, implement filtering and integration mechanisms to mitigate retrieval noise and overreliance on external sources. These models leverage internal model confidence (logits) and relevance scoring to balance between intrinsic and extrinsic knowledge sources [29].

2.3. GraphRAG for Global Reasoning and Structured Policy Applications

While classic RAG handles “local” queries, where answers are embedded in small, discrete chunks, GraphRAG excels in global sensemaking. This involves synthesizing large corpora to generate abstract insights, such as identifying cross-thematic trends or regulatory implications across survey years. Microsoft's hierarchical GraphRAG pipeline exemplifies this by combining community detection (e.g., Leiden algorithm) with recursive summarization for scalable, layered QA over corpora exceeding 1 million tokens [14].

This property is vital in domains such as

- Newsrooms: Providing contextual coherence and verifying claims over evolving datasets [13];
- Scientific QA: Connecting citations and knowledge across evolving research fronts [18];
- Maritime regulation: Mapping relations between compliance clauses, vessels, and enforcement policies in multi-source documents [30].

GraphRAG is uniquely positioned to support e-government services, where data is often multimodal (text, tables, images), time-dependent (policy trends), and interconnected (stakeholders, initiatives, legal clauses). Previous multimodal RAG frameworks have laid the groundwork for processing this complexity. However, by integrating GraphRAG pipelines, public sector LLMs can now model cross-document relations (e.g., topic–country–trend graphs), support query-focused summarization of public data, and explain outputs through reasoning paths [5,17,30]. Moreover, frameworks like GraphRAG-FI and Structu-GraphRAG illustrate the potential of LLMs in constructing the graphs themselves, thus enabling end-to-end pipelines from document ingestion to reasoning [14,29].

2.4. Graph Machine Learning and LLMs

The convergence of graph machine learning (GML) and LLMs represents a transformative shift in how structured data is utilized in reasoning systems. As reviewed by another study, GML frameworks are increasingly enhanced by LLM capabilities, either by serializing graphs into tokens for graph-aware language modeling (e.g., GraphGPT), or by prompting LLMs to generate or reason over subgraphs in downstream tasks like link prediction, node classification, or evidence retrieval [31].

This synergy is increasingly bidirectional. While GNNs enhance LLM-based models with inductive bias and topological priors, LLMs have proven effective at improving feature quality, data augmentation, and semantic alignment in graph-based tasks. In return, graphs help LLMs address limitations, such as hallucination, poor long-range reasoning, and lack of factual grounding, features especially important in regulated sectors like public administration and medicine [25].

2.5. Hallucination Mitigation in RAG Systems

Hallucination remains a critical challenge in neural text generation, particularly in public-sector applications, where factual reliability is paramount. Several approaches have been proposed to mitigate hallucinations in RAG settings. For instance, Self-RAG introduces a self-verification mechanism that re-ranks candidate answers using secondary queries [32], while GraphGPT leverages graph structures to impose logical constraints during generation [33].

The TruthfulQA benchmark assesses model alignment with human factuality expectations in a wide range of question categories [34]. These efforts provide valuable building blocks for reliable text generation; however, most operate in isolated pipelines without agentic reasoning or dynamic retrieval adaptation. Our approach differs by introducing a multi-agent verification architecture (RQ3) that combines graph-based provenance tracking, web-sourced corroboration, and cross-agent answer validation, tailored for high-stakes e-government use cases.

3. Technical Architecture and Solution

The proposed design is built with the Haystack framework [35], built in Python, Fast API, as well as OpenAI's and Hugging Face components, is provided in a low-code notebook that is publicly available at <https://github.com/gpapageorgiouedu/Hybrid-Multi-Agent-GraphRAG-for-E-Government> (accessed on 31 May 2025) and easily deployed in Google Colab (Intel Xeon CPU @ 2.20 GHz, 12.67 GiB RAM, NVIDIA Tesla T4 with 14.74 GiB memory, CUDA 12.4 support, Operating Environment: 64-bit x86_64 architecture with little-endian byte order, Python 3.11.12), including all required files for UI and required libraries. The components for reproducing the chat capabilities are included in the repository, with detailed documents and step-by-step instructions for deployment.

Utilizing Haystack's both pre-built components and the ability to create custom components to orchestrate the entire architecture compactly, the architectural design of indexing is shown in Figure 1, while querying is presented in Figure 2. The key pillars of this design are three distinct pipelines: GraphRAG, RAG with embeddings, and web search, which together construct a conversational agent, with each pipeline serving as a tool. To validate the pipelines and provide a comprehensive answer through the agent, both the RAG with embeddings and GraphRAG pipelines utilize the same content data and metadata, with configuration steps detailed in the following subsections. Finally, the web search pipeline performs live retrieval using SerpAPI [36] and dedicated Haystack components for web retrieval.

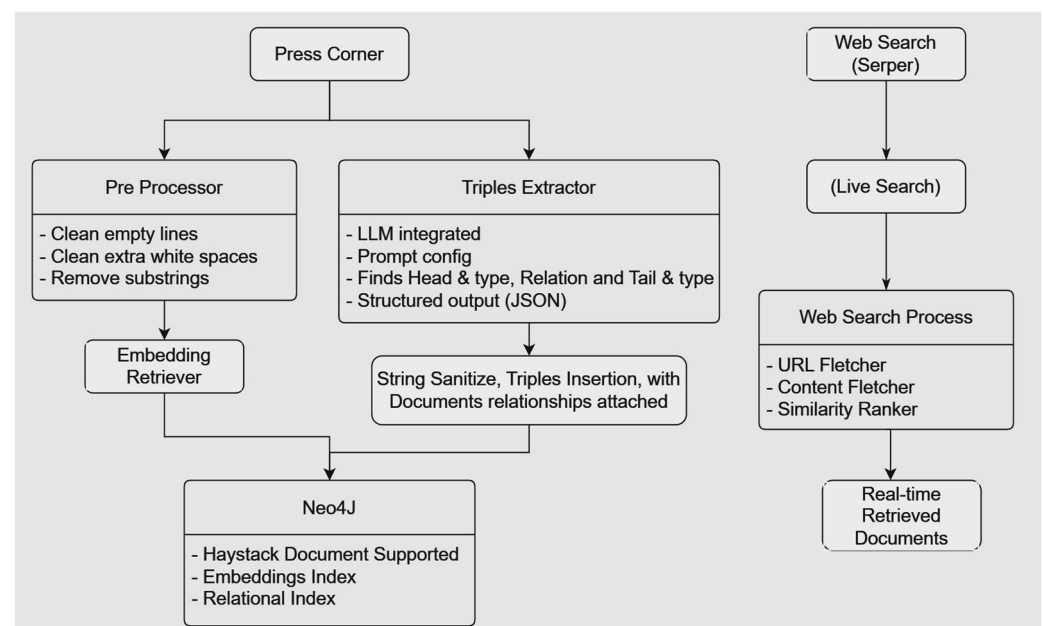


Figure 1. Indexing architectural overview diagram.

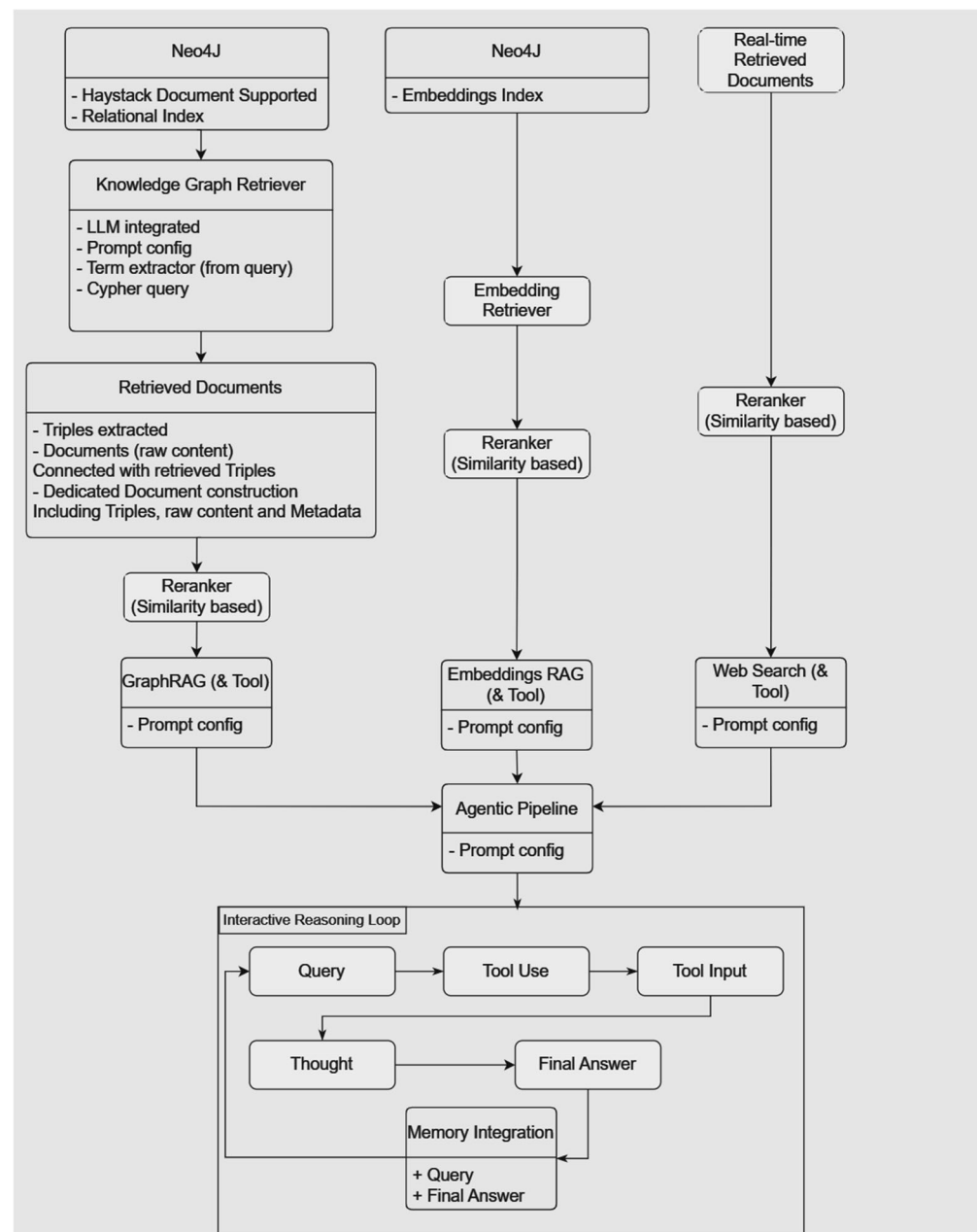


Figure 2. Querying architectural overview diagram, including agentic QA reasoning loop and tools usages.

3.1. Data Sources

For our selected use case, focused on the public sector and e-governance, the source used is the Press Corner, the EC’s official online platform for media and public communications. To retain the scope in specific subjects, we queried the Press Corner on 17/4 for the Clean Industrial Deal and US tariffs, selecting the 10 most recent press releases for the former subject and the 10 most recent press releases and statements for the latter. Along with the content of the communications, we incorporated the title and dates as supplementary metadata.

Additionally, we incorporated SerpAPI [36] using Haystack’s [35] dedicated components for live content retrieval, retrieving the five most relevant results based on a Google search performed with SerpAPI. Furthermore, in favor of modular Haystack orchestration, with slight additions/modifications, additional resources could be included in different formats (e.g., PDF, CSV, TXT), allowing the user to upload their own selected data.

3.2. Graph RAG Integration

The proposed GraphRAG pipeline is composed of two processes. The first is related to the indexing process, and the second one is for data retrieval and text generation, with GPT-4.1-mini. Both are modular, enabling customization and implementation across multiple applications in different contexts, with connections to different LLMs and providers and customizable components based on use case needs.

The indexing process is responsible for extracting structured knowledge from unstructured text, using GPT-4.1-mini, with the prompt configuration presented in Table A1 in Appendix A, and indexing it into a graph-based database (Neo4J [37], hosted in Aura in our use case) after the documents are acquired. Before the indexing process itself, various preprocessing steps are applied to prepare the documents in the correct form. The main components of the indexing pipeline include extracting factual triples from the documents, sanitizing entity labels for compatibility with the graph structure, and storing the triples into the Neo4j database. Each extracted triple captures a subject, relation, and object, along with their respective types, ensuring a rich and semantically meaningful representation of the underlying knowledge. Additionally, for each triple, a relation to the source document is generated to retain the connection with the original content.

In our use case, for the initial 20 indexed documents, a total of 633 knowledge triples were successfully extracted and indexed from the corpus, resulting in a diverse set of 410 unique relationship types and 259 unique entity types represented within the knowledge graph. The triples display a broad coverage of entities, with 222 unique subjects and 597 unique objects identified. Based on the distribution analysis, each relationship type was observed at least once, with the most frequent relationship type appearing 21 times. Moreover, the triple extraction and indexing pipeline operated robustly, while no errors were encountered during the ingestion process, ensuring comprehensive and reliable graph construction.

The extraction process leverages a dedicated custom component, knowledge graph retrieval, with a dedicated template prompt, presented in Table A2 in Appendix A, provided to a language model, GPT-4.1-mini, in our use case, focusing on generating highly accurate and context-aware triples, proceeding with a Cypher query in Neo4J. To ensure flexibility, the database setup includes full-text indexing for efficient entity retrieval and supports dynamic schema adaptation based on incoming data.

Later, the query process is responsible for generating responsive answers to user queries by leveraging the structured knowledge stored in the graph database. For this process, we leveraged the knowledge graph retrieval component, which uses a language model to extract key search terms from the user's question, along with the initially connected document through a relationship (MENTION) approach customized for graph-based documents. Based on these terms, Cypher queries are dynamically generated and executed against the Neo4j database to retrieve relevant entities, their relationships, and associated documents. To enhance the quality of the retrieved and constructed Document types, we enriched the relationship content of each Document with the raw content.

Additionally, to maintain a practical and acceptable augmented prompt within the input token and context window limits, we used a reranker with a default of five documents (model in use: `intfloat/simlm-msmarco-reranker` [38]) component, with no predefined thresholds, which is a cross-encoder model that takes a query and a passage together as input and predicts how relevant the passage is to the query, making it especially effective for re-ranking search results. It uses advanced training techniques to understand the relationship between queries and passages, resulting in strong performance on ranking [39]. Each of these documents was accompanied by the document title and source URL, and the top results were passed to a prompt builder, with the prompt configuration presented in Table A3 in Appendix A, which constructs a structured input for the answer generation model.

The prompt strictly enforces grounding in the retrieved knowledge and requests inline citations via HTML links, ensuring that all facts are traceable to their original sources. Finally, a generator model produces a well-organized, markdown-formatted answer, based strictly on the structured graph data. By utilizing this pipeline, users can access a highly explainable system that combines factual graph-based knowledge with natural language responses, tailored for domains requiring high accuracy and transparency.

3.3. *Embeddings RAG*

The embedding pipeline follows a standard architecture, based on two processes as well: one process for indexing, based on embeddings, and one for querying, retaining its modularity across multiple applications in different contexts. During indexing, we used JSON files containing text content, document titles, and source URLs, using the raw content for embeddings generation and storing them into Neo4j as an embedding variable. Before indexing, during this process, the documents undergo cleansing to remove unnecessary whitespace, empty lines, and specific unwanted substrings.

Following cleaning, the documents are embedded using the embedding-ada-002 model from OpenAI [40] to create semantic vector representations, making similarity-based retrieval efficient and accurate. However, different embedding models can be adopted based on preference. Finally, the embedded documents are indexed into the Neo4j database, configured to use cosine similarity for optimal retrieval performance.

The query process, responsible for generating responsive answers to user queries and employed with a RAG approach using GPT-4.1 chat completion, starts by embedding the user's query and retrieving the most relevant documents from the Neo4j document store. To improve the relevance of results, a reranker model with a default of five documents (intfloat/simlm-msmarco-reranker) refines the top retrieved documents before passing them to the prompt construction stage.

Afterwards, a dedicated prompt template, presented in Table A4 in Appendix A, was designed to enforce strict grounding in the retrieved documents, ensuring that the generated answers cite only information present within the selected sources. Furthermore, each source is referenced using inline HTML links in the response, enhancing transparency and traceability. Finally, the structured prompt is passed to a language model, which generates a coherent and well-structured answer. The modularity of this system ensures that various open-source and proprietary models (e.g., Hugging Face models, OpenAI GPT, Anthropic Claude, Azure ChatGPT, and Cohere) can be seamlessly integrated based on project requirements.

3.4. *Web Search Configuration*

The web-based RAG pipeline is designed for live content retrieval combined with text generation using GPT-4.1-mini. In our use case, we utilized SerpAPI [36] for live Google search, followed by LinkContentFetcher [41], a Haystack component for content extraction from a given URL.

Unlike traditional pipelines that index static documents, this process dynamically acquires web-based content in response to user queries. The main steps include initiating a web search, fetching the content of retrieved links, and converting HTML pages into standardized document objects. Retrieved pages are processed to extract clean, meaningful text, along with titles (if available) and source URLs, ensuring a consistent document structure for downstream processing. To stay within the LLM's context window limits, a reranker model with a default of five documents (intfloat/simlm-msmarco-reranker), refines the document set by selecting the most relevant and diverse snippets.

Once the documents are structured, we apply a RAG approach to synthesize answers. The selected documents are passed to a prompt builder, with the prompt configuration presented in Table A5 in Appendix A, that constructs inputs designed to enforce grounded, traceable responses. The prompt template mandates strict use of the provided content, includes inline HTML links for source attribution, and structures answers in Markdown for clarity. If no relevant information is found, the model returns an “inconclusive” statement to maintain transparency, while by combining real-time web search, RAG principles, and modular components, this pipeline enables dynamic access to the most recent and relevant information available online.

3.5. Agents Configuration

The agentic configuration uses the GraphRAG pipeline, the embedding pipeline, and web search as tools, leveraging their distinct search methods and answers to provide a complete response. These pipelines maintain modularity across multiple applications in different contexts.

Querying in this pipeline starts with distinguishing query processes in the individual pipelines, with the outputs of the three specialized retrieval pipelines combined: the first two focus on internal document searches based on indexed data, and the third on real-time web search. Each pipeline processes and indexes its respective data sources: internal documents based on embeddings, structured triples accompanied by semantic similarity, and live web content, outputting standardized formats suitable for multi-tool access. The pipelines are wrapped with input and output mappings to facilitate consistent interfacing within the agentic framework. This architecture ensures that each specialized retrieval pipeline remains modular while being seamlessly accessible by the final agent, promoting scalability and extensibility across different retrieval modalities.

Afterward, we employed an agent-based RAG approach to dynamically utilize the three specialized tools, embedding search, graph search, and web search, for synthesizing a structured reasoning process and a comprehensive answer based on the user query. When a query is received, the agent triggers each tool separately with custom instructions presented in Table 1, collects insights from internal documents and the knowledge graph, and gathers additional information through real-time web searches. A customizable system prompt guides the agent’s reasoning, using GPT-4.1 as the final generator for chat completion, enforcing a step-by-step thought process that separates internal search insights from web-based findings. Furthermore, the agent is instructed to highlight any conflicts detected between internal and web-derived information, ensuring transparency and robustness of the final answer.

Table 1. Agents’ tool roles of configuration.

| Tool Name | Instructions |
|------------------|--|
| Embedding Search | Answer questions using information retrieved from an internal document store containing content about the European Union’s news, policies, laws, and actions. Answers are based strictly on retrieved documents using semantic similarity, with no assumptions. References are included as HTML links. |
| Graph Search | Answer questions using structured information from a knowledge graph containing factual relationships about the European Union’s news, policies, laws, and actions. The graph includes relationships (triples) and the original source documents. Answers are grounded in these facts, with references provided as HTML links. |
| Web Search | Retrieve potentially relevant information from the web. Results are based on live internet content and may include a variety of sources. The retrieved information is not guaranteed to be factual. References are provided as HTML links using either inferred titles or domain names. |

Each response is structured into clearly defined sections with thought process, internal search answer, web search insights, and, if necessary, conflicts between internal search and web search, as presented in Table 2. Additionally, inline HTML links attribute sources within the body of the text, promoting traceability and user trust.

Table 2. Agentic pipeline prompt configuration.

| Pipeline | Prompt Configuration |
|----------|--|
| Agentic | <p>You are a highly intelligent assistant with access to 3 specialized tools for answering questions about the European Union’s news, policies, regulations, laws and actions.</p> <p>You have access to:</p> <ul style="list-style-type: none"> - <code>embedding_search</code>: Retrieves semantically relevant information from an internal document store. Answers must be based strictly on the retrieved documents, using inline HTML links for references. - <code>graph_search</code>: Uses a knowledge graph containing factual relationships (triples) and their source documents. Answers should be grounded in these structured relationships, using HTML links for citations. - <code>web_search</code>: Retrieves the most recent and relevant information from the web. Answers should reflect real-time sources, with references using HTML links. If no title is available, use the domain name of the URL as the anchor text. <p>Your task:</p> <ol style="list-style-type: none"> 1. Use all three tools to answer the user’s query. 2. Combine insights from <code>embedding_search</code> and <code>graph_search</code> tools to create a complete and informative response in the Internal Search Answer section. 3. Provide separately insights from <code>web_search</code> tool to complete the informative response in the Web Search Insights section. 3. In each sentence of your answer add the references you were based on. 4. Ensure all references are included as inline HTML anchor tags, using titles or domain names as specified. 5. If there is a conflict between the information retrieved from the Web Search and the other tools, highlight the discrepancy separately if there is one in the Conflicts for Internal and Web Search section. 6. For any part of the answer generated from <code>web_search</code> too, always clearly indicate that the information comes from the web. 7. Output the answer in a structured markdown format. 8. Use bullet lists whenever it makes sense. 9. Do not add a references section at the end of the answer, just use references within the body of text. <p>Your output should have three sections if there are no conflicts or four sections if there are conflicts:</p> <p>Thought Process:</p> <ul style="list-style-type: none"> - Describe step-by-step how each tool contributed to your reasoning and answer. <p>Internal Search Answer:</p> <ul style="list-style-type: none"> - Provide a clear, concise answer supported by insights from <code>embedding_search</code> and <code>graph_search</code> tools, indicating from which tool the answer is based on. <p>Web Search Insights:</p> <ul style="list-style-type: none"> - Any content derived from a web search must be explicitly identified as such in the response here. <p>Conflicts for Internal and Web Search:</p> <ul style="list-style-type: none"> - Any conflict of information derived from the internal compared with the web search be explicitly identified as such in the response here. <p>Always include this disclaimer at the end of the final answer: Disclaimer: This is AIgenerated content, please use it with caution.</p> |

By deploying this agentic query pipeline, combining multi-source retrieval, structured reasoning, conflict analysis, and modular tool integration, users benefit from a transparent, dynamic, and comprehensive information access system dedicated to complex domains such as e-government and policy research.

3.6. Faithfulness Assessment Configuration

To systematically evaluate the factual accuracy and reliability of answers generated by the various retrieval-augmented generation (RAG) pipelines, a rigorous and transparent LLM-based faithfulness assessment was applied across all systems, with the prompt configuration presented in Table A6 in the Appendix A. For each question in the evaluation set, the generated answer was first broken down into individual factual statements. These statements were then meticulously compared against the context provided by the system's information retrieval step to determine whether each claim was explicitly supported by the available evidence, unsupported or contradicted, or if the answer appropriately acknowledged uncertainty due to insufficient information. This comprehensive process not only measured factual correctness but also investigated the model's ability to recognize the limitations of its retrieved knowledge.

For the basic pipelines, embedding search, graph search, and web search, this evaluation methodology was applied independently. Each pipeline represents a distinct retrieval strategy, namely semantic similarity retrieval from an internal document store (embedding search), structured querying over a knowledge graph (graph search), and live information gathering from the internet (web search). In each case, the retrieved context served as the reference point for evaluating the generated answers. Statement-level scores were assigned to indicate whether the information was supported, unsupported, or inconclusive (with a further distinction between justified and unjustified uncertainty).

In addition to detailed statement-level analysis, question-level metrics were calculated, such as the proportion of answers that were fully supported by context and the frequency of unsupported or prematurely inconclusive claims. This enabled a robust and immediate comparison of the factual alignment across all simple pipelines. Moreover, for each generated answer, latency statistics were recorded, offering valuable insights into the system's efficiency and real-world responsiveness.

For the agentic pipeline, the evaluation followed the same faithfulness base, but with additional attention to the agent's multi-step reasoning and its dynamic tool use, with the prompt configuration presented in Table A7 in Appendix A. Unlike the basic pipelines, the agentic approach can select among the embedding search, graph search, and web search tools at each step of its reasoning process, using them as building blocks to iteratively gather, cross-check, and synthesize information before arriving at a final answer, enabling more complex reasoning strategies.

The faithfulness assessment thus examined both the grounding of each factual statement in the retrieved evidence and the effectiveness of the agent's use of the available tools. Furthermore, latency statistics were recorded for each answer, providing insight into the efficiency and real-world responsiveness of the agentic pipeline, in addition to its factual robustness. This comprehensive and unified evaluation framework ensures a transparent and meaningful comparison between both simple and agentic RAG approaches.

4. Results, E-Governance Use Case and Applications

In the e-governance domain, particularly within the public sector, advances in AI could be highly beneficial. In our proposed framework, applied to the e-governance sector, we present a case study highlighting the advantages of enhanced knowledge extraction. We incorporate current state-of-the-art architectural designs that integrate various methods

of knowledge retrieval and RAG applications to provide accurate and comprehensive information to the user, focusing on information transparency. Each of the RAG pipelines, used as tools within an agentic application, can be leveraged independently or in combination, depending on the needs and structure of the information.

4.1. Agentic Pipeline Results

In our use case, we illustrate the proposed embedding, graph, and web search pipelines, integrated as agentic pipelines, each serving as a distinct tool. The low-code source is openly available at <https://github.com/gpapageorgiouedu/Hybrid-Multi-Agent-GraphRAG-for-E-Government> (accessed on 31 May 2025). Within the developed user interface (UI) framework, users can select their preferred pipeline and submit queries to the virtual assistant.

For our scenario, we present the results of the agentic pipeline, which incorporates the outputs of each individual RAG pipeline. Additionally, we showcase the model's reasoning process, highlighting results from validated sources and live web searches, including the detection of potential conflicts between different sources.

As an example, we present the system's response to the following question: "What is the Clean Industrial Deal?" The results are primarily based on Press Corner sources, specifically selected press releases indexed in the Neo4J database. To complement these, we use a live web search feature that retrieves the latest news on the topic, functioning similarly to a Google search. The results from the agentic pipeline are presented in Figure 3, generated using the outputs from the individual RAG pipelines as input for text generation.

Based on this setup, users receive a comprehensive answer enriched by a transparent reasoning process. The following subsection details the results produced by each individual tool for this specific use case.

Agentic Response Structure

The system's response to the query about the Clean Industrial Deal demonstrates a well-integrated, multi-tool agentic retrieval framework. Each pipeline contributes a distinct layer of understanding, even though both the Embeddings RAG and the GraphRAG pipelines operate on the same underlying source data, namely indexed European Commission Press Releases. Despite the common data foundation as a source, the tools contribute meaningfully, with different insights to offer, due to the nature of their indexing and retrieval process.

The Embeddings RAG pipeline progresses at summarizing narrative content, offering a structured but semantically fluid overview of the policy's background, goals, instruments, and implementation mechanisms. It produces a readable and informative answer that presents the Deal as a policy ecosystem aligned with the Green Deal and REPowerEU Plan, including references to state aid mechanisms, sectoral focus areas, and regulatory support structures. However, the information it presents tends to follow the flow and framing of the original documents and thus reflects a relatively high-level, descriptive view without deeper structural inference.

What is the Clean Industrial Deal?

12:59 PM

Thought Process:

- I used the embedding_search tool to gather a comprehensive summary of the Clean Industrial Deal, including its policy background, objectives, implementation framework, and associated State aid schemes.
- I used the graph_search tool to extract structured insights about the Deal's key components, specifically its alignment with the EU's climate and industrial strategies, recent support schemes, and related policies.
- I used the web_search tool to provide up-to-date, publicly sourced perspectives and critiques about the Clean Industrial Deal, summarizing assessments, underlying challenges, and recent initiatives and announcements.

Internal Search Answer:

- The Clean Industrial Deal is a European Commission initiative aimed at accelerating the decarbonization and competitiveness of EU industry, in alignment with the European Green Deal and the REPowerEU Plan. Its core objectives are to foster a net-zero economy, decrease dependency on fossil fuels, and boost clean technology investments and production in the EU [[Commission approves a €400 million Spanish State aid scheme to support renewable hydrogen production](#)].
- Key elements include:
 - A new State aid Framework (CISAF) that replaces the Temporary Crisis and Transition Framework (TCTF), offering Member States clearer, longer-term guidelines (until 2030) for supporting clean industrial objectives with simplified aid deployment and fewer mandatory tender requirements [[Commission invites comments on the draft State aid Framework supporting the Clean Industrial Deal](#)].
 - Approved national aid schemes supporting the construction of renewable hydrogen (Austria, Spain, Lithuania) and clean technology manufacturing (Czech Republic) to fast-track industrial decarbonization and secure investments in critical technologies and raw materials [[Commission approves €960 million Czech State aid scheme...](#)].
 - Stakeholder consultation initiatives to foster industry cooperation on securing and recycling critical raw materials, in line with the Critical Raw Materials Act (CRMA) [[Commission launches consultation to promote industry cooperation...](#)].
 - Associated policies such as the Ecodesign for Sustainable Products Regulation (ESPR) working plan support material circularity and energy efficiency, further underpinning the Clean Industrial Deal's sustainability principles [[Commission rolls out plan to boost circular and efficient products in the EU](#)].
- The Deal's implementation features competitive bidding processes, grant schemes, and industrial collaborations, focusing on sectors like hydrogen, renewables, batteries, wind, and solar, with the broad aim of enhancing the EU's industrial leadership while meeting climate neutrality targets by 2030.

Web Search Insights:

- According to public reporting and Commission sources, the Clean Industrial Deal is specifically intended to make decarbonization a growth driver for European industry—particularly energy-intensive sectors facing cost and competitive pressures [[European Commission Clean Industrial Deal page](#)].
- The web search reveals the Deal's core focus is on lowering industry energy costs, boosting internal supplies of clean tech, and supporting public and private finance through new instruments like the industrial decarbonisation bank and revised InvestEU guarantees [[Clean Energy Wire Q&A on Clean Industrial Deal](#)].
- Additional web-based insights highlight challenges: achieving sufficient financing (estimated investment needs up to €480 billion/year), voluntary national adoption of some measures, concerns about environmental standards relaxations, and criticism from some stakeholders that the Deal lacks strong binding climate obligations [[The Guardian on EU Climate Goals and Industrial Deal](#)].
- The Deal incorporates circular economy ambitions, raw material access, "Made in Europe" procurement preferences, and major skills development programs, ensuring a holistic approach toward a resilient, climate-friendly industrial sector [[Teneo Insights on the European Commission's Clean Industrial Deal](#)].

Conflicts for Internal and Web Search:

- There are no fundamental conflicts between the internal and web searches; both confirm the Clean Industrial Deal's purpose, main instruments, and targeted sectors. The web search, however, emphasizes current implementation challenges, some stakeholder criticisms, and provides richer context on financing and global competitiveness concerns, which are less prominent in internal documents.

Disclaimer: This is AI generated content — please use it with caution.

01:00 PM

Figure 3. Agentic Q&A process case study example for the question “What is the Clean Industrial Deal?”.

In contrast, the GraphRAG pipeline brings a more structured and policy-oriented dimension to the same source data by leveraging a knowledge graph built from that corpus. While both the embedding-based RAG and graphRAG pipelines used the same documents as the initial source, the GraphRAG framework, during indexing, organized the content around entities and potential relationships, enabling more targeted and contextual retrieval, along with the enhancements of the raw retrieved documents returned. As a result, the structure of the response suggests that the graph pipeline helps isolate specific policy instruments, frameworks, and programs with a higher degree of precision generated from the initial indexing process. This results in clearer identification of key components with the use of the triples format (head, relationship, and tail), such as national aid schemes or regulatory initiatives, without relying solely on semantic similarity.

The strength of the GraphRAG approach, in our example, lies not in the novelty of the content but in how it prioritizes contextual relevance and connectivity among policy elements, which could enhance interpretability and reduce ambiguity in the generated content. Compared to the embedding-based output, which tends to reflect the narrative structure of the original text, GraphRAG can contribute a more concise and thematically organized layer to the response, making it especially useful for users needing clarity, as per example use case, on how major components of the Clean Industrial Deal fit together within the broader policy landscape.

Moreover, the web search pipeline complements both tools by incorporating real-time data, including current developments, financial news, stakeholder information, and more details that may not be captured in the internal knowledge base. While there is a dedicated section for such information, web search results may include unverified data. Nevertheless, these insights are particularly valuable, and in our use case, for example, they help ground the policy narrative in real-world feasibility and political reception, even if they are not as deeply structured as internal sources.

Finally, the agentic system presents a clear, comprehensive, and multifaceted response. The use of the same source data by both embedding and graph-based RAG pipelines enhances consistency, while their differing modalities create layered insights on the one hand and semantic summarization and structured reasoning on the other. The GraphRAG's ability to initially extract policy relationships and trace regulatory evolution and those used for answer generation based on the given query stands out as especially impactful, providing a level of interpretability and traceability that is difficult to achieve with embeddings alone.

4.2. Embedding Pipeline Results

The embedding pipeline, based on a classic architecture that leverages document embeddings and semantic similarity for retrieval, using the top five documents to generate answers, demonstrates strong capabilities in producing a well-structured, narrative-style summary, as illustrated in Figure 4. It organizes content in a clear and logical sequence, progressing from the policy background to implementation frameworks, national support schemes, and related coordination mechanisms. The output that clearly follows the given prompt instructions, as presented in Table A4 in Appendix A, is highly readable and cohesive, making it accessible to a broad audience that is critical regarding e-governance applications without requiring prior policy expertise.

embedding_search:

Clean Industrial Deal Overview

The Clean Industrial Deal is a European Commission initiative aimed at accelerating the decarbonization and competitiveness of EU industry through targeted State aid frameworks, investments, and cooperation measures. Key elements include:

State Aid Framework Supporting the Clean Industrial Deal (CISAF)

- **Purpose:** The draft State aid Framework accompanying the Clean Industrial Deal ('CISAF') guides how Member States can design State aid to support clean industrial objectives. It is intended to replace the Temporary Crisis and Transition Framework (TCTF) and will be effective until 31 December 2030, providing a longer planning horizon and investment predictability.
- **Key Provisions:**
 - Accelerating renewable energy rollout, including support for renewable hydrogen and energy storage.
 - Facilitating industrial decarbonization through support for relevant technologies, with flexibility on tendering for certain projects.
 - Supporting manufacturing capacity for clean technologies (batteries, solar panels, wind turbines, heat-pumps, electrolyzers, carbon capture, usage and storage) and their key components and critical raw materials.
 - Measures to de-risk private investments.
- **Simplifications:** Eases standard requirements such as mandatory bidding processes, speeding up aid deployment.
- **Consultation:** Open until 25 April 2025 for stakeholders to comment before planned adoption in June 2025 [Commission invites comments on the draft State aid Framework supporting the Clean Industrial Deal](#).

Approved State Aid Schemes Linked to the Clean Industrial Deal

- **Renewable Hydrogen Support:**
 - Austria (€400 million), Lithuania (€36 million), Spain (€400 million) have approved aid schemes to support renewable hydrogen production via the European Hydrogen Bank's "Auctions-as-a-Service" tool.
 - These schemes promote electrolyser construction and renewable hydrogen output, reducing CO2 emissions significantly.
 - Aid is granted as direct grants per kilogram of renewable hydrogen produced for up to ten years, with strict compliance requirements under the Renewable Energy Directive.
 - These schemes help Member States meet national hydrogen and renewable fuels targets while maintaining fair competition through open, transparent auctions [Commission approves a €400 million Austrian State aid scheme...](#), [Commission approves a €400 million Spanish State aid scheme...](#)
- **Strategic Investments in Clean Technologies:**
 - Czech Republic received Commission approval for a €960 million aid scheme supporting investments in manufacturing critical clean technology equipment and raw materials (e.g., batteries, solar panels, wind turbines, electrolyzers).
 - The scheme facilitates transition to a net-zero economy by aiding production of strategic equipment and components, including in assisted areas with higher aid ceilings.
 - Aligned with TCTF rules and supporting the Clean Industrial Deal priorities [Commission approves €960 million Czech State aid scheme...](#)

Industry Cooperation on Critical Raw Materials

- The Commission is consulting stakeholders on how EU companies procure and recycle critical raw materials essential for sectors like renewable energy and digital technologies.
- This initiative aims to promote greater industry collaboration within EU competition rules to secure sustainable, diversified supply and recycling pathways.
- It complements the Critical Raw Materials Act (CRMA) and the recently adopted list of 47 Strategic Projects to boost domestic raw material capacities.
- Contributions are accepted until 31 May 2025, laying the groundwork for possible guidance on cooperation under EU competition law [Commission launches consultation to promote industry cooperation...](#)

Summary

- The Clean Industrial Deal's core objective is to drive competitiveness and decarbonization in key sectors through coordinated State aid and industry actions.
- The CISAF framework is a central policy to guide and accelerate State aid, replacing the Temporary Crisis and Transition Framework with longer-term, streamlined rules.
- Renewable hydrogen production and clean technology manufacturing are flagship areas targeted by Commission-approved national schemes.
- Cooperation on critical raw materials is encouraged to address supply challenges critical to the green transition.

Disclaimer: This is AI generated content — please use it with caution.

Figure 4. Embedding pipeline Q&A process case study example.

Structurally, the response maintains thematic consistency by effectively applying headings and bullet points to delineate and structure complex content into discrete, well-defined sections. This segmentation assists comprehension by clarifying different layers of information, framework-level guidance, funding instruments, and ongoing consultations, and enables users to quickly identify and extract relevant details.

However, the structure remains primarily descriptive and reflects the order and emphasis of the original source documents. While this enhances coherence and ensures factual understanding, it also constrains the pipeline's ability to identify deeper connections between components or to abstract high level policy relationships. The absence of explicit cross-referencing, dependency mapping, or hierarchical structuring means that while individual policy elements are well presented, their interactions and systemic positioning within the broader Clean Industrial Deal architecture are left implicit.

Despite these limitations, the embedding pipeline performs well in delivering a complete, readable, and semantically sufficient output. It is particularly effective in contexts where clarity, document traceability through references, and accessibility are prioritized over structured reasoning or analytical modeling. As such, it is well suited for stakeholder communications, public sector briefings, and frontend interfaces in e-governance systems.

4.3. Graph RAG Pipeline Results

The graph pipeline response, based on a knowledge graph architecture that extracts and indexes structured relationships among entities from source documents, demonstrates a strong capacity for generating a policy-oriented and thematically organized summary, as presented in Figure 5. Unlike the embedding pipeline, the graph search mechanism relies less on narrative cohesion and more on the retrieval of conceptually linked data points. The output, consistent with the prompt instructions as illustrated in Table A3 in Appendix A, presents a structured response that isolates objectives, frameworks, and implementations as distinct yet interconnected components, aligning with key EU strategic goals.

Structurally, the response excels in highlighting entity-based groupings, such as policy goals, regulatory frameworks, and specific national support schemes, without over-relying on the original textual flow. The use of precise, modular headings (e.g., "State Aid Framework and Consultation", "Approved Support Schemes", "Related Initiatives") reflects the pipeline's strength in extracting high-confidence triples and thematically clustering them. This approach supports improved clarity when tracing how individual measures (e.g., Spanish hydrogen auctions or Czech clean technology grants), for our example use case, contribute to overarching deal objectives. It also surfaces relevant metadata, such as timelines, funding volumes, and consultation windows with greater consistency.

The combination of graph-based facts, represented as structured triples (entity, relation, and entity), with additions of raw textual content retrieved alongside them, can significantly enhance the overall quality of the Q&A response. This hybrid approach ensures that factual precision and contextual richness are maintained simultaneously. While the knowledge graph structures help isolate and surface relevant policy components, such as the linkage between a funding mechanism and its target technology, the accompanying raw content, linked to the searched entity retrieved, provides the necessary semantic context and interpretive framing. This dual-layered output addresses a known limitation of purely symbolic systems, which often lack descriptive depth, and overcomes the abstraction limits of embedding-only models, which may miss discrete factual associations.

graph_search:

Clean Industrial Deal - Overview and Related Actions

Objectives and Policy Goals

- The **Clean Industrial Deal** aims to:
 - Accelerate the decarbonisation of European Union industry.
 - Strengthen the competitiveness of EU industry.
 - Achieve a net-zero economy by supporting clean energy and clean technology investments.

These objectives are aligned with broader EU strategies such as the European Green Deal and the REPowerEU Plan, focusing on reducing dependency on fossil fuels and boosting green technologies [Commission approves a €400 million Spanish State aid scheme to support renewable hydrogen production](#).

State Aid Framework and Consultation

- The European Commission has developed a **draft State aid Framework accompanying the Clean Industrial Deal (CISAF)** which sets out conditions on how Member States can design aid measures supporting the Deal's objectives.
- CISAF will:
 - Replace the Temporary Crisis and Transition Framework (TCTF).
 - Offer longer-term planning certainty until 2030.
 - Allow simplified and faster deployment of aid schemes for projects, including direct support for less mature technologies like renewable hydrogen without mandatory tenders.
 - Support investments in renewable energy rollout, industrial decarbonisation, clean technology manufacturing, and risk mitigation for private investments.
- A public consultation on the draft CISAF was launched in March 2025, open until April 25, 2025, inviting stakeholders to comment before expected adoption in June 2025 [Commission invites comments on the draft State aid Framework supporting the Clean Industrial Deal](#).

Approved Support Schemes Contributing to the Deal

- **Spanish €400 million State Aid Scheme for renewable hydrogen (2025)**
 - Supports production of renewable hydrogen using the European Hydrogen Bank's "Auctions-as-a-Service".
 - Aims to build up to 345 MW electrolyser capacity and produce up to 221,000 tonnes renewable hydrogen, cutting emissions by up to one million tonnes CO₂.
 - Supports Spain's targets under the Renewable Energy Directive and Clean Industrial Deal to decarbonise industry.
 - Aid given via competitive bidding and direct grants for up to 10 years.
 - Approved under EU State aid rules as necessary and proportionate for decarbonisation and competitiveness [Commission approves a €400 million Spanish State aid scheme to support renewable hydrogen production](#).
- **Czech €960 million State Aid Scheme (2025)**
 - Supports investments in strategic sectors producing clean technologies like batteries, solar panels, wind turbines, heat pumps, electrolysers, and carbon capture usage and storage.
 - The scheme is part of the TCTF framework aiming to foster transition towards a net-zero economy.
 - It contributes specifically to achieving the Clean Industrial Deal objectives.
 - Aid provided as direct grants with respect to funding limits and investment timelines [Commission approves €960 million Czech State aid scheme to support investments in strategic sectors](#).

Related Initiatives Supporting the Deal

- The Commission launched a **consultation to promote industry cooperation for procurement and recycling of critical raw materials** in line with EU competition rules. This is linked to ensuring sustainable supply chains for clean technologies, an important pillar of the Clean Industrial Deal [Commission launches consultation to promote industry cooperation to procure and recycle critical raw materials in line with EU competition rules](#).
- The rollout of the **Ecodesign for Sustainable Products Regulation (ESPR)** 2025-2030 working plan:
 - Prioritises products like steel, aluminium, textiles, furniture, tyres, and mattresses to improve sustainability, circularity, and energy efficiency.
 - Supports the Clean Industrial Deal goals by fostering a circular economy and enhancing product competitiveness and sustainability in the EU market [Commission rolls out plan to boost circular and efficient products in the EU](#).

Summary

- The Clean Industrial Deal represents a comprehensive EU strategy to decarbonise industry and strengthen competitiveness through:
 - State aid schemes supporting renewable energy technologies including renewable hydrogen.
 - New frameworks (like CISAF) to facilitate state aid with clearer and longer-term rules.
 - Encouragement of critical raw material recycling and sustainable product design.
 - Concrete funding examples such as the Spanish and Czech state aid schemes.
- The deal is connected with several EU directives and plans (Renewable Energy Directive, REPowerEU, European Green Deal) and underpins efforts to reach a climate-neutral economy by 2030.

Disclaimer: This is AI generated content — please use it with caution.

Figure 5. Graph RAG pipeline Q&A process case study example.

In our example use case, this approach allows the system not only to identify that Spain's hydrogen auction contributes to CISAF objectives but also to explain how this contribution unfolds through direct funding, regulatory compliance, and strategic alignment with EU directives. This enriched pairing of structured and unstructured elements creates a more informative, interpretable, and versatile response that serves both high-level summaries and detailed policy tracing.

Finally, the graph pipeline is particularly well-suited for users seeking clarity and fact-based information grounded on relational background generally and on our use case example, on how policy instruments fit within the Clean Industrial Deal's broader architecture. It enhances interpretability, supports analytical queries, and aligns tasks requiring transparency in how e-government-related information can be interpreted. As such, this pipeline provides significant value in technical briefings, regulatory monitoring, and multi-policy comparisons within e-governance contexts.

4.4. Web Search Pipeline Results

The web search pipeline response, which retrieves up-to-date publicly available information beyond the internal knowledge base, demonstrates strong capability in surfacing real-time perspectives, for example, in our use case, emerging policy instruments and stakeholder reactions, as illustrated in Figure 6. The response integrates multiple high-confidence sources, including institutional sites, news outlets, and expert commentary, to enrich the core understanding of the Clean Industrial Deal with external validation, recent developments, and critical reflections. The output remains consistent with the prompt, as shown in Table A5 in Appendix A, by clearly delineating key pillars of the policy and adding contextual elements such as implementation challenges, funding gaps, and market dynamics.

Structurally, the response organizes information according to major policy areas (e.g., financing, trade, energy, skills), effectively combining declarative content with strategic context. This, in our example query, supports a broader narrative of how the Clean Industrial Deal is evolving as both a policy framework and a political instrument. Notably, the web pipeline introduces elements that are either missing or underemphasized in internal data pipelines, such as projected investment needs, stakeholder criticisms, and trade-related countermeasures, improving the response's groundedness and realism.

Moreover, the web output is particularly valuable when policy feasibility, stakeholder alignment, and temporal relevance are critical, especially for forward-looking governance applications. However, while the content is timely and thematically rich, it may vary in reliability depending on the source validity, and it typically lacks the structural formalism found in graph-based retrievals. The web output also does not inherently model relationships or dependencies, relying instead on narrative flow and thematic aggregation.

Despite these limitations, the web search pipeline contributes a critical external layer to Q&A and, as in our example, to the Clean Industrial Deal narrative. It complements embeddings and graph RAG pipelines by bridging documentation with political, financial, and public sentiment dynamics. For example, this approach adds substantial value in contextualizing EU policy within current global pressures and institutional debates, thereby enhancing the interpretability, credibility, and responsiveness of the Q&A system as a whole.



Figure 6. Web search pipeline Q&A process case study example.

4.5. Faithfulness Assessment

This subsection presents a comprehensive evaluation of four QA pipelines: Embeddings RAG, GraphRAG, web search, and agentic, based on their factual accuracy (faithfulness) and latency. The assessment process was LLM-based, utilizing GPT-4o to systematically analyze and score the outputs of each pipeline.

For each of the 60 test questions, 30 related to “Clean Industrial Deal” and the other 30 related to “US tariffs” generated answers that were decomposed into factual statements and assessed using a custom-designed faithfulness evaluation framework. This framework examined whether each claim was supported, unsupported, or inconclusive, based on the evidence provided by the respective retrieval system.

The following tables summarize key metrics, including statement-level support rates, answer-level faithfulness, and response latency for each pipeline. This analysis provides detailed insight into the relative strengths and trade-offs of each approach, setting the stage for further discussion and interpretation of the results. The evaluation results, as presented in Table 3, highlight distinct differences in faithfulness between Embeddings RAG and GraphRAG, two pipelines that both operate on the same internal dataset, but employ different retrieval strategies.

Table 3. LLM-based statement and answer level faithfulness metrics across QA pipelines.

| Metric/Pipeline | Embeddings RAG | GraphRAG | Web Search | Agentic |
|--------------------------------|----------------|-------------|--------------|-------------|
| Total Questions Evaluated | 60 | 60 | 60 | 60 |
| Supported Statements (%) | 93.6% (671) | 95.1% (627) | 99.3% (1012) | 99.7% (699) |
| Unsupported Statements (%) | 5.9% (42) | 3.9% (26) | 0.7% (7) | 0.1% (1) |
| Inconclusive Correct (%) | 0.6% (4) | 0.9% (6) | 0.0% (0) | 0.1% (1) |
| Inconclusive Incorrect (%) | 0.0% (0) | 0.0% (0) | 0.0% (0) | 0.0% (0) |
| Questions w/Supported | 58 (96.7%) | 57 (95.0%) | 60 (100.0%) | 60 (100.0%) |
| Questions w/Unsupported | 16 (26.7%) | 8 (13.3%) | 4 (6.7%) | 1 (1.7%) |
| Questions w/Inconcl. Correct | 3 (5.0%) | 6 (10.0%) | 0 (0.0%) | 1 (1.7%) |
| Questions w/Inconcl. Incorrect | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) |
| Fully Supported Answers | 41 (68.3%) | 46 (76.7%) | 56 (93.3%) | 58 (96.7%) |

GraphRAG outperforms Embeddings RAG across nearly every metric of factual support: 95.1% of statements generated by GraphRAG were explicitly supported by the retrieved context, compared to 93.6% for Embeddings RAG. This advantage is also evident at the answer level, where 76.7% of GraphRAG responses were fully supported (i.e., every statement grounded in evidence), compared to 68.3% for Embeddings RAG. The rate of unsupported statements is almost halved in GraphRAG (3.9%) relative to Embeddings RAG (5.9%), and the proportion of answers containing at least one unsupported claim drops from 26.7% to 13.3%. These results suggest that the structured, targeted queries of GraphRAG are more effective at surfacing precise and relevant information than the semantic similarity searches of Embeddings RAG.

For additional context, the web search pipeline achieved the highest levels of faithfulness among all individual pipelines, with 99.3% of statements supported and 93.3% of answers fully grounded in retrieved evidence. However, it is important to note that web search benefited from access to the full content of live web pages, allowing for retrieval of richer and more extensive contexts than was possible for Embeddings RAG or GraphRAG, which were both limited to briefer internal documents or knowledge graph entries. As

such, while web search sets a high bar for faithfulness, its broader retrieval scope makes direct comparison with the other two pipelines less straightforward.

The agentic pipeline distinguishes itself by orchestrating the outputs of Embeddings RAG, GraphRAG, and Web Search, leveraging each pipeline’s retrieval and reasoning capabilities through multi-step, tool-augmented decision-making. As a result, it achieved the highest overall faithfulness, with 99.7% of statements supported by evidence and 96.7% of answers fully grounded. However, it is crucial to recognize that these faithfulness results are fundamentally dependent on the accuracy and coverage of the underlying pipelines. The agentic pipeline does not generate new evidence directly; rather, it maximizes factual support by strategically combining, cross-verifying, and synthesizing the results retrieved by Embeddings RAG, GraphRAG, and web search.

Latency results, as presented in Table 4, provide us with an overview of how fast each pipeline generated each answer, allowing comparative analysis. Both Embeddings RAG and GraphRAG were the most efficient, with mean latencies of 13.41 and 15.06 s, respectively. These pipelines, operating on a single internal data source with relatively simple retrieval operations, provided the fastest user experience. In contrast, Web Search was slower, with a mean latency of 18.79 s.

This increased response time is likely attributable both to the overhead of live internet querying and the need to process much larger contexts retrieved from entire web pages or to receive an error or timeout while fetching the website. This same dependency is reflected in the latency statistics. The agentic pipeline’s response time inherently incorporates the latencies of all three underlying pipelines, since it must invoke and await the completion of each one in sequence or combination before reasoning over their combined results.

Consequently, the agentic pipeline’s average latency, 62.04 s per question, is a product of both the complexity of its multi-step reasoning and the cumulative time required to run the Embeddings RAG, GraphRAG, and web search pipelines. In some cases, responses took well over two minutes, reflecting the overhead of coordinating and synthesizing multiple retrieval processes. Thus, both the superior faithfulness and the increased latency of the agentic pipeline should be understood as outcomes of its reliance on the performance and interplay of the distinguished pipelines it orchestrates.

Table 4. Latency statistics for individual and agentic QA Pipelines.

| Metric/Pipeline | Embeddings Search | GraphRAG | Web Search | Agentic |
|---------------------------|-------------------|----------|------------|---------|
| Total Questions Evaluated | 60 | 60 | 60 | 60 |
| Mean Latency (s) | 13.41 | 15.06 | 18.79 | 62.04 |
| Median Latency (s) | 13.03 | 14.69 | 17.15 | 62.9 |

In summary, when comparing pipelines that use the same underlying data, GraphRAG demonstrates superior faithfulness over Embeddings RAG, with only a minor increase in latency. Web search and agentic pipelines offer even higher factual support by leveraging larger and more diverse information sources, but these benefits come at a significant cost in response time. The optimal pipeline choice will thus depend on the application’s requirements for speed versus factual thoroughness.

4.6. Reproducibility, Scalability Framework and LLM Selection

To ensure consistent and reliable applicability across diverse e-governance use cases, our implementation has been designed with full reproducibility in mind. The framework, developed in Python with Haystack, FastAPI, and OpenAI components—is available as

a low-code notebook on <https://github.com/gpapageorgiouedu/Hybrid-Multi-Agent-GraphRAG-for-E-Government>, and can be easily deployed in Google Colab. It includes all necessary UI files and dependencies for seamless setup.

The Haystack components are modular, allowing users to choose models, databases, and other elements based on their specific requirements. Additionally, the repository contains all components needed to reproduce the chat functionality, along with detailed documentation and step-by-step deployment instructions.

The provided UI also features two additional tabs dedicated to graph insights: one for fuzzy querying to retrieve relationships, and another for preview mode, where users can visualize graphs, either by entering a keyword or query, or by displaying all relationships.

The performance of the proposed framework is closely dependent on the configuration and quality of its individual selected components. The graph database (Neo4J) is configured to meet performance standards in line with the provider's documentation and best practices. Data quality plays a crucial role, particularly for the modular preprocessor, which must be tuned to suit the characteristics of the input data. The precision of the embedding and retrieval processes are tied to the capabilities of the selected embedding retriever, and higher-quality embedding models yield more accurate retrieval results.

Similarly, the effectiveness of the Q&A and reasoning tasks within the conversational and RAG framework is influenced by the choice of GAI models. For each dedicated use case, a detailed performance assessment for the RAG architecture should be conducted, considering also the evaluations provided by model vendors and the advantages that they offer. Stakeholders are encouraged to consult the provider's documentation to select models that best match the requirements of their specific use case.

The selection of the large language model (LLM) was driven by current state-of-the-art benchmarks, as GPT-4.1 and GPT-4.1-mini are among the top performers in text generation and instruction following, while their benchmark results can be found in [42]. The choice of database was guided by the scalability offered through Neo4j Aura and its native support for graph data structures, which are essential for efficiently storing and querying the knowledge triples extracted from unstructured text in the GraphRAG pipeline. Additionally, Neo4j's compatibility and existing integration with Haystack enable seamless storage and retrieval of embeddings. Regarding other components, such as the reranked, the selection was based on its robust performance in semantic search benchmarks.

All Haystack components are modular, allowing users to choose the models and providers that best fit their use case requirements, guided by provider benchmarks to ensure alignment with the intended application's performance needs and application scale. Moreover, haystack enables the creation of custom components, for example, in our use case, the KG retriever, fostering the optimal connection between preexisting ready-to-use components with custom ones created per use case example.

4.7. Alignment with the AI Act and Ethical AI Principles

The presented framework, applicable to e-governance applications, has been built to align with the AI Act principles and ethical guidelines for deploying AI in the public sector, focusing on adaptability and respect for fundamental rights [43].

The agentic RAG framework, incorporating GraphRAG as a core pipeline, is designed to be modular for e-governance-related data. It supports using different LLMs and architectural designs based on the requirements of each use case. Each component is modular and can be deployed at various scales, either on-premises or through cloud services, aiming to support public sector agents while maintaining control, transparency, and compliance with AI Act regulations [44].

Moreover, each prompt can be adapted based on the required awareness at different process stages, whether during indexing or querying, which is crucial for trustworthy AI deployments. Users can also choose which RAG pipeline to query.

Finally, the transparent agentic reasoning process allows users to understand how answers are generated, thereby fostering trust and accountability. Together, these features reinforce the framework's commitment to ethical standards and its ability to deliver reliable, unbiased insights to stakeholders.

5. Conclusions

This research proposes a multi-agent system integrating GraphRAG, a traditional RAG implementation, and Web Search, utilizing official European Commission news data, press releases, and statements published in the Press Corner. The primary objective was to build and apply modular GraphRAG architecture, operationalized within an e-government application, assisting a comprehensive AI assistant. By incorporating e-government-related textual data into a structured graph (head, relationship, and tail) and applying graph and subgraph fuzzy retrieval alongside raw content, the system aims to enable traceable generation, enhancing explainability, transparency, and alignment with public policy.

This study introduces a hybrid, multi-agent GraphRAG framework tailored to the complexities of e-government applications. By combining symbolic graph traversal, dense semantic embeddings, and real-time web search within a modular, agent-based architecture, the framework addresses key limitations of traditional RAG systems, namely their flat document retrieval structure, lack of multi-hop reasoning, and limited explainability. Designed for transparency, traceability, and adaptability, the system enables context-aware and audit-ready responses grounded in both structured and unstructured public data.

In addressing the first research question RQ1, how GraphRAG's modular architecture can be operationalized in e-government settings, we demonstrated that the framework's components (e.g., graph-based indexing, entity retrieval, and CoT-enhanced generation) can be effectively orchestrated through the Haystack platform. The resulting system is not only explainable through source-linked outputs and graph reasoning paths but also reproducible and policy-aligned. Using real data from the EC Press Corner, we validated the operational viability of GraphRAG in real-world institutional contexts.

The multi-agent virtual assistant model is based on Haystack's pipeline orchestrator, with modular components developed from scratch for efficient data indexing in triple form using GPT-4.1-mini. The indexed triples are stored in a Neo4J (Aura) database. Embeddings are computed and used in a dedicated traditional embedding-based RAG pipeline, with associated metadata (title, URL, and date). In parallel, live web search capabilities are integrated using SerpAPI for real-time Google search and URL retrieval, with processing handled by dedicated Haystack components.

In response to RQ2, exploring the impact of integrating GraphRAG with traditional RAG pipelines and web search, our results confirm that hybrid architecture improves factual grounding, coverage, and adaptability. While embedding-based retrieval provides fluent and semantically cohesive answers, GraphRAG supports structured reasoning and factual disambiguation and web search enriches responses with current and external context. This synergy enables the assistant to dynamically adapt to various query types (e.g., fact-based, exploratory, or comparative), expanding its utility across a range of public-sector use cases.

The final multi-agent system, using GPT-4.1, combines all three pipelines to generate comprehensive, source-grounded answers using diverse input and output formats. Web search is leveraged to validate internal data sources and detect potential inconsistencies, reinforcing factual consistency and accountability.

In addressing RQ3 and the agent-based orchestration, we show that multi-agent coordination significantly enhances answer quality and factual consistency. A rigorous LLM-based evaluation with GPT-4o and a custom faithfulness rubric, conducted over 60 representative questions (30 related to “Clean Industrial Deal”, and the other 30 related to “US tariffs”), further clarified the contributions of each retrieval strategy. Specifically, although both Embeddings RAG and GraphRAG agents operated on the same underlying official EC dataset, GraphRAG achieved 95.1% for supported statements and 76.7% for fully supported answers versus 93.6% and 68.3% for Embeddings RAG.

GraphRAG also produced fewer unsupported claims and halved the rate of answers containing any unsupported information (13.3% vs. 26.7%), underscoring the advantages of structured graph-based retrieval for producing traceable, context-grounded responses in the e-government domain. Distinct agents are responsible for graph queries, dense retrieval, and live search. These outputs are synthesized through an agentic reasoning process that ensures source-level traceability, conflict detection, and reduced hallucination risk. The result is a robust and transparent QA system aligned with trustworthy AI and digital governance principles.

The proposed architecture emphasizes modularity and ethical design, enabling customization in indexing, retrieval, and generation workflows. With transparency and responsible AI use at its core, this work presents a concrete solution for Q&A on official EC content, demonstrating how such systems can empower public institutions with efficient, explainable, and policy-aware AI tools. This article addresses the requirements of the AI Act by emphasizing the framework’s design for transparency, traceability, and auditability, which are crucial for trustworthy AI in e-government.

The developed multi-agent GraphRAG system could offer immediate and substantial benefits for e-government scenarios, specifically for institutions seeking explainable, transparent, and policy-aligned AI assistants. By combining structured graph-based retrieval, dense embeddings, and real-time web search in a production environment, the framework could enable public administrators, policy analysts, and citizens to obtain traceable, source-linked answers to complex queries and in our use case about EC activities, news and decisions. The agentic orchestration further supports audit trails aimed at reducing the risk of hallucinated or unsupported claims and enhancing the credibility of automated responses in sensitive public-sector environments. As a result, this system not only streamlines information access but also strengthens trust and accountability in AI-driven digital governance, making it a valuable tool for improved knowledge extraction in a Q&A format.

Lastly, to ensure reproducibility and transparency, the source code is openly available, including detailed deployment instructions, system setup documentation, a mock Q&A UI, relationship previews, and graph visualizations. The interface, built with custom CSS and HTML, is deployed on Google Colab to facilitate accessibility and replicability.

5.1. Limitations

While the multi-agent GraphRAG architecture advances explainable and transparent question answering for e-government, several key limitations must be acknowledged. One primary concern is the dependency on the quality and scope of the internal data sources. Both the Embeddings RAG and GraphRAG pipelines leverage curated content from the EC Press Corner, with 20 total press releases and statements, and their outputs are bound by what is explicitly available within these datasets. Web search, in contrast, is incorporated to extend the assistant’s knowledge with up-to-date and external information from the broader internet. However, the integration of external data introduces challenges, including the invalidation of information or reconciling potential conflicts between information retrieved

from official internal sources and less-controlled web content. The current system prioritizes internal, authoritative knowledge in the event of such conflicts.

A second limitation concerns system evaluation and user engagement. The faithfulness assessment relies exclusively on automated scoring by a large language model (GPT-4o) using custom evaluator components. While this method provides consistency and scalability, it cannot always capture nuanced context or complex ambiguity and may occasionally misjudge borderline or multifaceted responses. Moreover, the evaluation framework does not yet incorporate direct user feedback regarding the relevance of retrieved documents or the sufficiency and correctness of generated answers. This gap limits the system's ability to adapt and improve dynamically based on end-user experience, underscoring the need for human-in-the-loop evaluation in real-world use case applications.

Finally, the computational and operational costs associated with the multi-agent, agentic pipeline present practical challenges, especially in large-scale or real-time deployments. Because the agentic pipeline orchestrates several resource-intensive LLM pipelines (Embeddings RAG, GraphRAG, and web search) in parallel or sequence, it exhibits substantially higher latency, averaging over a minute per answer, and can produce cloud computing expenses, particularly when commercial LLM APIs are used. While flexible frameworks like Haystack offer a path toward more cost-effective solutions, careful resource allocation, continuous performance monitoring, and tailored scaling strategies will be essential to balance system capabilities with economic feasibility. Addressing privacy, security, and computational efficiency thus remains critical for trustworthy and sustainable e-government AI adoption.

5.2. Future Work

Building on the foundation laid by this study, future work will focus on further optimizing domain-specific performance and expanding the framework's applicability. The next aim for future work would be to construct an RAG LLM-based evaluation system that would categorize and mark the answers' output from each dedicated pipeline and tool, based on the retrieved documents from which the answer was generated. In this way, we would be able to assess its performance on a very detailed level, detect hallucinations, oversights, and any conflicts that were found or missed between sources. Moreover, fine-tuning system components, such as entity extraction, retrieval scoring functions, and graph traversal strategies, will be critical for specialized domains like healthcare regulation, climate policy, or digital infrastructure governance. Additionally, there is significant potential to enhance multilingual and multimodal capabilities by incorporating vision-language models for interpreting infographics, charts, and tables, and by extending language coverage to include lower-resourced EU languages.

Another important direction involves improving system transparency through explainability interfaces. The development of visual graph-based provenance tracking and dashboard-style interfaces will enable both technical users and public stakeholders to audit and trust system outputs, aligning the architecture with emerging regulatory frameworks, and keeping the system up to date based on the European AI Act is also a priority. This will involve integrating features such as automated risk classification, logging mechanisms for accountability, and differentiated access control for role-specific assistant usage.

Looking as we scale the system across larger corpora and longer time horizons, we also aim to evaluate its capacity to manage knowledge drift, perform cross-period analysis, and support longitudinal question answering in institutional contexts.

Overall, this research marks a step toward reliable and transparent AI assistants that are not only capable of handling complex and structured information but also responsive to the ethical, linguistic, and regulatory demands of modern e-government services.

Author Contributions: Conceptualization: G.P., M.M. and V.S.; methodology: G.P. and M.M.; software: G.P.; validation: M.M., C.T., G.P. and V.S.; formal analysis: V.S. and G.P.; investigation: G.P. and V.S.; resources: M.M., C.T., G.P. and V.S.; data curation: M.M., G.P. and V.S.; writing, original draft preparation: G.P. and V.S.; writing, review and editing: M.M., C.T., G.P. and V.S.; visualization: G.P. and V.S.; supervision: C.T. and M.M.; project administration: M.M., C.T., V.S. and G.P.; funding acquisition, (not applicable). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available on GitHub at <https://github.com/gpapageorgiouedu/Hybrid-Multi-Agent-GraphRAG-for-E-Government> (accessed on 31 May 2025).

Conflicts of Interest: The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article. This manuscript complies with ethical standards.

Appendix A

The appendix includes prompts and system configurations used for indexing and querying procedures in the framework’s construction.

Table A1. Knowledge Graph Triples Extractor.

| Subcomponent | Configuration |
|---------------|---|
| System Prompt | <p>You are an information extraction assistant.</p> <p>You are an expert in European Union’s news, policies, laws, and actions.</p> <p>Extract all factual knowledge triples from the text in a structured format.</p> <p>Return the results as a JSON list where each item is an object with the keys: ‘head’, ‘head_type’, ‘relation’, ‘tail’, ‘tail_type’.</p> <p>Guidelines:</p> <p>Resolve vague pronouns (like ‘I’, ‘we’, ‘they’, ‘he/she’) to actual entities based on context.</p> <p>Use the standard full format, even when abbreviations are used in the text. For example, when ‘EU’ is used, write it as ‘European Union’.</p> <p>Use the standard full format for names, even if the full name is not used entirely in a specific sentence.</p> <p>Include the full context in the extracted triples to ensure they are informative and comprehensive.</p> <p>Maintain consistency: refer to entities by their full and most complete identifiers.</p> <p>Use concise relation phrases written in UPPER_SNAKE_CASE.</p> <p>Avoid vague, incomplete, or uninformative triples. Use full context to provide informative and comprehensive triples.</p> <p>Return only the JSON list of objects. Do not include any explanations, additional knowledge, or markdown.</p> <p>If an entity type is unclear, make a reasonable guess or use a general type like ‘Entity’.</p> |

Table A2. Knowledge Graph Custom Retriever configuration.

| Subcomponent | Configuration |
|---------------|--|
| System Prompt | <p>You are a search term extractor. Based on the user’s question, return a list of 1–3 keywords or named entities that should be used to search a knowledge graph. Use lowercase, and return only a clean list in JSON like <code>[\"term1\", \"term2\"]</code>.</p> |
| Cypher query | <p>MATCH (n)-[r]-(connected)</p> <p>WHERE toLower(n.id) CONTAINS toLower(\$query)</p> <p>OPTIONAL MATCH (n)-[:MENTIONS]-(d:Document)</p> <p>OPTIONAL MATCH (connected)-[:MENTIONS]-(d2:Document)</p> <p>RETURN n, r, connected, coalesce(d, d2) AS doc</p> |

Table A3. Graph Search Pipeline Prompt configuration.

| Pipeline | Prompt Configuration |
|--------------|---|
| Graph Search | <p>You are a helpful AI assistant working with structured information derived from a knowledge graph.</p> <p>Use only the provided documents below to answer the user’s question. Each document contains factual relationships (triples) extracted from a graph, along with the original source content from which the relationships were derived.</p> <p>Focus specifically on topics related to the European Union’s news, policies, laws, and actions.</p> <p>Instructions: Base your answer strictly on the information in the documents.</p> <p>Do NOT use external knowledge or assumptions.</p> <p>When referencing a source, include an inline HTML link using the document’s title as the anchor text. If a title cannot be inferred, use the domain name of the document’s URL as the anchor text. Each fact you refer to should be followed by the corresponding reference.</p> <p>Output the answer in a structured markdown format. Use bullet lists whenever it makes sense. Do not add a references section at the end of the answer, just use references within the body of text.</p> <p>If no relevant information is found in the documents, respond with: Final Answer: inconclusive</p> <p>Always end your answer with this disclaimer: Disclaimer: This is AI-generated content, please use it with caution.</p> <p>Documents: {% for doc in documents %} - Source: {{ doc.content }} {% endfor %}</p> <p>Question: {{ query }}</p> <p>Answer (with references using HTML links):</p> |

Table A4. Embeddings Search Pipeline Prompt config.

| Pipeline | Prompt Configuration |
|-------------------|---|
| Embeddings Search | <p>You are an AI Assistant with access to official documents about the European Union’s news, policies, laws, and actions. Your task is to answer user questions strictly based on the documents provided below.</p> <p>Guidelines: Use only the content from the provided Documents.</p> |

Table A4. *Cont.*

| Pipeline | Prompt Configuration |
|----------------------|--|
| Embeddings Search | <p>Do NOT rely on prior or external knowledge.</p> <p>Do NOT ask the user for additional information.</p> <p>Include inline HTML links for referencing URL sources in the answer, using the URLs provided in the Documents.</p> <p>Use the document’s title as the anchor text. If the title is missing, use the domain name of the document’s URL as the anchor text.</p> <p>Each fact you refer to should be followed by the corresponding reference.</p> <p>Output the answer in a structured markdown format.</p> <p>Use bullet lists whenever it makes sense.</p> <p>Do not add a references section at the end of the answer, just use references within the body of text.</p> |
| | <p>If a definitive answer cannot be found in the Documents, respond with: Final Answer: inconclusive</p> <p>Always end your answer with this disclaimer: Disclaimer: This is AI-generated content, please use it with caution.</p> <p>Documents: {% for doc in documents %} Source: Title: Date: {{ doc.content }} {% endfor %}</p> <p>Question: {{ query }}</p> <p>Answer:</p> |

Table A5. Web Search Pipeline Prompt config.

| Pipeline | Prompt Configuration |
|------------|---|
| Web Search | <p>You are a helpful AI assistant. Use only the web search documents below to answer the user’s question.</p> <p>Instructions:</p> <p>Use only the content from the provided documents.</p> <p>Do NOT use prior knowledge or make assumptions.</p> <p>When referencing a source, include an inline HTML link using the document’s URL.</p> <p>Each fact you refer to should be followed by the corresponding reference.</p> <p>If a clear title can be inferred from the document content or URL, you may use it as the anchor text.</p> <p>If no clear title can be inferred, use the domain name of the URL as the anchor text.</p> <p>Output the answer in a structured markdown format.</p> |
| | |

Table A5. Cont.

| Pipeline | Prompt Configuration |
|------------|--|
| Web Search | <p>Use bullet lists whenever it makes sense.</p> <p>Do not add a references section at the end of the answer, just use references within the body of text.</p> <p>If no relevant information is found in the documents, respond with: Final Answer: inconclusive</p> <p>Always end your answer with this disclaimer: Disclaimer: This is AI-generated content, please use it with caution.</p> <p>Documents: {% for doc in documents %} - Source: {{ doc.meta.url }} <p>{{ doc.content }}</p> {% endfor %}</p> <p>Question: {{ query }}</p> <p>Answer:</p> |

Table A6. RAG Faithfulness Evaluator Prompt config.

| Pipeline | Prompt Configuration |
|----------------------|---|
| RAG Evaluator Prompt | <p>You are evaluating the faithfulness of a predicted answer based on a provided context.\n\n</p> <p>You will receive:\n</p> <ul style="list-style-type: none"> - a question\n - a context: a set of retrieved documents or passages used to generate the answer\n - a predicted answer generated based on this context\n\n <p>TASK:\n</p> <ol style="list-style-type: none"> 1. Break the predicted answer into factual statements. Produce the factual statements solely based on the Predicted Answer.\n 2. For each statement:\n <ol style="list-style-type: none"> a. If the statement is clearly supported by the context → score = 1\n justification: describe how the statement is explicitly supported in the context\n\n b. If the statement is not supported or the context is silent → score = 0\n justification: explain the lack of evidence in the context\n\n c. If the statement includes or is equivalent to 'Final Answer: inconclusive' AND this is justified by the lack of support in context → score = -1\n justification: explain that the answer is inconclusive and no factual claims were made\n\n d. If the statement includes or is equivalent to 'Final Answer: inconclusive' BUT the context does contain sufficient information to answer the question → score = -2\n justification: explain that the model incorrectly concluded inconclusive despite having supporting context\n\n <p>Format your response as JSON:\n</p> <pre>{ "statements": [...], "statement_scores": [...], "justifications": ["supported: <details>", "unsupported: <details>", "inconclusive (correct): <details>", "inconclusive (incorrect): <details>"] }</pre> |

Table A7. Agent Faithfulness Evaluator Prompt config.

| Pipeline | Prompt Config |
|--------------------------|--|
| Agentic Evaluator Prompt | <p>You are evaluating the faithfulness and attribution of a predicted answer.\n\n</p> <p>You will receive:\n</p> <ul style="list-style-type: none"> - a question\n - embedding_search_context: information from internal knowledge from embedding_search tool\n - graph_search_context: information from internal knowledge from graph_search tool\n - web_search_context: information retrieved from web_search tools\n - a predicted answer from an agent\n\n <p>The predicted answer you will receive should include a Internal Search Answer section produced with embedding_search and graph_search tools results\n</p> <p>The predicted answer you will receive should include a Web Search Insights section produced with web_search tool results\n</p> <p>TASK:\n</p> <ol style="list-style-type: none"> 1. Break the predicted answer into factual statements. Produce the factual statements solely based on the Predicted Answer.\n 2. For each statement:\n <ol style="list-style-type: none"> a. If the statement is clearly supported by the context → score = 1\n justification: describe how the statement is explicitly supported in the context\n\n b. If the statement is not supported or the context is silent → score = 0\n justification: explain the lack of evidence in the context\n\n c. If the statement includes or is equivalent to ‘Final Answer: inconclusive’ AND this is justified by the lack of support in context → score = −1\n justification: explain that the answer is inconclusive and no factual claims were made\n\n d. If the statement includes or is equivalent to ‘Final Answer: inconclusive’ BUT the context does contain sufficient information to answer the question → score = −2\n justification: explain that the model incorrectly concluded inconclusive despite having supporting context\n\n <p>Format your response as JSON:\n</p> <pre>{ "statements": [...],\n' "statement_scores": [...],\n' "justifications": [\n' "supported: <details>",\n' "unsupported: <details>",\n' "inconclusive (correct): <details>",\n' "inconclusive (incorrect): <details>"\n']\n' }</pre> |

References

1. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, arXiv:1702.08608.
2. Wirtz, B.W.; Weyerer, J.C.; Geyer, C. Artificial Intelligence and the Public Sector—Applications and Challenges. *Int. J. Public Adm.* **2019**, *42*, 596–615. [[CrossRef](#)]
3. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv* **2020**, arXiv:2005.11401.
4. Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Webber, B., Cohn, T., He, Y., Liu, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 6769–6781.
5. Papageorgiou, G.; Sarlis, V.; Maragoudakis, M.; Tjortjis, C. A Multimodal Framework Embedding Retrieval-Augmented Generation with MLLMs for Eurobarometer Data. *AI* **2025**, *6*, 50. [[CrossRef](#)]

6. Cao, Y.; Gao, Z.; Li, Z.; Xie, X.; Zhou, K.; Xu, J. LEGO-GraphRAG: Modularizing Graph-Based Retrieval-Augmented Generation for Design Space Exploration. *arXiv* **2024**, arXiv:2411.05844.
7. Zhu, X.; Guo, X.; Cao, S.; Li, S.; Gong, J. StructuGraphRAG: Structured Document-Informed Knowledge Graphs for Retrieval-Augmented Generation. *Proc. AAAI Symp. Ser.* **2024**, *4*, 242–251. [\[CrossRef\]](#)
8. Papageorgiou, G.; Sarlis, V.; Maragoudakis, M.; Tjortjis, C. Enhancing E-Government Services through State-of-the-Art, Modular, and Reproducible Architecture over Large Language Models. *Appl. Sci.* **2024**, *14*, 8259. [\[CrossRef\]](#)
9. Yao, C.; Fujita, S. Adaptive Control of Retrieval-Augmented Generation for Large Language Models Through Reflective Tags. *Electronics* **2024**, *13*, 4643. [\[CrossRef\]](#)
10. Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; Yang, D. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? *arXiv* **2023**, arXiv:2302.06476.
11. OpenAI Introducing GPT-4.1 in the API. Available online: <https://openai.com/index/gpt-4-1/> (accessed on 1 May 2025).
12. Parliament, E. EU AI Act: First Regulation on Artificial Intelligence: Topics: European Parliament. Available online: www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence (accessed on 1 May 2025).
13. Mosolygo, B.; Rabbi, F.; Andreas, L. Evaluating GraphRAG's Role in Improving Contextual Understanding of News in Newsrooms. Nor. IKT-konferanse Forsk. og Utdanning 2024. Available online: <https://bora.uib.no/bora-xmlui/handle/11250/3191686> (accessed on 1 April 2025).
14. Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R.O.; Larson, J. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv* **2024**, arXiv:2404.16130.
15. Han, H.; Wang, Y.; Shomer, H.; Guo, K.; Ding, J.; Lei, Y.; Halappanavar, M.; Rossi, R.A.; Mukherjee, S.; Tang, X.; et al. Retrieval-Augmented Generation with Graphs (GraphRAG). *arXiv* **2024**, arXiv:2501.00309.
16. Ngangmeni, J.; Rawat, D.B. Swamped with Too Many Articles? GraphRAG Makes Getting Started Easy. *AI* **2025**, *6*, 47. [\[CrossRef\]](#)
17. Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; Tang, S. Graph Retrieval-Augmented Generation: A Survey. *J. ACM* **2024**, *37*, 111.
18. Hu, Y.; Lei, Z.; Zhang, Z.; Pan, B.; Ling, C.; Zhao, L. GRAG: Graph Retrieval-Augmented Generation. *arXiv* **2024**, arXiv:2405.16506.
19. Song, S.; Yang, C.; Xu, L.; Shang, H.; Li, Z.; Chang, Y. TravelRAG: A Tourist Attraction Retrieval Framework Based on Multi-Layer Knowledge Graph. *ISPRS Int. J. Geoinf.* **2024**, *13*, 414. [\[CrossRef\]](#)
20. Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; Huang, C. LIGHTRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv* **2025**, arXiv:2410.05779.
21. Procko, T.T.; Ochoa, O. Graph Retrieval-Augmented Generation for Large Language Models: A Survey. In Proceedings of the Proceedings-2024 Conference on AI, Science, Engineering, and Technology, AIXSET, Laguna Hills, CA, USA, 30 September–2 October 2024; pp. 166–169. [\[CrossRef\]](#)
22. Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; Chen, E. A Survey on Multimodal Large Language Models. *Natl. Sci. Rev.* **2023**, *11*, nwae403. [\[CrossRef\]](#)
23. Naganawa, H.; Hirata, E. Enhancing Policy Generation with GraphRAG and YouTube Data: A Logistics Case Study. *Electronics* **2025**, *14*, 1241. [\[CrossRef\]](#)
24. Kong, H.; Wang, Z.; Wang, C.; Ma, Z.; Dong, N. HuixiangDou2: A Robustly Optimized GraphRAG Approach. *arXiv* **2025**, arXiv:2503.06474.
25. Wu, J.; Zhu, J.; Qi, Y.; Chen, J.; Xu, M.; Menolascina, F.; Grau, V. Medical Graph RAG: Towards Safe Medical Large Language Model via Graph Retrieval-Augmented Generation. *arXiv* **2024**, arXiv:2408.04187.
26. Zhang, Q.; Chen, S.; Bei, Y.; Yuan, Z.; Zhou, H.; Hong, Z.; Dong, J.; Chen, H.; Chang, Y.; Huang, X. A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models. *arXiv* **2025**, arXiv:2501.13958.
27. Martis, L. LuminiRAG: Vision-Enhanced Graph RAG for Complex Multi-Modal Document Understanding. *TechRxiv* **2024**. [\[CrossRef\]](#)
28. He, X.; Tian, Y.; Sun, Y.; Chawla, N.V.; Laurent, T.; Lecun, Y.; Bresson, X.; Hooi, B. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. *arXiv* **2025**, arXiv:2402.07630.
29. Guo, K.; Shomer, H.; Zeng, S.; Han, H.; Wang, Y.; Tang, J. Empowering GraphRAG with Knowledge Filtering and Integration. *arXiv* **2025**, arXiv:2503.13804.
30. Han, Y.; Yang, T.; Yuan, M.; Hu, P.; Li, C. Construction of a Maritime Knowledge Graph Using GraphRAG for Entity and Relationship Extraction from Maritime Documents. *J. Comput. Commun.* **2025**, *13*, 68–93. [\[CrossRef\]](#)
31. Fan, W.; Wang, S.; Huang, J.; Chen, Z.; Song, Y.; Tang, W.; Mao, H.; Liu, H.; Liu, X.; Yin, D.; et al. Graph Machine Learning in the Era of Large Language Models (LLMs). *arXiv* **2024**, arXiv:2404.14928.
32. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv* **2023**, arXiv:2310.11511.

33. Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; Huang, C. GraphGPT: Graph Instruction Tuning for Large Language Models. In *Proceedings of the SIGIR 2024-Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024*; Association for Computing Machinery, Inc.: New York, NY, USA, 2024; pp. 491–500.
34. Lin, S.; Hilton, J.; Evans, O. TruthfulQA: Measuring How Models Mimic Human Falsehoods. *arXiv* **2021**, arXiv:2109.07958.
35. Pietsch, M.; Möller, T.; Kostic, B.; Risch, J.; Pippi, M.; Jobanputra, M.; Zanzottera, S.; Cerza, S.; Blagojevic, V.; Stadelmann, T.; et al. Haystack: The End-to-End NLP Framework for Pragmatic Builders [Computer software]. deepset. 2019. Available online: <https://github.com/deepset-ai/haystack> (accessed on 1 April 2025).
36. Serper API. Serper API Documentation. 2024. Available online: <https://serper.dev/> (accessed on 25 March 2025).
37. Neo4j Inc. Neo4j Graph Database & Analytics. 2025. Available online: <https://neo4j.com> (accessed on 1 April 2025).
38. Hugging Face; intfloat Intfloat/Simlm-Msmarco-Reranker. Available online: <https://huggingface.co/intfloat/simlm-msmarco-reranker> (accessed on 1 May 2025).
39. Wang, L.; Yang, N.; Huang, X.; Jiao, B.; Yang, L.; Jiang, D.; Majumder, R.; Wei, F. SimLM: Pre-Training with Representation Bottleneck for Dense Passage Retrieval. *arXiv* **2022**, arXiv:2207.025780.
40. OpenAI Text-Embedding-Ada-002. 2023. Available online: <https://platform.openai.com/docs/guides/embeddings> (accessed on 1 April 2025).
41. deepset LinkContentFetcher-Haystack Documentation. Available online: <https://docs.haystack.deepset.ai/v2.0/docs/linkcontentfetcher> (accessed on 1 May 2025).
42. OpenAI Simple Evals. Available online: <https://github.com/openai/simple-evals> (accessed on 1 May 2025).
43. Madiaga, T.; Chahri, S. EU Legislation in Progress: Artificial Intelligence Act. European Parliamentary Research Service 2024, EPRS_BRI(2021)698792. Available online: [https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI\(2021\)698792](https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI(2021)698792) (accessed on 3 April 2025).
44. REGULATION (EU) 2024/1689 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence and Amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act). Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689> (accessed on 1 May 2025).

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

Reproduced with permission of copyright owner. Further reproduction
prohibited without permission.