OG-RAG: Ontology-grounded retrieval-augmented generation for large language models

Kartik Sharma[*]
Georgia Institute of Technology
ksartik@gatech.edu

Peeyush Kumar
Microsoft Research
peeyush.kumar@microsoft.com

Yunqing Li[*]
Lenovo
connie555514@gmail.com

Abstract

While LLMs are widely used for generic tasks like question answering and search, they struggle to adapt to specialized knowledge, such as industrial workflows in healthcare, legal, and agricultural sectors, as well as knowledge-driven tasks such as news journalism, investigative research, and consulting without expensive fine-tuning or sub-optimal retrieval methods. Existing retrieval-augmented models, such as RAG, offer improvements but fail to account for structured domain knowledge, leading to suboptimal context generation. Ontologies, which conceptually organize domain knowledge by defining entities and their interrelationships, offer a structured representation to address this gap. This paper presents **OG-RAG**, an Ontology-Grounded Retrieval Augmented Generation method designed to enhance LLM-generated responses by anchoring retrieval processes in domain-specific ontologies. OG-RAG constructs a hypergraph representation of domain documents, where each hyperedge encapsulates clusters of factual knowledge grounded using domain-specific ontology and retrieves a minimal set of hyperedges for a given query using an optimization algorithm. Our evaluations demonstrate that OG-RAG increases the recall of accurate facts by 55% and improves response correctness by 40% across four different LLMs. Additionally, OG-RAG enables 30% faster attribution of responses to context and boosts fact-based reasoning accuracy by 27% compared to baseline methods. We release the code at https://github.com/microsoft/ograg2.

1 Introduction

Large language models (LLMs) have advanced the capabilities of question-answering systems, search engines, and generic chatbots (Perplexity, 2024; ChatGPT, 2024) . However, they face significant challenges with fact-based adaptation, particularly in domains that rely on precise, domain-specific data, e.g, industrial workflows in healthcare, legal, and agricultural sectors, as well as knowledge work such as news journalism, web based investigative research, consulting, and more. (Cascella et al., 2023; Thirunavukarasu et al., 2023; Singhal et al., 2023; Guha et al., 2024; Wang et al., 2024; Balaguer et al., 2024). To overcome these limitations, off-the-shelf LLMs can be either fine-tuned for specific domains (Bommasani et al., 2021) or paired with external tools or documents (Lewis et al., 2020; Schick et al., 2024). However, fine-tuning is computationally expensive and requires extensive data curation, making it a less practical solution (Balaguer et al., 2024; Ovadia et al., 2023). On the other hand, retrieval-based approaches, such as RAG (Lewis et al., 2020; Sarthi et al., 2024; Zhang et al., 2024b; Borgeaud et al., 2022; Karpukhin et al., 2020; Edge et al., 2024), use domain-agnostic embeddings to retrieve query-relevant information from domain-specific documents for answering. Although promising, these methods fail to capture the deep conceptual relationships and nuanced facts required for accurate domain-specific retrieval.

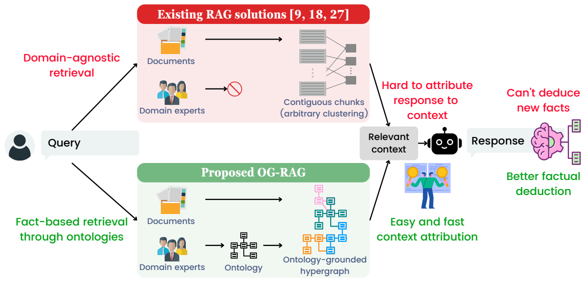


Figure 1: Comparison of the proposed Ontology-Grounded Retrieval Augmented Generation (OG-RAG) with existing RAG solutions.

Each domain organizes its knowledge and terminology in distinct ways, which cannot be generalized across different fields (Mernik et al., 2005). For example, in industrial workflows, facts and relationships are carefully curated and structured

[*]Work done during internship at Microsoft Research

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32962–32981
November 4-9, 2025 ©2025 Association for Computational Linguistics

into domain-specific frameworks, while in knowledge work and investigative research, ontologies serve as templates for organizing and analyzing facts and concepts (Jackson, 1990; Guarino et al., 2009). Current LLMs struggle to adapt to these diverse structures, limiting their accuracy and effectiveness in specialized domains. Another major issue is that users often struggle to trace generated responses back to the relevant context. Furthermore, many specialized domains follow strict procedural rules, and the current techniques fail to reliably deduce accurate conclusions based on this established domain knowledge. This gap presents a major challenge to the wider applicability of LLMs in specialized workflows.

In this paper, we present OG-RAG (Ontology-Grounded Retrieval Augmented Generation), which enhances LLMs by integrating domain-specific ontologies for fact-based adaptation. Ontologies, which define key entities and their relationships within a domain, provide structured representation that is essential for adapting to complex and evolving information landscapes. OG-RAG ground the retrieval process in these ontologies for **improved response accuracy**, **flexible fact-based adaptation**, and enabling **verifiable context attribution**. In particular, we form hypergraph representations of domain documents, where each hyperedge groups related factual knowledge. Using a greedy algorithm, the engine retrieves a minimal set of hyperedges for a given query that forms a compact context for the LLM. Our evaluations of OG-RAG within the agriculture and news domains demonstrate that OG-RAG boosts fact recall by 55% and response correctness by 40% across four LLMs. Users attribute responses 30% faster and better, while fact-based reasoning improves correctness by 27%. These results demonstrate OG-RAG's effectiveness in delivering reliable, fact-based answers across specialized domains.

2 Related Work

Retrieval methods. In addition to the traditional retrieval augmented generation (RAG) (Lewis et al., 2020), graph-based approaches have also been proposed. These include GraphRAG (Edge et al., 2024), RAPTOR (Sarthi et al., 2024), and other knowledge graph-based frameworks such as LlamaIndex [1] and Neo4J [2]. They have advanced LLM performance by leveraging structured knowledge graphs to organize and retrieve contextually relevant information. However, these approaches rely on ad-hoc extraction of entities and domain-specific information, often without grounding in domain expertise. Recent papers (Li et al., 2024; Luo et al., 2023; He et al., 2024) explore retrieving and reasoning over structured knowledge bases or existing knowledge graphs to further enhance factual retrieval. However, these require access to a complete knowledge graph, which is often prohibitive. Our proposed approach instead leverages domain experts-defined semantic schema to ground unstructured documents for improved retrieval. A contemporaneous work, SiReRAG (Zhang et al., 2024a), further combines the strengths of RAPTOR's semantic similarity structure and entity-driven GraphRAG structure, motivating future extensions of combining OG-RAG's domain-specific structure with a general semantic-similarity structure to create a domain-adaptive and generalizable unified RAG system.

Domain adaptation for RAG. Adapting general-purpose RAG systems to specific domains is a challenging task and previous techniques have involved end-to-end training of retriever and generator (Siriwardhana et al., 2023), backbone sharing (Guan et al., 2024), self-improving (Xu et al., 2025), and domain-specialized retriever training (Xu et al., 2024; Cai et al., 2024). While these approaches enhance domain adaptation, they involve retraining and limit the adoption of these technologies to arbitrary domains. Thus, domain ontologies present a unique opportunity to ground and adapt the retrieval process. Traditionally, ontologies have been created to provide a consistent and clear framework for organizing domain knowledge (Guarino et al., 2009; Jackson, 1990). With the advent of LLMs, there promise has been identified in automated learning of ontologies and to extract useful domain-specific information (Babaei Giglou et al., 2023). Other works incorporate ontology information to verify RAG outputs, rather than structurally guiding retrieval or generation (Zhao et al., 2024). Bran et al. (2025) and Xiao et al. (2024) propose ontology-guided RAG but their focus is primarily on scientific documents and automated ontology construction and entity typing tasks respectively. Our work bridges this gap in the literature by leveraging ontology for general-purpose domain-specific retrieval.

[1] https://tinyurl.com/y7t2mt8s
[2] https://tinyurl.com/2rd38usb

Attribution. To enhance the interpretability and reliability of the LLM responses, it is important to attribute their generation to trustworthy sources. One way is to generate text with citations but prior work has shown limitations of existing zero-shot approaches (Gao et al., 2023) and specially-trained models (Khalifa et al., 2024). Furthermore, other forms of attribution are also explored since citations require users to search over a full page to verify the claims in the generated response, which is undesirable. Thus, locally-attributable methods (Slobodkin et al., 2024) and human-in-the-loop (Kamalloo et al., 2023) strategies have also been proposed. While these approaches provide sentence-level attribution, complementary benefits can be achieved through interpretable RAG contexts. OG-RAG provides easy-to-attribute contexts that require only a little effort from the users to trace the generation of the response.

Deductive reasoning. Traditional rule-based reasoning systems provide interpretable and easily controllable ways to deduce novel conclusions from a given input (Jackson, 1990; Saparov et al., 2023). However, they lack the flexibility and generalization capabilities of neural models like LLMs. On the other hand, LLMs are prone to arbitrary hallucinations in deductive reasoning, which can be problematic in structured workflows (Wang et al., 2024; Saparov et al., 2023). OG-RAG combines the structured precision of fact-based reasoning with neural flexibility by anchoring unstructured text to domain-specific vocabulary, enabling LLMs to apply domain-specific rules more effectively.

3 Background

An **ontology** is a formal representation of key entities and their relationships within a domain. For example, in the agriculture domain, entities like crops, soil, and weather conditions are defined, along with relationships such as "crop is grown in a region" or "soil has moisture level". Earlier foundational efforts have led to the widespread standardization of domain-specific ontologies in many industries, such as healthcare [3] retail [4] , and energy [5]. It differs from *taxonomy* or *classifications* as it allows for richer relationships between entities that need not be hierarchical. More formally,

Definition 1 *An* ***ontology*** $\mathcal{O} \subseteq \mathcal{S} \times \mathcal{A} \times (\mathcal{S} \cup \{\phi\})$ *consists of a set of triples that relate a set of entities* $\mathcal{S}$ *using a set of attributes* $\mathcal{A}$*, where* $(s, a, v) \in \mathcal{O}$ *denotes that the subject entity* s *has an attribute* a*, and the value* $v := v_{\mathcal{O}}(s, a)$ *is either:* ***(1)*** *Another entity* $s' \in \mathcal{S}$*, or* ***(2)*** *An unspecified domain value, denoted by* ϕ*. Here,* $v := v_{\mathcal{O}}(s, a)$ *represents the value of the attribute* a *for entity* s*, which is either another entity within the ontology or an undefined (unspecified) text or data.*

For example, consider a subject entity s = "Crop", that can have the attribute a_1 = "is grown in", which maps it to another object entity $v_{\mathcal{O}}(s, a_1) = s'$ = "Crop Region". Additionally, the same entity s can have another attribute a_2 = "has name", which maps it to an arbitrary text, denoted as $v_{\mathcal{O}}(s, a_2) = \phi$, indicating that this value is unspecified and can be any relevant text or name in the domain. These unspecified values can thus be filled by extracting relevant knowledge from domain-specific documents $\mathcal{D}$.

However, different parts of the documents may provide distinct yet valid text/data values related to the same ontology entity. For example, one section may talk about Soybean crops grown in the Northwest region, while another talks about the one grown in the Northeast region. To represent this variability, we introduce factual blocks, each capturing a localized subject–attribute–value triple grounded in a specific context. This mechanism allows us to preserve the contextual distinction between values tied to the same ontology entity, avoiding overwriting or flattening these differences during grounding. Therefore, we model the extracted information $\mathcal{I} := \mathcal{D}(\mathcal{O})$ using a set of self-contained *factual-blocks* $F \in \mathcal{D}(\mathcal{O})$, *i.e.*,

Definition 2 ***Ontology-mapped data*** $\mathcal{I} := \mathcal{D}(\mathcal{O})$ *is a set of factual-blocks, where each factual-block* F *represents a set of ontology relationships* $(s, a, v) \in F$*, where the value* v *is derived as follows: If value* $v_{\mathcal{O}}(s, a) = \phi$ *then* $v \in \mathcal{V}$ *is extracted from the document text; otherwise* $v = v_{\mathcal{O}}(s, a)$ *is the value provided by the ontology.*

Thus, ontology-mapped data represents *self-contained* and *ontology-grounded* information extracted from domain-specific documents. For example, a factual-block F might represent that: a term s ="Seed" is a_1 = "of crop" $v(s, a_1)$ ="Soybean" is a_2 = "is grown in" $v(s, a_2)$ = (s' ="Crop Region", which a_3 = "has a name" of $v(s', a_3)$ = "Northwest Region"). We employ the ontology

[3] https://tinyurl.com/3e8pc2xr
[4] https://tinyurl.com/u5x2nck4
[5] https://tinyurl.com/4re4xekx

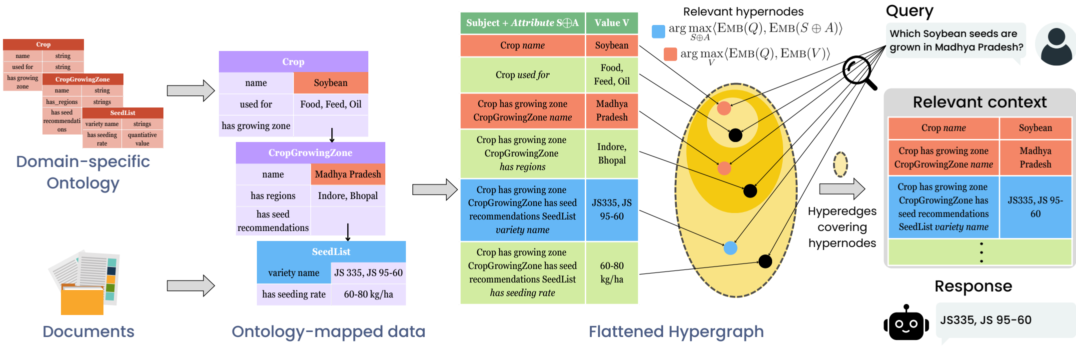


Figure 2: OG-RAG: Ontology-Grounded Retrieval-Augmented Generation

mapping prompt described in Appendix C.2, splitting documents into large chunks of 8192 tokens. Using this setup, we then generate the JSON-LD through GPT-4o's structured output capability.

4 OG-RAG

Then, we propose Ontology-Grounded Retrieval Augmented Generation (OG-RAG) that integrates ontologies, *i.e.*, formal representations of domain-specific concepts and their relationships, into the retrieval process. Unlike existing approaches that rely on general-purpose embeddings or ad-hoc context generation without grounding in domain expertise, OG-RAG leverages ontology-driven hypergraph retrieval to dynamically adapt LLMs to structured knowledge bases and complex domain-specific queries. Figure 2 shows the high-level pipeline of the proposed method and we describe each component in more detail below.

4.1 Hypergraph Construction

The first part includes mapping the general domain-specific documents $\mathcal{D}$ onto a given ontology $\mathcal{O}$ and converting the available information into a set of factual blocks F. To do this, we leverage GPT-4o and prompt it to fill domain-specific values into the ontology from the chunked parts of the documents, thus, converting the documents into an ontology-grounded format suitable for retrieval. The complete prompt is provided in Appendix C.2.

4.1.1 Hypergraph Transformation

First, we note that a factual-block $F \in \mathcal{D}(\mathcal{O})$ is a nested structure by definition since a value can be mapped to another subject-attribute pair in it. This can be challenging to process directly and efficiently. To address this, we flatten the structure so that each factual-block F in the ontology-mapped data $\mathcal{I}$ is converted into a set of flattened factual-blocks $\overline{F}$, making the information easier to handle without significant loss of detail. Algorithm 1 below outlines the flattening process, which is also illustrated in Figure 2.

Algorithm 1 Flattening a factual block

Require: Factual block F, Concatenation $\oplus$.
Ensure: A set of flattened factual-blocks $\overline{F} \leftarrow$ $\text{Flatten}(F)$ flattens any nested information present in F.

```
procedure FLATTEN(F)
    F̄ ← {}
    F̄_0 ← {(s ⊕ a, v) : (s, a, v) ∈ F, v ∈ V, (s', a', s) ∉ F}.   ▷ can be directly flattened
    F̄ ← F̄ ∪ {F̄_0}.
    for (s, a, s') ∈ F \ F̄_0 do.
        if s' ∈ S then
            F_{s'} ← F̄_0 ∪ {(s ⊕ a ⊕ s' ⊕ a', v') : (s', a', v') ∈ F}.
            F̄ ← F̄ ∪ FLATTEN(F_{s'}).
        end if
    end for
    return F̄
end procedure
```

We define each flattened factual-block as a hyperedge $e \in \mathcal{E}$, where a hyperedge e connects multiple hypernodes $\{n_i \in \mathcal{N}\}$, where each hypernode $n_i \in \overline{F}$ is a key-value pair where the key concatenates the nested entities and their attributes. Importantly, this flattening process preserves the integrity of the information without introducing data loss. This allows OG-RAG to capture multi-dimensional relationships between facts, unlike simpler graph-based models that only handle pairwise connections. We can now convert the extracted information from the documents into a more structured hypergraph, defined as follows:

Definition 3 *A **hypergraph** $\mathcal{H}(\mathcal{I}) := (\mathcal{N}, \mathcal{E})$ consists of hypernodes $\mathcal{N}$ and hyperedges $\mathcal{E}$, such that each hyperedge $e \in \mathcal{E}$ is an arbitrary set of hypernodes. Defining $\mathcal{P}(X)$ as the power set of X and $\bigoplus X$ as the set that is formed by concatenating the strings within each element of the set X, we have the hyperedges $\mathcal{E} \subseteq \mathcal{P}(\mathcal{N})$ and the hypernodes $\mathcal{N} \subseteq [\bigoplus \mathcal{P}(\mathcal{S} \times \mathcal{A})] \times \mathcal{V}$, where $\times$ is the cartesian product.*

With this definition, a hypernode is essentially a key-value pair and we declare a hyperedge to be a true *fact* grounded in domain-specific data, where

Definition 4 *A **fact** is a logical assertion between two entities - subject and object, through a functional attribute, which can be evidentially verified to be either true or false.*

Now, let us consider two hypernodes, $n_1(s_1 \oplus a_1, v_1)$ =(Crop has name, Soybean) and $n_2(p_2 \in \bigoplus \mathcal{P}(\mathcal{S} \times \mathcal{A}), v_2)$ =(Crop has growing zone CropGrowingZone with name, Northwest) forming an hyperedge e =((Crop has name, Soybean), (Crop has growing zone CropGrowingZone with name, Northwest)) can be represented as a simplified **fact**: *hasGrowingZone*(Crop has name Soybean) = *Northwest*, which can be evidentially verified to be True or False.

Algorithm 2 OG-RAG

Require: Query Q, Domain-specific Ontology $\mathcal{O}$, Documents $\mathcal{D}$, Sentence embedding function $\mathbf{Z}$, LLM $\mathcal{M}_0$, Maximum length L
Ensure: Retrieved context $\mathcal{C}_{\mathcal{H}}(Q)$ is grounded in the ontology and relevant to the query
procedure OG-PREPROCESS($\mathcal{O}, \mathcal{D}, \mathcal{M}_0$)
  $\mathcal{I} \leftarrow$LLM $\mathcal{M}_0$(Mapping Prompt, $\mathcal{D}, \mathcal{O}$)
  $\mathcal{H}(\mathcal{I}) \leftarrow$ Hypergraph with edges $\bigcup_{F \in \mathcal{I}}$ FLATTEN(F).
end procedure
procedure OG-RETRIEVE($Q, \mathcal{H}(\mathcal{I}), \mathbf{Z}, k, L$)
  $\mathcal{N}, \mathcal{E} \leftarrow$ nodes, edges of hypergraph $\mathcal{H}(\mathcal{I})$.
  $\mathcal{N}_S(Q) \leftarrow \text{topk}_{(s,a,v) \in \mathcal{N}} \langle \mathbf{Z}(s \oplus a), \mathbf{Z}(Q) \rangle$.
  $\mathcal{N}_V(Q) \leftarrow \text{topk}_{(s,a,v) \in \mathcal{N}} \langle \mathbf{Z}(v), \mathbf{Z}(Q) \rangle$.
  $\mathcal{N}(Q) \leftarrow \mathcal{N}_S(Q) \cup \mathcal{N}_V(Q)$.
  $\mathcal{C}_{\mathcal{H}}(Q) \leftarrow \{\}$
  while $(|\mathcal{N}(Q)| > 0) \vee (|\mathcal{C}_{\mathcal{H}}(Q)| < L)$ **do**
    $\mathcal{C}_{\mathcal{H}}(Q) \leftarrow \mathcal{C}_{\mathcal{H}}(Q) \cup \arg\max_{e \in \mathcal{E}} |\{n \in \mathcal{N}(Q) : n \in e\}|$
  end while
  return $\mathcal{C}_{\mathcal{H}}(Q)$
end procedure

4.2 Hypergraph-based retrieval

With the hypergraph constructed on domain-specific information, *i.e.*, $\mathcal{H}(\mathcal{I}(\mathcal{D}, \mathcal{O}))$, OG-RAG is now ready to retrieve relevant context based on user query Q that can support the LLM in generating accurate, domain-specific responses. Algorithm 2 presents the complete pre-processing and retrieval algorithm while the complexity analysis is provided in Appendix A.2.

4.2.1 Relevant Hypernodes

We first identify the set of hypernodes relevant to a given query. Using Definition 3, a hypernode $n \in \mathcal{N}$ can be represented as a key-value pair that comes from the elements in the sets $\mathcal{S}$, $\mathcal{A}$, $\mathcal{V}$. A hypernode can then be considered relevant to a query if: (1) the query pertains to an attribute a of the term s, or (2) the query focuses on an object with specific values v. In other words, a hypernode is relevant if either the similarity between the key (representing concatenated entities and attributes) and the query Q is high, or the similarity between v (the value) and the query Q is high. OG-RAG finds two sets of query-relevant hypernodes: $\mathcal{N}_S(Q)$ and $\mathcal{N}_V(Q)$ to represent the two sets respectively. In particular, $\mathcal{N}_S(Q)$ denotes the top k hypernodes with the highest similarity between their attributed term, *i.e.*, $s \oplus a$ and the query Q in the vector space $\mathbf{Z}$. Similarly, $\mathcal{N}_V(Q)$ represents the top k hypernodes with the highest similarity between their value v and the query Q. Thus, for each query, we extract $2 \cdot k$ relevant hypernodes.

4.2.2 Relevant Hyeredges as Context

We form the relevant context as the set of hyperedges $\mathcal{C}_{\mathcal{H}}(Q) \subset \mathcal{E}$ that minimally cover the relevant hypernodes, $\mathcal{N}(Q) = \mathcal{N}_S(Q) \cup \mathcal{N}_V(Q)$. This is formulated as an optimization problem and solved in a greedy manner. Since the objective of minimizing the number of hyperedges is linear under a matroid constraint, the solution can be shown to be optimal (Korte et al., 2011). Specifically, we maintain a dictionary that maps each hypernode $n \in \mathcal{N}$ to the set of hyperedges that it is a part of, *i.e.*, $\mathcal{E}(n)$, where $e \in \mathcal{E}(n) \implies n \in e$. In each iteration, we add the hyperedge that covers the largest number of uncovered nodes to the context and remove those nodes from further consideration. This process is repeated until either we have L hyperedges or all the relevant nodes are covered. In this way, the context is constructed as a collection of up to L hyperedges representing *facts* relevant to the given query. By organizing information into hyperedges, OG-RAG is able to group related facts together, ensuring that the retrieved context is both compact and comprehensive, capturing all necessary facts to support accurate LLM responses, while optimizing for efficiency.

5 Experimental Setup

Datasets. We evaluate OG-RAG in two distinct domain categories that involve specialized workflows: (a) Industrial workflows, with a focus on the agriculture domain, where precise, data-driven decisions are critical for crop management and resource allocation, and (b) Knowledge work, where we evaluate it on research and analysis tasks in the news domain. For the agriculture domain, we utilize two high-quality datasets. These comprise 85 documents prepared by agriculture experts, focusing on the crop cultivation of Soybean and Wheat in India [6]. For the news domain, we use the publicly available dataset from Multi-hop RAG (Tang and Yang, 2024) to generate a hard subset of RAGAS reasoning questions, focusing complex and multifaceted news stories. This results in a total of more than $45K$ entities' attributes relationships. We provide exemplary excerpts from the datasets along with more statistics in Appendix B. For discussion on scalability to larger datasets, please refer Appendix A.3. Note that we avoid any comparison with general RAG datasets since they are specifically designed to contain data from various domains that does not align with the main goal of our work to enhance the domain-specific adaptation.

Ontology. We use a semi-automated approach to construct the ontology for both domains, which reflects the broader applicability of OG-RAG in specialized workflows. For the agriculture domain, the ontology was generated using a proprietary ontology learning module, which was then reviewed and verified by multiple experts specializing in crop cultivation. For the news domain, we modify the existing Simple News and Press (SNaP) ontology[7]. The complete ontologies for both domains are provided in Appendix C along with more details.

Large Language Models. We consider 4 large language models for zero-shot query answering while adding the retrieved context from different methods: 2 closed-box models [8] (GPT-4o-mini and GPT-4o) and 2 open-source models [9] (Llama-3.1-8B and Llama-3.1-70B). These models have been chosen for their remarkable understanding and ability to reason in natural language. We consider 4096 completion tokens and a temperature of 0.

Baselines. We compare OG-RAG against three leading retrieval-based methods, representing state-of-the-art approaches to context retrieval and generation, to demonstrate its effectiveness: **(1) RAG (Lewis et al., 2020)** that retrieves relevant contiguous chunks from documents using maximum inner product search, **(2) RAPTOR (Sarthi et al., 2024)** that clusters similar chunks and summarizes them through an LLM for additional information, **(3) GraphRAG (Edge et al., 2024)** that forms a knowledge graph of the documents by entity and relationship extraction and clustering them into semantic communities.

We use the text-embedding-3-small [8] as the sentence embedding function across all retrieval methods and GPT-4o as the LLM (*i.e.*, $\mathcal{M}_0$) for preprocessing. For each method, we find the top $\{2, 5\}$ contexts and and select the one with the highest performance. We select these values based on a tradeoff between coverage and input length constraints. For more details and analyses on all hyperparameters, refer Appendix D.1 and E.1.

Metrics. Building on the RAGAS framework (RAGAS, 2024), we use the following metrics to assess the quality of the retrieved context and the generated responses while using text-embedding-3-small as the embedding model and GPT-4o as the LLM.

1. **Context Recall (C-Rec):** Proportion of claims in the ground-truth answer that can be attributed to the retrieved context.
2. **Context Entity Recall (C-ERec):** Proportion of entities in the ground-truth answer that are present in the retrieved context.
3. **Answer Similarity (A-Sim):** Similarity between the generated response and the ground-truth answer in the embedding space.
4. **Answer Correctness (A-Corr):** A combination of answer similarity (defined above) and factual similarity, which is the F1-score between the

[6]Dataset available here: https://github.com/agaronagoovi/Multihop-Agri-QA-dataset

[7]https://iptc.org/thirdparty/snap-ontology/

[8]https://openai.com/index

[9]https://ai.meta.com/blog/meta-llama-3-1/

Method	Soybean		Wheat		News	
	C-Rec	C-ERec	C-Rec	C-ERec	C-Rec	C-ERec
RAG	0.22	0.08	0.14	0.04	0.01	0.01
RAPTOR	0.54	0.19	0.85	0.29	**0.82**	0.46
GraphRAG	0.41	0.14	0.78	0.05	-	-
OG-RAG	**0.84**	**0.41**	**0.95**	**0.34**	**0.82**	**0.52**

Table 1: **Quality of contexts retrieved by different methods for domain-specific query-answering. We found the 95% confidence interval to be ≤ 0.05 for all metrics, representing small margin of error. It is not reported here. The symbol '-' denotes that the computation did not complete within 1 day.**

claims in the ground-truth answer and those in the generated response.

5. **Answer Relevance (A-Rel):** Measures how easily the original question can be inferred from the generated response.

6 Experiments

6.1 Query answering

Question Generation. We generate a set of question/answer pairs using the RAGAS framework (RAGAS, 2024) to validate the factual accuracy of our proposed method. RAGAS prompts off-the-shelf LLM to generate questions of varying difficulty, each with the corresponding ground-truth answers and contexts. Specifically, we generate up to 100 unique questions from RAGAS focused on multi-hop reasoning abilities, which is commonly required in specialized domain tasks. Examples of these generated questions, along with their ground-truth answers, are provided in Appendix E.2.

6.1.1 Does OG-RAG retrieve more useful contexts?

A context is deemed useful for a query if it provides sufficient information to derive the ground-truth response. We evaluate this using Context Recall and Context Entity Recall. Table 1 compares the performance of different retrieval methods across three datasets. OG-RAG outperforms the baselines in almost all cases, boosting the recall of correct claims by 55% and recall of correct entities by 110%. The only exception is the News dataset where OG-RAG matches the context recall performance of RAPTOR but still delivers better performance. Note that here we select the best of either top-2 or 5 contexts for each method and defer individual performance to Appendix E.1.

6.1.2 Does OG-RAG help generate factually accurate responses?

A useful context should lead to more factual and precise response when incorporated into the query for various LLMs. We evaluate this by comparing how closely the generated responses/answers align with the ground-truth answer when added as context across different LLMs. Table 2 presents the results of response correctness, similarity, and relevance for the 3 datasets. OG-RAG consistently outperforms the baselines, significantly improving answer correctness by 40%, and answer relevance by 16%. The only notable exception where OG-RAG slightly underperforms is in the Answer Relevance for Wheat and Soybean datasets in GPT-4o and Llama-3-70B. This is likely due to the broad scope of the retrieved context, which can sometimes introduce extraneous information. This can be possibly mitigated through further fine-tuning of the hypergraph retrieval mechanism, adjusting the level of detail to suit the complexity of the queries expected. We leave domain-specific optimization for future work, as the current approach already delivers good responses across all datasets.

6.1.3 Is OG-RAG efficient?

Finally, we demonstrate that OG-RAG is computationally efficient by comparing its pre-processing and per-query retrieval times with other methods across different datasets. Table 3 shows that OG-RAG performs nearly as efficiently as a simple RAG method, with only a minimal increase of at most 2 seconds during querying time despite being at least 100% better in factual accuracy. OG-RAG is also shown to have significantly lower computational time than more competitive baselines such as RAPTOR and GraphRAG at both the pre-processing and query stages, particularly highlighted by a 50% drop in the pre-processing times. This efficiency is particularly critical for real-time applications, such as agricultural monitoring systems, legal research, and automated news fact-checking, where quick retrieval and processing of domain-specific knowledge is essential.

6.2 Context attribution

Survey design. To assess how effectively the proposed method aids humans in verifying facts within LLM-generated responses, we conduct a human study measuring the time taken to verify whether the given context supports the generated response. We randomly select 10 queries from the

Method	Soybean			Wheat			News		
	A-Corr	A-Sim	A-Rel	A-Corr	A-Sim	A-Rel	A-Corr	A-Sim	A-Rel
Llama-3-8B									
RAG	0.26	0.59	0.22	0.26	0.65	0.23	0.15	0.52	0.08
RAPTOR	0.34	**0.66**	0.59	**0.54**	**0.76**	0.67	**0.53**	0.74	0.68
GraphRAG	0.26	0.63	0.52	0.43	0.35	0.27	-	-	-
OG-RAG	**0.40**	0.65	**0.60**	**0.54**	0.73	**0.72**	0.52	**0.76**	**0.69**
Llama-3-70B									
RAG	0.27	0.59	0.19	0.26	0.65	0.14	0.17	0.58	0.09
RAPTOR	0.41	0.70	**0.64**	0.58	**0.77**	**0.75**	0.39	0.72	0.64
GraphRAG	0.30	0.65	0.55	0.47	0.37	0.29	-	-	-
OG-RAG	**0.54**	**0.75**	0.56	**0.63**	**0.77**	0.73	**0.51**	**0.77**	**0.67**
GPT-4o-mini									
RAG	0.29	0.66	0.59	0.33	0.73	0.66	0.34	0.73	0.64
RAPTOR	0.34	0.68	**0.85**	0.51	0.77	**0.88**	0.51	0.77	**0.88**
GraphRAG	0.25	0.63	0.65	0.35	0.70	0.85	-	-	-
OG-RAG	**0.48**	**0.72**	0.77	**0.62**	**0.78**	0.85	**0.62**	**0.78**	0.85
GPT-4o									
RAG	0.31	0.62	0.29	0.29	0.69	0.28	0.27	0.67	0.20
RAPTOR	0.34	0.68	0.68	0.59	**0.79**	**0.89**	0.58	0.84	**0.76**
GraphRAG	0.26	0.63	0.63	0.35	0.70	0.86	-	-	-
OG-RAG	**0.48**	**0.72**	**0.79**	**0.62**	**0.79**	0.79	**0.66**	**0.86**	0.73

Table 2: **Quality of the answers generated by different LLMs using different retrieval methods. We found the 95% confidence interval to be ≤ 0.05 for all metrics, so it is not reported here. The symbol '-' denotes that the computation did not complete within 1 day.**

Method	Soybean		Wheat		News	
	$T_{\text{pre}} \downarrow$	$T_{\text{query}} \downarrow$	$T_{\text{pre}} \downarrow$	$T_{\text{query}} \downarrow$	$T_{\text{pre}} \downarrow$	$T_{\text{query}} \downarrow$
RAG	11.41	2.49	10.55	2.36	449.21	3.56
RAPTOR	71.66	4.81	61.56	4.38	1513.57	5.45
GraphRAG	157.04	5.95	307.37	5.65	>1 day	-
OG-RAG	29.61	3.75	47.76	4.09	655.15	4.12

Table 3: **Efficiency of different retrieval methods on domain-specific query-answering. T_{pre} and T_{query} denote the average pre-processing and query time in seconds. We found the variance to be within 5 seconds, so it is not reported here.**

agriculture dataset and present the responses generated by GPT-4o using both RAG and OG-RAG, each paired with their respective contexts. We exclude RAPTOR due to its content similarity with RAG, and GraphRAG due to its prohibitive context length. Participants are asked to evaluate the level of factual support the context provides for the response on a scale of 1-4, where 1 corresponds to "no support" and 4 corresponds to "full support" for all claims. We also track the time each participant takes to complete this task. Each participant is shown 10 questions, consisting of 5 random queries, each paired with both RAG and OG-RAG responses and contexts in a randomized order. To ensure fairness, each query is presented an equal number of times across all participants. Examples of the survey design can be found in Appendix F.

Results. We recruited a total of 16 participants, aged 18-34, and familiar with LLMs within a university campus after approving their consent. Table 4 presents the average time taken and the level of support participants attributed to the contexts. We observed that OG-RAG significantly reduced the time required by 28.8% and increased the human-attributed support by 29.6% on average. Furthermore, the **median token count** for OG-RAG responses is 229 (Soybean) and 261 (Wheat), versus 264 and 278 respectively for standard RAG. These numbers show no meaningful difference in length in both approaches, while humans still find OG-RAG answers easier to attribute to context. Thus, OG-RAG not only enables faster fact verification but also provides more robust and clear contexts, making the system more user-friendly and reliable for context fact attribution.

Method	Time taken $\downarrow$	Support [1-4] $\uparrow$
RAG	61.15 ± 28.48	2.67 ± 0.30
OG-RAG	$\mathbf{43.50 \pm 18.08}$	$\mathbf{3.46 \pm 0.19}$

Table 4: **Comparison of the time taken and support given by humans for the generated responses to the contexts produced by RAG and OG-RAG, presented with 95% confidence intervals.**

6.3 Factual Deduction

Deductive Facts. We assess OG-RAG's ability to enhance deductive reasoning in LLMs by evaluating how well it can generate new conclusions based on a set of predefined facts. These facts, grounded in domain-specific ontologies, provide the framework for reasoning tasks that require multi-step logic. Specifically, for this experiment we use six agricultural facts to deduce CO2 emissions, as this information is not directly available in the documents. These facts are partially derived from industry sources on the relationship between fossil fuels, pesticides, and greenhouse gases [10]:

1. Farm area in the North Eastern Hill zone is 1 hectare or ha.
2. Farm area in North Plain Hill zone is 2 hectares or ha.
3. Herbicide production is calculated by multiplying the farm area by the recommended herbicide quantity.
4. 1 kg of herbicide production results in 18.22—26.63 kg of CO2e emissions.
5. 1 kg of insecticide production results in 14.79-–18.91 kg of CO2e emissions.
6. 1 kg of fungicide production results in 11.94—29.19 kg of CO2e emissions.

Question Generation. To create the evaluation test set, we prompt GPT-4o following the RAGAS guidelines (RAGAS, 2024) to generate questions that require the application of deductive facts

[10]Adapted in part from https://tinyurl.com/mw3jxhxk

and a randomly sampled chunk from the ontology-mapped data to generate the responses. Specifically, we use the following prompt:

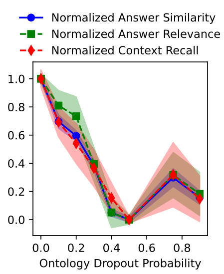
```
Given the following data and a set of deductive
rules, generate a hard question that requires the
application of the rules on the data to generate
the answer.
Data: < Domain-specific data >
Rules: < Fixed set of rules >
Question:
```

Next, we make two additional LLM calls to generate the corresponding answer and assign a rating from 1 to 10, evaluating how well the question tests the application of the rules on the data to derive the answer. We select 10 questions that receive a rating of at least 7. A full list of generated questions is provided in Appendix E.2.

Results. Table 5 presents the results of factual deductions across two agriculture datasets, using GPT-4o and GPT-4o-mini as the underlying LLMs. In all cases, except two, the OG-RAG context substantially improves the correctness, similarity, and relevance of the generated answers compared to baseline methods. This demonstrates that OG-RAG is more effective at supporting deductive reasoning from a fixed set of facts. One exception is in the Soybean dataset for answer relevance which again points to a slightly less pertinent answer due to a broader retrieved context by OG-RAG. Overall, these results confirm that OG-RAG provides a more robust context for deducing new facts. We also note an unexpected low performance of RAPTOR on Soybean with GPT-4o where the retrieved context leads GPT-4o to inaccurate final answers as measured by RAGAS metrics. This shows how deductive reasoning can be sensitive to small changes in the retrieved contexts since the same context works alright for GPT-4o-mini.

6.4 Ontology sensitivity

We also test the sensitivity of our results on the quality of ontology by randomly dropping certain attributes from the ontology with a probability. The adjoining figure shows the min-max normalized values of different metrics on Soybean at varying dropout probabilities. We observe that the results are preserved until about half of the information in the ontology is dropped.



Method	Soybean			Wheat		
	A-Corr	A-Sim	A-Rel	A-Corr	A-Sim	A-Rel
GPT-4o-mini						
RAG	0.46	0.89	0.66	0.41	0.92	0.64
RAPTOR	0.42	0.89	**0.81**	0.50	0.92	0.74
GraphRAG	0.44	0.91	0.83	0.49	0.93	0.82
OG-RAG	**0.50**	**0.92**	0.75	**0.53**	**0.94**	**0.83**
GPT-4o						
RAG	0.44	0.90	0.56	0.42	0.92	0.54
RAPTOR	0.01	0.11	0.03	0.41	0.91	0.74
GraphRAG	0.48	0.92	**0.84**	0.44	0.90	0.73
OG-RAG	**0.56**	**0.92**	0.75	**0.47**	**0.94**	**0.83**

Table 5: **Comparison of different retrieval methods in their ability to support deductive reasoning from pre-defined rules in different LLMs.**

7 Conclusion

In this work, we propose OG-RAG, a novel hypergraph-based retrieval method for domain adaptation of LLMs using ontology-grounded retrieval-augmented generation. OG-RAG has wide applicability in domains, including healthcare, law, agriculture, journalism, and research. Experiments on agriculture and news datasets show OG-RAG improves factual accuracy, accelerates answer attribution, and strengthens rule-based reasoning. By offering greater flexibility and control over how context is retrieved and utilized, OG-RAG paves the way for more adaptable and reliable language systems to incorporate controlled vocabulary and structured evidence retrieval. This not only enhances user comprehension of generated responses but also facilitates smoother integration of LLMs into industrial workflows and knowledge work. Future works can explore automated or semi-automated ontology construction techniques to build these frameworks in an end-to-end fashion, ensuring broader applicability of retrieval-augmented models across diverse domains.

Acknowledgements

This research was supported by the TRAPI service at Microsoft to access the LLM endpoints. We also extend our gratitude to the 16 participants of our study that helped us in validating the claims of attribution efficiency. Finally, we thank the anonymous reviewers in previous ARR cycles, researchers at Microsoft Research: Emre Kiciman, Swati Sharma, and Leonardo Nunes, and fellow interns for their useful feedback.

Limitations

Ontology creation. Since we require domains to have well-defined ontologies for effective domain adaptation and retrieval, we are limited by the availability of these ontologies. Here, we take the advantage of legacy systems developed in most domains such as knowledge work and industrial workflows, that have already developed detailed ontologies. However, this dependency can be a limitation in newer and fast-developing domains such as technology. We thus leave it for future works to abstract the domain knowledge as ontologies in an automated manner (Babaei Giglou et al., 2023). Our proposed hypergraph-based retrieval strategy can directly benefit from these innovations as we establish its efficacy to accurately and succinctly answer domain-specific questions given its ontology.

Human experiment. All 16 participants were university-educated people aged 18–34 with familiarity with LLMs. While larger-scale studies would yield more robust conclusions, our user study serves as a targeted, cost-conscious evaluation to gather evidence of context attribution as aligned with the scope of our work.

References

Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. 2023. Llms4ol: Large language models for ontology learning. In *International Semantic Web Conference*, pages 408–427. Springer.

Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O Nunes, and 1 others. 2024. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv e-prints*, pages arXiv–2401.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, and 1 others. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Andres M Bran, Alexandru Oarga, Matthew Hart, Magdalena Lederbauer, and Philippe Schwaller. 2025. Ontology-retrieval augmented generation for scientific discovery.

Fengyu Cai, Xinran Zhao, Tong Chen, Sihao Chen, Hongming Zhang, Iryna Gurevych, and Heinz Koeppl. 2024. Mixgr: Enhancing retriever generalization for scientific domain through complementary granularity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10369–10391.

Marco Cascella, Jonathan Montomoli, Valentina Bellini, and Elena Bignami. 2023. Evaluating the feasibility of chatgpt in healthcare: an analysis of multiple clinical and research scenarios. *Journal of medical systems*, 47(1):33.

ChatGPT. 2024. https://chatgpt.com/. [Accessed 08-10-2024].

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Kaisi Guan, Qian Cao, Yuchong Sun, Xiting Wang, and Ruihua Song. 2024. Bsharedrag: Backbone shared retrieval-augmented generation for the e-commerce domain. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1137–1158.

Nicola Guarino, Daniel Oberle, and Steffen Staab. 2009. What is an ontology? *Handbook on ontologies*, pages 1–17.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Peter Jackson. 1990. *Introduction to expert systems*. Addison-Wesley Longman Publishing Co., Inc.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Muhammad Khalifa, David Wadden, Emma Strubell, Honglak Lee, Lu Wang, Iz Beltagy, and Hao Peng. 2024. Source-aware training enables knowledge attribution in language models. *arXiv preprint arXiv:2404.01019*.

Bernhard H Korte, Jens Vygen, B Korte, and J Vygen. 2011. *Combinatorial optimization*, volume 1. Springer.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Mufei Li, Siqi Miao, and Pan Li. 2024. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.

Marjan Mernik, Jan Heering, and Anthony M Sloane. 2005. When and how to develop domain-specific languages. *ACM computing surveys (CSUR)*, 37(4):316–344.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

Perplexity. 2024. https://www.perplexity.ai/. [Accessed 08-10-2024].

RAGAS. 2024. https://docs.ragas.io/en/stable/. [Accessed 08-10-2024].

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using ood examples. *Advances in Neural Information Processing Systems*, 36:3083–3105.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *arXiv preprint arXiv:2401.18059*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. Attribute first, then generate: Locally-attributable grounded text generation. *arXiv preprint arXiv:2403.17104*.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024. Can llms reason with rules? logic scaffolding for stress-testing and improving llms. *arXiv preprint arXiv:2402.11442*.

Jinfeng Xiao, Linyi Ding, James Barry, Mohab Elkaref, Geeth De Mel, and Jiawei Han. 2024. Orag: Ontology-guided retrieval-augmented generation for theme-specific entity typing. In *First Conference on Language Modeling*.

Ran Xu, Hui Liu, Sreyashi Nag, Zhenwei Dai, Yaochen Xie, Xianfeng Tang, Chen Luo, Yang Li, Joyce C Ho, Carl Yang, and 1 others. 2025. Simrag: Self-improving retrieval-augmented generation for adapting large language models to specialized domains. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11534–11550.

Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Yanqiao Zhu, May Dongmei Wang, Joyce Ho, Chao Zhang, and Carl Yang. 2024. Bmretriever: Tuning large language models as better biomedical text retrievers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22234–22254.

Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu.

2024a. Sirerag: Indexing similar and related information for multihop reasoning. *arXiv preprint arXiv:2412.06206*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024b. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Chengshuai Zhao, Garima Agrawal, Tharindu Kumarage, Zhen Tan, Yuli Deng, Ying-Chih Chen, and Huan Liu. 2024. Ontology-aware rag for improved question-answering in cybersecurity education. *arXiv preprint arXiv:2412.14191*.

Appendix

A Method details

A.1 Retrieval-Augmented Generation

Given a user query Q and the relevant context as found above, we prompt the LLM $\mathcal{M}$ to use this context to answer the query as $\mathcal{M}(\mathcal{P}(Q, \mathcal{C}_{\mathcal{H}}(Q)))$, where $\mathcal{P}$ denotes the corresponding textual prompt:

```
Given the context below, generate the answer to
the given query. Note that the context is provided
as a list of valid facts in a dictionary format.
Context: Line-separated retrieved context C_H(Q)
Query: User-given query Q
Answer:
```

A.2 Complexity Analysis

Algorithm 2 outlines the full procedure of the proposed method which consists of two main components: (1) OG-PREPROCESS, applied to the set of documents once, and (2) OG-RETRIEVE, used to retrieve the relevant context for each query.

A.2.1 Query Complexity

Assume the context size for the LLM $\mathcal{M}_0$ is N_C. The ontology $\mathcal{O}$, which can be written in a JSON-LD or textual format, has a length $|\mathcal{O}|$, where the attributes are mapped to their corresponding ranges in the natural language vocabulary. OG-PREPROCESS phase may involves several LLM calls depending on the number of document chunks, specifically, $|\mathcal{D}|/N_C + |\mathcal{O}|$ number of calls. We do not make any additional LLM calls during the querying time in the OG-RETRIEVE procedure.

A.2.2 Time Complexity

We ignore the time taken by LLM calls while calculating the time complexity, as this is accounted for under query complexity. Thus, the time complexity of the OG-PREPROCESS step only involves the hypergraph transformation by flattening the mapped data. Let us assume we have $|\mathcal{I}|$ factual-blocks derived from the documents, and each factual-block has a maximum length of $|F|_{max} = O(|\mathcal{O}|)$. We consider two cases: **(1) Minimal or No Nesting**: In this case, the time complexity is determined by step 4 in the algorithm, leading to a complexity of $O(|\mathcal{O}||\mathcal{I}|)$, **(2) Maximum Nesting**: In this scenario, step 4 may result in an empty set. Thus, each factual-block F can be recursively flattened $\log |\mathcal{O}|$ times while searching through the entire set, leading to a time complexity of $O(|\mathcal{I}||\mathcal{O}| \log |\mathcal{O}|)$.

A.2.3 Space Complexity

The only storage required is for the hypergraph structure $\mathcal{H}(\mathcal{I})$, which is directly proportional to the number of hyperedges $|\mathcal{E}| = |\overline{\mathcal{I}}|$.

A.3 Scalability

We note that large documents may lead to extremely large hypergraphs with millions of hyperedges. However, most of the time is taken up by the pre-processing hypergraph creation step due to additional LLM calls and flattening. A desirable property of our method is that this only incurs a one-time cost along with the embedding pre-computation. At query time, our method will thus only involve a maximum inner product search over the number of hypernodes and hyperedges. If the dataset leads to a creation of a huge hypergraph, this can sometimes be prohibitive but even with 1000-sized hypergraph, we find stable and efficient performance (in Table 3 and 2).

B Datasets

Complexity of the data. While the number of documents in each dataset may appear modest, we emphasize that the knowledge base derived from these documents, via ontology grounding, is significantly richer and more complex than the raw document count suggests. The agricultural ontology (Figure 3) consists of 76 nodes (44 leaf nodes), while the news ontology (Figure 4) includes 40 nodes (19 leaf nodes). These ontologies are used to ground the document content, resulting in large and highly structured hypergraphs. Table 6 provides detailed statistics illustrating the scale and complexity of these hypergraphs. Thus, even the filtered News dataset produces a hypergraph with 7,497 nodes and 4,573 hyperedges, with individual hyperedges connecting up to 22 nodes and node degrees reaching as high as 560. These properties indicate a highly interconnected structure, where retrieval is far from trivial. The resulting scale simulates the kinds of complexity expected in larger real-world knowledge graphs.

C Ontology

C.1 Ontology Creation

For the agricultural domain, we use a proprietary ontology reviewed and verified by multiple experts on crop cultivation following the steps in the literature on ontology (Jackson, 1990; Guarino et al., 2009). For the news domain, we modify the already

Dataset	# Hypernodes	# Hyperedges	Av. nodes per hyperedge	Min. nodes per hyperedge	Max. nodes per hyperedge	Av. degree	Min. degree	Max. degree
Soybean	282	209	6.08	2	9	4.51	1	198
Wheat	253	253	5.61	3	9	5.61	1	253
News	7497	4573	9.67	0	22	5.89	1	560

Table 6: Dataset statistics

existing Simple News and Press (SNaP) ontology by simplifying it by removing the image attributes and flattening some hierarchies. Specifically, we simplify its structure by excluding certain classes, such as those related to images, videos, and the "stuff" hierarchy. Instead, we allow an asset to be linked to multiple events, and each event can be associated with multiple organizations and persons. These ontologies, as shown in Figures 3 and 4, are created by the experts to keep the abstract information of the domain, such that they can work across different documents from that domain. This can be verified by the fact that one agricultural ontology works for both Soybean and Wheat, while one News ontology is applicable to various news documents. Furthermore, multiple domains have organized their knowledge as ontologies (Jackson, 1990; Guarino et al., 2009), and we leverage these advancements directly with our method, as highlighted through the use of SNaP ontology.

C.2 Ontology Mapping Prompt

```
 Here is a context definition for a crop
cultivation ontology.
Context Definition:
{context_definition}
———-
Generate a JSON-LD using the following
data and the above context definition
for crop cultivation ontology.
Use '@graph' object namespace for the
data in JSON-LD.
Be comprehensive and make sure to fill
all of the data.
Keep nesting to the minimum and still be
able to disambiguate.
If there are multiple subfields
enumerated in a 'List' namespace then do
not combine them in a single subfield,
keep them as separate subfields to
disambiguate.
Ensure that you populate all items in the
'List' namespace, do not leave any item.
Do not include any explanations or
apologies in your response.
Do not add any other text other than the
generated JSON-LD in your response.
Generate in Json format.
————-
Data:

{data}
————
JSON-LD json:
```

C.3 Examples

Figures 3 and 4 show agriculture and news ontologies used in the work.

D Additional experimental details

All experiments were conducted using Python 3.8.12 on an Ubuntu 18.04 PC with an Intel Xeon E5-2698 v4 CPU @ 2.20GHz and 512 GB RAM. As our method primarily relies on OpenAI API queries, GPU infrastructure was not required.

D.1 Baselines

1. **RAG (Lewis et al., 2020):** RAG (Retrieval-Augmented Generation) retrieves query-relevant document chunks by embedding them into a vector space and then finding the context based on the maximum chunk-query A-Sim.
2. **RAPTOR (Sarthi et al., 2024):** RAPTOR clusters document chunks into hierarchical structures and uses an LLM to summarize the clusters as additional context. For this experiment, we set the tree depth to 3 and use the collapsed-tree retrieval strategy.
3. **GraphRAG (Edge et al., 2024):** GraphRAG retrieves from a knowledge graph constructed using an LLM by extracting entities and relationships and clustering them into semantic communities. We use default graph construction prompts and local search with community level as 2 for retrieval. Other parameters are: **Encoding:** cl100k_base, **Entity Extraction:** NLTK-based, **Vector Store:** Lancedb, **Graph Layout:** Zero graph, **Community Detection:** Graph intelligence-based, **Entity Types:** Organization, Person, Geo, Event, **Prompts:** Default

Dataset	Metric	OG-RAG Top-2	OG-RAG Top-5	RAG Top-2	RAG Top-5	Raptor Top-2	Raptor Top-5
Soybean	A-Corr	0.43	0.48	0.31	0.31	0.34	0.34
	A-Sim	0.70	0.72	0.62	0.63	0.68	0.68
	A-Rel	0.69	0.79	0.29	0.29	0.69	0.85
	C-Rec	0.74	0.84	0.15	0.22	0.54	0.59
	C-ERec	0.34	0.41	0.06	0.08	0.20	0.23
Wheat	A-Corr	0.66	0.62	0.25	0.29	0.54	0.60
	A-Sim	0.78	0.79	0.69	0.70	0.78	0.79
	A-Rel	0.77	0.79	0.26	0.28	0.87	0.89
	C-Rec	0.89	0.95	0.04	0.14	0.76	0.85
	C-ERec	0.31	0.34	0.03	0.04	0.34	0.29

Table 7: Senstivity of the number of contexts. We keep $k = 5$ when $L = 2$ for OG-RAG and $k = 10$ when $L = 5$.

for each workflow, **Search Parameters:** 0.5 text unit proportion, 0.1 community proportion, top 10 mapped entities and relationships.

E Additional results

E.1 Parameter sensitivity

Table 7 shows how the results change if we consider top-2 or top-5 contexts in each method.

E.2 Evaluation Questions

E.3 Soybean

Question: Which pest can be controlled with Imidacloprid 48 FS pesticide in soybean crops?
OG-RAG Answer: The pest that can be controlled with Imidacloprid 48 FS pesticide in soybean crops is the White Grub.

Question: What type of soil should be avoided for growing soybeans due to its medium to high nutrient holding capacity and medium water holding capacity?
RAPTOR Answer: The context does not explicitly mention a specific type of soil that should be avoided for growing soybeans due to its medium to high nutrient holding capacity and medium water holding capacity. However, it does provide recommendations for managing different soil types:

- In saline soils (pH >7.5), gypsum at 150-200 kg/ha should be applied along with organic manure. - In acidic soils (pH <6.5), lime at 600 kg/ha is recommended.

Based on this information, it can be inferred that while soybeans can be grown in various soil types, specific amendments are needed for saline and acidic soils to optimize growth. Therefore, no particular soil type is explicitly advised to be avoided, but appropriate soil management practices are necessary for different soil conditions.
OG-RAG Answer: Sandy soil should be avoided for growing soybeans due to its medium to high nutrient holding capacity and medium water holding capacity.

Question: What is the name of the crop that requires seeds with a germination rate of at least 80%, should be harvested when 85-90% pods have turned brown and lost their green color, and should be stored in a cool, dry place using hermetic storage bags for long-term storage?
RAPTOR Answer: The context provided does not mention a crop that requires seeds with a germination rate of at least 80%, should be harvested when 85-90% pods have turned brown and lost their green color, and should be stored in a cool, dry place using hermetic storage bags for long-term storage. The context specifically discusses soybean harvesting, threshing, and storage practices, including a germination rate of at least 70%, harvesting when 95% of pods change color, and storage recommendations. Therefore, based on the given context, there is no crop that matches the criteria described in the question.
OG-RAG Answer: The name of the crop is Soybean.

Question: Which soybean crop is recommended for the Central Zone and has an early sowing time on January 1st, 2016?
RAPTOR Answer: The context provided does not mention any soybean crop recommended for the Central Zone with an early sowing time on January 1st, 2016. The recommended sowing times for the Central Zone are from 20th June to 5th July. Therefore, there is no soybean crop recommended for the Central Zone with a sowing time of January 1st, 2016.
OG-RAG Answer: The soybean crop recommended for the Central Zone with an early sowing time starting on January 1st, 2016, is

the variety "JS 20-69."

Question: What is the temperature range for soybean germination and how does it compare to the temperature requirements for flowering and grain filling?
RAPTOR Answer: The provided context does not specify the exact temperature range required for soybean germination, flowering, or grain filling. Therefore, based on general agronomic knowledge:

1. **Soybean Germination**: Soybeans typically germinate best at soil temperatures between 15°C to 30°C (59°F to 86°F). Optimal germination usually occurs around 25°C (77°F).

2. **Flowering and Grain Filling**: For flowering and grain filling, soybeans generally require slightly higher temperatures. The optimal temperature range for these stages is typically between 20°C to 30°C (68°F to 86°F). Temperatures above 30°C (86°F) can negatively impact these processes, especially if they are sustained over long periods.

In summary, while soybean germination can occur at a range of temperatures starting from around 15°C, the optimal conditions for flowering and grain filling are generally within a similar but slightly higher range, emphasizing the importance of maintaining moderate temperatures throughout the growing season for optimal yield.
OG-RAG Answer: The temperature range for soybean germination is 15-30°C. For flowering, the temperature requirement is 20-30°C, and for grain filling, it is also 20-30°C. This means that the temperature range for germination is slightly broader on the lower end compared to flowering and grain filling, which have the same temperature requirements.

E.4 Wheat

Question: What is the recommended moisture level for harvesting wheat and how should the grain be dried before storage?
OG-RAG Answer: The recommended moisture level for harvesting wheat is 25%. Before storage, the grain should be dried by spreading it on tarpaulin plastic sheets in bright sunlight to a moisture level below 12

Question: Why conduct a seed germination test for wheat in relation to food production?
OG-RAG Answer: Conducting a seed germination test for wheat is essential to ensure a minimum germination rate of 85%. This is crucial for achieving an optimum plant population, which directly impacts the yield. By verifying the germination status of seeds, farmers can ensure that they plant seeds that are likely to grow successfully, leading to better food production and higher yields.

Question: What are the recommended practices for harvesting and drying wheat grains in the NEP Zone before storage?
OG-RAG Answer: The recommended practices for harvesting and drying wheat grains in the North Western Plains (NEP) Zone before storage are as follows:

1. **Harvesting:** - Harvest when the moisture level of the grain is 25- Wheat should be harvested 4-5 days before it is dead ripe. - Morning is the best time for harvesting. - For quick harvesting, use a combine harvester to avoid losses due to shattering and lodging. - In case of manual harvesting, make bundles and dry them for 3-4 days before threshing with a thresher.

2. **Drying:** - Before storage, dry the grain by spreading it on tarpaulin plastic sheets in bright sunlight. - Ensure the moisture level is reduced to below 12% before storage.

Question: What are the recommended practices for harvesting and drying wheat grains in the Northern Hills Zone before storage?
OG-RAG Answer: In the Northern Hills Zone, the recommended practices for harvesting and drying wheat grains before storage are as follows:

1. **Harvesting:** - Harvest when the moisture level of the grain is 25- Wheat is generally manually harvested, but a combine harvester can be used for quick harvesting to avoid losses due to shattering and lodging. - Harvest 4-5 days before the wheat is dead ripe.

- Morning is the best time for harvesting. - In case of manual harvesting, make bundles and dry them for 3-4 days before threshing with a thresher.

2. **Drying:** - Before storage, dry the grain by spreading it on tarpaulin plastic sheets in bright sunlight to reduce the moisture level below 12%.

E.5 News

Question: Which earbuds are stylish and see-through?
OG-RAG Answer: The earbuds that are stylish and see-through are the "Nothing Ear Stick."

Question: What else debuts with 'Hurricane Season' on Netflix Nov. 1?
OG-RAG Answer: On Netflix Nov. 1, along with "Hurricane Season," the following titles debut:

- Locked In (Netflix Film) - Mysteries of the Faith (Netflix Documentary) - Nuovo Olimpo (Netflix Film) - Till Murder Do Us Part: Soering vs. Haysom (Netflix Documentary) - Wingwomen (Netflix Film) - 13 Going on 30 - 13 Hours: The Secret Soldiers of Benghazi - 60 Days In: Season 5 - The Addams Family - The Big Lebowski - The Change-Up - Desperado - Downsizing - Drag Me to Hell - Love in the Wild: Season 1 - Madea's Family Reunion - The Mummy (2017) - Paul Blart: Mall Cop - Paul Blart: Mall Cop 2 - Pitch Perfect - The Scorpion King - Scott Pilgrim vs. the World - Sherlock Gnomes - Six Feet Under: Seasons 1-5 - Sixteen Candles - The Social Network - Ted 2 - Victorious: Season 3 - Whiplash

Question: How do price-match policies curb customer loss in big sales?
OG-RAG Answer: Price-match policies help curb customer loss during big sales by ensuring that customers can get the best price without needing to shop around at multiple retailers. By offering to match lower prices found elsewhere, retailers like Best Buy, Walmart, Dell, eBay, and Lowe's aim to keep customers satisfied and loyal, preventing them from purchasing from competitors. This strategy is particularly effective during major sales events like Black Friday and Cyber Monday, when price competition is intense. By guaranteeing competitive pricing, retailers can maintain their customer base and potentially attract new customers who are looking for the best deals.

Question: Who wrote about AI risks and the creative class?
OG-RAG Answer: Daniel Tencer wrote about AI risks and the creative class.

Question: Who hit the Hawks' game-winning buzzer-beater vs. the Taipans?
OG-RAG Answer: Tyler Harvey hit the Hawks' game-winning buzzer-beater vs. the Taipans.

Question: Which Chrome extension by Steven Tey aids in sharing AI replies?
OG-RAG Answer: The Chrome extension by Steven Tey that aids in sharing AI replies is called "ShareGPT."

Question: Which artists thrived under Michelle Jubelirer at Capitol Music Group?
OG-RAG Answer: The artists that thrived under Michelle Jubelirer at Capitol Music Group include Ice Spice and The Beatles.

Question: Why are JLab earphones seen as a budget-friendly backup with EQ presets?
OG-RAG Answer: JLab earphones are considered a budget-friendly backup because they offer great value by including three different EQ presets directly into the earphones. This feature enhances the listening experience without a high cost, making them an attractive option for those seeking an affordable alternative to their main wireless earphones.

E.6 Factual Deduction

Question: Calculate the total CO2e emissions from herbicide production for a farm located in the North Eastern Plains Zone, using the recommended herbicide for Phalaris minor.

Assume the farm area is 1 hectare. Provide the range of CO2e emissions based on the given data and rules.
OG-RAG Answer: To calculate the total CO2e emissions from herbicide production for a farm located in the North Eastern Plains Zone, we need to follow these steps:

1. **Identify the Recommended Herbicide and Quantity:** - The recommended herbicide for Phalaris minor is Clodinafop. - The recommended quantity is 60 gm a.i. (active ingredient) per hectare.

2. **Calculate the Total Herbicide Required:** - Since the farm area is 1 hectare, the total herbicide required is 60 grams.

3. **Convert Grams to Kilograms:** - 60 grams is equivalent to 0.06 kilograms.

4. **Calculate CO2e Emissions:** - The CO2e emissions for herbicide production range from 18.22 to 26.63 kg of CO2e per kilogram of herbicide. - For 0.06 kg of herbicide, the CO2e emissions are calculated as follows: - Lower bound: 0.06 kg * 18.22 kg CO2e/kg = 1.0932 kg CO2e - Upper bound: 0.06 kg * 26.63 kg CO2e/kg = 1.5978 kg CO2e

Therefore, the total CO2e emissions from herbicide production for the farm are in the range of 1.0932 to 1.5978 kg CO2e.

Question: Calculate the total potential CO2e emissions from herbicide, insecticide, and fungicide production for a farm located in the North Plain zone, assuming the recommended herbicide quantity is 2 kg per hectare, insecticide quantity is 1.5 kg per hectare, and fungicide quantity is 1 kg per hectare. Use the average CO2e values for your calculations.
OG-RAG Answer: To calculate the total potential CO2e emissions from herbicide, insecticide, and fungicide production for a farm located in the North Plain zone, we need to follow these steps:

1. **Determine the farm area**: The farm area in the North Plain Hill zone is 2 hectares.

2. **Calculate the total quantity of each type of chemical used**: - Herbicide: 2 kg/ha × 2 ha = 4 kg - Insecticide: 1.5 kg/ha × 2 ha = 3 kg - Fungicide: 1 kg/ha × 2 ha = 2 kg

3. **Use the average CO2e values for each type of chemical**: - Herbicide: Average CO2e = (18.22 + 26.63) / 2 = 22.425 kg CO2e/kg - Insecticide: Average CO2e = (14.79 + 18.91) / 2 = 16.85 kg CO2e/kg - Fungicide: Average CO2e = (11.94 + 29.19) / 2 = 20.565 kg CO2e/kg

4. **Calculate the total CO2e emissions for each type of chemical**: - Herbicide: 4 kg × 22.425 kg CO2e/kg = 89.7 kg CO2e - Insecticide: 3 kg × 16.85 kg CO2e/kg = 50.55 kg CO2e - Fungicide: 2 kg × 20.565 kg CO2e/kg = 41.13 kg CO2e

5. **Calculate the total potential CO2e emissions**: - Total CO2e = 89.7 kg + 50.55 kg + 41.13 kg = 181.38 kg CO2e

Therefore, the total potential CO2e emissions from herbicide, insecticide, and fungicide production for the farm in the North Plain zone is 181.38 kg CO2e.

F Survey design

Figure 5 shows the design of our survey with the introduction and one exemplary question.

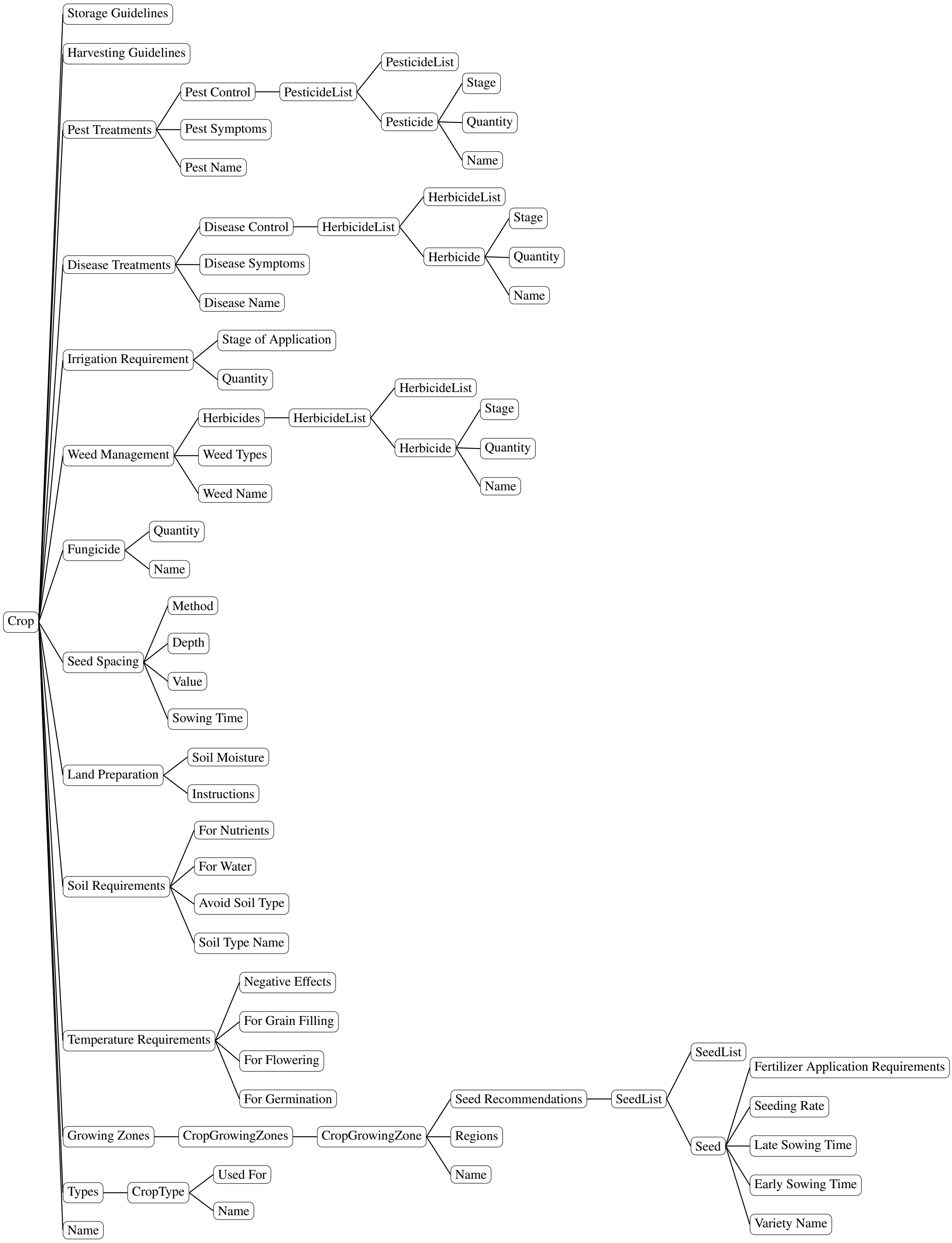


Figure 3: Agriculture ontology

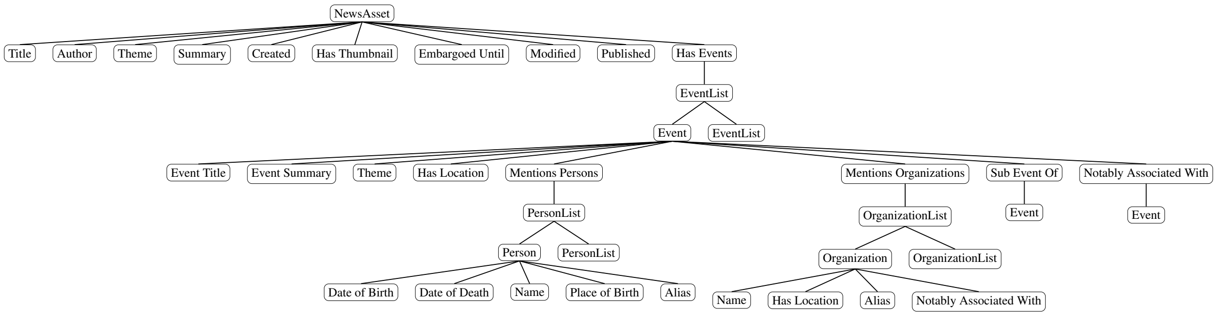
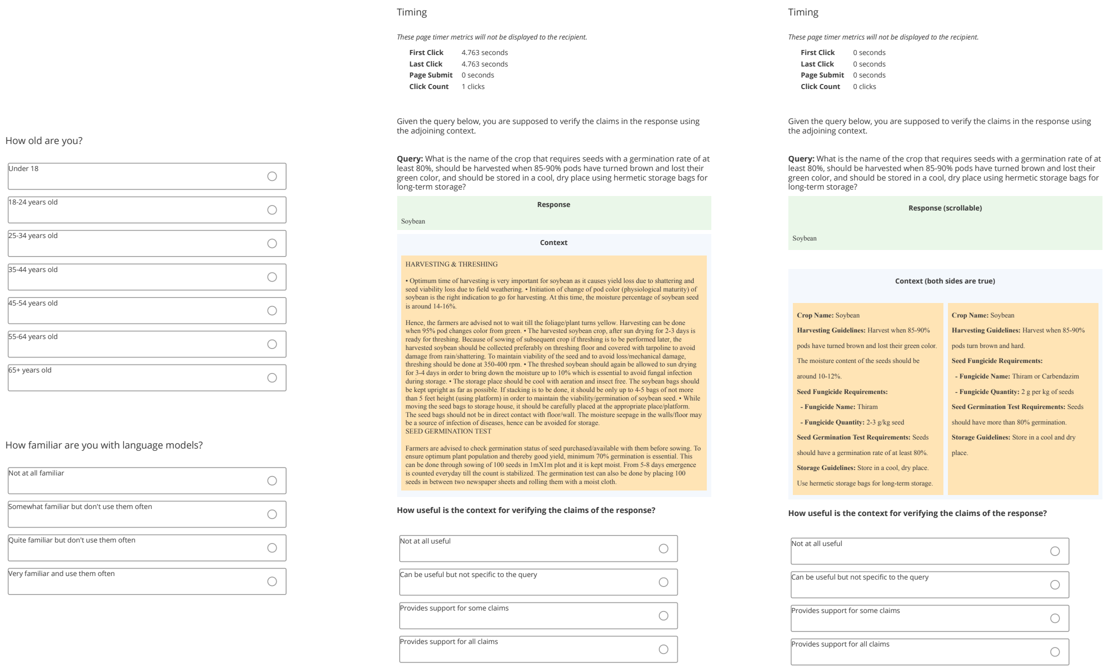


Figure 4: News ontology



(a) Introduction page (b) RAG context attribution (c) OG-RAG context attribution

Figure 5: Survey design