KONDA: An LLM-based Tool for Semantic Annotation and Knowledge Graph Creation Using Ontologies for Research Data

Soo-Yon Kim[1,*,†], Martin Görz[1,†] and Sandra Geisler[1]

[1]*Data Stream Management and Analysis, RWTH Aachen University, Ahornstr. 55, Aachen, 52074, Germany*

Abstract

The increasing demand for FAIR (Findable, Accessible, Interoperable, and Reusable) research data management (RDM) practices has underscored the need for tools that support semantic annotation and structuring of datasets. However, integrating ontologies and knowledge graphs into research workflows often presents a technical, cognitive, and time-consuming challenge for researchers. We investigate how large language models (LLMs) can be leveraged to lower this barrier by guiding researchers through the ontology-aligned annotation and transformation of heterogeneous research data into knowledge graphs. To this end, we design and implement a modular, domain-agnostic tool that is able to process and annotate various types of research data by combining methods such as vector-based semantic matching, human-in-the-loop validation, and integration with terminology lookup services and graph databases. The tool provides a guided end-to-end workflow, from data and context ingestion to entity and relation extraction, ontology-based annotation, and graph assembly, designed to support non-expert users through an intuitive, lightweight web interface. Results from a usability survey show that the tool is considered practical and useful. We argue that LLMs, when embedded in an interactive and user-centered system, can significantly enhance the accessibility and effectiveness of semantic research data management. Future work may investigate tool performance in real-world settings.

Keywords

semantic annotation, knowledge graphs, ontologies, large language models, research data management

1. Introduction

The growing volume, complexity, and relevance of research data for knowledge production have made effective data management a central concern across scientific disciplines. In response, the FAIR principles (Findable, Accessible, Interoperable, and Reusable) [1] were introduced to promote data management practices for enhanced transparency and reproducibility.

Among the goals of FAIR, interoperability plays a central role. It enables the meaningful exchange and integration of data across systems, disciplines, and institutions [2]. One important strategy for achieving interoperability is the use of semantic technologies [3, 4]. Structured and semantically enriched data improve the unambiguous comprehensibility and machine-readability of datasets and support automated reasoning [5]. Knowledge graphs, in particular, make it possible to visualize and explore relationships within data and support querying across entities and contexts [6]. In increasingly data-driven research environments, aligning heterogeneous datasets through shared ontologies and creating expressive knowledge graphs contributes to cumulative knowledge production and interdisciplinary collaboration.

However, achieving semantic interoperability often falls short in practice [7]. Creating semantically structured data involves complex processes such as ontology-based annotation and the creation of knowledge graphs. These processes do not only require conceptual and technical expertise with ontologies and graph-based technologies, but can also be highly time-consuming, which poses a barrier for researchers working under limited time and resource constraints [8, 9]. In addition, deciding which

5th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment, Nov 2025, Nara, Japan

[*] Corresponding author.

[†] These authors contributed equally.

kim@dbis.rwth-aachen.de (S. Kim); martin.goerz@rwth-aachen.de (M. Görz); geisler@dbis.rwth-aachen.de (S. Geisler)
https://sandra-geisler.de/ (S. Geisler)
0000-0001-5975-0031 (S. Kim); 0009-0009-6087-0183 (M. Görz); 0000-0002-8970-6282 (S. Geisler)

concepts to use and how to apply them often requires an initial coordination effort among collaborators, particularly in interdisciplinary projects. This adds another layer of complexity to ontology-based work, even before any technical implementation begins.

Although a wide range of ontologies and semantic web frameworks and tools exist[1,2,3], their integration into everyday research workflows remains limited. The tools frequently require data in specific formats or assume a high level of data quality, which is often unrealistic in heterogeneous research contexts. Further, many tools are designed for specific domains and therefore offer limited applicability outside their original scope. Lastly, solutions are often technically cumbersome to operate and offer limited support for users without prior experience in semantic technologies. As a result, much of the research data produced today remains poorly described, which limits its potential for integration, discovery, and reuse across domains.

Addressing this gap calls for a tool that provides low-barrier access to semantic technologies, supports diverse data types and domains, and enables the gradual annotation and transformation of data without demanding extensive upfront modeling. Large language models (LLMs) offer promising capabilities in this regard, as they can be operated with natural language, are able to process various types of data, and can conduct semantic matching.

Building on these considerations, we developed KONDA (Knowledge graph creation and ONtology enrichment for research DAta), a tool that builds on LLM functionalities to facilitate semantic annotation and knowledge graph creation in research contexts. It supports users in transforming research data into semantically enriched graph representations through a guided, end-to-end workflow. The system integrates modular key functionalities into a single web interface, including data upload, terminology lookup, ontology embedding, entity and relation extraction, semantic similarity matching, graph visualization, and export options, with continuous human-in-the-loop validation.

The contributions of the tool presented in this paper are as follows. (1) The tool is agnostic with respect to both domain and data format. (2) It enables the reuse of existing ontologies. (3) Semantic annotation is semi-automated through a combination of LLM-based recommendations and human-in-the-loop validation. (4) A knowledge graph construction component supports the visual exploration of the semantically enriched data. (5) All steps are unified under a single web interface. (6) The modular architecture supports self-hosting of the tool and the deployment of various LLMs.

To assess the tool's usability and practical value, we conducted a feasibility demonstration and a user study focusing on the accessibility of the workflow, the quality of the generated suggestions, and the overall fit with real-world research practices. The results indicate that KONDA functions as intended, and is perceived as useful and approachable, even by users without prior experience in semantic technologies. The findings support the potential of LLM-based systems to lower the entry threshold for semantic data management and to foster broader adoption of FAIR-aligned practices.

The remainder of the paper is structured as follows. Section 2 reviews related work in ontology-based annotation and knowledge graph creation, and the application of language models in semantic research data management. Section 3 develops a solution concept. Section 4 describes the system architecture and implementation of KONDA. Section 5 presents the evaluation design and results. Section 6 discusses key insights and limitations. Section 7 summarizes the paper and gives implications for future development.

2. Related Work

Various tools and approaches have been developed that address semantic technologies for research data management using LLMs. These works cover areas such as ontology-based annotation, knowledge graph generation, and holistic semantic research data management systems. In the following, we review related work and position it in relation to the goals and design of KONDA.

[1] https://protege.stanford.edu/ (accessed on 25.07.2025)

[2] https://bioportal.bioontology.org/annotator (accessed on 25.07.2025)

[3] https://inception-project.github.io/ (accessed on 25.07.2025)

Ontology-based annotation. Recent works have explored how LLMs can support the semantic annotation tasks of entity and relation recognition and their linking to ontology concepts. Tools such as BioLinkerAI [10] and frameworks such as NSSC [11] focus on the medical domain, where entities identified in clinical or biomedical texts are linked to terms of the Unified Medical Language System[4]. NeOn-GPT [12] combines a systematic ontology engineering methodology with prompt-based interactions to translate domain descriptions into formal representations. OntoKGen [13] provides an interactive workflow for identifying domain-relevant concepts and relations in technical documents and helps users formalize these into custom ontologies. These approaches apply different methods: Some combine rule-based methods for entity and relationship extraction with LLM-based methods for selecting and disambiguating candidate concepts based on contextual information. Other systems show how LLMs can support the construction of semantic resources from text. However, there are some limitations. Many tools rely on the existence of predefined rules and established ontologies, requiring a substantial amount of a-priori symbolic knowledge and making the solutions not suitable for use across diverse research domains. In the works surrounding ontology integration, there is a focus primarily on generating new ontologies, and little support is offered for integrating existing ontologies into data annotation workflows.

Knowledge graph creation. Further works explore how LLMs can support the construction of knowledge graphs. Yang et al. [14] have proposed a method for having LLMs expand ontologies based on competency questions and subsequently construct knowledge graphs. In OntoKGen [13], ontologies can be custom-built and then be used to generate knowledge graphs. Vizcarra et al. [15] have developed a system for constructing a knowledge graph which describes user-product interactions. While these approaches present effective solutions for knowledge graph construction, they underscore the importance of ontology integration in ensuring that the resulting graphs are semantically meaningful. However, their reliance on domain-specific prior work for the integration of ontologies limits the scalability of the suggested pipelines across different domains.

End-to-end semantic data management systems. Few works address the holistic (semi-)automated support of semantic research data management. The Leibniz Data Manager [16] integrates automatic components for the annotation of data with concepts e.g. from Wikidata[5], its integration into a knowledge graph, and querying functions. Further systems such as AGENTiGraph [17] and the materials science platform NOMAD [18], while offering functionalities for semantic data management, focus the automated support rather to downstream tasks such as analysis. Hence, there is a general lack of systems that provide end-to-end, (semi-)automated pipelines for semantic research data management workflows.

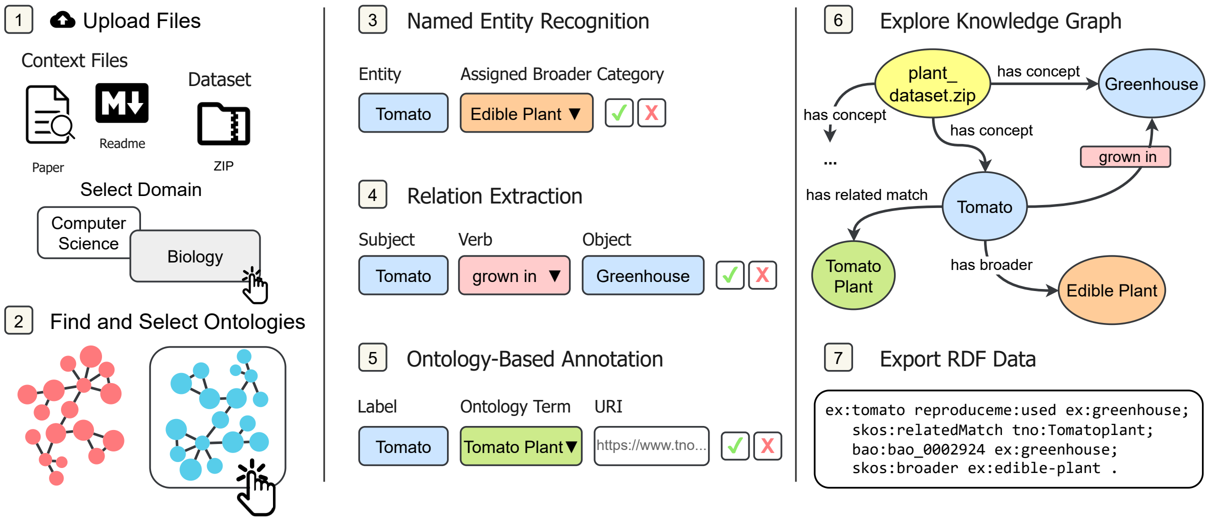
While these existing systems cover a range of tasks related to semantic annotation and graph construction, important aspects remain underdeveloped. First, few approaches support the semantic transformation of heterogeneous research data in a way that is applicable across domains and data formats. Second, existing tools rarely facilitate the reuse of established ontologies during annotation, often focusing on the creation of new ontologies instead. Third, available solutions are often fragmented across different tools and interfaces, lacking a unified system for end-to-end ontology-based annotation and knowledge graph creation, with a specific focus on low-barrier usability for time- and resource-constrained researchers. Together, these gaps motivate the development of a system that combines these components into a single, accessible workflow.

3. Methodology

To address these gaps, we propose KONDA, a tool designed to support researchers in structuring their data using ontologies and generating machine-readable knowledge graphs.The conceptual foundation of KONDA centers around a modular pipeline architecture, in which LLMs are embedded into a transparent, human-in-the-loop workflow. The core design goal is to preserve semantic rigor while

[4] https://www.nlm.nih.gov/research/umls/index.html (accessed on 23.07.2025)

[5] www.wikidata.org/ (accessed on 25.07.2025)



```
ex:tomato reproduceme:used ex:greenhouse;
   skos:relatedMatch tno:Tomatoplant;
   bao:bao_0002924 ex:greenhouse;
   skos:broader ex:edible-plant .
```

Figure 1: Overview of the app workflow for annotating and transforming research data into a knowledge graph. The process includes uploading files and selecting a domain, finding relevant ontologies, extracting entities and relations, annotating entities with ontology terms (with user validation at each stage), and finally exploring and exporting the resulting knowledge graph.

reducing cognitive and technical overhead, enabling users to guide, validate, and refine the semantic transformation of their data.

At the highest level, the tool guides the researcher through a sequence of seven conceptual stages, depicted in Figure 1: (1) data ingestion, (2) ontology selection, (3) entity and (4) relation extraction, (5) ontology-based annotation, (6) structured knowledge graph assembly, and (7) export. Each stage is conceptually modular, with clearly defined intermediate results that are presented to the user in natural language for inspection and revision, the mechanism of which is depicted in Figure 6 in Section A of the appendix. This design ensures not only transparency and autonomy for researchers, but also adaptability to different research domains and varying degrees of semantic web familiarity.

The process begins with the ingestion of heterogeneous research materials. As illustrated in stage (1) of Figure 1, users upload both a research dataset and a collection of supplementary documentation (e.g., README files, related publications, notes, or protocols) that contextualize the dataset. Because raw research data often lacks sufficient semantic context, the tool uses few-shot prompting with an LLM to generate summaries of the accompanying documentation [19]. This stage creates a compact, coherent representation of the research context, which is reused across subsequent tasks to ground entity recognition, relation modeling, and ontology alignment. This design decision emerged from our early stage experiments, where the presence or absence of context dramatically affected the relevance and specificity of extracted entities.

Once the semantic context has been established, users proceed to the ontology selection stage (2). To support this stage, the tool offers a keyword-based ontology search interface integrated with a terminology lookup service[6] to query relevant domain ontologies. Researchers may select ontologies from this search or upload custom ontologies as needed. This stage ensures that subsequent extraction and alignment tasks are grounded in vocabularies that reflect the intended domain.

In stage (3), KONDA prompts the LLM to identify a set of relevant entities that capture the key concepts in the dataset, guided by both the extracted context and the chosen ontologies. To structure these entities more effectively, they are grouped into broader semantic categories that provide scaffolding for interpretation and alignment. For example, in a community gardening dataset, the tool may extract "Chamomile" as an entity and assign it the category "Herbal Plant". These categories improve ontology mapping in stage (5). The extracted entities and categories are presented to the user through an

[6] https://terminology.tib.eu/ts/ontologies (accessed on 25.07.2025)

interactive interface. Each extraction can be reviewed, edited, or extended, ensuring human-in-the-loop validation. Users may adjust the number of extracted items to control complexity as well as granularity. The inputs remain in natural language at this stage, therefore lowering the entry barrier while guiding alignment in downstream stages to ensure both usability and semantic consistency.

Next, the tool identifies relationships between the validated entities in stage (4). Prompted again with contextual information and the current list of entities and categories, the LLM proposes subject, predicate, object triples that represent meaningful semantic links. Relationship labels are initially suggested in plain language, such as "is grown in" or "is used for", and are subject to user revision. Only entities validated in the previous stage are used as subject and object nodes, ensuring referential integrity across the graph. This iterative model avoids ambiguous relationships and allows the researcher to use their use case-specific knowledge to validate results.

While entity and relation extraction form the conceptual backbone of the knowledge graph, representing them as free-form text alone does not guarantee interoperability. To address this, the tool includes a dedicated ontology-based annotation stage. Here, each free-form entity, category, and relationship is mapped to a formal ontology term using a two-step strategy, shown in stage (5) in Figure 1. First, the tool performs a vector-based similarity search: embeddings are computed for both user-defined terms and ontology labels (including descriptions) using an embedding model. Second, these candidates are matched through LLM-assisted reasoning: the model evaluates conceptual fit between the user term and the ontology candidate, taking into account the dataset's context summaries. Final suggestions are presented to the user, who can approve, override, or reject each mapping. If no suitable match is found, users may fall back to their original term, preserving semantic intent while maintaining a clear record of unmatched items.

A key design consideration is to support a flexible representation of user-defined terms and their ontology alignments in the final graph. During annotation, users can choose whether to replace their original label with the matched ontology term, which is useful when an exact or highly appropriate match is found, or to retain the original label and link it to the ontology term via a looser semantic relation (e.g., *skos:relatedMatch*). This approach maintains both formal interoperability and the expressive nuance of domain-specific terminology, accommodating cases where exact matches do not exist or where preserving original phrasing is important for interpretation.

The final stages are the knowledge graph assembly stage (6), in which all prior components, entities, categories, relations, and ontology annotations are combined into a structured, machine-readable graph, and the export stage (7). The graph is centered around a dataset node that links to contextual documents, extracted entities (coined *concepts*), and their broader *categories*. Edges between entities reflect validated relationships, and each node optionally links to an ontology term via explicit match relations. The schema is depicted in Figure 7 in Section A of the appendix and is designed to be compatible with SKOS[7] and other common vocabularies, ensuring downstream compatibility with query engines, metadata registries, and discovery platforms. All outputs are exportable and conform to standard formats, supporting integration into broader scientific knowledge graph infrastructures.

In summary, KONDA supports ontology-based annotation and knowledge graph generation by embedding LLMs into an interactive, user-centered pipeline. It transforms heterogeneous, informal research data into semantically structured, ontology-aligned graphs while keeping the researcher in control at every step. The next section details the implementation of this conceptual workflow, including tool architecture, backend components, and the orchestration of LLM tasks and user interaction.

4. Implementation

Building upon the conceptual workflow outlined in the previous chapter, the tool was implemented as a modular web-based application designed to be a user-friendly and guided tool for researchers. The implementation emphasizes accessibility, responsiveness, and adaptability to different domains and dataset formats. These priorities are reflected in both the tool's architecture and its choice of technologies.

[7] https://www.w3.org/2004/02/skos/ (accessed on 23.07.2025)

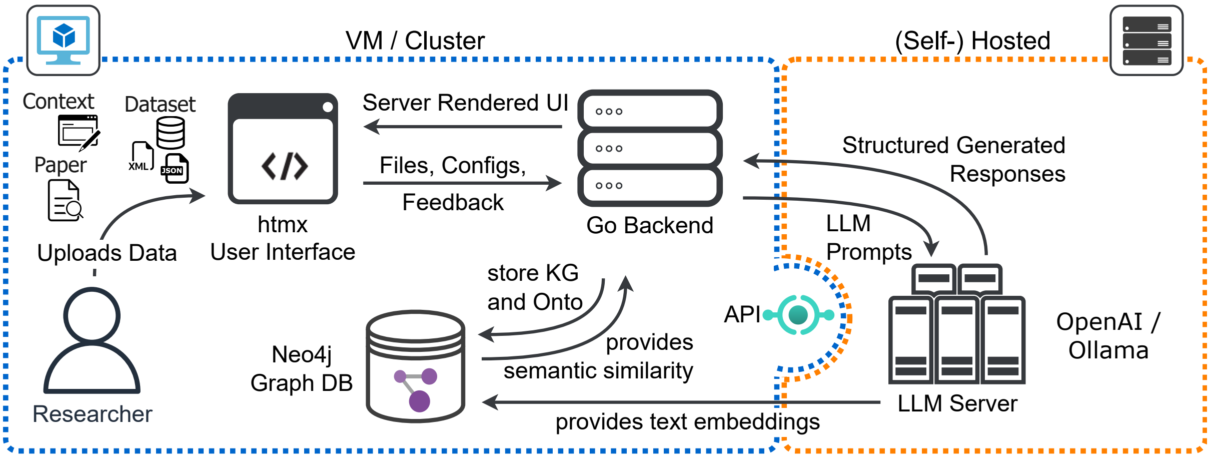


Figure 2: System architecture of the large language model (LLM)-based tool for research data management (RDM), illustrating a researcher uploading data through an HTMX-based user interface (UI). The backend, built in Go, processes the data and communicates with various LLM servers (OpenAI, or models in Ollama) to generate structured outputs and stores them in a Neo4j graph database. This modular setup allows the flexible LLM selection, supporting both commercial and self-hosted configurations.

This chapter details how the conceptual components, i.e., file preprocessing, contextual enrichment, semantic structuring, ontology alignment, and knowledge graph construction, were translated into a fully operational tool, while maintaining human-in-the-loop validation.

The application follows a client-server architecture in which responsibilities are clearly separated between a dynamic, form-driven web frontend and a processing-intensive backend. This separation allows for asynchronous background processing while keeping the user experience lightweight and responsive. Researchers interact with the tool through a stepwise interface composed of dynamically updated forms, while the backend handles file extraction, interaction with LLMs, vector similarity searches, and graph database operations, as shown in Figure 2.

The frontend was developed using a minimalist stack that avoids complex frameworks, in favor of technologies that support a sequential interaction model. A combination of HTMX[8] and Go's[9] native templating system is used to render dynamic forms without requiring a single-page application framework. HTMX enables partial page updates triggered by user input or server responses, keeping the interface reactive without heavy client-side logic. TailwindCSS[10] and AlpineJS[11] are used to provide styling and minor interactivity, such as toggling inputs or dynamically updating sections. This approach focuses on providing a guided user interaction through sequentiality while allowing for dynamic feedback and validation at each stage of the pipeline.

The backend is implemented in Go and is responsible for all computational tasks. The server manages session-specific workspaces, processes uploaded files, orchestrates background tasks, generates and interprets LLM prompts, and constructs the final knowledge graph in a graph database. Each researcher session is isolated, with its own temporary storage and database instance, ensuring data privacy and system stability.

Once a user initiates a session by uploading files, the backend begins processing them asynchronously. A file type detection mechanism is used to determine the appropriate file extractor for each input. Extractors are format-specific and capable of operating in either full or sampled mode depending on file size. For example, large CSV files are processed by extracting headers and a sample of rows to preserve structure without exceeding LLM input limits, while PDF or Markdown files are converted into plain text for summarization. These extracted segments are then fed into structured LLM prompts

[8] https://htmx.org/ (accessed on 23.07.2025)
[9] https://go.dev/ (accessed on 23.07.2025)
[10] https://tailwindcss.com/ (accessed on 23.07.2025)
[11] https://alpinejs.dev/ (accessed on 23.07.2025)

designed to generate dataset and context summaries. The resulting summaries form the semantic basis for downstream processing and are reused across prompts to reduce cost and redundancy.

To manage these asynchronous operations, the tool introduces a task orchestration layer in which each transformation step (e.g., summarization, extraction, or alignment) is defined as a discrete task with explicit dependencies. If a user modifies an input or updates a configuration, dependent tasks are automatically invalidated and marked for recomputation. This approach ensures that all outputs remain consistent with the current state of the data while supporting a responsive and fluid user experience.

LLM integration is abstracted behind an API layer supporting OpenAI and Ollama style API calls[12]. Prompts are constructed using Go templates that encapsulate few-shot examples, context, and output format instructions. For each major transformation task, i.e., entity recognition, relation extraction, and ontology annotation, the tool constructs structured few-shot prompts in a consistent format. The output is always constrained to JSON, enabling reliable parsing and integration into the backend. While the tool was primarily tested with Azure-hosted models[13] for production stability, the API interface supports any OpenAI-compatible provider, including self-hosted alternatives with Ollama. This allows researchers to select between commercial and private deployment scenarios based on their requirements for performance, cost, or data sensitivity.

A key implementation detail lies in the handling of ontology alignment. All user-uploaded ontologies, as well as those selected through the in-app ontology search interface, are parsed and stored in a dedicated Neo4j[14] graph database. Neo4j was selected for its ability to efficiently store and traverse graph-structured data, its flexible node and edge modeling, and its capability to easily generate and store text embeddings. Each ontology class, attribute, or property, and, if available, its description, is embedded using OpenAI's text-embedding-3-large[15] model. These embeddings are stored in a vector index within the graph database and used to retrieve semantically similar candidates when annotating extracted entities and relations. A two-level structure is used: a session-specific vector store for selected domain-specific ontologies, and a shared foundational store that covers domain-agnostic vocabularies. This layered approach ensures broad term coverage while prioritizing relevant domain-specific matches.

The matching process combines semantical and reasoning-based techniques. First, cosine similarity is used to retrieve candidate matches from the vector stores. The matches are ranked by their similarity score. Then, the best ranked candidates are passed to an LLM along with the dataset and context summaries. The LLM selects the most appropriate ontology term for each extracted element, based on semantic fit. This dual-step matching process balances scalability with contextual understanding and supports the alignment of informal labels to formal vocabularies across disciplines.

The final knowledge graph is constructed using a Neo4j instance dedicated to each user session. This graph captures the structure and semantics of the dataset, enriched by context and ontology alignment. Nodes are created for the dataset itself, context files, extracted entities, categories, and matched ontology terms. Relationships such as "has concept", "has broader", and predicate triples between entities are encoded as edges, using either matched ontology URIs or generated URIs based on the user's configured namespace. Where applicable, SKOS vocabulary is used to represent hierarchical relationships and ontology matches, enabling future inference or reasoning.

Neo4j also provides the infrastructure for in-app visualization and export. The final graph is rendered in an interactive viewer, allowing researchers to explore and verify the result. Export options include RDF/XML[16], Turtle[17], and JSON-LD[18] formats, ensuring compatibility with semantic web standards.

To manage computational resources, the tool implements a cleanup routine that removes session-specific data, including files, summaries, vector indexes, and Neo4j instances, after a period of inactivity. This ensures scalability across users while maintaining a consistent and responsive experience.

[12] https://ollama.com/blog/openai-compatibility (accessed on 23.07.2025)

[13] https://azure.microsoft.com/en-us/products/ai-services/openai-service (accessed on 23.07.2025)

[14] https://neo4j.com/ (accessed on 23.07.2025)

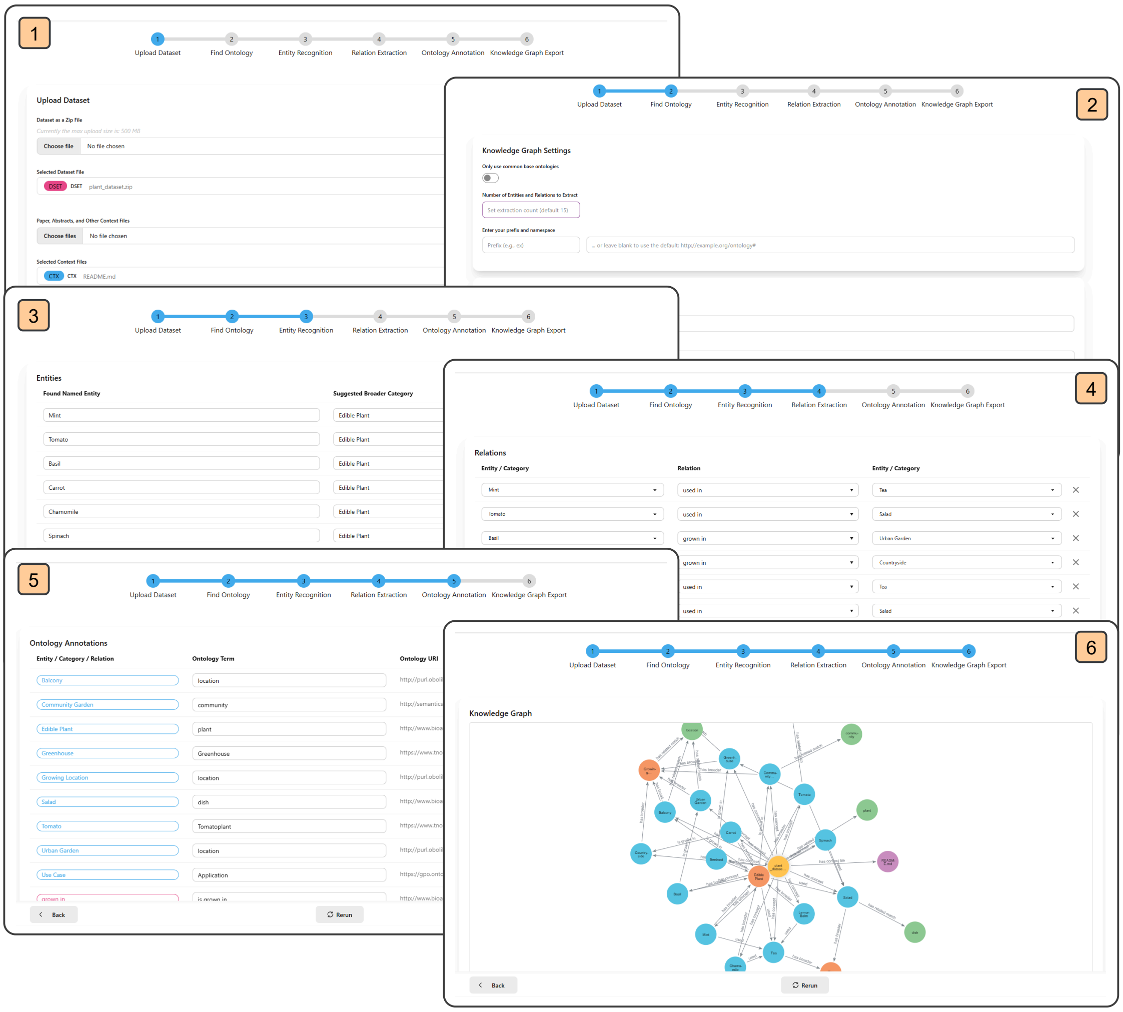
[15] https://platform.openai.com/docs/models/text-embedding-3-large (accessed on 26.09.2025)

[16] https://www.w3.org/TR/rdf-syntax-grammar/ (accessed on 23.07.2025)

[17] https://www.w3.org/TR/turtle/ (accessed on 23.07.2025)

[18] https://json-ld.org/ (accessed on 23.07.2025)

A collage of the different screens of the interface is depicted in Figure 3.



Upload Dataset

Dataset as a Zip File

Currently the max upload size is: 500 MB

Choose file No file chosen

Selected Dataset File

DSET DSET plant_dataset.zip

Paper, Abstracts, and Other Context Files

Choose files No file chosen

Selected Context Files

CTX CTX README.md

Knowledge Graph Settings

Only use common base ontologies

Number of Entities and Relations to Extract

Set extraction count (default 15)

Enter your prefix and namespace

Prefix (e.g., ex)

... or leave blank to use the default: http://example.org/ontology#

Entities

Found Named Entity	Suggested Broader Category
Mint	Edible Plant
Tomato	Edible Plant
Basil	Edible Plant
Carrot	Edible Plant
Chamomile	Edible Plant
Spinach	Edible Plant

Relations

Entity / Category	Relation	Entity / Category
Mint	used in	Tea
Tomato	used in	Salad
Basil	grown in	Urban Garden
	grown in	Countryside
	used in	Tea
	used in	Salad

Ontology Annotations

Entity / Category / Relation	Ontology Term	Ontology URI
Balcony	location	http://purl.oboli
Community Garden	community	http://semantics
Edible Plant	plant	http://www.bioa
Greenhouse	Greenhouse	https://www.tno
Growing Location	location	http://purl.oboli
Salad	dish	http://www.bioa
Tomato	Tomatoplant	https://www.tno
Urban Garden	location	http://purl.oboli
Use Case	Application	https://gpo.onto
grown in	is grown in	http://www.bioa

Knowledge Graph

Figure 3: A step-by-step collage of the KONDA application interface, illustrating the full user workflow: from dataset upload and ontology selection to entity recognition, relation extraction, ontology-based annotation, knowledge graph construction, and export. Each numbered screen represents a sequential stage in the conceptual pipeline, with the two last stages merged in one screen.

In summary, the implementation effectively realizes the conceptual framework outlined previously by combining minimalistic, sequential user interaction with powerful backend automation. Through a modular architecture, robust background processing, provider-flexible LLM integration, and ontology-aware graph construction, the tool transforms the abstract process of semantically enriching data into a guided and transparent application for researchers. The next chapter evaluates this implementation with a feasibility and a user study to assess its applicability and usability.

5. Evaluation

To assess the feasibility, effectiveness, and usability of the proposed system, we conducted two complementary evaluations. First, we examined whether the tool can successfully transform realistic research data into semantically enriched knowledge graphs using a structured example scenario. Second, we carried out a user study to evaluate both the usability and the acceptance of the tool among researchers from diverse disciplinary backgrounds.

5.1. Functional Demonstration of the Transformation Pipeline

Plant Name	Plant Part	Use	Recommender	Location
Mint	Leaf	Tea	Emily Rose	Urban Garden
Tomato	Fruit	Salad	Liam Stone	Greenhouse

Table 1
Excerpt of the dataset listing plants, their utilized parts, associated use, recommender names, and locations.

To assess whether the tool delivers on its core objective of transforming research data into semantically structured, ontology-aligned knowledge graphs, we conducted a detailed example run using a synthetic dataset. The aim of this demonstration is to evaluate its effectiveness when applied to a realistic, interpretable scenario.

The dataset originates from a fictional community gardening initiative, containing records of edible plants, such as "Mint" and "Tomato", along with their typical uses, growing locations, and the individuals who recommended them. An excerpt is given in Table 1, and the full dataset is provided in Table 2 in Section B of the appendix. This CSV file is accompanied by a short README that provides background on the dataset, clarifies the meaning of each column, and gives narrative examples to contextualize the data. The README file is given in Section B.2 of the appendix.

Once uploaded, the tool processed both files and generated semantic interpretations without further user input. For instance, plant names were correctly identified as content-bearing entities, and grouped under the broader label "Edible Plant". Similarly, values like "Greenhouse", and "Urban Garden" were grouped as "Growing Locations", while "Tea" and "Salad" were categorized as "Use Cases". These extractions proved coherent and appropriate when reviewed in the context of the dataset and accompanying description.

Crucially, the tool also inferred meaningful relations between these entities. For example, it proposed triples such as "Tomato – is grown in – Greenhouse" and "Mint – used in – Tea", correctly capturing relationships that are implicit in the tabular structure and narrative context. These relations were both syntactically well-formed and semantically valid, indicating that the pipeline accurately captured relevant domain knowledge.

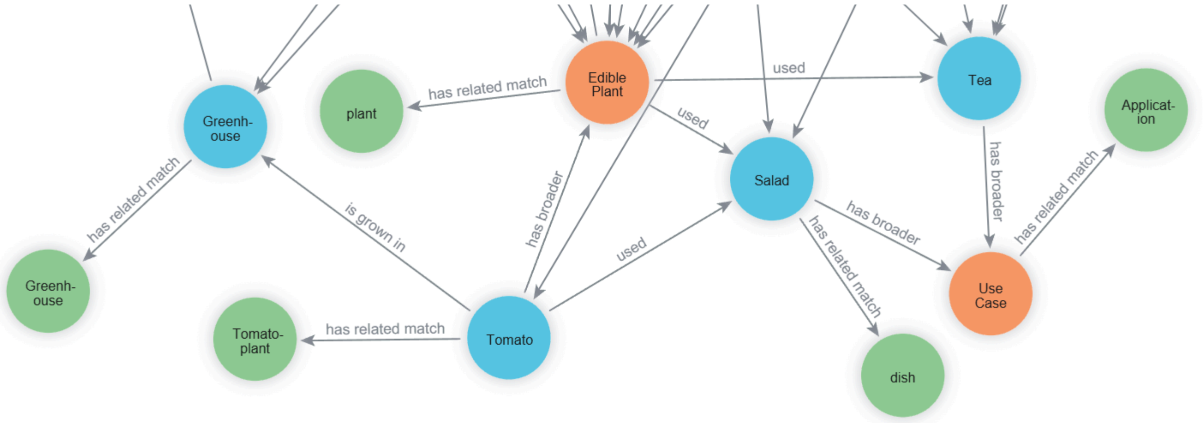


Figure 4: Resulting knowledge graph excerpt from KONDA for the described example.

To evaluate semantic enrichment, we selected two ontologies for alignment: the Common Greenhouse Ontology (CGO)[19] and the BioAssay Ontology (BAO)[20]. The tool matched "Greenhouse" directly to "CGO:Greenhouse", and successfully aligned "Tomato" with "CGO:Tomato Plant". Additionally, the

[19] https://gitlab.com/ddings/common-greenhouse-ontology/-/blob/main/Ontologies/CGO-v2.4-current-stable-version.ttl?ref_type=heads (accessed on 23.07.2025)

[20] https://bioportal.bioontology.org/ontologies/BAO (accessed on 23.07.2025)

predicate “grown in” was matched to “BAO:BAO_0002924”, demonstrating that the tool not only finds appropriate classes, but also meaningful property mappings. Importantly, when concepts such as “Mint” lacked a suitable match in the selected ontologies, the tool retained them using a custom identifier, ensuring no information was discarded in the process.

The resulting knowledge graph consisted of 27 nodes and 53 edges. An excerpt is shown in Figure 4. Each extracted element was either matched to an ontology term or clearly retained with its original label. Relations between entities were labeled with readable, accurate predicates, and semantic groupings such as categories were preserved to enhance interpretability. The graph was rendered interactively and could be exported in standard semantic web formats, as depicted fully in Section B.3 of the appendix.

This example run confirms that the tool can produce coherent, semantically enriched knowledge graphs from real-world inspired data with minimal intervention, setting the stage for a deeper evaluation of quality and usability in the subsequent sections.

5.2. Usability and Acceptance

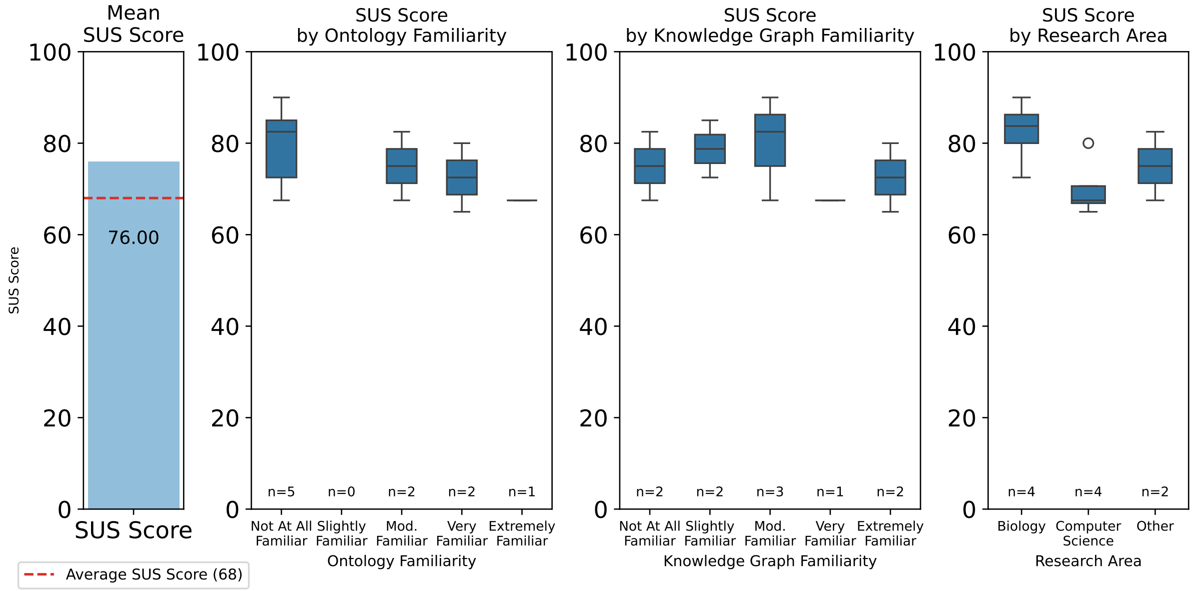


Figure 5: System Usability Scale (SUS) results for the app. The overall mean SUS score was 76, exceeding the usability benchmark of 68. Scores varied by research area, with biology and “other” users rating higher than computer science. Usability perceptions showed little dependence on prior familiarity with semantic technologies, indicating accessibility for non-experts.

To assess whether the tool meets its research objective of providing a user-friendly experience suitable for researchers of varying backgrounds, a structured usability study was conducted. The evaluation targeted researchers from diverse disciplines and differing levels of expertise with semantic technologies, and focused on determining whether the researchers could successfully complete the semantic enrichment pipeline and found the tool intuitive, understandable, and effective in practice.

A total of 10 participants from various domains, including biology, computer science, materials science, and economics, took part in the study. All participants were affiliated with RWTH Aachen University and had prior exposure to working with research data. The study was conducted in three phases: a questionnaire on the participant’s background, a guided example task using a sample dataset, and a structured post-task survey. The example task required participants to upload the aforementioned plant dataset and its context, run the pipeline through entity extraction, ontology-based annotation, and graph export, and inspect the results.

To quantitatively assess usability, the System Usability Scale (SUS) was used alongside additional Likert-scale questions that addressed more specific aspects of usability, functionality, and perceived impact [20]. The SUS yielded a mean score of 76, surpassing the commonly accepted usability benchmark of 68 [21]. This result indicates strong overall usability across participants. Furthermore, the average rating for overall usability was 8.8 out of 10 (SD: 0.42), with 9 being the most commonly selected value, suggesting high satisfaction with the interface and workflow.

A central result for evaluating the accessibility of the tool is shown in Figure 5, which splits SUS scores by participants' familiarity with ontologies, knowledge graphs, and research domain. Most notably, participants who self-reported being "Not at all familiar" with ontologies or knowledge graphs still gave high usability ratings, comparable to those with moderate or high familiarity. This suggests that we reached the objective of lowering the barrier to entry for non-experts. The guided interface, structured prompts, and stepwise validation mechanisms enabled users with no prior exposure to semantic technologies to complete the full pipeline effectively.

The figure also highlights differences across disciplinary backgrounds. While researchers from biology and other non-informatics fields reported the highest SUS scores, users from computer science also rated the tool positively. These results suggest that the tool's usability is robust across varying research domain backgrounds.

Likert-scale responses further reinforced these results. Participants rated the intuitiveness of the interface at 4.6/5 (SD: 0.52), the clarity of instructions at 4.5/5 (SD: 0.71), and reported they would feel comfortable to use the tool again in future projects (mean: 4.5/5, SD: 0.53). Most users agreed that they were able to complete the task without issues and understood the purpose of each pipeline step.

Free-text feedback echoed these positive trends. Participants highlighted the ease of reviewing extracted entities and the clarity of the interface. Suggested improvements focused on secondary aspects rather than core functionality, such as improving inline help, refining the ontology selection experience, and more clearly separating entities from relations during annotation.

In summary, the survey results strongly support the conclusion that the tool enables non-expert users to create semantically enriched knowledge graphs with minimal onboarding through the transparent, modular, and well-guided interface. High SUS scores across diverse familiarity levels and research backgrounds, combined with positive qualitative feedback, support that the tool is both usable and accessible.

6. Discussion

The evaluation results indicate that KONDA significantly contributes to low-barrier ontology-based semantic annotation and knowledge graph construction. The tool provides an accessible interface that allows researchers from different domains and with varying levels of technical expertise to create semantically enriched representations of scientific data. High usability scores across disciplines and user backgrounds support the system's applicability for use in various research contexts.

A particular strength of KONDA is its capacity to process heterogeneous input formats and support semantic structuring without requiring prior experience with semantic web technologies. The integration of key annotation and transformation steps, such as terminology lookup, ontology alignment, and graph construction, into a single interface enables a guided workflow that helps translate informal and semantically unstructured data into formal representations. The reuse of existing ontologies contributes to interoperability even in the absence of prior agreement on shared terminologies.

Nonetheless, certain limitations remain. The quality of semantic matching between identified entities and ontology concepts depends on the capabilities of the underlying language model. While this approach is effective in many cases, it can produce imprecise or overly general matches, particularly when relevant concepts are underrepresented in the selected ontology. In addition, although the tool is designed to support data from different domains, its effectiveness depends on the availability and coverage of suitable ontologies. Domains with limited ontology support may require additional effort to achieve meaningful annotations.

The tool's performance with larger datasets and more complex graph structures also remains a challenge. While the manual validation step is important, it introduces additional effort for users which grows with the size of the dataset. While the current implementation supports minimal interactive use, future developments should consider optimization methods for scalability.

Lastly, the current system offers few functionalities for supporting the exploration of the resulting knowledge graphs. Advanced post-processing or querying mechanisms on the knowledge graph remain therefore underexplored.

7. Conclusion

This paper introduced KONDA, a tool for semantic annotation and knowledge graph construction for research data. It leverages the integration of LLMs to provide an accessible, user-friendly workflow. The tool supports researchers in processing heterogeneous data, reusing existing ontologies, and producing structured semantic representations without requiring prior expertise in semantic technologies. An initial user study confirmed the usability and applicability of the system across a range of different user contexts.

Future work may focus on evaluating the tool in real-world settings. Improvement of domain-specific performance may be explored by integrating recommendation mechanisms for ontologies. In addition, the implementation of persistent ontology embeddings may reduce redundancy, improve efficiency, and enhance matching quality in repeated use scenarios. The integration of further functionalities for the support of FAIR-aligned practices may be considered, such as enabling FAIR Digital Objects as export formats. Finally, expanding the tool's capabilities to support tasks such as semantic search, filtering, or integration with reasoning engines may increase its value for further research workflows.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o in order to: Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. D. Wilkinson, M. Dumontier, I. J. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. Da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons, The fair guiding principles for scientific data management and stewardship, Scientific data 3 (2016) 160018. doi:10.1038/sdata.2016.18.

[2] D. Hodapp, A. Hanelt, Interoperability in the era of digital innovation: An information systems research agenda, Journal of Information Technology 37 (2022) 407–427. doi:10.1177/02683962211064304.

[3] H. Rahman, M. I. Hussain, A comprehensive survey on semantic interoperability for internet of things: State–of–the–art and research challenges, Transactions on Emerging Telecommunications Technologies 31 (2020). doi:10.1002/ett.3902.

[4] B. H. de Mello, S. J. Rigo, C. A. Da Costa, R. Da Rosa Righi, B. Donida, M. R. Bez, L. C. Schunke, Semantic interoperability in health records standards: a systematic literature review, Health and technology 12 (2022) 255–272. doi:10.1007/s12553-022-00639-w.

[5] L. Westhofen, C. Neurohr, M. Butz, M. Scholtes, M. Schuldes, Using ontologies for the formalization and recognition of criticality for automated driving, IEEE Open Journal of Intelligent Transportation Systems 3 (2022) 519–538. doi:10.1109/OJITS.2022.3187247.

[6] S. Appukuttan, L. L. Bologna, F. Schürmann, M. Migliore, A. P. Davison, Ebrains live papers - interactive resource sheets for computational studies in neuroscience, Neuroinformatics 21 (2023) 101–113. doi:10.1007/s12021-022-09598-z.

[7] T. Benson, G. Grieve, Why interoperability is hard, in: T. Benson, G. Grieve (Eds.), Principles of Health Interoperability, Health Information Technology Standards, Springer International Publishing, Cham, 2021, pp. 21–40. doi:10.1007/978-3-030-56883-2{\textunderscore}2.

[8] A. O. Alkhamisi, M. Saleh, Ontology opportunities and challenges: Discussions from semantic data integration perspectives, in: 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), IEEE, 2020, pp. 134–140. doi:10.1109/CDMA47397.2020.00029.

[9] S. Reichmann, T. Klebel, I. Hasani-Mavriqi, T. Ross-Hellauer, Between administration and research: Understanding data management practices in an institutional context, Journal of the Association for Information Science and Technology 72 (2021) 1415–1431. doi:10.1002/asi.24492.

[10] A. Sakor, K. Singh, M.-E. Vidal, Biolinkerai: Capturing knowledge using llms to enhance biomedical entity linking, in: M. Barhamgi, H. Wang, X. Wang (Eds.), Web Information Systems Engineering – WISE 2024, volume 15439 of *Lecture Notes in Computer Science*, Springer Nature Singapore, Singapore, 2025, pp. 262–272. doi:10.1007/978-981-96-0573-6{\textunderscore}19.

[11] Á. García-Barragán, A. Sakor, M.-E. Vidal, E. Menasalvas, J. C. S. Gonzalez, M. Provencio, V. Robles, Nssc: a neuro-symbolic ai system for enhancing accuracy of named entity recognition and linking from oncologic clinical notes, Medical & biological engineering & computing 63 (2025) 749–772. doi:10.1007/s11517-024-03227-4.

[12] N. Fathallah, A. Das, S. de Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, Neon-gpt: A large language model-powered pipeline for ontology learning, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), The Semantic Web: ESWC 2024 Satellite Events, volume 15344 of *Lecture Notes in Computer Science*, Springer Nature Switzerland, Cham, 2025, pp. 36–50. doi:10.1007/978-3-031-78952-6{\textunderscore}4.

[13] M. S. Abolhasani, R. Pan, Ontokgen: A genuine ontology and knowledge graph generator using large language model, in: 2025 Annual Reliability and Maintainability Symposium (RAMS), IEEE, 2025, pp. 1–6. doi:10.1109/RAMS48127.2025.10935139.

[14] H. Yang, L. Xiao, R. Zhu, Z. Liu, J. Chen, An llm supported approach to ontology and knowledge graph construction, in: 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2024, pp. 5240–5246. doi:10.1109/BIBM62325.2024.10822222.

[15] J. Vizcarra, S. Haruta, M. Kurokawa, Representing the interaction between users and products via llm-assisted knowledge graph construction, in: 2024 IEEE 18th International Conference on Semantic Computing (ICSC), IEEE, 2024, pp. 231–232. doi:10.1109/ICSC59802.2024.00043.

[16] A. Sakor, M. Brunet, E. Iglesias, A. Rivas, P. D. Rohde, A. Kraft, M.-E. Vidal, Integrating knowledge graphs and neuro-symbolic ai: Ldm enables fair and federated research data management, in: W. Nejdl, S. Auer, O. Karras, M. Cha, M.-F. Moens, M. Najork (Eds.), Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA, 2025, pp. 1044–1047. doi:10.1145/3701551.3704125.

[17] X. Zhao, M. Blum, R. Yang, B. Yang, L. M. Carpintero, M. Pina-Navarro, T. Wang, X. Li, H. Li, Y. Fu, R. Wang, J. Zhang, I. Li, Agentigraph: An interactive knowledge graph platform for llm-based chatbots utilizing private data, 2024. URL: https://arxiv.org/abs/2410.11531. arXiv:2410.11531.

[18] L. Sbailò, Á. Fekete, L. M. Ghiringhelli, M. Scheffler, The nomad artificial-intelligence toolkit: turning materials-science data into knowledge and understanding, npj Computational Materials 8 (2022). doi:10.1038/s41524-022-00935-z.

[19] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, D. Sontag, Large language models are few-shot clinical information extractors, 2022. URL: https://arxiv.org/abs/2205.12689. arXiv:2205.12689.

[20] John Brooke, Sus: A 'quick and dirty' usability scale, in: P. W. Jordan, B. Thomas, I. L. McClelland, B. Weerdmeester (Eds.), Usability Evaluation In Industry, CRC Press, 1996, pp. 207–212. doi:10.1201/9781498710411-35.

[21] J. R. Lewis, J. Sauro, Item benchmarks for the system usability scale., Journal of Usability studies 13 (2018).

A. Conceptual Design Details

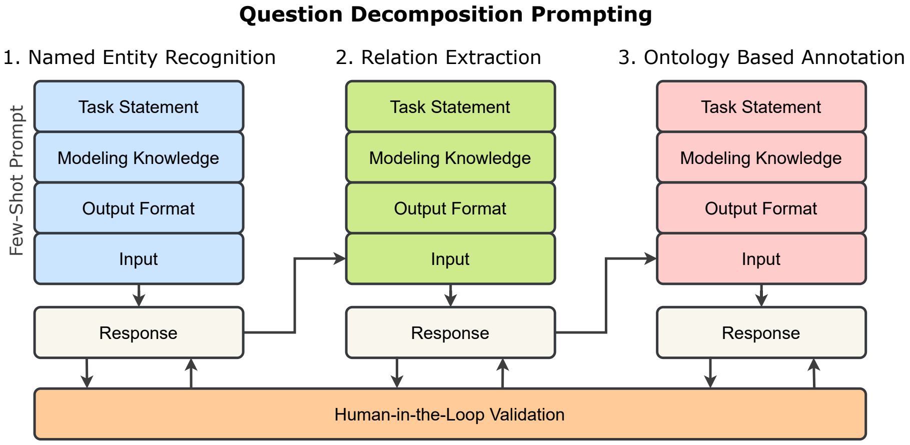

Question Decomposition Prompting
1. Named Entity Recognition
2. Relation Extraction
3. Ontology Based Annotation
Few-Shot Prompt
Task Statement
Modeling Knowledge
Output Format
Input
Response
Human-in-the-Loop Validation

Figure 6: The pipeline breaks down the task into three focused LLM prompts: entity recognition (ER), relationship extraction (RE), and ontology-based annotation (OBA). Each prompt uses few-shot examples and a structured format, followed by human-in-the-looop validation to refine results at every stage. This modular approach improves accuracy, interpretability, and user control throughout the pipeline.

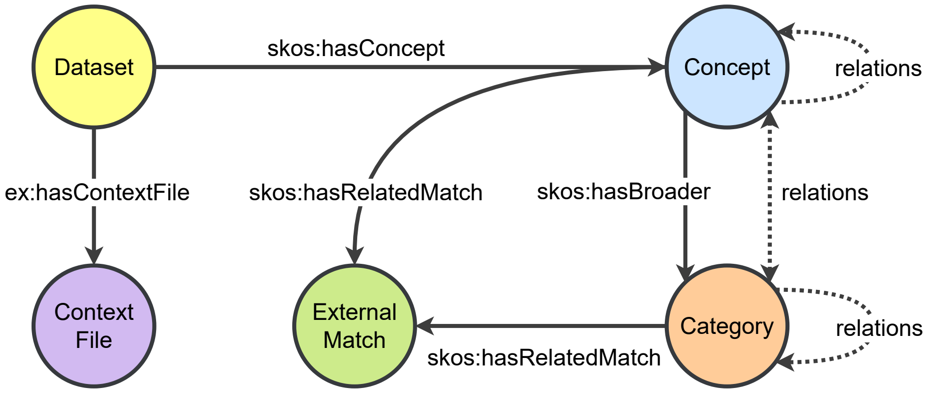


Figure 7: Schema of the resulting knowledge graph. The graph centers on a Dataset node, which links to Context File nodes and extracted Concepts. Concepts are associated with broader Category nodes, and may be semantically aligned to external ontology terms by providing a term match. We utilize SKOS properties to support reasoning and inference. The dotted lines for relations are optional edges that are found during relationship extraction (RE) for further semantic enrichment.

B. Full Synthetic Example

B.1. CSV File

Plant Name	Plant Part	Use	Recommender	Location
Mint	Leaf	Tea	Emily Rose	Urban Garden
Tomato	Fruit	Salad	Liam Stone	Greenhouse
Basil	Leaf	Cooking	Sofia Bloom	Balcony
Carrot	Root	Cooking	Noah Fields	Countryside
Chamomile	Flower	Tea	Olivia Fern	Urban Garden
Spinach	Leaf	Salad	Emily Rose	Community Garden
Beetroot	Root	Salad	Noah Fields	Countryside
Lemon Balm	Leaf	Tea	Sofia Bloom	Balcony

Table 2
Running example dataset listing plants, their utilized parts, associated use, recommender names, and locations

B.2. README File: Community Garden Edible Plants Log

This dataset is based on notes collected from a local community gardening initiative, where members share their favorite edible plants, how they use them, and where they grow them.

Each entry includes:

- The name of the plant,
- the part of the plant that is commonly used (e.g., leaf, fruit, root, or flower),
- what it is typically used for (e.g., culinary, herbal, or fresh consumption),
- the name of the person who recommended the plant,
- and the growing location (e.g., greenhouse, balcony, or countryside).

For example:

- Emily Rose grows mint and spinach in an urban garden and uses them for tea and salad.
- Sofia Bloom gardens on the balcony and recommends basil and lemon balm for cooking and tea.
- Noah Fields contributes countryside plants like carrots and beets, both used in hearty meals and fresh dishes.

The dataset reflects diverse preferences, gardening styles, and cultural uses for everyday edible plants.

B.3. Knowledge Graph Export

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix gpo: <https://gpo.ontology.link/> .
@prefix ex: <http://example.org/ontology#> .
@prefix obo: <http://purl.obolibrary.org/obo/> .

ex:tea skos:broader ex:use-case .

ex:plant_dataset-zip ex:has-file ex:readme-md;
  ex:has-concept ex:tomato, ex:mint, ex:community-garden, ex:carrot, ex:beetroot, ex:greenhouse,
    ex:chamomile, ex:balcony, ex:urban-garden, ex:salad, ex:basil, ex:spinach, ex:lemon-balm,
    ex:tea, ex:countryside .

ex:beetroot <http://www.bioassayontology.org/bao#BAO_0002924> ex:greenhouse;
  skos:broader ex:edible-plant .
```

```
ex:community-garden skos:broader ex:growing-location;
  skos:relatedMatch <http://semanticscience.org/resource/SIO_001064> .

ex:countryside skos:broader ex:growing-location .

ex:growing-location skos:relatedMatch obo:AEON_0000155 .

ex:urban-garden skos:relatedMatch obo:AEON_0000155;
  skos:broader ex:growing-location .

ex:carrot skos:broader ex:edible-plant;
  <http://www.bioassayontology.org/bao#BAO_0002924> ex:countryside .

ex:tomato reproduceme:used ex:salad;
  skos:relatedMatch <https://www.tno.nl/agrifood/ontology/common-greenhouse-ontology#Tomatoplant>;
  <http://www.bioassayontology.org/bao#BAO_0002924> ex:greenhouse;
  skos:broader ex:edible-plant .

ex:basil skos:broader ex:edible-plant;
  <http://www.bioassayontology.org/bao#BAO_0002924> ex:urban-garden .

ex:salad skos:relatedMatch <http://www.bioassayontology.org/bao#BAO_0010016>;
  skos:broader ex:use-case .

ex:lemon-balm reproduceme:used ex:tea;
  skos:broader ex:edible-plant .

ex:edible-plant <http://www.bioassayontology.org/bao#BAO_0002924> ex:balcony, ex:community-garden;
  reproduceme:used ex:salad, ex:tea;
  skos:relatedMatch <http://www.bioassayontology.org/bao#BAO_0000605> .

ex:spinach reproduceme:used ex:salad;
  skos:broader ex:edible-plant .

ex:chamomile reproduceme:used ex:tea;
  skos:broader ex:edible-plant .

ex:greenhouse skos:broader ex:growing-location;
  skos:relatedMatch <https://www.tno.nl/agrifood/ontology/common-greenhouse-ontology#Greenhouse> .

ex:balcony skos:broader ex:growing-location;
  skos:relatedMatch obo:AEON_0000155 .

ex:mint reproduceme:used ex:tea;
  skos:broader ex:edible-plant .

ex:use-case skos:relatedMatch gpo:GPO_f97d2d4d_86db_4fd5_9e35_5bb363cd91c9 .
```

C. Online Resources

The original implementation of the tool is available as a Coscine resource from the corresponding author upon reasonable request. The interface of the tool is available from the corresponding author to users with institutional credentials upon reasonable request. External access to the interface is restricted due to security requirements.

- Coscine GitLab resource: http://hdl.handle.net/21.11102/f2df4398-c216-4db3-a141-29b82f072e1b,
- KONDA interface: https://cloud22.dbis.rwth-aachen.de,
- GitHub repository: https://github.com/dsma-org/konda.