A Provenance-Aware Approach for Representing and Querying Domain Entities in Knowledge Graphs

Umayer Reza[1], Torsten Hahmann[1,*] and Jinwu Wang[2]

[1]School of Computing and Information Science, University of Maine
[2]Forest Products Laboratory, U.S. Forest Service

Abstract

With the increasing number of scientific publications, constructing knowledge graphs from scientific literature often requires automated entity extraction techniques, such as named entity recognition (NER). Although domain entities can be extracted using task-specific NER models and general-purpose or fine-tuned large language models (LLMs), these approaches remain error-prone. Therefore, automatically generated domain entities often require expert validation, while existing workflows provide limited support for preserving the provenance of annotation and review activities. As a result, the connection between extracted entities, their textual context, and the decisions that shape their final representation in a knowledge graph is frequently lost. This demonstration presents a provenance-aware workflow for constructing and maintaining knowledge graphs of domain entities through the integration of AnNER, a lightweight tool for entity annotation and review, and OnNER, an ontology for representing entities in scholarly publications and their provenance. The proposed approach captures entity mentions, labels, review decisions, contributors, timestamps, and textual context as interconnected RDF resources, enabling the complete history of annotation and review activities to be preserved and queried.

Keywords

Knowledge Graph Construction, Domain Entities, Annotation Provenance, AnNER, OnNER

1. Introduction

Knowledge graphs have become an important mechanism for organizing and querying knowledge extracted from large text corpora [1, 2]. In specialized domains, such as scientific research, they can help make domain-specific entities and concepts more accessible for search, analysis, and knowledge discovery. However, the usefulness of such knowledge graphs depends not only on the information they contain but also on the ability to understand where that information originated, how it was extracted, and whether it has been reviewed and validated. The construction of such knowledge graphs relies on the accurate identification of domain-specific entities and concepts within textual sources. Recent advances in named entity recognition (NER) and, more recently, large language models (LLMs) have substantially improved the automated extraction of domain-specific entities from text [3, 4]. Nevertheless, automatically generated entities remain prone to errors, inconsistencies, and domain-specific ambiguities. Consequently, expert review remains essential for constructing reliable knowledge graphs from textual sources. To increase transparency and trust, entity recognition and review workflows must preserve detailed provenance information about annotations, including their textual context, contributors, and review decisions.

Existing annotation tools [5, 6, 7] provide mechanisms for creating and managing annotations, but typically focus on the final annotation result and offer only limited support to represent the complete history of annotation and review activities in a semantically meaningful way. For example, they may support annotating or reviewing by one or multiple domain experts, but only share the final set of annotations rather than the whole annotation and review history. Ontologies can provide rich semantic representations of extracted entities, but ontologies constructed using NLP techniques rarely preserve

Proceedings of FOIS 2026 Satellite events co-located with the 16th International Conference on Formal Ontology in Information Systems (FOIS 2026), September 21-25, 2026, Vitória, Brazil

*Corresponding author.
a.reza@maine.edu (U. Reza); torsten.hahmann@maine.edu (T. Hahmann); jinwu.wang@usda.gov (J. Wang)
0000-0003-4013-3513 (U. Reza); 0000-0002-5331-5052 (T. Hahmann); 0000-0002-8363-1689 (J. Wang)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-27

CEUR-WS.org/Vol-4342/paper11.pdf

how concepts were derived from entities identified in the source texts, or how those concepts were subsequently reviewed and curated. As a result, the connection between annotation activities and constructed ontologies (e.g., through ontology learning) or knowledge graphs is often only implicit, which hinders making corrections or other updates. In short, ontology construction workflows from text sources may preserve the current label and entity span but provide very limited support for reconstructing earlier annotation decisions, identifying who made each decision and when, or linking those decisions to the precise document fragment from which the entity was extracted. This makes it difficult to answer questions such as: Which automatically extracted entities still require expert review? Who accepted or rejected a particular annotation, and when? Were there competing annotations for certain entities? How did an entity's label or text span change during review? From which document and textual context did the annotation originate?

To overcome these limitations, we developed two complementary components whose interplay in constructing knowledge graphs of domain entities is demonstrated here. OnNER [8] provides a semantic model for representing domain entities, their textual context, their labels, and the provenance of annotation and review decisions. AnNER [9] is a lightweight annotation and review environment that uses the core structure of OnNER to semantically store information about annotation review activities performed through its user interface. Together, they provide the key components for a provenance-aware approach in which annotation activities, review decisions, and semantic representations remain tightly connected. We show how the AnNER-OnNER pipeline supports traceable knowledge graph construction by integrating the semantic representation of document structure and content with that of extracted and reviewed domain entities, their annotations, and associated provenance. More broadly, this constitutes an ontology-based approach for storing, updating, and querying annotations of domain entities and their provenance in an RDF-based knowledge graph, so that not only the final annotations but also their histories can be queried in detail.

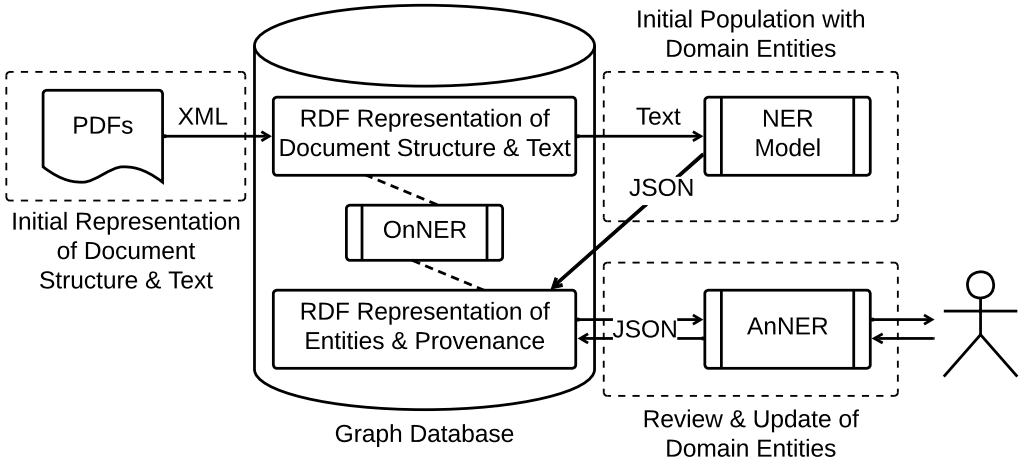

Figure 1: High-level workflow of the demonstrated system. Scientific PDFs are first transformed into XML and represented as RDF describing document structure and text. The extracted text is then processed by an NER model to generate initial entity annotations, which are exchanged in JSON format together with additional information such as unique identifier for each entity, responsible annotator's name, and annotation creation timestamp. The annotations are transformed into an RDF representation of entities and their provenance and are linked to the RDF representation of the source documents. The resulting document and annotation representations are presented in AnNER for human review and update. Following review, the revised information about the validated annotations is transformed into an RDF representation that preserves annotation provenance and maintains links to the RDF representation of the source documents. All RDF representations are OnNER-based, while XML, Text, and JSON serve as the primary interchange formats between the system components.

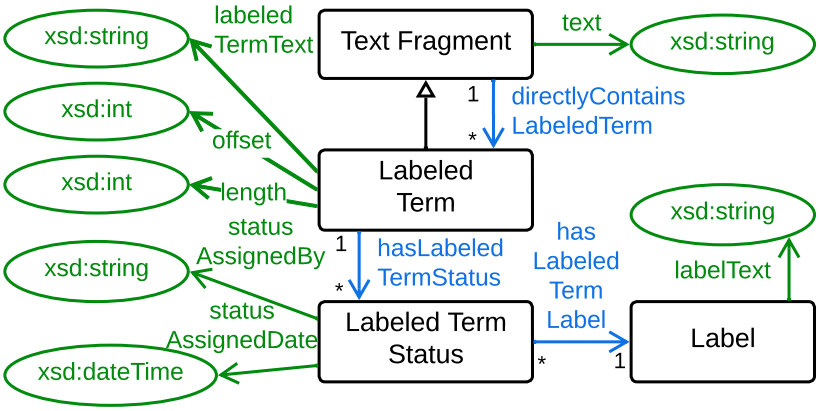

Figure 2: Overview of the OnNER ontology for representing entities, their associated labels, review statuses, and provenance information. The four labeled term statuses currently included are `Candidate`, `Accepted`, `Rejected`, and `Suggested`. Reproduced from [9].

2. The AnNER-OnNER Pipeline for Knowledge Graph Construction

The AnNER-OnNER pipeline is a provenance-aware system for transforming automatically extracted domain entities into reviewed, queryable knowledge graph. Its central purpose is not only to identify domain entities in text, but also to preserve the provenance of how those entities were extracted, reviewed, updated, and linked back to their textual context. Figure 1 provides an overview of the demonstrated pipeline. While the complete knowledge graph construction pipeline includes components for document processing (i.e., to process a PDF into an XML and further into an RDF representation) and automated entity extraction (NER), the demonstration concentrates on the constructing and updating of the knowledge graph when annotations are formally encoded, reviewed by experts (in AnNER), and transformed into a queryable knowledge graph.

2.1. Semantic Model Underlying the Pipeline

OnNER provides the semantic model used by the pipeline to represent annotated entities, their labels, their review history, and the text fragments in which they occur. The ontology introduces a core pattern (see Figure 2) consisting of four central concepts: `Text Fragment`, `Labeled Term`, `Labeled Term Status`, and `Label`. A `Text Fragment` represents a portion of text, such as a paragraph, that contains one or more annotated entities. Each annotated entity is represented as a `Labeled Term` which is also a `Text Fragment` and is associated with one or more `Labeled Term Status` instances that capture the evolution of annotation decisions over time. OnNER currently defines four types of `Labeled Term Status`: `Candidate`, `Accepted`, `Rejected`, and `Suggested`. `Candidate` represents an annotation that is awaiting expert review. `Accepted` indicates that a reviewer has validated the annotation, whereas `Rejected` indicates that the annotation has been determined to be incorrect or unsuitable. `Suggested` represents a new or revised annotation proposed during the review process, for example, after modifying its text span or assigned label. Finally, `Label` represents the semantic categories assigned to entities.

An alternative semantic model, the W3C Web Annotation Data Model [10], also provides a general framework to link annotations to resources or selected parts of resources and to record provenance information such as creators and timestamps. However, it is less granular and detailed than OnNER in that it does not capture review-status pattern of individual entities or the nested and ordered structure of scholarly publications represented in OnNER needed to support querying by parts of text (e.g., for finding entities that co-occur in the same or two subsequent paragraphs).

The knowledge graph instantiates this semantic model by representing annotations as RDF resources rather than as isolated text spans. This allows preserving the connection between an entity mention, the document fragment in which it occurs, the label assigned to it, and the sequence of annotation or

```
data:Publication_10.1016_j.powtec.2014.04.016 rdf:type onner:ScholarlyPublication ;
    onner:publicationTitle 'Effect of drying conditions on cellulose ...'^^xsd:string ;
    onner:publicationDate '2014-04-15'^^xsd:date ;
    onner:doi '10.1016/j.powtec.2014.04.016'^^xsd:string ;
    onner:directlyContainsDocumentPart
        data:10.1016_j.powtec.2014.04.016_A,
        data:10.1016_j.powtec.2014.04.016_S1,
        data:10.1016_j.powtec.2014.04.016_S2,
        data:10.1016_j.powtec.2014.04.016_S3,
        data:10.1016_j.powtec.2014.04.016_S4,
        data:10.1016_j.powtec.2014.04.016_S5 .

data:10.1016_j.powtec.2014.04.016_A rdf:type onner:Abstract ;
    onner:nextDocumentPart data:10.1016_j.powtec.2014.04.016_A_P1 ;
    onner:directlyContainsDocumentPart data:10.1016_j.powtec.2014.04.016_A_P1 .

data:10.1016_j.powtec.2014.04.016_A_P1 rdf:type onner:Paragraph ;
    onner:positionInParentDocumentPart '1'^^xsd:nonNegativeInteger ;
    onner:nextDocumentPart data:10.1016_j.powtec.2014.04.016_S1 ;
    onner:paragraphText 'The microstructure of CNC particles depends ...'^^xsd:string .
```

Figure 3: Example RDF representation of a scholarly document in OnNER. The publication, section, and paragraph are represented as interconnected RDF resources, preserving the document structure and enabling precise links between annotations and their source text.

review decisions made about it. As a result, the pipeline supports provenance-aware querying of both the entities themselves and the activities through which they were created and validated.

2.2. RDF Representation of Documents and Domain Entities

The workflow begins with scholarly documents that have already been transformed into a structured RDF representation (see Figure 3) following OnNER's document model. In addition to a unique document identifier, each component of the document is assigned its own unique identifier, enabling annotated entities to remain linked to their original textual context throughout the workflow. The RDF representation further captures hierarchical relationships among document components, including the organization of sections and their subsections, as well as the sequence of paragraphs within a document, to support navigation across the document structure and facilitate precise provenance tracking. Consequently, annotations generated and reviewed during the workflow can be associated not only with the source document as a whole, but also with the specific document fragments from which they were derived. This document-centric representation provides the foundation for the provenance-aware annotation and review processes described in the subsequent stages of the pipeline.

The domain entities identified through an extraction process are subsequently transformed into an RDF representation (see Figure 4) aligned with the core concepts of OnNER. Each entity mention is assigned a unique identifier, enabling individual occurrences of the same term appearing in different locations within a document to be represented independently. Entities are associated with their textual span, classification label, extraction timestamp, the model or human contributor responsible for the extraction, and the paragraph in which they occur. This representation preserves the link between entities and their original textual context while capturing provenance information about how each annotation was generated. Automatically extracted entities are initially assigned the status `Candidate`, indicating that they have not yet undergone expert validation. The resulting RDF representation provides the foundation for the subsequent review process, enabling annotation status and provenance information to be updated while maintaining explicit links to both the source document and the originating extraction activity.

2.3. Review Update, and Knowledge Graph Maintenance

After creating the initial RDF representation of the documents and domain entities, the pipeline supports human-in-the-loop validation. This review stage enables domain experts to inspect extracted entities,

```
data:10.1016_j.powtec.2014.04.016_S2.1_P1 onner:directlyContainsLabeledTerm
    data:10.1016_j.powtec.2014.04.016_S2.1_P1_E1_Rv1,
    data:10.1016_j.powtec.2014.04.016_S2.1_P1_E2_Rv1,
    data:10.1016_j.powtec.2014.04.016_S2.1_P1_E3_Rv1,
    data:10.1016_j.powtec.2014.04.016_S2.1_P1_E1 .

data:10.1016_j.powtec.2014.04.016_S2.1_P1_E1_Rv1 rdf:type onner:LabeledTerm ;
    onner:labeledTermText 'Inter-nanoparticle forces'^^xsd:string ;
    onner:offset '0'^^xsd:nonNegativeInteger ;
    onner:length '25'^^xsd:nonNegativeInteger ;
    onner:labeledTermDirectlyContainedBy data:10.1016_j.powtec.2014.04.016_S2.1_P1 ;
    onner:hasLabeledTermStatus data:Suggested_10.1016_j.powtec.2014.04.016_S2.1_P1_E1_Rv1 .

data:Suggested_10.1016_j.powtec.2014.04.016_S2.1_P1_E1_Rv1
    rdf:type onner:SuggestedStatus ;
    onner:statusAssignmentDate '2026-06-12T20:24:42Z'^^xsd:dateTime ;
    onner:statusAssignedBy data:THahmann ;
    onner:hasLabeledTermLabel data:Label_4 .

data:Label_4 rdf:type onner:Label ;
    onner:fromLabelingSchema data:Labeling_Schema ;
    onner:labelText 'PROPERTY'^^xsd:string .
```

Figure 4: Example RDF representation of an entity annotation and its provenance. The entity is associated with its textual context, review status, and extraction source, enabling traceability of annotation decisions throughout the review workflow.

JSON (Before Review)

```
[
 "10.1016_j.powtec.2014.04.016_S2.1_P3",
 "CNC nanoparticles with parallel orientation ...",
 {
     "entities": [
         [
             "10.1016_j.powtec.2014.04.016_S2.1_P3_E1",
             0,
             17,
             [
                 [
                     "Candidate",
                     "MAT_ENT_STRUCT",
                     "2026-02-18T07:20:09Z",
                     "gpt-4o"
                 ]
             ]
         ],
     ]
 }
]
```

JSON (After Review)

```
[
 "10.1016_j.powtec.2014.04.016_S2.1_P3",
 "CNC nanoparticles with parallel orientation ...",
 {
     "entities": [
         [
             "10.1016_j.powtec.2014.04.016_S2.1_P3_E1",
             0,
             17,
             [
                 [
                     "Candidate",
                     "MAT_ENT_STRUCT",
                     "2026-02-18T07:20:09Z",
                     "gpt-4o"
                 ],
                 [
                     "Accepted",
                     "MAT_ENT_STRUCT",
                     "2026-06-12T20:24:42Z",
                     "THahmann"
                 ]
             ]
         ],
     ]
 }
]
```

Figure 5: Simplified JSON representation of an extracted entity after initial annotation (left) and after acceptance by one reviewer (right). Other reviewers might add additional status updates (e.g., another *Accept* or a *Reject* or reverting to *Candidate* status) or add new entities altogether.

revise or reject annotations, or add new ones as necessary; all while recording the provenance of all annotation decisions. AnNER supports this validation process through an interactive annotation and review environment that operates on a JSON format aligned with the OnNER data model. This format (see Figure 5) preserves all the information required to represent entities, including their corresponding paragraph text, identifiers for both the paragraphs and the entities, as well as assigned labels, status records, contributors, and timestamps. This enables the subsequent transformation into the OnNER-based RDF representation.

Before starting the review process, the RDF representation of entities is transformed into JSON and loaded into AnNER, which currently operates on JSON data. Users, who are typically domain experts, can accept, reject, modify, or suggest annotations through the review interface (see Figure 6). Because

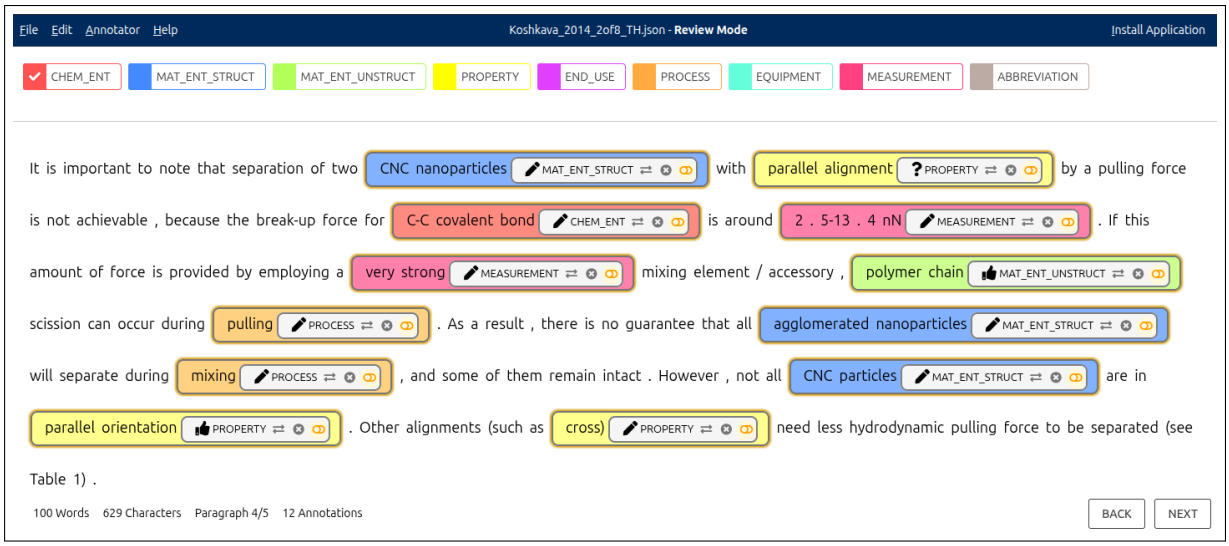

Figure 6: The review interface of AnNER. It facilitates the validation and correction process for automatically extracted domain entities.

of the fine-grained entity-level provenance, review can be done by a single expert, or by multiple experts, each of them either reviewing only the annotations they are comfortable with or specific sets of annotations (e.g., one document or one section therein)[1]. Each review action generates a new status record that captures the review outcome, contributor identity, timestamp, and assigned label. Rather than overwriting previous decisions, the system preserves all review records, thereby maintaining a complete history of annotation evolution.

Following review, the updated annotations are transformed back into RDF by instantiating the corresponding OnNER concepts and relationships. The resulting knowledge graph contains representations of document content, annotated entities, review decisions, and their associated provenance. Consequently, users can query not only the validated entities themselves but also how, when, and by whom annotation decisions were made throughout the review process. This provenance-preserving review process distinguishes the approach from conventional annotation workflows that retain only the final annotation state.

3. Querying the Knowledge Graph

By explicitly representing domain entities together with their review decisions, contributors, timestamps, and textual context in RDF, OnNER enables SPARQL queries that extend beyond simple keyword retrieval and may also involve provenance information. Selected examples that demonstrate the kind of queries that allow users to explore the domain entities (reviewed or not) holistically and in context are discussed next. These queries operate not only on the identified entities themselves but also on their annotation history and associated provenance information.

1. Retrieving annotations awaiting expert review. Users may identify annotations that have not yet undergone expert review by retrieving all annotations that have only a `Candidate` status (see Figure 7). Such queries help users monitor the growth of the knowledge graph and identify entities that still require validation.

2. Retrieving recently annotated entities. Users may also retrieve entities that have been annotated within a specific time period. For example, the query shown in Figure 8 returns entities whose annotation

[1]It is also possible that different reviewers review the same set of annotations independently, though merging decisions, e.g., by computing inter-annotator agreement, is not directly supported but could be done by querying the knowledge graph for entities with competing review decisions.

```
PREFIX ↔
SELECT DISTINCT ?entity ?text ?offset ?length WHERE {
    ?entity rdf:type onner:LabeledTerm ;
            onner:labeledTermText ?text ;
            onner:offset ?offset ;
            onner:length ?length ;
            onner:hasLabeledTermStatus ?status .
    ?status rdf:type onner:CandidateStatus .

    FILTER NOT EXISTS {
        ?entity onner:hasLabeledTermStatus ?otherStatus .
        ?otherStatus rdf:type ?otherType .
        VALUES ?otherType {
            onner:AcceptedStatus onner:RejectedStatus onner:SuggestedStatus
        }
    }
} ORDER BY ?entity
```

	entity	text	offset	length
1	data:10.1016_j.powtec.2014.04.016_S2.1_P4_E2	"parallel alignment"	"70"^^xsd:nonNegativeInteger	"18"^^xsd:nonNegativeInteger

Figure 7: SPARQL query for retrieving entity annotations that have not yet undergone expert review. The query identifies entities whose only assigned status is *Candidate*.

```
PREFIX ↔
SELECT ?entity ?text ?classification ?date ?assignedBy WHERE {
    ?entity onner:labeledTermText ?text ;
            onner:hasLabeledTermStatus ?status .
    ?status onner:hasLabeledTermLabel ?label ;
            onner:statusAssignmentDate ?date ;
            onner:statusAssignedBy ?assignedBy .
    ?label onner:labelText ?classification .

    FILTER (?date >= (NOW() - "P3M"^^xsd:duration))
} ORDER BY DESC(?date)
```

	entity	text	classification	date	assignedBy
1	data:10.1016_j.powtec.2014.04.016_S2.1_P1_E1_Rv1	"Inter-nanoparticle forces"	"PROPERTY"	"2026-06-12T20:24:42Z"^^xsd:dateTime	data:THahmann
2	data:10.1016_j.powtec.2014.04.016_S2.1_P1_E2_Rv1	"interaction energy"	"PROPERTY"	"2026-06-12T20:24:42Z"^^xsd:dateTime	data:THahmann
3	data:10.1016_j.powtec.2014.04.016_S2.1_P1_E3_Rv1	"preparation"	"PROCESS"	"2026-06-12T20:24:42Z"^^xsd:dateTime	data:THahmann
4	data:10.1016_j.powtec.2014.04.016_S2.1_P1_E1	"polymer nanocomposite systems"	"MAT_ENT_STRUCT"	"2026-06-12T20:24:42Z"^^xsd:dateTime	data:THahmann

Figure 8: SPARQL query for retrieving entity annotations whose statuses were assigned within the last three months.

status was assigned during the last three months. Such queries support the analysis of annotation activity over time and enable users to monitor recent additions to the knowledge graph.

3. Retrieving the review history of an entity. In addition, users may examine the review history of a specific entity. All review actions associated with an entity can be retrieved together with their assigned labels, contributors, and timestamps (see Figure 9). These details allow users to reconstruct the evolution of the review actions that led to the entity's current status.

While the example queries presented here illustrate some types of provenance and document-context questions supported by OnNER, readers are referred to the OnNER repository (https://github.com/

```
PREFIX ↔
SELECT ?termText ?statusType ?classification ?assignedBy ?date WHERE {
    data:10.1016_j.powtec.2014.04.016_S2.1_P1_E5
        onner:labeledTermText ?termText ;
        onner:hasLabeledTermStatus ?status .

    ?status sesame:directType ?statusType ;
            onner:hasLabeledTermLabel ?label ;
            onner:statusAssignedBy ?assignedBy ;
            onner:statusAssignmentDate ?date .
    ?label onner:labelText ?classification .
} ORDER BY ?date
```

	termText	statusType	classification	assignedBy	date
1	"CNC"	onner:CandidateStatus	"MAT_ENT_STRUCT"	data:gpt-4o	"2026-02-18T07:20:09Z" ^^xsd:dateTime
2	"CNC"	onner:SuggestedStatus	"ABBREVIATION"	data:THahmann	"2026-06-12T20:24:42Z" ^^xsd:dateTime

Figure 9: SPARQL query for retrieving the complete review history of a specific entity annotation. The query returns the sequence of status assignments associated with the annotation.

theSKAILab/OnNER) for a broader and more diverse set of competency questions and corresponding SPARQL queries covering annotation lifecycle tracking, review provenance, temporal analysis, and entity-context exploration.

4. Use Case: CelloGraph

To illustrate and validate the practical utility of the AnNER-OnNER pipeline, we use examples from the development of CelloGraph, a knowledge graph of domain entities extracted from scientific publications in the field of sustainable cellulosic materials. CelloGraph aims to support knowledge discovery by capturing materials, chemicals, properties, processes, applications, and other domain-specific entities appearing in the scientific literature. Its construction requires not only the identification of relevant entities but also mechanisms for validating those entities and tracing their origins.

While comprehensive domain knowledge discovery in CelloGraph relies on the integration of OnNER with domain ontologies, this demonstration focuses specifically on provenance-related queries enabled by the knowledge graph constructed using AnNER and OnNER. Users can retrieve entities within their textual context, identify entities whose labels or spans were modified during review, examine review decisions, and inspect the contributors and timestamps associated with annotation activities. For example, users can determine who recently reviewed or approved entities labeled as properties, identify which reviewer validated annotations for a specific document, retrieve documents containing annotations that remain unreviewed, and investigate whether the same term has been annotated inconsistently across different contexts by the same reviewer. The pipeline also supports the reconstruction of the complete review history of individual entities, enabling users to examine the sequence of annotation decisions over time. By preserving annotation provenance alongside domain knowledge, the framework enhances the transparency, traceability, and trustworthiness of the resulting knowledge graph of cellulosic materials.

5. Summary and Conclusion

We have presented the AnNER-OnNER pipeline as a provenance-aware framework for the review, representation, and querying of domain entities extracted from textual sources. OnNER provides a semantic model for representing entities, their textual context, labels, and review history, while AnNER

utilizes and populates this semantic model during the annotation and review of domain entities. In the process, AnNER captures the complete provenance of annotation activities. Together, these two components are key enablers for constructing a traceable RDF-based knowledge graph of domain entities extracted from textual sources that preserves information about contributors, review decisions, timestamps, and textual context.

In future work, we plan to further develop the CelloGraph ecosystem shown in Figure 1 by integrating domain ontologies and additional knowledge graph construction components. This will extend the framework beyond provenance-aware annotation management to support ontology-driven knowledge discovery and knowledge graph construction. Ultimately, the completed CelloGraph workflow will provide an end-to-end pipeline spanning document processing, entity extraction, expert review, semantic representation, provenance tracking, and knowledge graph querying.

Acknowledgments

This research was supported in part by the U.S. Department of Agriculture through a National Institute of Food and Agriculture (NIFA) grant, Award 2021-67022-34366 and by the Forest Products Laboratory of the U.S. Forest Service.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI's GPT-4 for grammar and spelling checks. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] A. Hogan, et al., Knowledge Graphs, ACM Computing Surveys 54 (2021) 1–37. doi:10.1145/3447772.

[2] L. Zhong, J. Wu, Q. Li, H. Peng, X. Wu, A Comprehensive Survey on Automatic Knowledge Graph Construction, ACM Computing Surveys 56 (2023) 1–62. doi:10.1145/3618295.

[3] I. Keraghel, S. Morbieu, M. Nadif, Recent Advances in Named Entity Recognition: A Comprehensive Survey and Comparative Study, arXiv preprint arXiv:2401.10825 (2024). doi:10.48550/arXiv.2401.10825.

[4] D. Xu, et al., Large Language Models for Generative Information Extraction: A Survey, Frontiers of Computer Science 18 (2024). doi:10.1007/s11704-024-40555-y.

[5] R. Islamaj, D. Kwon, S. Kim, Z. Lu, TeamTat: a collaborative text annotation tool, Nucleic Acids Research 48 (2020) W5–W11. doi:10.1093/nar/gkaa333.

[6] S. Dobbie, et al., Markup: A Web-Based Annotation Tool Powered by Active Learning, Frontiers in Digital Health 3 (2021). doi:10.3389/fdgth.2021.598916.

[7] A. Loreggia, S. Mosco, A. Zerbinati, SenTag: a Web-based Tool for Semantic Annotation of Textual Documents, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, 2022, pp. 13191–13193. doi:10.1609/aaai.v36i11.21724.

[8] U. Reza, X. Zhang, T. Hahmann, OnNER: An Ontology for Semantic Representation of Named Entities in Scholarly Publications, in: Formal Ontology in Information Systems, volume 394, 2024, pp. 32–46. doi:10.3233/FAIA241291.

[9] U. Reza, N. Pease, J. Wang, T. Hahmann, AnNER: Supporting Efficient Entity Annotation and Review Workflows for Knowledge Graph Construction, in: Proceedings of the 13th Knowledge Capture Conference 2025, 2025, pp. 52–55. doi:10.1145/3731443.3771347.

[10] R. Sanderson, P. Ciccarese, B. Young, Web Annotation Data Model, Technical Report, World Wide Web Consortium, 2017. URL: https://www.w3.org/TR/2017/REC-annotation-model-20170223/.