OnNER: An Ontology for Semantic Representation of Named Entities in Scholarly Publications

Umayer REZA [a], Xuelian ZHANG [a] and Torsten HAHMANN [a,1]

[a] *Spatial Computing, School of Computing and Information Science, University of Maine, USA*

ORCiD ID: Umayer Reza https://orcid.org/0000-0003-4013-3513, Xuelian Zhang https://orcid.org/0000-0002-0708-4070, Torsten Hahmann https://orcid.org/0000-0002-5331-5052

Abstract. A significant portion of scientific knowledge resides within scholarly publications, both in print and digital formats. Recent advancements in natural language processing and information extraction techniques have enhanced the accessibility of this knowledge for further automated querying and processing. Structured and semantically-aware representations, such as ontologies, play a crucial role in simplifying and integrating access to this vast pool of knowledge. While several ontologies have been developed to capture the structure and discourse of scientific publications, there is a notable scarcity of ontologies dedicated to succinctly representing the full range of terminology prevalent in scholarly literature.

This paper introduces the Ontology for Named Entity Representation (OnNER) to address this gap. OnNER is designed to capture named entities – the terms identified and labeled using named entity recognition (NER) methods – from scholarly publications. The ontology provides a structured representation of how these entities are labeled together with relevant semantic context, such as their location within a document. We detail the design of OnNER and demonstrate how it facilitates advanced querying of named entities' presence and collocation within and across publications.

Keywords. Named Entity Recognition, Ontology, Scholarly Publication, Information Extraction

1. Introduction

Scholarly publications are being produced in rapidly increasing quantities, making it challenging even for dedicated researchers to keep pace. While standards for semantically tagging publications with metadata about authors, dates, subjects, or keywords are widely available, it remains difficult to automatically process the *content* of publications at sufficiently granular scales that makes specific pieces of knowledge accessible to targeted search and queries by both computers and humans.

Information Extraction (IE) approaches [1, 2] aim to improve access to information from unstructured textual sources by automatically constructing structured representa-

[1] Corresponding Author: torsten.hahmann@maine.edu

tions that can be more flexibly and efficiently queried than unstructured text. Named entity recognition and classification (short: NER) [3] is a particular popular IE technique, whose recent improvements have led to more wide-spread adoption for extracting domain-specific terms [4] from text sources. Examples span various domain such as biomedicine [5] or chemistry [6, 7]. Likewise, NER can be leveraged for the automated or semi-automated construction of ontologies (see, e.g., [8, 9]).

However, the standard outputs produced by NER tools are too simplistic to readily support advanced querying, especially across documents. While NER approaches have been heavily used as part of pipelines to automatically construct ontologies or link entities to existing ontologies or knowledge graphs [1, 10], the identified named entities are typically not shared as a resource for other purposes or alternative approaches. An ontology that explicitly represents named entities would make them more accessible for further processing and advanced querying. Simultanously, it would increase the transparency of both NER and downstream tasks, making changes to entities or their classification more traceable. Such an ontology would be broadly useful for any kind of terminology extraction, entity linking, or ontology learning, regardless of whether implemented as fully automated or human-in-the-loop approaches.

Use case For instance, consider a scenario where a material science researcher seeks to investigate the relationship between the shape of nanoparticles used to make a material and the material's permeability. Merely searching for publications that mention both terms anywhere might yield results that are not directly relevant Instead, a more effective approach would involve identifying sections or paragraphs within publications that discuss the terms together. By doing so, the search results would be more specific and pertinent. Moreover, this method would encompass relevant information from publications that do not primarily focus on the impact of nanoparticle shape on material permeability. It's worth noting that publications containing relevant insights may not necessarily list "nanoparticle shape" or "permeability" as keywords and they may not be among the publication's most descriptive or frequent terms.

Objective Towards these goals, we develop the **On**tology for **N**amed **E**ntity **R**epresentation (OnNER) to represent *named entities* – which are single or compound nouns or noun phrases (i.e. groups of word that function like a noun) – that have been identified by NER tools or by manual tagging of plain text. OnNER is designed to not just represent the named entities and their classification labels, but also other aspects of their semantics, such as their location within the structure of documents, needed for querying not just which and where named entities occur in documents, but also how they appear together. The ontology is intended to be populated with the named entities identified by NER tools to construct knowledge graphs that we can subsequently query using SPARQL. To guide the development, we collected 16 competency questions (available from the OnNER GitHub repository) that exemplify the various ways of how we expect the ontology will be used for querying. They include, for example, the following:

1. Which "chemicals" are mentioned in conjunction with "permeability"? In which publications and paragraphs?
2. What are the most recent publications that mention at least three times one of "bacterial cellulose" or "BC nanofibers".
3. Retrieve all paragraphs from publications since 2017 that include named entities labeled as "application" in conjunction with the property "tensile strength".

4. What named entities have been assigned classification labels by an NER System that have been corrected by human labelers?

These queries suggest two key requirements for OnNER:

1. Named entities and their location in a piece of text at different levels of granularities (e.g. document-, section-, or paragraph-level) to support seamless querying them by terms or labels across multiple documents;
2. Labeling metadata and provenance such as the applied labeling scheme, information about the labeler (a system or human), when it was labeled, and the purpose or status (e.g. reviewed, removed, etc.) of the labeling;

Vision The work presented here is only a first step towards our vision of traceable entity linking where the named entities in their OnNER representation are then explicitly tied to domain ontologies for: (1) **Querying corpora for named entities** using the terminology provided by existing domain ontologies and their links to named entities; and (2) **Growing ontologies** by identifying named entities that are not linked to existing terms from a domain ontology as candidates terms that may be added to the ontology after review by subject matter experts.

Next, we will provide more background on named entity recognition and ontology learning, and discuss related ontologies (Sec. 3) before presenting the details of OnNER in Sec. 4. Sec. 5 show how we instantiated, verified and validated the ontology.

2. Background

OnNER simultaneously leverages and supports recent advances in information extraction (IE), which encompasses both machine-learning approaches and more traditional natural language processing techniques for extracting structured information from text sources, such as written documents or web pages, that provide enormous amounts of information though with only little or no formal structure [1, 2]. Two of the most common and widely used IE techniques are *named entity recognition and classification* (NER or NERC) [3] and *relation extraction* (RE). Because of their maturity[2] and conceptual simplicity, NER approaches are particularly attractive as tools for supporting the construction or expansion of knowledge graphs and ontologies or specific subtasks therein, such as terminology extraction or named entity linking (NEL) [10, 11]. Next, we will review NER and how named entities are represented before discussing NER's role in ontology and knowledge graph construction.

2.1. Named Entity Recognition

Surprisingly, the formats used to store and share named entities remain simplistic and do not readily faciliate large-scale, efficient analysis of such entities. The prevailing formats typically involve simple tab- or comma-separated text files where each line or object describes one named entity. For example, consider the following (partial) sentence from [12]: "... the objective of the present review is to decipher and comprehensively discuss the role of the *nanoparticle shape* [...] on the *modulation* of the *mass transfer properties* in *nanocomposites*, as a function of *filler volume fraction* and in the light of the *nanocomposite structure* achieved." The representation of the first four named entities from this passage may appear as follows:

[2] NER approaches now consistently achieve impressive F-scores above 90% for common types of named entities such as persons or places.

```
214  232     property      nanoparticle shape
307  317     process       modulation
327  352     property      mass transfer properties
356  371     material      nanocomposites
```

Each line comprises (1) the labeled word(s), (2) the assigned label, and (3) the start and end positions (in characters) within the text. The labels "property", "process", and "material" originate from a predefined labeling schema, also referred to as a tagging schema, with which the NER model was trained. For instance, the CoNLL 2023 NER dataset [13] uses four distinct labels: PER (person), LOC (location), ORG (organization), and MISC (miscellaneous). Many domain-specific labeling schemata, such as the CHEMDNER [7] tags for chemicals entities or NERO's [5] labels for biomedical entities use finer-grained labels tailored to the domain, such as "(chemical) formula". OnNER aims to provide a unified semantic format for storing and querying information about named entities across documents and independent of the employed labeling schemata.

2.2. Ontology Construction and Population using Named Entity Recognition

Traditionally, ontologies have been created manually by domain experts in a labor-intensive and time-consuming process, which does not scale well. Ontology learning (OL) [14, 15] expedites the development of domain-specific formal or linguistic ontologies by partially or fully automating central tasks such as terminology or concept extraction [4] or by learning taxonomic and other relations [8,9,16–18]. NER is equally central to tasks that automatically populating ontologies with instances from text sources or that link named entities to existing ontologies or knowledge graphs [11]. OnNER is uniquely focused on serving as an intermediate semantic representation that explicitly represents identified and classified named entities in an ontological format before further processing. This offers the major benefit that changes in how entities are detected and classified are completely decoupled from how entities are eventually linked or added to graphs.

3. Related Ontologies

Relevant existing ontologies fall into four distinct categories based on what they primarily represent: (1) named entities, (2) the structure of scholarly publications; (3) the meaning of the content of publications; and (4) references and other links between documents. Existing ontologies for describing the structure, content, and links between scholarly publications have already been reviewed comprehensively by [19], here we cover only work that is most closely related to OnNER and that has informed its development.

Representing named entities The Named Entity Recognition Ontology (NERO) [5] is designed to represent named entities that are recognized from a large text corpus, but it focuses specifically on the needs and kinds of named entities from the biomedical domain, aiming to bridge the related yet distinct terminologies from molecular biology, genetics, biochemistry, and medicine. NERO does not provide a generic semantic representation of named entities that is reuseable in other domains.

Representing the structure of documents This includes a number of ontologies from the Semantic Publishing and Referencing (SPAR) suite of ontologies [20][3]. From the SPAR

[3] All SPAR ontologies can be found at http://www.sparontologies.net/ontologies

suite, the Document Component Ontology (DoCO) [21] provides classes for describing the structural components that appear in documents, such as sentences, paragraphs, sections, abstract, bibliography, and text chunks. DoCO is closely tied to (1) the Disclosure Element Ontology (DEO) [22] that supplies classes for describing rhetorical components (e.g. background, materials, methods, conclusion) of documents; (2) the Collections Ontology (CO) [23] for representing lists; and (3) the Pattern Ontology (PO) [24] for basic structural elements, such as containers, table or blocks. While many of the classes provided by DoCO are directly relevant to OnNER, its tight integration with PO, CO and DEO and, in particular, the mixing of aspects of the visual and semantic structure unnecessarily complicate the reuse of its classes, such as *Textchunk*, and would create substantial representational overhead without clear benefits to our project. Short of direct reuse, we decided to subclass any relevant DoCO classes to support future alignment of instance data, while not having to import DoCO and DEO fully. We also forego DoCO's representation of the order of document parts using lists from CO; opting for a more intuitive representation that separates nesting from order and ultimately simplifies retrieving, for example, the paragraph prior to or following a given paragraph.

Representing the content of documents At the coarsest scale, the content of a publication can be described via metadata. Ontologies for encoding metadata of scholarly publications and other documents include, amongst others, the widely used Dublin Core Metadata schema [25] that includes terms to specify information such as the authors, contributors, publication date, the abstract, and the full citation of a publication. But capturing metadata is not the primary concern for OnNER.

Approaches to capture the content of publications at finer scales focus on capturing the embedded *discourse structure*. This includes the Ontology of Rhetorical Blocks (ORB) [26] and the finer-grained Discourse Elements Ontology (DEO) [22]. Both structure documents into frontmatter, body, and backmatter and distinguish more specific document parts, such as methods, materials, results or conclusion, based on what the convey. These distinctions are independent yet fully compatible with OnNER's focus on representing named entities from documents if one is interested in queries about specific discourse elements. However, correctly identifying and representing the discourse structure from documents is far from trivial and currently hinders populating the ontology with discourse element. Work on Core Scientific Concepts (CoreSC) [27] has tested the automatic population of the discourse structure from biomedical articles but employs different discourse elements.

At the finest scale, a publication can be divided into what are called *nano-publication* [28, 29], which are the smallest units of publishable and citeable information. A fundamental difference to OnNER is that nano-publications are more like the relations between named entities rather than the named entities themselves. While the extraction of nano-publications from the literature requires joint entity and relation extraction, which is still quite error-prone, nano-publications are powerful for sharing and referencing pieces of scientific knowledge outside the scope of traditional publications. Substantial synergies could be realized by explicitly representing the named entities from nano-publications. OnNER could be used for that purpose.

Representing references and other links between documents Bibliographic entities and references across scholarly documents are already modeling comprehensively by BiRO [30]. BiBO [31] models them in a more limited way, focusing on the type of document that is reference. The Citation Typing Ontology (CiTO) [32] describing and distinguishes

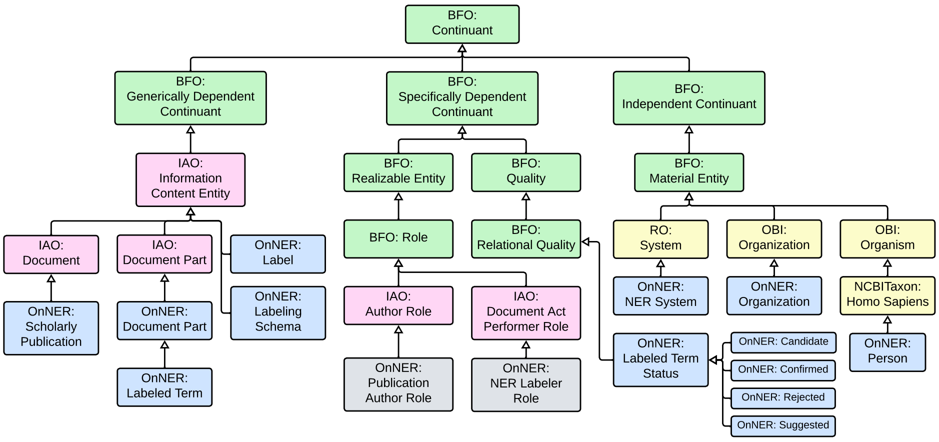

Figure 1. OnNER's higher-level classes (in blue) and their alignment with BFO (in green) and IAO (in pink). OnNER also reuses high-level concepts from RO, OBI, and NCBITaxon (in yellow), which are also aligned with BFO. *Publication Author Role* and *NER Labeler Role* are grayed out as they are not directly utilized in OnNER. The alignment with other ontologies is not shown here.

different types of citations within a paper. We reuse its most general *cites* relation, which allows using any of its more fine-grained subproperties with OnNER without complications. While we include *Bibliography* and *BibliographicEntry* as classes and *references* and *cites* in our ontology, these are rather tangential concepts within OnNER.

4. The Conceptual Design of OnNER

We now introduce and characterize the key concepts and relations required for OnNER, grouped around two key requirements: (1) describing the "named entities" themselves (Section 4.1); and (2) locating them within the structure of a document (Sections 4.2 through 4.4). Subsequently, Section 4.5 briefly discusses citations, bibliographic entries and what they refer to.

Fig. 1 shows how we align OnNER with foundational concepts from Basic Formal Ontology (BFO) and the Information Artifact Ontology (IAO) [33] to improve OnNER's semantic clarity, reusability, and integration with other BFO-aligned ontologies. We also have explored reuse and alignment with relevant classes and relations from related ontologies, specifically the SPAR Ontologies (DoCO, DEO, BiRO, CiTO). Because concepts from SPAR are intended for fairly broad use, we subclass them in order to not restrict the meaning of the reused concepts by our additional axioms in a way that conflicts with the original intent and scope of the reused ontologies.

4.1. Named Entities

In the center of Figure 2 is the class *Labeled Term*[4] that represents named entities; it has two essential data properties: *labeledTermText* denotes the labeled word or group of words, *offset* the starting position within the text fragment. The *length* is explicitly

[4] To improve readability, we add whitespaces to all class names throughout the paper though no whitespaces are used in the URIs in the ontology proper. Property names are shown unaltered.

stored as well. The object property *directlyContainsLabeledTerm* identifies which *Text Fragment* the *Labeled Term* is contained in.

Each *Labeled Term* is associated with a *Label* only indirectly via the reified relation *Labeled Term Status*. This allow revisions to the assigned *Label* to review the labels generated by NER systems, to allow disagreement among labelers and to correct classification errors later on. Thus, a *Labeled Term* may be associated via *hasLabeledTermStatus* with potentially multiple *Labeled Term Status*es of the same or different *Label*s. Each status is associated with exactly one *Label* via *hasLabeledTermLabel* and one labeler – the system, person or organization who performed the labeling – via *statusAssignedBy*. This object property is a shortcut that results from the three-hop link (formalized using a `propertyChainAxiom`) to a role (*NER Labeler Role*), the realized occurrent (i.e., the labeling activity), and finally the actual participant – an *NER System* (a subclass of *obo:RO_0002577*), *Person* or *Organization* – who performed the labeling. Subclasses of *Labeled Term Status*, such as *Candidate Status* and *Confirmed Status*, are used to distinguish different statuses. The former indicates that the named entity and its label still require review while the latter is used after review and confirmation by a domain expert. Additional custom status classes can be added as needed.

Each *Label* belongs to a particular *Labeling Schema* (sometimes called a tagging schema) that encompasses a set of distinct labels for a particular domain or purpose. Additional information about *Labeling Schema*s can be attached using Dublin Core [25] or similar metadata standards.

Nested Named Entities Some NER systems allow so-called "nested" named entities that contain other named entities as parts thereof. This is in contrast to the more common assumption of "flat" entities which do not have any named entity parts. In the example from Sec. 2.1, "nanoparticle shape" could be treated as a nested named entity that contains "nanoparticle" (classified as "structure") and "shape" (classified as "property") as parts that are named entities themselves.

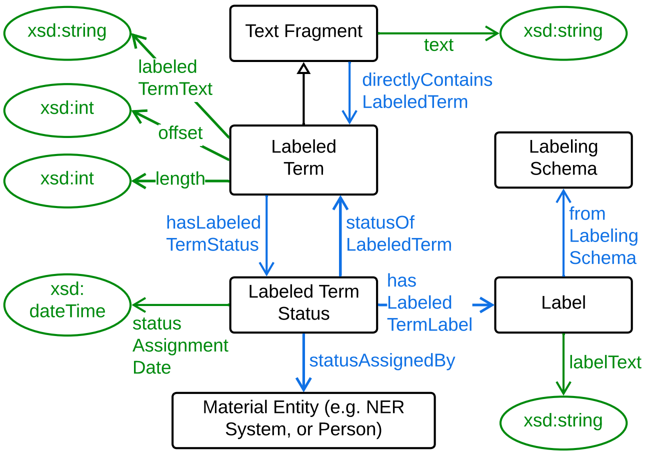

Figure 2.: *Labeled Term* is the central concept in OnNER describing named entities. It is associated with its location in the text and one or more labeling statuses, each of which indicate a status (using subclasses), a label and associated labeling schema, a labeler and the date and time when it was labeled. Object properties are shown in blue and data properties in green.

To model this distinction, we introduce *Compound Labeled Term* and *Atomic Labeled Term* as disjoint subclasses of *Labeled Term* that are or are not, respectively, linked to another *Labeled Term* via the *directlyContainsLabeledTerm* property.

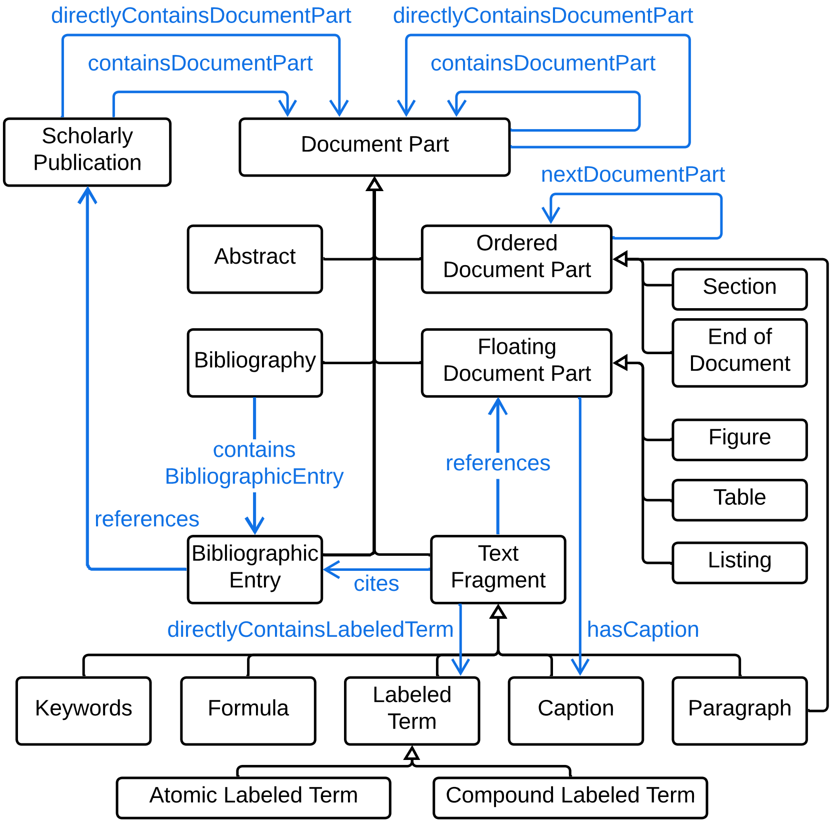

Figure 3. The subclasses of *Document Part* and the key relationships among them and to *Scholarly Publication* that encode the location of named entities (*Labeled Terms*) within the structure of the document.

4.2. Documents, Sections and Other Parts

The class *Scholarly Publication* represents the entire content of a scholarly publication rather than the artifact itself. It is a subclass of *IAO:Document* and, indirectly, of *IAO:Information Content Entity* from the IAO [33]. Datatype properties capture metadata information such as the title, DOI, publication venue, and publication date of a *Scholarly Publication* to be used for targeted queries. We connect a publication to its authors by modeling them as roles, namely *Publication Author Role* from IAO. But since we're not directly interested in the role entities, we create an object property *authorOfPublication* that directly links the *Person* to the *Scholarly Publication*s they authored.

Scholarly publications primarily consist of sections of text, as well as tables, figures and other floating elements. We introduce *Document Part* (see top of Fig. 3) that serves a common superclass for all of these different parts of a document we want to model, including *Labeled Term* and its broader class *Text Fragment*. Many of its subclasses (e.g. *Section*, *Abstract*, *Figure*, *Table*, and *Formula*) are also subclasses of DoCO classes of the same name.

The primary elements for structuring the main text within documents are *Section*s, which can be nested but also contain various other document parts such as *Paragraph*s and *Formula*s. Sections typically contain a section title and often a section number, which are captured by data properties.

Other Document Parts Tables, figures, and listings typically do not fit directly into the order of sections and paragraphs. To distinguish them, we introduce disjoint classes: *Ordered Document Part*s of which *Section* and *Paragraph* (discussed further below) are subclasses and *Floating Document Part*s. The latter cannot contain other non-floating *Document Parts* except for *Text Fragments*.

Similarly, *Abstract* and *Bibliography* often look like sections in a publication, but are not modeled as such as they cannot contain subsections and they are – despite their common location at the beginning or end of a document – outside the regular order of a document's text. The *Abstract* class refines DoCO's and IAO's *Abstract* classes and may contain one or more paragraphs. A *Bibliography* is unique in that it is composed only of *Bibliographic Entries* (via the *containsBibliographicEntry* object property) but does not contain other *Document Part*s.

Nesting of Document Parts OnNER's *containsDocumentPart* object property is central to capturing the nesting structure among *Document Part*s. It is modeled as a subproperty of the *contains* relation from SPAR's Pattern ontology [24]. It captures whenever a *Document Part* is contained in a *Scholarly Publication* or another *Document Part*. While *containsDocumentPart* applies to direct and indirect multi-level containment, we introduce the subproperty *directlyContainsDocumentPart* to capture only the immediate containment relations between, e.g., a publication and its main sections or between a section and its direct subsections. *containsDocumentPart* is then the transitive closure thereof that relates a *Scholarly Publication* or *Document Part* to *all* its nested *Sections* and other *Document Parts*, no matter how deeply they are nested. We also provide inverses of both relations.

4.3. Text Fragments as the Smallest Document Parts that Contain Named Entities

We introduce *Text Fragment* to capture smaller semantically meaningful pieces of text within a document. These are the *Document Parts* that may directly contain *Labeled Term*s. We distinguish five disjoint subclasses of *Text Fragment*. The first is *Paragraph*, which is also a subclass of DoCO's *Paragraph* class and represents paragraphs of text primarily from *Sections* or the *Abstract*.

The second subclass is *Caption*, which is also a subclass of DEO's *Caption* class and differs from *Paragraph* in that it is part of a *Floating Document Part* such as a table or figure, which it describes. This relationship is captured by *hasCaption* as a subproperty of *directlyContainsDocumentPart*.

*Formula*s are specialized pieces of text that represent mathematical expressions, chemical formulas, etc. They are modeled as subclass of *Text Fragment* and may appear within *Paragraph*s or even *Caption*s. *Formula* is not intended to include long sets of equations, which are treated as *Listing*s instead. Depending on the domain of interest, an entire *Formula*, like H_2O may be a named entity (a *Labeled Term*) or contain one ore multiple *Labeled Term*s.

Keywords are another subclass of *Text Fragment*. They typically appears with the abstract and may also contain one or more *Labeled Term*s.

Finally, for modeling consistency the class *Labeled Term* itself is a subclass of *Text Fragment*, which captures the smallest pieces of text of interest to OnNER. We introduce the object property *directlyContainsLabeledTerm* as a subproperty of *directlyContainsDocumentPart* that precisely picks out which *Text Fragment* each *Labeled Term* is directly contained in.

4.4. Relative Order between Document Parts

Until now, we only modeled the nesting of *Document Parts* but not their order. Explicitly representing the order of, e.g., paragraphs lets us ask questions about subsequent named entities even when they are not in the same paragraph or, more general, about named entities in subsequent paragraphs or sections. To help with such queries or to reconstruct the entire main text in its correct order, we don't require absolute locations but only relative locations. We use the object property *nextDocumentPart* that links any *Ordered Document Part* – either a *Section* or *Paragraph* – to its subsequent *Document Part*. The next *Document Part* of a *Section* is either the *Paragraph* that starts the *Section* or the next subsection that starts immediately without any text in between. For a *Paragraph*, the next *Document Part* is another *Paragraph* or, if it is the last *Paragraph* in its *Section*, a new *Section*. The very last *Paragraph* of a publication points via *nextDocumentPart* to an instance of the dedicated class *EndOfDocument* that explicitly indicates the end of the regular text. Only *Floating Document Part*s, a *Bibliography* or similar non-ordered *Document Part*s may follow after in the printed or digital document.

4.5. Links Across Documents: Bibliography, References and Citations

Scholarly writing often uses citations for justifying the presented ideas or pointing to related work. We use the *cites* relationship – modeled as a subproperty of CiTO's *cites* – to relate a *Text Fragment* to a *Bibliographic Entry* that specifies the full details of the cited work. The *Bibliographic Entries* of a document are often listed in a single *Bibliography*[5] located at the end of a document, but certain citation style may also include *Bibliographic Entries* as a foot or at the end of a *Section*. We do not restrict their location but require a *Bibliography* – if a document has one – to only consists of *Bibliographic Entries*. The *Scholarly Publication* that a *Bibliographic Entry* points to can be indicated by the *references* object property, which is a subproperty of BiRO's *references*.

5. Evaluation

OnNER has been formalized as an OWL2 ontology in the OWL2-DL profile using the turtle syntax. It is publicly available on GitHub at https://github.com/thahmann/OnNER. Its current version (covering only an TBox) includes 66 classes, 46 object properties, 24 data properties, and 284 logical axioms. The axioms include 79 subclasses, 6 disjoint classes, 22 sub object properties, 20 inverse object properties, 16 asymmetric and irreflexive object properties, 33 domain and 32 range restriction for object properties, and 2 transitive object properties.

The ontology has been tested for common issues, such as missing domain or range restrictions, or the existence of inverses for symmetric properties, using the OntOlogy Pitfall Scanner (OOPS!) [34]. Some issues identified in earlier versions were corrected. We carefully examined the remaining pitfalls raised by OOPS! and ensured that they do not pose any serious issues.

Before starting to instantiate the ontology with data, we verified its logical consistency using the Pellet OWL2 reasoner [35] that is integrated with Protégé. No inconsistencies were found, we further inspected all inferences (such as inferred subclass relationships) in Protégé to ensure that no unexpected inferences result.

[5] *Bibliography* and *Bibliographic Entry* are modeled as subclasses of the similar concepts *Bibliographic Collection* and *Bibliographic Record* from BiRO.

5.1. Populating the Ontology to Build a Knowledge Graph

Our next verification step involves instantiating the ontology with data, i.e. creating an ABox, and checking that it is consistent with the TBox. This kind of *external verification* ensures that (1) the ontology can be used together with the kind of datasets it is intended to be used with and that (2) no inconsistencies emerge once the classes and properties are instantiated.

For this verification, we populate the ontology (i.e., the TBox) with data that has been automatically generated from five scholarly publications. As source files we use PDF version of these publications, which are then converted to an XML format using a customized pipeline using the GROBID tool[6], which extracts bibliographic information and metadata, references, citations and the structure of the text of PDF documents. We then generate triples that represent the publication itself, essential metadata, the abstract and the structure and content of the main text. In the process, we generate URIs for each publication (using its DOI) and each of its sections and paragraphs. For simplicity, we currently do not create instances for figures, other floating elements, and the bibliography yet as they contain few relevant named entities. The text of each paragraph is then passed to an NER model[7] that produces a list of named entities, their labels and positions within the paragraph. Those are triplified and added to the RDF file that represents that particular publication[8].

Each publication results in one Turtle file that import the OnNER ontology. Altogether, our example ABox generated from five selected publications spans 252 paragraphs with a total of 4,548 labeled terms. The deployed knowledge graph (ABox and TBox) comprised 228,878 triples. The bottom-left of Fig. 4 shows an example of a single named entity with the text "nanoparticle shape" that is identified by the URI *data:10.1016_j.memsci.2018.03.085_1-3-3*. It is an instance of *Labeled Term*; its relationship to the paragraph, section and publication (the example is from [12]) it is contained in are shown at the top of Fig. 4. The bottom-right shows that the *Labeled Term* instance has been identified as a candidate of the label "PROPERTY", which is a label from our "CelloGraph" *Labeling Schema* CelloGraph and has been assigned by the NER system called here "Cellulosic_NER_Model" on March 15, 2024.

We loaded the ABox together with the ontology from Protégé and reran the Pellet reasoner to ensure that no inconsistencies emerge and that no unexpected consequences, especially among the type and subclass relationships, are inferred by the reasoner.

5.2. Validation: Querying the Knowledge Graph

Our last but most important evaluation step tests the ontology's adequacy forexpressing and answering the competency questions that guided its design. We first loaded the TBox and sample ABox into Ontotext's GraphDB[9], a triple store that supports SPARQL queries and OWL2 inferencing. Subsequently, we translated our guiding competency questions that reflect the range of intended uses of the ontology, into SPARQL queries[10]

[6] https://github.com/kermitt2/grobid

[7] We use a custom NER model trained using spaCy's [36] NER model training pipeline that uses labels relevant to cellulose materials.

[8] The entire process is implemented in Python;the code and the NER model are available from the project's GitHub repository at https://github.com/thahmann/OnNER

[9] https://www.ontotext.com/products/graphdb

[10] The full set of SPARQL queries are shared in the `evaluation` folder of the OnNER repository.

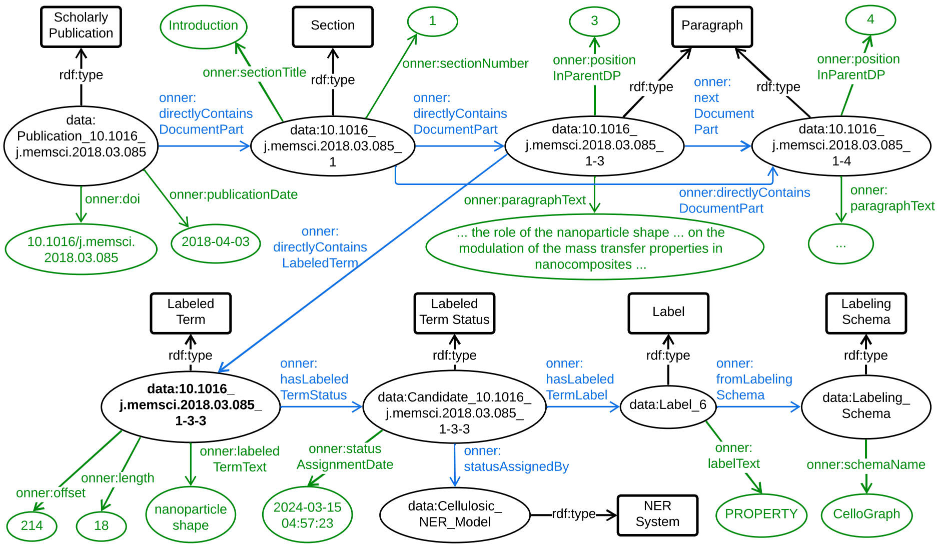
Scholarly Publication
rdf:type
data: Publication_10.1016_ j.memsci.2018.03.085
onner:doi
10.1016/j.memsci. 2018.03.085
onner:publicationDate
2018-04-03
Introduction
onner:sectionTitle
Section
rdf:type
1
onner:sectionNumber
onner: directlyContains DocumentPart
data:10.1016_ j.memsci.2018.03.085_ 1
onner: directlyContains DocumentPart
3
onner:position InParentDP
Paragraph
rdf:type
rdf:type
onner: next Document Part
4
onner:position InParentDP
data:10.1016_ j.memsci.2018.03.085_ 1-3
data:10.1016_ j.memsci.2018.03.085_ 1-4
onner:paragraphText
onner:directlyContains DocumentPart
onner: paragraphText
... the role of the nanoparticle shape ... on the modulation of the mass transfer properties in nanocomposites ...
...
onner: directlyContains LabeledTerm
Labeled Term
rdf:type
data:10.1016_ j.memsci.2018.03.085_ 1-3-3
onner:offset
214
onner:length
18
onner:labeled TermText
nanoparticle shape
onner: hasLabeled TermStatus
Labeled Term Status
rdf:type
data:Candidate_10.1016_ j.memsci.2018.03.085_ 1-3-3
onner:status AssignmentDate
2024-03-15 04:57:23
onner: statusAssignedBy
data:Cellulosic_ NER_Model
rdf:type
NER System
onner: hasLabeled TermLabel
Label
rdf:type
data:Label_6
onner: labelText
PROPERTY
onner: fromLabeling Schema
Labeling Schema
rdf:type
data:Labeling_ Schema
onner:schemaName
CelloGraph

Figure 4. A small example illustrating how OnNER's ABox is populated.

and executed them in the knowledge graph. For example, one competency questions asks: *Which "chemicals" are mentioned in conjunction with "permeability"? In which publications and paragraphs?*. It can be best expressed in OnNER as: *Retrieve all the labeled terms that have been assigned the label "chemical" are mentioned in the same paragraph as a named entity with the text "permeability". Also return the paragraphs and publications where this happens*, which in turn is encoded using the SPARQL query shown below. The query returns chemicals such as lignin ($C_{81}H_{92}O_{28}$), carbon monoxide (CO_2) and carbon dioxide (CO_2).

```
SELECT ?chemical ?paragraph ?publication WHERE {
    ?publication onner:containsDocumentPart ?paragraphID .
    ?paragraphID rdf:type onner:Paragraph ;
        onner:paragraphText ?paragraph ;
        onner:directlyContainsLabeledTerm ?ne1 , ?ne2 .

    ?ne1 onner:labeledTermText ‘permeability’^^xsd:string .

    ?ne2 onner:labeledTermText ?chemical ;
        onner:hasLabeledTermStatus ?ne2Status .
    ?ne2Status onner:hasLabeledTermLabel ?label2 .
    ?label2 onner:labelText ‘CHEMICAL’^^xsd:string . }
```

Listing 1: Example of one of OnNER's guiding competency questions as SPARQL query. It asks for any named entities labeled as "CHEMICAL" that appear together (in the same paragraph) with a named entity containing the text "permeability'.

6. Discussion and Summary

Motivated by the need for explicitly capturing detailed information about named entities in scholarly publications, we have introduced OnNER as a semantic model for describing named entities and where they appear. It is guided by a set of competency questions that specify the different types of queries OnNER is intended to support.

OnNER is aligned with higher-level concepts from the Basic Formal Ontology (BFO) and Information Artifact Ontology (IAO) to make sure the model adheres to established ontological concepts and to maximize integration with other BFO-aligned ontologies. Likewise, OnNER's classes reuse the names (though with different namespaces) and connect via subclass relations to existing ontologies about document structure and citations, such as DoCO, DEO, CiTO, and BiRO. More direct reuse is avoided because of significant differences in scope and because we axiomatize many of those concepts more stringently.

OnNER has been populated with data that has been generated automatically from the PDF versions of five publications. Translating and executing the competency questions as SPARQL queries over the ontology and proves that the ontology is suitable for its intended purpose. Moreover, it showcases its versatility for posing fine-grained questions about the named entities that have been identified in the literature. All steps of populating the ontology are fully automated, thus enabling the rapid construction of large knowledge graphs of named entities for various domains with the help of domain-specific NER tools. Unlike prior work, OnNER is a domain-independent representation of named entities that can accommodate various domains, NER tagging schemata, and NER tools.

In the future, we plan to use OnNER as the underlying representation for an open-source NER tagging tool that supports reviewing and correcting named entities and their labels. We also plan to drastically expand OnNER's capabilities by connecting the named entities explicitly to concepts from existing or emerging domain ontologies.

References

[1] Martinez-Rodriguez JL, Hogan A, Lopez-Arevalo I. Information extraction meets the semantic web: a survey. Semantic Web. 2020;11(2):255-335.

[2] Jiang J. Information extraction from text. In: Aggarwal C, Zhai C, editors. Mining Text Data. Springer; 2012. p. 11-41.

[3] Nadeau D, Sekine S. A survey of named entity recognition and classification. Lingvisticae Investigationes. 2007;30(1):3-26.

[4] Damle D, Uren V. Extracting significant words from corpora for ontology extraction. In: Intern. Conf. on Knowledge Capture (K-CAP). ACM; 2005. p. 187–188.

[5] Wang K, Stevens R, Alachram H, Li Y, Soldatova L, King R, et al. NERO: a biomedical named-entity (recognition) ontology with a large, annotated corpus reveals meaningful associations through text embedding. npj Systems Biology and Applications. 2021;7(1).

[6] Eltyeb S, Salim N. Chemical named entities recognition: a review on approaches and applications. J of Cheminformatics. 2014;6:1-12.

[7] Krallinger M, Rabal O, Leitner F, Vazquez M, Salgado D, Lu Z, et al. The CHEMDNER corpus of chemicals and drugs and its annotation principles. J of Cheminformatics. 2015;7(1):1-17.

[8] Navigli R, Velardi P. Learning domain ontologies from document warehouses and dedicated web sites. Comput Linguist. 2004;30(2):151–179.

[9] Drymonas E, Zervanou K, Petrakis EGM. Unsupervised ontology acquisition from plain texts: the OntoGain system. In: Intern. Conf. on Applications of Natural Language to Inf. Syst. (NLDB). vol. 6177 of LNCS. Springer; 2010. p. 277-87.

[10] Shen W, Wang J, Han J. Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering. 2015;27(2):443-60.

[11] Tedeschi S, Conia S, Cecconi F, Navigli R. Named Entity Recognition for Entity Linking: What works and what's next. In: Findings of the Association for Computational Linguistics: EMNLP 2021; 2021. p. 2584-96.

[12] Wolf C, Angellier-Coussy H, Gontard N, Doghieri F, Guillard V. How the shape of fillers affects the barrier properties of polymer/non-porous particles nanocomposites: A review. Journal of Membrane Science. 2018;556:393-418.

[13] Tjong Kim Sang E, De Meulder F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Conf. on Natural Language Learning; 2003. p. 142-7.

[14] Buitelaar P, Cimiano P, Magnini B. Ontology learning from text: An overview. Ontology learning from text: Methods, evaluation and applications. 2005;123.

[15] Watrobski J. Ontology learning methods from text - an extensive knowledge-based approach. Procedia Computer Science. 2020;176:3356-68.

[16] Al-Aswadi F, Chan HY, Gan KH. Automatic ontology construction from text: a review from shallow to deep learning trend. Artif Intell Rev. 2020;53:3901–3928.

[17] Koho M, Leal R, Ikkala E, Tamper M, Rantala H, Hyvönen E. Building lightweight ontologies for faceted search with named entity recognition: Case WarMemoirSampo. In: Intern. Workshop on Knowledge Graph Generation From Text (TEXT2KG 2022). vol. 3184 of CEUR-WS; 2022. p. 19-35.

[18] Revenko A, Mireles V, Breit A, Bourgonje P, Moreno-Schneider J, Khvalchik M, et al. Learning ontology classes from text by clustering lexical substitutes derived from language models1. In: Towards a Knowledge-Aware AI. IOS Press; 2022. p. 155-69.

[19] Ruiz-Iniesta A, Corcho Ó. A review of ontologies for describing scholarly and scientific documents. In: Workshop on Semantic Publishing at ESWC 2014. vol. 1155 of CEUR-WS; 2014. .

[20] Peroni S, Shotton D. The SPAR Ontologies. In: Intern. Semantic Web Conference (ISCW); 2018. .

[21] Constantin A, Peroni S, Pettifer S, Shotton D, Vitali F. The Document Components Ontology (DoCO). Semantic Web. 2016;7(2):167-81.

[22] Shotton D, Peroni S. The Discourse Elements Ontology (DEO); 2015.

[23] Ciccarese P, Peroni S. The Collections Ontology: creating and handling collections in OWL 2 DL frameworks. Semantic Web. 2014;5(6):515-29.

[24] Iorio AD, Peroni S, Poggi F, Vitali F. Dealing with structural patterns of XML documents. J of the American Society for Information Science and Technology. 2014;65(9):1884–1900.

[25] DCMI Usage Board. DCMI Metadata Terms; 2020. Available from: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.

[26] Ciccarese P, Grosza T, Clark T, Waard A. Ontology of Rhetorical Blocks (ORB); 2011.

[27] Liakata M, Saha S, Dobnik S, Batchelor C, , Rebholz-Schuhmann D. Automatic recognition of conceptualization zones in scientific articles and two life science applications. Bioinformatics. 2012;28(7):991–1000.

[28] Groth P, Gibson A, Velterop J. The anatomy of a nanopublication. Information Services & Use. 2010;30(1-2):51-6.

[29] Kuhn T, Meroño-Peñuela A, Malic A, Poelen JH, Hurlbert AH, Ortiz EC, et al. Nanopublications: A Growing Resource of Provenance-Centric Scientific Linked Data. In: Intern. Conf. on e-Science. IEEE Computer Society; 2018. p. 83-92.

[30] Di Iorio A, Nuzzolese AG, Peroni S, Shotton DM, Vitali F. Describing bibliographic references in RDF. In: Workshop on Semantic Publishing at ESWC 2014. vol. 1155 of CEUR-WS; 2014. .

[31] D'Arcus B, Giasson F. Bibliographic Ontology (BIBO) in RDF; 2006. Available from: https://www.dublincore.org/specifications/bibo/bibo/.

[32] Shotton D. CiTO, the citation typing ontology. In: Journal of Biomedical Semantics. vol. 1. Springer; 2010. p. 1-18.

[33] Ceusters W, Smith B. Aboutness: Towards Foundations for the Information Artifact Ontology. In: Intern. Conf. on Biomedical Ontology (ICBO 2015). vol. 1515 of CEUR-WS; 2015. .

[34] Poveda-Villalón M, Gómez-Pérez A, Suárez-Figueroa MC. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. Intern J on Semantic Web and Inf Syst2̂014;10(2):7-34.

[35] Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y. Pellet: a practical OWL-DL reasoner. Web Semantics: science, services and agents on the World Wide Web. 2007;5(2):51-3.

[36] Honnibal M, Montani I, Van Landeghem S, Boyd A. spaCy: Industrial-strength Natural Language Processing in Python; 2020.