DeLTA: A Description Logic–based Annotation Schema for Constructing Expressive OWL DL Axioms from Text

Ildar Baimuratov[1,2]

[1] L3S Research Center, Leibniz University Hannover, Germany
[2] TIB – Leibniz Information Centre for Science and Technology, Hannover, Germany

Abstract

Enriching an ontology with complex axioms enables the inference of new knowledge, contextualized querying, and consistency checking, but it is error-prone and time-consuming. A further challenge is that domain experts are rarely ontology engineers, while ontology engineers often lack sufficient domain knowledge. This "knowledge acquisition bottleneck" has led to a prevalence of inexpressive ontologies, highlighting the need to facilitate the acquisition of expressive description logic axioms. In this work, we propose a domain-independent text annotation schema that maps directly to OWL DL syntax, enabling the algorithmic generation of expressive axioms from the annotations. Annotated text allows domain experts to participate in the loop, ensuring the trustworthiness of the resulting knowledge base. Additionally, the annotation process can later be automated using NLP once a sufficient number of annotations have been completed. We demonstrate the applicability and scalability of the approach through two distinct use cases: building requirements and scientific claims.

Keywords

Description Logics, Text annotation, Knowledge acquisition, OWL DL

1. Introduction

Relevance. Successfully developing an ontology requires careful capture of relevant domain knowledge, which is an error-prone and time-consuming task. An additional challenge is that domain experts are rarely experts in ontology engineering, and conversely, ontology engineers are often not sufficiently familiar with the domain being modeled. This difficulty in capturing knowledge is known as the "knowledge acquisition bottleneck" [1]. This bottleneck has resulted, among other consequences, in a prevalence of inexpressive ontologies. For example, Wang et al. [2] surveyed 1,261 OWL ontologies and found that only 12% had the expressivity of OWL DL, and only 33% had at least the expressivity of OWL Lite. In this work, we address the need to facilitate the acquisition of expressive description logic (DL) axioms from natural language text, including logical connectives, negation, role restrictions, and other features.

Limitations of existing approaches. Current methods for facilitating knowledge acquisition from natural language text include NLP-based knowledge extraction and machine learning–based ontology learning. However, these approaches are mostly limited to inexpressive ontologies and often lack accuracy. Additionally, in both cases, domain experts have no option to correct or validate the extracted semantics. In contrast, an annotation-driven approach, where natural language text is visually aligned with semantic tags and the annotations are algorithmically translated into a knowledge representation framework, enables, on the one hand, human interaction through annotation editing, and on the other hand, support from NLP and ML techniques once a sufficient number of annotations has been completed. Nevertheless, existing annotation-driven approaches for knowledge acquisition from natural language remain limited in expressive power and are bound to specific domains, such as law, building construction, or scholarly knowledge.

DL 2026: 39th International Workshop on Description Logics, July 17–19, 2026, Lisbon, Portugal
ildar.baimuratov@l3s.de (I. Baimuratov)
0000-0002-6573-131X (I. Baimuratov)

Contributions. The contributions of this work are: (1) a domain-independent text annotation schema that maps to OWL DL syntax, and (2) an algorithm for the automatic construction of OWL DL axioms from annotated text. Our approach is validated through a proof-of-concept implementation and its application to two use cases from different domains: building requirements and scientific claims.

2. Related Work

Before discussing related work, we clarify which research is *not* relevant to our study:

- Numerous studies on semantic annotation that do not go beyond named entity recognition, entity linking, semantic role labeling, or relation extraction, i.e., work that does not address the construction of complex, syntactically valid expressions.
- Markups such as Description Logic Markup Language[1] and OWL DL itself, which focus on encoding description logics in a machine-readable format rather than visually aligning them with natural language via text annotation.
- Studies of logics and the semantics of annotations themselves, such as detecting conflicting annotations [3], providing formal semantics of markup elements [4], or developing formal frameworks for linguistic annotations [5].
- The acquisition of DL axioms from non-textual sources, such as structured data [6].

2.1. Automatic Knowledge Extraction from Text

To the best of our knowledge, there is no domain-independent approach that enables the acquisition of expressive DL axioms from natural language via text annotation. However, we review approaches for extracting DL axioms from text that do not rely on annotation. Text2Onto [7] is a domain-independent ontology learning framework based on Probabilistic Ontology Model. Welty and Murdock [8] leveraged information extraction from text to populate knowledge bases suitable for reasoning. Völker et al. [9] presented a method for extracting expressive OWL DL axioms based on a syntactic analysis of natural language definitions. DOG4DAG (Dresden Ontology Generator for Directed Acyclic Graphs) [10] allows for the generation of terms, sibling, definition, and parent-child relationships from text. Sánchez et al. [11] proposed a methodology for learning relational axioms, such as functional, inverse, symmetric, transitive, reflexive, and inverse-functional properties, using linguistic patterns and web search engines. De Azevedo et al. [12] described an approach for automatic construction of OWL DL ontologies with ALC expressivity. Rios-Alvarado et al. [13] proposed a method for acquiring taxonomic, disjoint, and equivalent relations from text using a syntactic dependency parser, a clustering algorithm, and WordNet. Gyawali et al. [14] addressed the enrichment of existing OWL DL ontologies with complex axioms using syntactic parsing and lexicon mapping. There are also studies on ML-based, end-to-end translation of natural language into DL axioms. Petrucci et al. [15] trained a Recurrent Neural Network (RNN) to extract OWL formulas from text. In a follow-up study, Petrucci et al. [16] used an encoder–decoder architecture. More recently, Rossiello et al. [17] leveraged the BART model [18]. These approaches focus primarily on the automatic discovery of concepts, relations, and simple structures from text corpora, rather than human-in-the-loop construction of expressive OWL DL representations. The two approaches are complementary rather than competing. Knowledge extraction systems such as Text2Onto can assist by suggesting annotations for specific tags, while the proposed schema provides a framework for the curated construction of complex OWL axioms from these annotations.

2.2. Annotation-driven Methods from Specific Domains

We also consider annotation-based approaches for extracting expressive axioms from natural language text that, however, were designed for specific application domains and do not aim at OWL DL expressivity.

[1] https://co4.inrialpes.fr/xml/dlml/

Building construction. Despite a wide use of rule-based approaches for automatic compliance checking in building construction, only few annotation-based approaches were proposed, including RASE, with a translation to SWRL [19], and the approach by Li et al. [20] that involves annotating regulatory text with a set of semantic roles and converting them into a knowledge graph.

Law. Several approaches have employed text annotation for the formalization of legal texts. Nazarenko et al. [21] annotated legal text with elements related to LegalRuleML. Libal [22] proposed an annotation tool and Legal Linguistic Templates, which are translated into Prolog clauses.

Scholarly knowledge. In the domain of scholarly knowledge representation, the use of text annotation is more established, however, the target representations are usually unexpressive. Liakata et al. [23] presented annotation schemas CoreSC, based on the EXPO ontology, and AZ-II, focusing on connections between papers. Sateli and Witte [24] developed a vocabulary for labeling sentences with rhetorical entities such as *Claim* and *Contribution*. Several annotation schemas have been implemented in LaTeX, including SALT [25], PROV [26], sTEX [27], RDFTeX [28], and SciKGTeX [29], which enable exporting annotations in RDF format.

3. Method

In this section, we present DeLTA — a domain-independent **De**scription **L**ogic-based **T**ext **A**nnotation schema, and the algorithm for translating the annotations into OWL DL axioms.

3.1. Annotation Schema

The annotation schema includes span-based tags and arrows connecting them, organized into several annotation layers. Each annotation layer corresponds to a function that maps a token in the input text to a tag from a predefined tag set. In this sense, layers represent independent, parallel annotation dimensions over the same token sequence. In text annotation environments, annotation layers can be stacked and differentiated via color coding. The DeLTA annotation schema presented in this paper consists of two layers: Semantic Types and Semantic Roles.

Semantic Types. The Semantic Types layer includes tags corresponding to syntactic elements of the OWL DL language, enabling the construction of class restrictions. We primarily use Manchester syntax[2] for tag names, although some are slightly modified to improve readability for domain experts. For example, `owl:ObjectProperty` is represented by the tag *Relation*, and `owl:DataProperty` corresponds to *Property*. The mapping of these tags to OWL syntax is provided in Table 1. It can be seen that Table 1 contains no tag corresponding to `owl:intersectionOf`. This is because intersections between classes are applied by default during translation into OWL and therefore do not require explicit annotation. In terms of DL, the list of Semantic Types corresponds to the ALC logic extended with full existential restrictions ($\mathbf{E}$), qualified number restrictions ($\mathbf{Q}$), role compositions ($\circ$), and datatypes ($\mathbf{D}$).

Additionally, the Semantic Types layer includes the specific annotation arrows between tags: *Domain*, *Range*, and *Of*, which do not map to OWL syntax but facilitate the algorithmic construction of DL axioms. *Domain* and *Range* are used to connect *Relation* and *Property* tags, being binary directed mappings, while *Of* is used to connect unary or undirected operations, such as *Not*, *Comparison*, various restrictions, and *Or*. Their possible starting and ending tags are listed in Table 2. For example, in the case of an n-ary disjunction, a token representing the disjunction is annotated with the tag *Or* and connected to each disjunct via an *Of* arrow.

[2] https://www.w3.org/TR/owl2-manchester-syntax/

Table 1
Mapping of Semantic Types to the OWL DL syntax

Tag	OWL
Literal	rdfs:Literal
Class	owl:Class
Not	owl:complementOf
Or	owl:unionOf
Relation	owl:ObjectProperty
Property	owl:DataProperty
Some	owl:someValuesFrom
Only	owl:allValuesFrom
Number	owl:maxCardinality, owl:minCardinality, owl:cardinality
Comparison	xsd:minInclusive, xsd:minExclusive, xsd:maxInclusive, xsd:maxExclusive

Table 2
Arrows connecting Semantic Type tags

Arrow	Start	End
Domain	Relation, Property	Class, Relation
Range	Relation	Class, Relation, Property
	Property	Literal
Of	Not, Or	Class, Relation, Property
	Comparison	Literal
	Some, Only, Number	Relation, Property

Semantic Roles. Semantic Roles are required for connecting restricted classes, resulting from annotated Semantic Types, into TBox axioms. In this work, we focus on asserting only General Class Inclusion (GCI), implemented in OWL as the relation `owl:SubClassOf`. To annotate this relation, two semantic roles are introduced: (1) *Subject*, which represents the subclass, and (2) *Restriction*, which represents the superclass. The name *Restriction* indicates that the superclass usually expresses a class restrictions imposed on the subject class. The arrow between *Subject* and *Restriction*, corresponding to the `owl:SubClassOf` relation, is annotated with the tag *To*, resulting in the triple $< Subject, To, Restriction >$.

Annotation interface. To implement the proposed annotation schema, we use the INCEpTION annotation environment [30]. One of the key advantages of this tool is its ability to import ontologies, which can then be used as annotation tags. Additionally, this tool supports the integration of ML-based recommenders, which can streamline the annotation process in future work. The original regulations are imported into INCEpTION in TXT format, and the annotations are exported in WebAnno TSV v3.3. The annotation environment, as well as other related software implementing the proposed approach, is available on GitHub[3].

3.2. Translation to OWL DL

The algorithm for transforming the annotations into OWL DL code is divided into several steps: (1) annotation preprocessing, (2) generating elementary OWL entities, (3) lexical mapping, (4) constructing class definitions, and finally, (5) constructing axioms. The algorithm takes as input the annotations in a tabular format exported from INCEpTION. In this table, each row corresponds to a textual token, indexed by its order of appearance in the text, and each column represents a distinct annotation layer, with the annotated tags as values. The translation algorithm is implemented in Python using the Owlready2 library [31].

Preprocessing. Before generating OWL code, the data undergoes preprocessing. Tokens are grouped and extracted into separate tables corresponding to the annotation layers, Semantic Types (ST) and Semantic Roles (SR), so that each table row corresponds to exactly one tag.

[3]https://github.com/ldrbmrtv/delta

Generating entities. After obtaining separate tables ST and SR, the next step involves generating elementary OWL entities based on the tags *Class*, *Relation* and *Property* from ST, according to the mappings in Table 1. Additionally, OWL classes are created based on the tags *Subject* and *Restriction* from SR. The original tokens from the text are assigned as labels for the generated entities.

Lexical mapping. To reduce the complexity of the annotation, we employ lexical mappings for certain tags from ST to OWL without loss of accuracy. Specifically, we define two mappings: $Card$ and $Constr$. The mapping $Card : T \rightarrow N$ associates word-based tokens T with integers N, for example, the word "two" is mapped to the integer 2. These integers are later used for constructing cardinality restrictions. The mapping $Constr : T \rightarrow \{<, \leq, \geq, >\}$ is used to define constrained data types. For instance, a token such as "not more than", labeled with the tag *Comparison*, is mapped through $Constr$ to `xsd:maxInclusive`. This approach relieves annotators from manually entering integers or selecting from the four possible constraints, when these values can be recognized automatically.

Constructing class definitions. Next, we use the remaining tags from ST to construct class definitions for the classes corresponding to the *Subject* and *Restriction* tags. For brevity, we refer to the combined set of object properties and data properties as predicates P. First, we identify the set of classes and literals R that appear at the end of predicate chains, i.e., those that are ranges but not domains for other predicates. We then apply backward recursion to iteratively construct expressions E using the ST table until it is empty. In each iteration, we check whether any *Not* or *Or* tags are associated with elements $r \in R$ via the *Of* arrow and apply `owl:complementOf` or `owl:unionOf`, respectively. Additionally, if r is a *Literal*, we check whether it is associated with a *Comparison* tag. If so, we construct a constrained datatype using this literal and the associated datatype constraint. For each r, we identify its incoming predicates through *Range* arrows and determine, for every such predicate $p \in P$, its associated restriction, *Some*, *Only*, or *Number*, using the *Of* arrow. If p has no associated restriction, we assign *Some* by default if it belongs to the *Subject* in SR, and *Only* if it belongs to the *Requirement*. The identified restriction, predicate, and range are combined into an expression e. Each e is added to the set E, while the original r is removed from R. Then, we identify the domains D of the predicate p using the *Domain* arrow and, if they are not empty, pass them to the next iteration of the algorithm. Once all predicate chains are fully processed, the expressions in E are joined with the remaining elements in R. The final set E forms the complete set of building blocks for defining the *Subject* and *Restriction* classes. Because the algorithm is fairly complex, we provide a more formal outline in Algorithm 1. Being recursive, it can handle annotations of arbitrary complexity.

Constructing axioms. Using all expressions E, we construct axioms that accurately capture the semantics of the original text and enable automated reasoning. To achieve this, for each $sr \in SR$, we identify the corresponding subset $E_{sr} \subset E$ based on textual token inclusion. The elements of E_{sr} are then combined using the `owl:intersectionOf` relation to form a single complex class, which is declared equivalent to the given sr. Finally, we use the *To* arrow between semantic roles in SR to establish an `owl:subClassOf` relation between the classes representing subjects and restrictions. As a result, the proposed algorithm, despite certain limitations, translates an annotation into machine-interpretable OWL DL code.

4. Validation

Our approach addresses the extraction of expressive DL axioms from natural language via text annotation. It features (1) domain independence of the annotation schema and (2) algorithmic translation of annotations into expressive DL axioms. Accordingly, the validity criteria of our approach are not based on quantitative metrics typically used in text annotation, such as inter-annotator agreement. Annotations produced by different annotators may be identical yet still fail to yield meaningful DL expressions. Instead, to demonstrate domain independence, we consider two use cases from different domains: building requirements and scientific claims. To demonstrate the completeness of the translation algorithm, we selected examples that cover all Semantic Types, except for the *Or* tag (see subsection 5.1), and we illustrate step by step the application of the algorithm to these examples. Finally, we ensure that the generated OWL DL expressions are syntactically correct and enable decision-making support through automated reasoning.

4.1. Use Case 1: Building Requirements

DL-based representations of building requirements enable computational support for matching buildings with requirements, automated compliance checking, and consistency checking of requirements. As an example of a building requirement, we use a provision from Clause 1.24 of UK Approved Document F1[4]:

Example 1. *If the dwelling only has one habitable room, a minimum ventilation rate of 13 l/s should be used.*

This requirement appears in the CODE-ACCORD corpus [32], which contains building regulations annotated with named entities and relationships.

Annotation. Figure 1 demonstrates a DeLTA annotation of Example 1. The Semantic Type tags are yellow, and Semantic Role tags are purple. Besides classes, relations and properties, this requirement involves expressive DL constructs such as a universal restriction, a cardinality restriction, a numerical value, and a datatype constraint. However, this requirement does not illustrate the use of other tags, such as *Not*, *Or* and *Some*. Some of these are involved in the second use case.

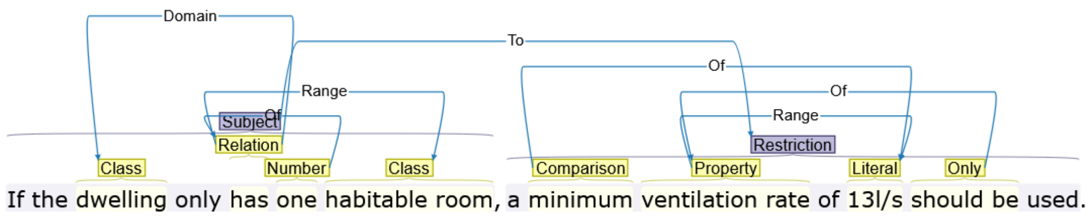

Figure 1: DeLTA annotation of a building requirement

Preprocessing. Table 3 presents the Semantic Type tags, and Table 4 presents the Semantic Role tags extracted from the annotation in Figure 1. Each row in both tables corresponds to a single annotated tag. The tables include columns for the tag's index in its order of appearance, the textual token annotated with the tag, and the tag label itself. In addition, both tables contain columns representing semantic arrows, whose values indicate the indices of the tags to which the arrows point. Accordingly, the Semantic Type table also includes columns for the arrows *Domain*, *Range* and *Of*, while the Semantic Role table contains only one additional column for the arrow *To*.

Generating entities. Based on the *Class*, *Relation*, and *Property* tags in the Semantic Type table, we create corresponding OWL classes, object properties, and data properties. Additionally, we create OWL classes for both *Subject* and *Restriction* from the Semantic Role table. Figure 2 demonstrates all elementary OWL entities generated from the annotation in Figure 1.

Lexical mapping. The example contains *Number* and *Comparison* tags, whose tokens should be mapped to integers and numerical constraints, respectively. Thus, two mappings are required: (1) $Card$: "One" $\rightarrow$ 1, and (2) $Constr$: "minimum" $\rightarrow$ `xsd:minInclusive`. The resulting integer 1 and `xsd:minInclusive` are then used in further processing instead of the tokens "One" and "minimum".

[4]https://assets.publishing.service.gov.uk/media/61deba42d3bf7f054fcc243d/ADF1.pdf

Table 3
Semantic Types for the annotation in Figure 1

Index	Token	Tag	Domain	Range	Of
1	dwelling	Class	-	-	-
2	has	Relation	1	4	-
3	one	Number	-	-	2
4	habitable room	Class	-	-	-
5	minimum	Comparison	-	-	7
6	ventilation rate	Property	-	-	-
7	13	Literal	-	-	-
8	should be	Only	-	-	6

Table 4
Semantic Roles for the annotation in Figure 1

Index	Token	Tag	To
1	If the dwelling only has one habitable room	Subject	1
2	a minimum ventilation rate of 13 l/s should be used	Restriction	-

Constructing class definitions. Using the semantic arrows *Domain*, *Range*, *Of*, and recursion to handle nested structures, the remaining tags in the table of Semantic Types, i.e., *Number*, *Comparison*, *Literal*, and *Only*, are combined into complex expressions. These expressions are then assigned to OWL classes representing *Subject* and *Restriction* as class definitions. The following listing in OWL Manchester Syntax defines the *Subject* of the example:

```
Subject EquivalentTo: Dwelling and (has exactly 1 HabitableRoom)
```

The second listing defines the Restriction:

```
Restriction EquivalentTo: ventilationRate only xsd:float[>= 13.0f]
```

These expressions capture the precise semantics of the original requirement, allowing automated reasoning over the domain knowledge.

Constructing axioms. Finally, we connect the *Subject* and *Restriction* OWL classes using `owl:subClassOf`, corresponding to the annotated *To* arrow. Figure 3 illustrates the resulting OWL representation of the requirement from Example 1, generated from the annotation shown in Figure 1.

In this way, a DeLTA annotation is algorithmically transformed into OWL DL code. When reasoning over this representation, a *Dwelling* that has exactly one habitable room will be automatically classified under the *Subject* class. If it has a ventilation rate below 13, an inconsistency will be detected, because the dwelling is required to belong to the *Restriction* class as a superclass but does not satisfy the class definition, see the reasoning explanation in Figure 4. This indicates a violation of the requirement.

4.2. Use Case 2: Scientific Claims

Another case we use to validate our approach is the representation of scientific claims. Similarly to building requirement formalization, DL-based representations of scientific claims provide several advantages, such as deriving consequences, evidence-based verification, and consistency checking of scientific theories. As an example, we consider the following claim:

Example 2. *The use of ultrasound irradiation in biodiesel transesterification reactions does not require any additional heating.*

This claim is part of a dataset of scientific claims[5] collected by Bucur et al. [33] for formalization in higher-order logic.

[5] https://github.com/LaraHack/linkflows_claims_dataset

Algorithm 1 Constructing class definitions

Require: ST, R, E
  for $r \in R$ **do**
    $x \leftarrow Of(r)$
    if $x = Not$ **then**
      $r \leftarrow \neg r$
    end if
    if $x = Or$ **then**
      $Y = \{y : y = Of(x)$, where $y \neq r\}$
      $r \leftarrow \{r\} \cup Y$
      $R \leftarrow R \setminus Y$
    end if
    if $r = Literal$ **then**
      $constraintDataType = Of(r)$
      $r \leftarrow constraintDataType(r)$
    end if
    $P \leftarrow range(r)$
    for $p \in P$ **do**
      $restriction \leftarrow Of(p)$
      $e \leftarrow expression(restriction, p, r)$
      $E \leftarrow E \cup \{e\}$
      $R \leftarrow R \setminus \{r\}$
      $D \leftarrow domain(p)$
      if $D \neq \emptyset$ **then**
        Apply Algorithm 1 to ST, D, E
      end if
    end for
  end for
  $E \leftarrow E \cup R$

Figure 2: Elementary entities generated from Figure 1

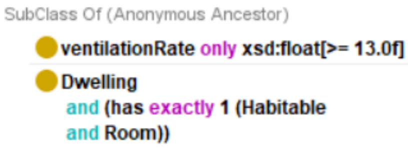

Figure 3: Final OWL representation generated for the annotation in Figure 1

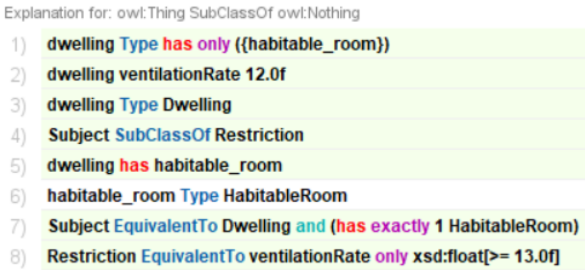

Figure 4: An example of building compliance checking using the generated OWL DL representation

Annotation. Figure 5 illustrates the DeLTA annotation of Example 2. In contrast to the annotation of the building requirement in Figure 1, this scientific claim involves the *Not* and *Some* tags. Consequently, only the *Or* tag remains uncovered by the examples, and it is discussed in the Discussion section.

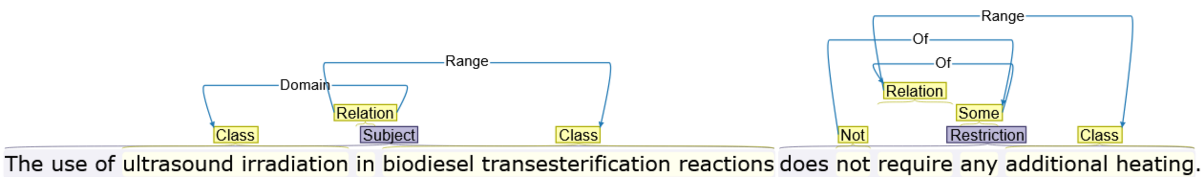

Figure 5: DeLTA annotation of a scientific claim

Preprocessing. Table 5 lists the tags of the Semantic Type layer, and Table 6 provides the Semantic Role tags parsed from the annotation in Figure 5.

Generating entities. Figure 6 illustrates the elementary OWL entities generated from the Semantic Type tags in Table 5 and the Semantic Role tags in Table 6. In this case, only classes and object properties are created, with no data properties.

Lexical mapping. Since the annotation includes no *Number* or *Comparison* tags, no lexical mappings are required.

Table 5
Semantic Types for the annotation in Figure 5

Index	Token	Tag	Domain	Range	Of
1	ultrasound irradiation	Class	-	-	-
2	in	Relation	1	3	-
3	biodiesel transester-ification reactions	Class	-	-	2
4	not	Not	-	-	5
5	require	Relation	-	7	7
6	any	Some	-	-	5
7	additional heating	Class	-	-	-

Table 6
Semantic Roles for the annotation in Figure 5

Index	Token	Tag	To
1	The use of ultra-sound irradiation in biodiesel trans-esterification re-actions	Subject	1
2	does not require any additional heating	Restriction	-

Constructing class definitions. Similarly to the previous example, the *Not* and *Some* tags, connected with *Of* arrows, are used in the recursive construction of definitions for the OWL classes corresponding to Semantic Roles. The following listing defines the *Subject*:

```
Subject EquivalentTo: UltrasoundIrradiation and (in some
            BiodieselTransesterificationReactions)
```

The listing below defines the *Restriction*:

```
Restriction EquivalentTo: not (require some AdditionalHeating)
```

It is noticeable that within the *Restriction* part, universal restrictions are typically used for requirements, whereas for scientific claims, existential restrictions are employed. Thanks to the expressive power of description logics, to which DeLTA corresponds, the same annotation and logical framework can be applied to both use cases. This would not be possible with less expressive formalisms, such as Horn clauses.

Constructing axioms. Figure 7 illustrates the final OWL representation of the claim from Example 2, resulted from connecting the *Subject* class to the *Restriction* class using `owl:subClassOf`.

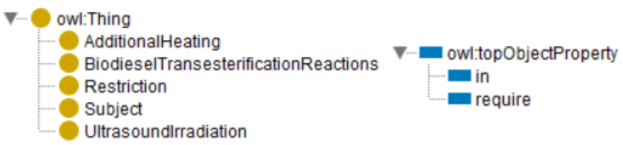

Figure 6: Elementary entities generated from Semantic Types in Table 5

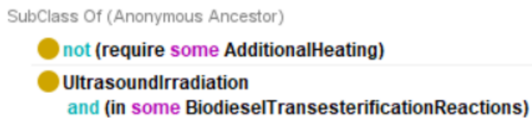

Figure 7: Final OWL representation generated for the annotation in Figure 5

This DL-based representation of the scientific claims enables, for example, computational consistency checking of scientific theories. If another paper were to claim that the use of ultrasound irradiation in biodiesel transesterification reactions *requires* additional heating, and this claim was formalized in the same way, automated reasoning would detect that the *Subject* is unsatisfiable. This occurs because it would belong simultaneously to the *Restriction* and its complement, as illustrated in the reasonin explanation in Figure 8.

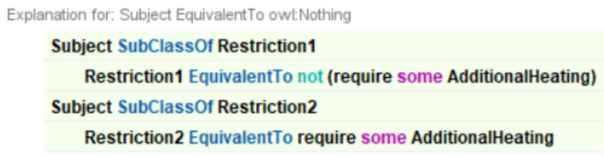

Figure 8: An example of scientific claims consistency checking with the generated OWL DL representations

4.3. Integration with Domain Ontologies

While this work emphasizes the domain independence of the approach, we also demonstrate how it can be integrated with existing domain ontologies. Such integration can support modeling decisions, for example, whether a concept should be represented as a Class, a Property, or decomposed into smaller concepts. For this purpose, an additional annotation layer of Terms may be introduced, which uses concepts from domain ontologies as annotation tags. During the translation of annotations into OWL, the Terms layer is processed separately during the preprocessing stage, similarly to the Semantic Types and Semantic Roles layers. At the stage of generating elementary OWL entities, the classes corresponding to the annotated Terms are imported from the respective domain ontologies. Relationships between these imported classes and the entities generated from the Semantic Types layer are then established via `owl:subClassOf`, based on token substring inclusion. Additionally, if an annotated relation in the domain ontology is functional or transitive, the generated OWL code will inherit these characteristics.

For example, in the building construction domain, Building Information Models (BIM) are commonly represented using the Industry Foundation Classes (IFC)[6]. Their OWL serialization, ifcOWL [34], can be used as a source of annotation tags in the Terms layer. An illustration is provided in Figure 9, where the annotation shown in Figure 1 is extended with a layer of IFC terms (highlighted in green). In this example, the word "room" is annotated with the *IfcSpace* tag. However, if the IFC model included a class specifically representing habitable rooms, the entire phrase "habitable room" could be annotated with that class instead. Since the IFC term is textually embedded within the span "habitable room", which is annotated as *Class*, the following axiom is asserted:

```
habitableRoom owl:subClassOf IfcSpace
```

Such connections are essential for aligning the formalized axioms with domain data, such as BIM models. The annotation of the Terms layer can be supported by NLP techniques, including named entity recognition, entity linking, and relation extraction, as well as ontology learning systems such as Text2Onto.

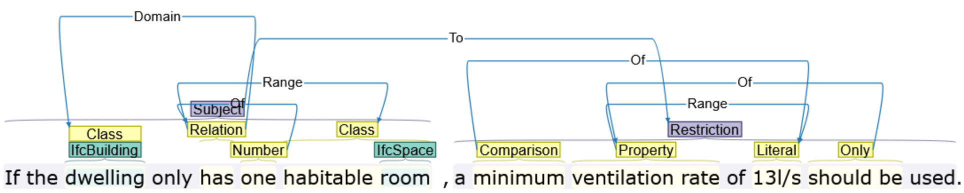

Figure 9: Annotation of Example 1 with an additional layer of IFC classes

5. Discussion

In this section, we discuss the design decisions behind DeLTA, as well as the advantages of the proposed approach, its limitations, and plans for future work.

[6]https://technical.buildingsmart.org/standards/ifc

5.1. Design Trade-offs

Handling disjunction in DeLTA. We found it difficult to illustrate the use of the *Or* tag in DeLTA. In practice, the *Or* tag is often unnecessary because, when a disjunction appears in a *Subject*, the disjuncts can instead be annotated as separate subjects, resulting, after translation into OWL, in separate classes without loss of semantic meaning. The *Or* tag is only required for proper semantic modeling when a disjunction appears in the *Restriction* part, which is rarely the case.

Transparency and cognitive load. DeLTA is designed to make the translation of annotations into DL algorithmic and transparent, thereby justifying its complex annotation schema. However, we consider the use of lexical mappings for several DL operations, such as cardinality restrictions and datatype constraints, to be a reasonable compromise that reduces cognitive load for annotators without sacrificing the scalability of the approach. Additionally, cognitive load is further reduced by dividing the list of categories into separate annotation layers. It could be reduced further through the use of interactive user interfaces, for example, by hiding individual layers, tags, or arrows. However, such human–computer interaction aspects are beyond the scope of the present work.

Schema extensibility. The annotation schema presented in this paper was driven by the use cases we considered. It does not include all existing OWL DL constructs, but it is sufficient to represent building requirements and scientific claims. However, the proposed approach is scalable, and the annotation schema can be extended when higher expressivity is required. This can be achieved by adding tags to the Semantic Type layer for additional OWL entity constructs, such as `owl:oneOf` for nominals, or by adding arrows to the Semantic Role layer for other axiom types, such as *owl:equivalentTo* for modeling definitions, or `rdf:type` for membership assertions.

5.2. Advantages of the Approach

Text annotation facilitates knowledge capture. Domain experts are required for knowledge capture, however, they cannot additionally be experts in computer science and knowledge representation. We did not conduct a user study to evaluate how DeLTA annotation supports formalization from a user-experience perspective. However, prior annotation schemas in domains such as law, construction, and science indicate that annotation-driven knowledge capture is desirable. This is because the annotation visually demonstrates how natural-language text is decomposed into formal semantic components for computational interpretation.

Advances in expressivity and generalizability. Our approach advances the state of the art in knowledge capture in two directions: increased expressive power of the target logical framework and domain-independence of the annotation schema. While existing annotation-based approaches can generate SHACL, SWRL or Prolog expressions, DeLTA, by contrast, supports expressive axioms at the OWL DL level, thereby enabling advanced computational reasoning and broad applicability across diverse use cases. Having a domain-independent annotation schema allows the reuse of the same methodology, infrastructure and software across different domains.

AI support for annotation and human-in-the-loop. Once a sufficient number of annotations have been completed, the annotation process can be facilitated using NLP, machine learning and large language models (LLMs). However, computational support for such use cases as building compliance checking or scientific claim verification can be conducted only in a transparent, trustworthy and controllable manner and thus cannot be passed entirely to LLMs, since they are not deterministic. Our annotation-based approach places the human-in-the-loop at the most effective point, since annotations can be created, reviewed and refined by domain experts, prior to the repeated use of the rules.

5.3. Limitations

Lexical mapping. The proposed approach relies on lexical mappings to generate cardinality restrictions and restricted data types. Any new languages, phrasings or typos in the text require manual processing. However, in future work, this mapping could be facilitated using NLP tools that enable fuzzy mapping.

Implicit semantics of natural language. Not all semantics are explicitly present in natural language and, therefore, can sometimes be difficult to annotate. This is particularly true for role restrictions, disjunctions and conjunctions. Our approach handles such cases by assigning default values, but this can potentially lead to inaccuracies in formalization. These inaccuracies must be addressed through predefined translation configurations or human intervention.

6. Conclusion and Future Work

In this work, we proposed DeLTA — a domain-independent annotation schema that enabled the algorithmic translation of annotations into description logic and their realization in OWL DL. We demonstrated the domain independence of the approach by applying it to two use cases: building requirements and scientific claims. Additionally, we discussed the trade-offs in the DeLTA design, how the approach could be integrated with domain ontologies, how it facilitated knowledge capture, and its limitations. The potential directions for future work are as follows:

AI support for annotation. We will create corpora of annotated text for various use cases and apply and evaluate various NLP algorithms and ML models to automate the annotation process. These algorithms and models can be integrated into the INCEpTION annotation environment as tag recommenders, preserving the human-in-the-loop approach.

Expanding the list of use cases. In this work, we demonstrated the application of DeLTA to two use cases: building requirements and scientific claims. We aim to expand the list of use cases for our approach, for example, the legal domain is a promising option. However, collaboration with domain experts in the corresponding areas will be required.

Deepening the existing use cases. We are also interested in deepening the demonstrated use cases by applying the formalized rules in various practical decision-making tasks, such as real-world building compliance checking and consistency checking of scientific claims.

Acknowledgments

We would like to acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – EXC 2163/1 - Sustainable and Energy Efficient Aviation – Project-ID 390881007. This study was also supported by the German Federal Ministry of Education and Research (BMBF) through the projects InSuKa (13XP5196F) and DIPONI (13XP5227E).

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] J. Cullen, A. Bryman, The knowledge acquisition bottleneck: time for reassessment?, Expert Systems 5 (1988) 216–225.
[2] T. D. Wang, B. Parsia, J. Hendler, A survey of the web ontology landscape, in: International Semantic Web Conference, Springer, 2006, pp. 682–694.
[3] S. Ghimire, F. Lévy, S. Hölldobler, Reasoning with text annotations (2011).
[4] S. Peroni, A. Gangemi, F. Vitali, Dealing with markup semantics, in: Proceedings of the 7th International Conference on Semantic Systems, 2011, pp. 111–118.
[5] S. Bird, M. Liberman, A formal framework for linguistic annotation, Speech communication 33 (2001) 23–60.
[6] A. Ozaki, Learning description logic ontologies: Five approaches. where do they stand?, KI-Künstliche Intelligenz 34 (2020) 317–327.
[7] P. Cimiano, J. Völker, Text2onto: A framework for ontology learning and data-driven change discovery, in: International conference on application of natural language to information systems, Springer, 2005, pp. 227–238.
[8] C. Welty, J. W. Murdock, Towards knowledge acquisition from information extraction, in: International Semantic Web Conference, Springer, 2006, pp. 709–722.
[9] J. Völker, P. Hitzler, P. Cimiano, Acquisition of owl dl axioms from lexical resources, in: European Semantic Web Conference, Springer, 2007, pp. 670–685.
[10] T. Wächter, G. Fabian, M. Schroeder, Dog4dag: semi-automated ontology generation in obo-edit and protégé, in: Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences, 2011, pp. 119–120.
[11] D. Sánchez, A. Moreno, L. Del Vasto-Terrientes, Learning relation axioms from text: An automatic web-based approach, Expert Systems with Applications 39 (2012) 5792–5805.
[12] R. R. De Azevedo, F. Freitas, R. G. Rocha, J. A. A. De Menezes, C. M. D. O. Rodrigues, G. D. F. e Silva, An approach for learning and construction of expressive ontology from text in natural language, in: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), volume 1, IEEE, 2014, pp. 149–156.
[13] A. B. Rios-Alvarado, I. Lopez-Arevalo, E. Tello-Leal, V. J. Sosa-Sosa, An approach for learning expressive ontologies in medical domain, Journal of medical systems 39 (2015) 1–15.
[14] B. Gyawali, A. Shimorina, C. Gardent, S. Cruz-Lara, M. Mahfoudh, Mapping natural language to description logic, in: The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14, Springer, 2017, pp. 273–288.
[15] G. Petrucci, C. Ghidini, M. Rospocher, Ontology learning in the deep, in: European knowledge acquisition workshop, Springer, 2016, pp. 480–495.
[16] G. Petrucci, M. Rospocher, C. Ghidini, Expressive ontology learning as neural machine translation, Journal of Web Semantics 52 (2018) 66–82.
[17] G. Rossiello, M. F. M. Chowdhury, N. Mihindukulasooriya, O. Cornec, A. M. Gliozzo, Knowgl: Knowledge generation and linking from text, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 16476–16478.
[18] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.
[19] T. H. Beach, Y. Rezgui, H. Li, T. Kasim, A rule-based semantic approach for automated regulatory compliance in the construction sector, Expert Systems with Applications 42 (2015) 5219–5231.
[20] S. Li, J. Wang, Z. Xu, Automated compliance checking for bim models based on chinese-nlp and knowledge graph: an integrative conceptual framework, Engineering, Construction and Architectural Management 32 (2025) 3832–3856.

[21] A. Nazarenko, F. Levy, A. Wyner, An annotation language for semantic search of legal sources, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[22] T. Libal, Legal linguistic templates and the tension between legal knowledge representation and reasoning, Frontiers in Artificial Intelligence 6 (2023) 1136263.

[23] M. Liakata, S. Teufel, A. Siddharthan, C. Batchelor, Corpora for the conceptualisation and zoning of scientific papers, in: N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, D. Tapias (Eds.), Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), European Language Resources Association (ELRA), Valletta, Malta, 2010. URL: https://aclanthology.org/L10-1440/.

[24] B. Sateli, R. Witte, Semantic representation of scientific literature: bringing claims, contributions and named entities onto the linked open data cloud, PeerJ Computer Science 1 (2015) e37.

[25] T. Groza, S. Handschuh, K. Möller, S. Decker, SALT - Semantically annotated LaTeX for Scientific Publications, in: European Semantic Web Conference, Springer, 2007, pp. 518–532. doi:10.1007/978-3-540-72667-8_37.

[26] L. Moreau, P. Groth, Provenance of Publications: A PROV Style for LaTeX, in: Proceedings of the 7th USENIX Conference on Theory and Practice of Provenance, USENIX Association, 2015, p. 2. doi:10.5555/2814579.2814581.

[27] M. Kohlhase, Using LaTeX as a Semantic Markup Format, Mathematics in Computer Science 2 (2008) 279–304. doi:10.1007/s11786-008-0055-5.

[28] L. Martin, A. Henrich, RDFtex In-Depth: Knowledge Exchange between LATEX-Based Research Publications and Scientific Knowledge Graphs, Int. J. Digit. Libr. 25 (2023) 517–535. doi:10.1007/s00799-023-00370-5.

[29] C. Bless, I. Baimuratov, O. Karras, Scikgtex-a latex package to semantically annotate contributions in scientific publications, in: 2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL), IEEE, 2023, pp. 155–164.

[30] J.-C. Klie, M. Bugert, B. Boullosa, R. E. de Castilho, I. Gurevych, The inception platform: Machine-assisted and knowledge-oriented interactive annotation, in: Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, Association for Computational Linguistics, 2018, pp. 5–9. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

[31] J.-B. Lamy, Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies, Artificial Intelligence in Medicine 80 (2017) 11–28. doi:https://doi.org/10.1016/j.artmed.2017.07.002.

[32] H. Hettiarachchi, A. Dridi, M. M. Gaber, P. Parsafard, N. Bocaneala, K. Breitenfelder, G. Costa, M. Hedblom, M. Juganaru-Mathieu, T. Mecharnia, et al., Code-accord: A corpus of building regulatory data for rule generation towards automatic compliance checking, Scientific data 12 (2025) 170.

[33] C.-I. Bucur, T. Kuhn, D. Ceolin, J. Van Ossenbruggen, Expressing high-level scientific claims with formal semantics, in: Proceedings of the 11th Knowledge Capture Conference, 2021, pp. 233–240.

[34] P. Pauwels, W. Terkaj, Express to owl for construction industry: Towards a recommendable and usable ifcowl ontology, Automation in Construction 63 (2016) 100–133. doi:https://doi.org/10.1016/j.autcon.2015.12.003.