IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No 3., June, 2024, pp. 1125-1139
Journal homepage: http://iieta.org/journals/isi

A Review of Semantic Annotation in the Context of the Linked Open Data Cloud

Khalid Jawad Kadhim*, Asaad Sabah Hadi

College of Information Technology, University of Babylon, Babylon 51002, Iraq

Corresponding Author Email: khalidj.sw.phd@student.uobabylon.edu.iq

https://doi.org/10.18280/isi.290330

Received: 5 August 2023
Revised: 22 January 2024
Accepted: 2 February 2024
Available online: 20 June 2024

Keywords:
semantic annotation, Linked Open Data (LOD), Semantic Web, metadata, Resource Description Framework (RDF), SPARQL Protocol and RDF Query Language (SPARQL), literature review

ABSTRACT

Semantic annotation, a pivotal technology facilitating comprehension within textual data, serves as the process of appending supplementary information or metadata to text, thereby augmenting its meaning. Given the inherent ambiguity of natural language, which renders the data susceptible to multiple interpretations, the task of discerning the intended meaning from raw data proves significantly more challenging than interpreting structured text. This ambiguity necessitates mechanisms to render text comprehensible to both machines and humans, thereby enabling the efficient extraction and innovation of various subjects. In response to these challenges, considerable research efforts have been dedicated to advancing the methodologies of text annotation. This review explores the role of semantic annotation in addressing contextual ambiguity, underspecified semantic representations, formal semantics, and the resolution of semantic ambiguities. By integrating additional data into the text, semantic annotation establishes a synergy with the Linked Open Data (LOD) framework, thereby providing context and enhancing machine readability. LOD, a practice of publishing structured data on the web to facilitate interlinking and utility, benefits from semantic annotation as it improves data publishing, linking, and enrichment processes. This enhancement directly contributes to the precision of web search results. The literature on semantic annotation, encompassing tools, methods, and techniques, as well as its relationship with LOD, is meticulously reviewed. This paper employs a systematic approach to select pertinent articles, highlighting state-of-the-art methods in semantic annotation, including deep learning and ontology-based techniques. The exploration aims to delineate the evolution of semantic annotation practices and their consequential impact on the LOD ecosystem, underscoring the mutual enrichment of both fields.

1. INTRODUCTION

Semantic annotation is one of the modern technologies that is beneficial to people because it helps them understand various things in any text [1]. Semantic annotation is also considered the process of adding additional information or metadata to the text to enhance its meaning. The huge input data is, of course, ambiguous, so anyone requires to understand raw data, which is quite difficult, rather than formal text [2]. It is also significant to make the text have the ability to be machine- and human-readable so that the operation of extracting and innovating on several subjects will stimulate researchers to do a huge number of studies to develop the process of text annotation. This paper is a review of what is done in the literature on semantic annotation, consisting of the old research, taking into account the tools, methods, and techniques about semantic annotation and their relationship with the LOD.

Semantic annotation is considered a very significant contemporary technology that is used in text mining to give the selected text clarity and understanding rather than an ambiguous form [3]. Semantic annotation can be defined as a comment that is added to the text, image, or diagram to enrich the target information, which can be formed as a text description, underlines, highlights, images, links, and so on [4]. In software programming, it means text comments embedded in the code that are ignored while running. We have to mention how to create annotations on the text in four important stages: Pre-processing is the initial step in the annotating process. Pre-processing entails cleaning and formatting the data in a way that is appropriate for the annotation task in order to get it ready for annotating. Before the data can be annotated, this may entail eliminating undesirable characters or symbols, translating the data to a particular format, and locating any problems or errors. Pre-processing guarantees that the data is prepared for annotation and that the annotation process will go well, making it a crucial step. Annotation guidelines creation is the second stage of annotation [5]. The rules or standards that describe how the data should be annotated are known as annotation guidelines. The categories or labels that will be used for annotation, the conventions to follow, and any guidelines or standards that must be followed during the annotation process are all outlined in these documents. Annotation rules are important because they guarantee that the annotation process is consistent, that the annotated data is of good quality, and that it can be used successfully for the objectives for which it was annotated. Annotation itself is the third step in the annotation process. Applying the annotation standards to the data in order to categorize or label it with the desired categories or labels is known as annotation. This could entail identifying and categorizing identified entities, including persons, locations, and organizations, or giving the text a sentiment or emotion. Human annotators can manually annotate documents by hand, or algorithms of machine learning and other technologies of Natural Language Processing (NLP) have the ability to automatically annotate documents. Although manual annotation is frequently regarded as the best practice, it can be expensive and time-consuming. Although it may be quicker and more effective, automatic annotation might not be as precise or dependable as manual annotation. Quality control is the fourth and last step in the annotating process [6]. For the purpose of ensuring that the annotated data is precise, reliable, and of the highest caliber, quality control entails reviewing and assessing it. This may entail error analysis, which involves identifying and fixing flaws in the annotated data, or inter-annotator agreement studies, which involve many annotators annotating the same data to evaluate the reliability of the annotations. Quality control is important since it makes sure the annotated data is trustworthy and useful for the intended uses. Document annotation is a process that may be manual, semi-automatic, or fully automatic, so manual annotation gives the user the ability to add annotations to the website and share them with others. Figure 1 illustrates the annotation process on the raw text as an example. Figure 2 shows the representation of annotation and semantic annotation.

In the Semantic Web, the data are collected in a huge manner, so that in annotation process the text must be segmented in order to be easily annotated and then to be understood, Figure 3 illustrate this operation.

The raw data have an ambiguity concept in which it is difficult to distinguish what the meaning of the text is by human or machine; for example, a specific word may have different meanings, so the annotation process gives the disambiguation for the text. There are three types of annotations: the first is formal, which is considered machine-readable; the second is informal, which is human-readable; and the third is ontological, which is both human- and machine-readable. The formal annotation is very close to being a semantic annotation [7].

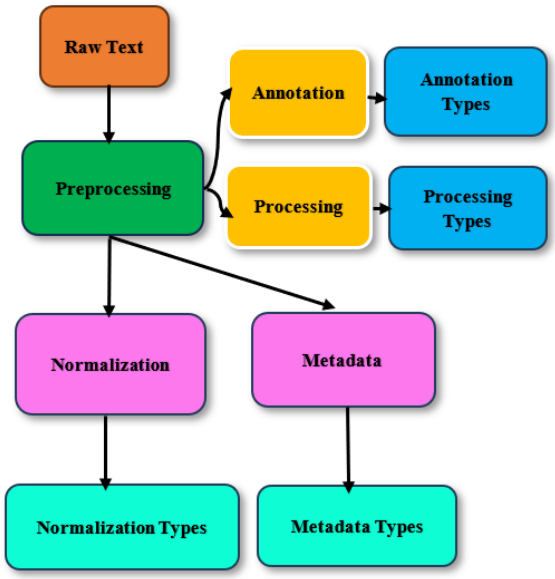


Figure 1. Annotation process

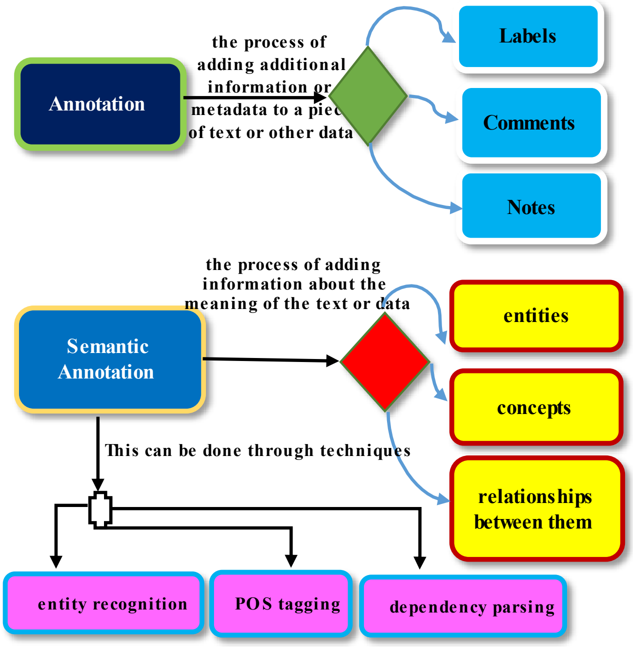

Annotation
the process of adding additional information or metadata to a piec of text or other data
Labels
Comments
Notes
Semantic Annotation
the process of adding information about the meaning of the text or data
entities
concepts
relationships between them
This can be done through techniques
entity recognition
POS tagging
dependency parsing

Figure 2. Annotation and semantic annotation representation

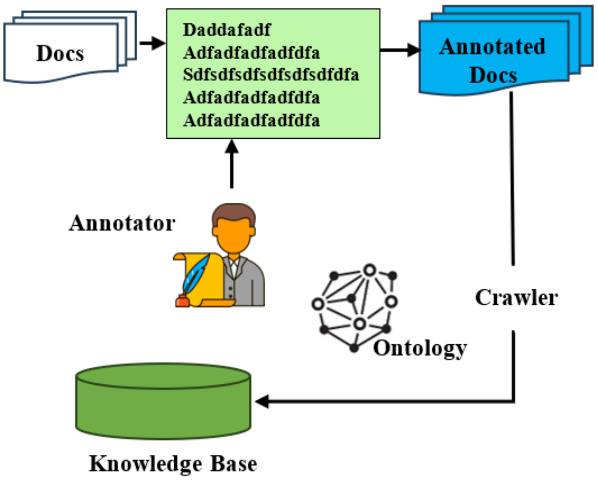


Figure 3. document annotation with semantic data

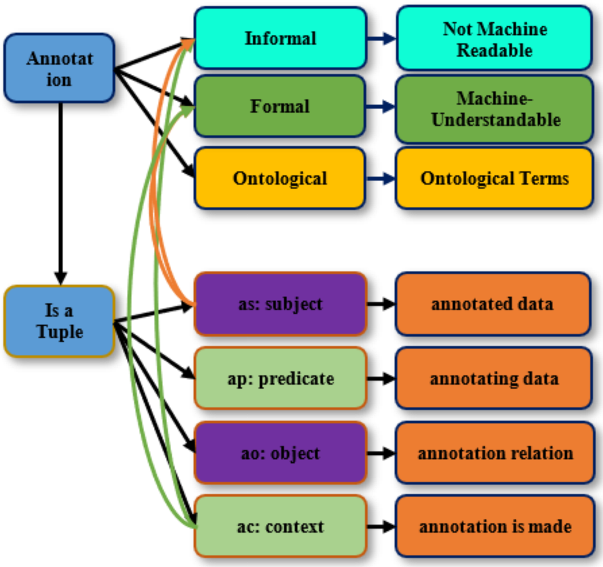


Figure 4. Semantic annotation architecture

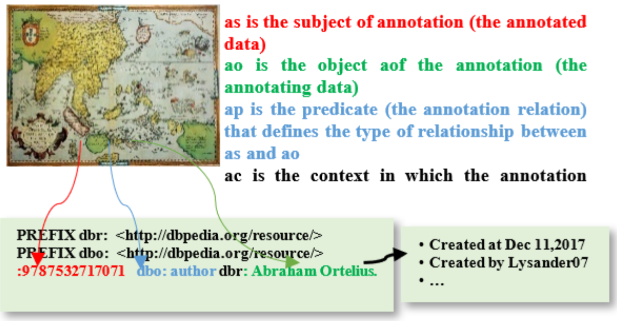

as is the subject of annotation (the annotated data)
ao is the object aof the annotation (the annotating data)
ap is the predicate (the annotation relation) that defines the type of relationship between as and ao
ac is the context in which the annotation
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/resource/>
:9787532717071 dbo: author dbr: Abraham Ortelius.
• Created at Dec 11,2017
• Created by Lysander07
• …

Figure 5. Semantic annotation

Semantic annotation is the process of adding information about the meaning of the text or data. Semantic annotation is defined as the process of adding additional linguistic information to the available linguistic forms to make them more descriptive [8]. Semantic annotation could also be defined as the process of adding semantic information to linguistic functions. Suppose we have two sets of objects, documents, and formal representations. In this way, two functions can also be defined. The first function that maps the documents into formal representations is called annotation, and the second function that maps formal representations to documents is called indexing. In semantic annotation, there are three important aspects that must be identified: entities, concepts, and relationships between them. Figure 4 shows the semantic annotation architecture.

Semantic annotation is a tuple that has (a_s (Subject = the annotated data), a_p (predicate = the annotation relation), a_o (object = is the annotation relation), and a_c (context = the relationship between the a_s and a_o in which the annotation was made). Figure 5 illustrates the semantic annotation.

In general, semantic annotation has a body and target; the body is the text that is required to be annotated from the whole text. The target includes the source text and selector, and the target of a specific resource has a URI (Unified Resource Identifier). The selector has three parts: the first is the text positioner selector, the second is the start position, and the third is the end position of the body, as shown in Figure 6.

The technique of tagging documents with pertinent concepts and adding information to them that connects the content to ideas shown in a knowledge network is known as semantic annotation. This facilitates the discovery, comprehension, and reuse of unstructured content. Here are some significant details about its significance [9, 10].

Better information retrieval, Documents with semantic tags are simpler to locate, comprehend, integrate, and reuse thanks to semantic annotation, which enhances information retrieval. Smarter knowledge management, by transforming material into a more manageable data source, contributes to improved knowledge management. Enables Semantic Search and Content Aggregation: among the most popular uses made possible by semantic annotation are automated relationship discovery, information aggregation, and semantic search. Facilitates Automated Operations: computers can carry out tasks like categorization, linking, inferencing, searching, filtering, and so on with the help of semantic annotations. Enhances Interoperability and reusability—the integration of data required for model parameterization and validation, as well as the interoperability and reusability of models—are all improved by semantic annotation. Important in Computer Vision and Information Retrieval: In computer vision and information retrieval, semantic annotation is crucial. It offers comprehensive semantic details regarding unseen pictures, including the semantic kind and related visual relationships. The objective of this review is to find out the latest techniques and tools used in semantic annotation and to discover what the researchers do. The primary goal of this review is to answer the questions and get a clearer understanding of semantic annotation. Improving knowledge management by turning the knowledge into smarter and more manageable data is one of the impacts of semantic annotation on text data [11]. A better representation of ambiguity while dealing with noisy and inaccurate parsers, as well as semantic annotation, increases the amount of extracted data and provides a better representation of ambiguity.

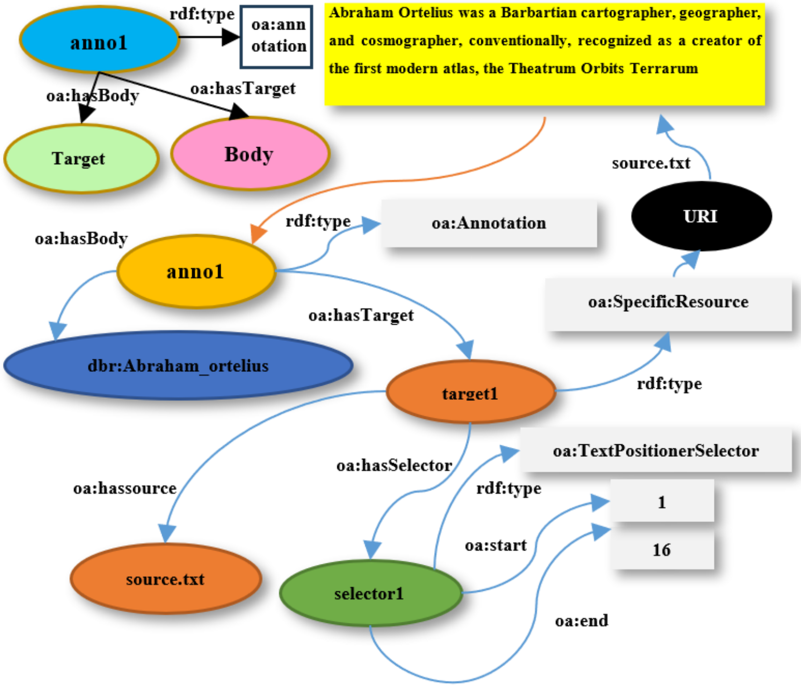


Figure 6. Semantic annotation components

The further impact of semantic annotation is improving discovery and interoperability, which promotes reusability and reproducibility. One of the most significant semantic annotation approaches, which is considered the state of the art, is Named Entity Recognition (NER). This technique aims to classify and identify named entities in the input corpus, such as persons, organizations, and locations. Deep learning methods like Bidirectional Long Short-Term Memory (BILSTM) and the Bidirectional Encoder Representation from Transformers (BERT) model have achieved important improvements in NER tasks in terms of capturing contextual information. This research focuses on a specific semantic annotation technique in which we will use deep learning for word embedding and NER and map them with LOD for evaluation purposes.

2. PROBLEM STATEMENT

Most research mentions that the input data for Semantic Web and ontology-based systems is huge; some of these data are structured and some are unstructured. On this occasion, there is difficulty navigating through the input data when these inputs are text, document, or any other format [12]. There are some studies that mention that the large amount of data suffers from data ambiguity, which means each word has multiple meanings. This makes it difficult for machines to understand the intended meaning of the text and can lead to errors or inaccuracies in natural language understanding and information extraction tasks. For this reason, some studies take into account the contextual meaning of the terms to help computers understand and analyze the text more effectively [13].

Another issue with semantic annotation is the high cost of manual annotation, which can be time-consuming and resource-intensive. Automated annotation can help to reduce these costs, but there are still challenges associated with achieving high levels of accuracy and consistency with automated annotation, particularly when dealing with complex and diverse data sources [14].

Some researchers mentioned that the data suffers from heterogeneity, such as data from different sources having different formats, ontologies, and vocabularies [15, 16] mentioned that the semantic annotation process requires high-quality data to ensure accurate and consistent annotation, and the quality of the data can be impacted by factors such as errors, omissions, and inconsistencies [17, 18] illustrates that semantic annotation requires regular maintenance and updating to ensure that the annotation remains accurate and relevant over time, especially as the data evolves.

From all above, the goal of this research is to smooth the navigation through the text by using the semantic annotation tools to reduce errors and ambiguity in the input text. For this reason, the dataset should be chosen carefully to ensure the quality, consistency, and homogeneity of the data, and finally to reduce the time consumed while using the manual annotation. These gaps can be resolved by using deep neural networks and machine learning methods.

3. LOD

LOD is one of the Semantic Web's (it also has the familiar name Web of Data) basic foundations. The Semantic Web is all about connecting datasets in ways that humans and machines can comprehend. The recommended practices for establishing these linkages are provided by Linked Data. Linked data, in other terms, is a combination of designing essentials for exchanging interconnected, machine-readable data on the Web [19].

Two sets of design guidelines for sharing machine-readable interrelated data on the Web are linked data and LOD. Machine-readable, semantic data that a machine can "understand" is referred to as linked data. Semantics come from links, and the majority of linked data development is concentrated on ontologies that give words meaning and tools that interpret data as microdata, RDFa, or RDF. LOD is created when Linked Data and Open Data (freely used and disseminated data) are joined. A type of LOD database is an RDF database, such as GraphDB from Ontotext. Large datasets from many sources may be handled by it, and it can connect them to Open Data, which facilitates effective data-driven analytics and knowledge discovery.

The term LOD refers to a collection of best practices for posting and connecting structured data on the internet. The data is open in the sense that it can be freely accessed, used, and reused by anyone for any purpose. LOD enables data to be integrated and joined from many sources, allowing for more meaningful and valuable information to be derived from it. The main technical components of LOD include the use of URIs to identify resources, the use of RDF to represent data in a graph-based format, and the use of standard RDF vocabularies to describe the relationships between resources. The use of standards such as SPARQL and RDF makes the data accessible and queried uniformly [20].

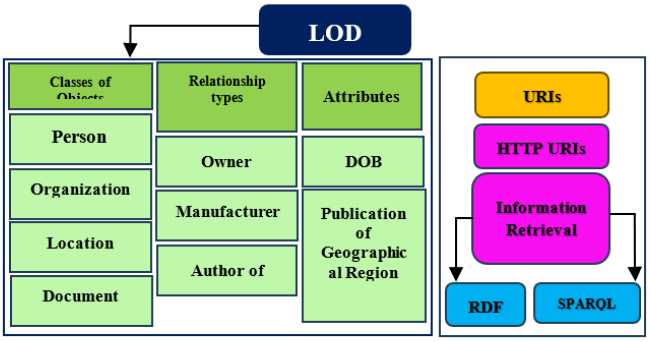


Figure 7. LOD

Figure 7 illustrates that the LOD includes classes of objects such as information about (person, organization, location, and document). While the relationship type includes, as an example, information about (owner, manufacturer, and author of the book, LODs have attributes such as date of birth, population of the geographic region, and so on. In conclusion, the information that has been available in LOD is very huge and varied.

Resource Description Framework, or RDF, is a paradigm that may be used to convey information about both concrete items and abstract ideas. It uses a graph structure to represent the relationships between things. Anything may be described with RDF, including people, pets, things, and ideas of all kinds. Statements with the following syntax are used to convey information: <subject>, <predicate>, <object>. The relationship between the subject and the object is expressed in these sentences. Resources include both the topic and the object [21]. RDF Query Language and Protocol, or SPARQL, is an RDF query language that may be used to obtain and modify data that is stored in RDF format23. Users can use it to query data from any data source that can be mapped to RDF, including databases. Your data is internally written as triples with subjects, predicates, and objects and is seen by SPARQL as a directed, labeled graph [22].

Unified Resource Identifiers, or URIs, in LOD Every resource on the Semantic Web has a URI. URIs are distinct identifiers used to refer to content that has been posted online. They are essential to the Findability, Accessibility, Interoperability, and Reusability (FAIR) of statistical data. By serving as particular "addresses," URIs facilitate the location of data. The URI allows a user to easily access the data after it has been located. URIs offer a common format that facilitates machine-to-machine communication by making it simple to combine data with other information [23]. LOD faces some challenges and limitations, such as a lack of distinction among building materials, a lack of clarity and specificity, and a lack of alignment with the level of information needed. Maintaining visual quality and optimizing efficiency are two factors that must be balanced while adopting LOD. To obtain the required outcomes in real-time rendering applications, overcoming the obstacles and limits calls for a mix of technical know-how, creative considerations, and meticulous optimization [24].

4. ONTOLOGY

Within the fields of philosophy and computer science, ontology pertains to the formalization of knowledge in a certain area or subject. It is a methodical framework for classifying and arranging the ideas, things, connections, and attributes that make up a given domain [25]. The formal definition of ontology is represented as a structure $O: = (C, \leq C, R, \leq R)$ consisting of (1) two disjoint sets C and R called the Concept identifier and the Relation identifier respectively; (2) a partial order $\leq C$ on C called concept hierarchy or taxonomy; (3) a function $\sigma: R \rightarrow C X C$ called signature; and (4) a partial order $\leq R$ on R called relation hierarchy. Figure 8 depicts an example of the ontology of foods.

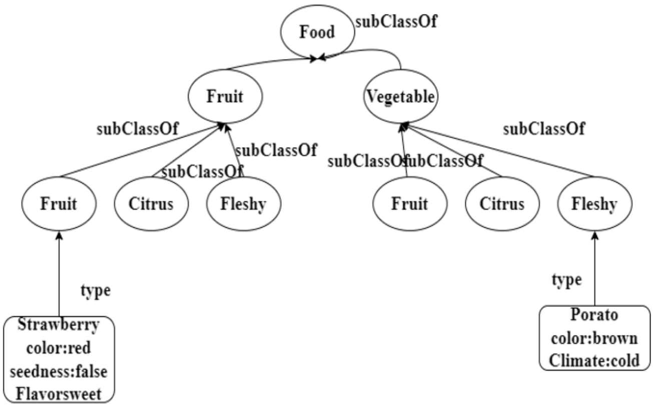


Figure 8. An Example Ontology of Food

Ontology, commonly referred to as knowledge management, is a model that depicts knowledge as a collection of concepts within a certain domain and records the connections between them. It compiles information inside an organization as a model, which is then queried by the user to address challenging questions and highlight connections throughout the organization [26]. People nowadays have to reach more data in one day than most people have in their entire lives in the past decades. The big issue is that these data exist in many different formats, and all of these data have been captured in many different forms, which makes it nearly sophisticated to be understood in terms of existing relationships between different data. In the current world, determining how policies are captured in written documents, how those policies link to the business processes documented in models, and how those business processes relate to the data stored in databases is extremely complex. Data must be formed in a way that permits these sorts of relationships to be discovered; ontology apprehends this data in such a manner that these relationships can be seen.

The RDF and Web Ontology Language (OWL) are the two standards that modulate the designation of ontologies. Ontologies, according to RDF and OWL, consist of two fundamental components: classes and relationships. Figure 9 illustrates how the class of person has a relationship with the class organization; the relationship type is Has Employer. The combination of these classes and the relationship is called the triple; this triple consists of subject, predicate, and object. All the components needed in any domain and their relationships are specified using an ontology. Since tags are meaningless on their own unless they are placed in some kind of context, it is futile to annotate the page with tags that are not connected to an ontology. By offering a formal framework for classifying, expressing, and managing information inside an organization or a particular topic, ontology plays a vital role in knowledge management. Ontology helps in organizing knowledge by providing a formal structure that defines relationships, entities, concepts, and properties in a specific domain. By representing knowledge in a consistent and standardized manner, ontologies enable efficient storage, retrieval, and navigation of information. Furthermore, ontology ontologies provide a shared lexicon and semantic structure to facilitate knowledge exchange and interoperability. They enable cooperation and communication between many people, groups, and systems by supplying a common understanding of ideas and connections. Information may be exchanged and combined more easily because of ontologies, which facilitate the integration of knowledge from diverse sources [27]. Ontologies have the ability to map and convert data across various forms and formats. Ontologies facilitate the conversion and alignment of data across multiple formats by specifying the connections and mappings between concepts in the ontology and the relevant components in other data sources. This facilitates data harmonization and standardization, which facilitates data analysis, querying, and interpretation.

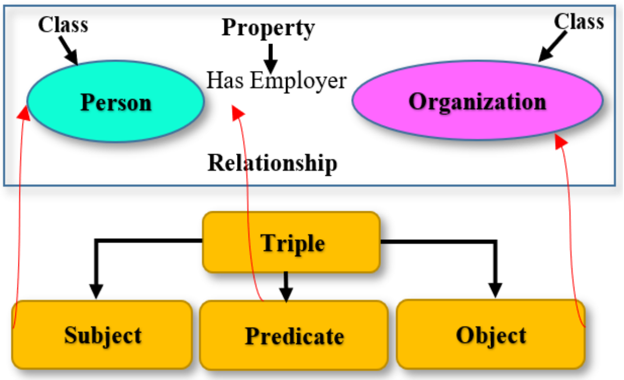


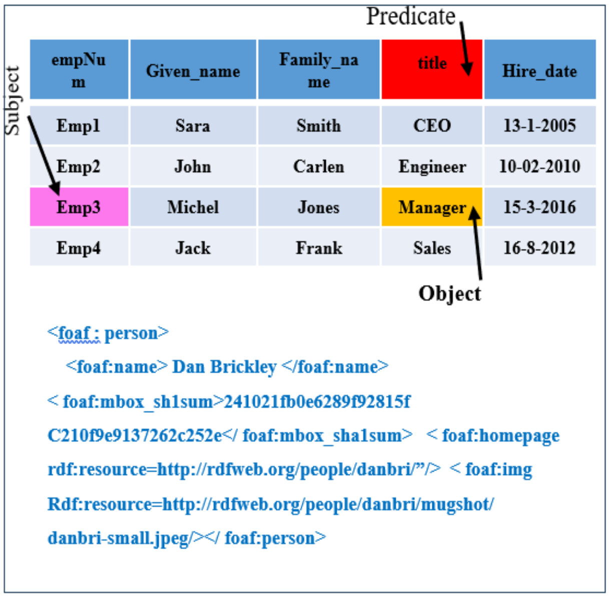
Figure 9. An Ontology representation

5. SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) is a standard query language for retrieving information or data from RDF databases. This query language is used for querying and manipulating the RDF data from the web and allows users to retrieve and modify the data that is saved in triple stores (RDF databases) in a similar way to how SQL is used with relational databases. SPARQL provides an expressive way to describe complex queries across multiple RDF graphs and has become a standard for querying linked data on the web [28]. Figure 10 illustrates how to pick each element of the query and write it in the formal format of SPARQL so that all elements at the end will be written in the same way. Here is a simple example of a SPARQL query that retrieves the names and ages of individuals who are described as instances of the FOAF: Person class in an RDF dataset The query in this example makes use of the predefined prefixes FOAF and RDF. The RDF namespace is denoted by the RDF prefix, whereas the Friend of a Friend (FOAF) vocabulary namespace is denoted by the FOAF prefix.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT?name ?age
WHERE {
    ?person rdf:type foaf:Person .
    ?person foaf:name ?name .
    ?person foaf:age ?age .
}
```

Table 1 shows a brief difference between SQL and SPARQL in terms of data model, querying capabilities, data representation, and scope and use cases.



empNum	Given_name	Family_name	title	Hire_date
Emp1	Sara	Smith	CEO	13-1-2005
Emp2	John	Carlen	Engineer	10-02-2010
Emp3	Michel	Jones	Manager	15-3-2016
Emp4	Jack	Frank	Sales	16-8-2012

```
<foaf : person>
   <foaf:name> Dan Brickley </foaf:name>
< foaf:mbox_sh1sum>241021fb0e6289f92815f
C210f9e9137262c252e</ foaf:mbox_sha1sum>  < foaf:homepage
rdf:resource=http://rdfweb.org/people/danbri/"/>  < foaf:img
Rdf:resource=http://rdfweb.org/people/danbri/mugshot/
danbri-small.jpeg/></ foaf:person>
```

Figure 10. SPARQL representation

Table 1. Difference between SQL and SPARQL

Purpose	SQL	SPARQL
Data Model	SQL works with structured data that has predetermined schemas and is intended for use with relational databases. Data is represented by tables with rows and columns, and functions like choosing, adding, updating, and removing data are supported.	Based on a graph model, SPARQL is intended for searching RDF (Resource Description Framework) data. Flexible data modeling is possible with RDF as it expresses data as triples (subject-predicate-object). To access data, SPARQL queries navigate the graph structure.
Querying Capabilities	A wide range of procedures are available in SQL for querying and working with structured data. It can do intricate operations including sorting, joining, aggregating, filtering, and grouping. Subqueries and views are additional elements of SQL that allow for more complex data processing.	Specifically created for RDF data queries, SPARQL has strong graph traversal and pattern matching features. In addition to supporting filtering, optional patterns, graph pattern composition, and other features, it enables querying based on subject-predicate-object patterns. Additionally, SPARQL has tools for combining and organizing results.
Data Representation	works with data that is tabulated and has rows and columns in relational database tables. Usually, the data is arranged into structured schemas that have predetermined data types and limitations.	RDF data, which is used by SPARQL, is information represented as triples (subject-predicate-object). RDF can describe complex connections and semantics and enables extendable and flexible data structuring.
Scope and Use Cases	SQL is widely used in traditional relational database management systems (RDBMS) and is suitable for applications that deal with structured and tabular data. It is commonly used in business applications, data analysis, reporting, and transaction processing.	RDF data, which is frequently utilized in knowledge graphs, ontology-based systems, linked data, Semantic Web applications, and linked data repositories, may be queried and altered using SPARQL. With the use of strong graph-based queries, SPARQL makes it possible to navigate intricate relationships and find new data.

6. RDF

RDF is defined as the criterion for modeling and interchanging information on the web. It is a way of describing resources and their relationships in a machine-readable format. RDF data is represented as a graph of triples, where every triple consists of a subject, a predicate, and an object. RDF is used in a variety of applications, such as the Semantic Web, knowledge management, and data integration. A question might be arising: why RDF? The answer to this question is incorporating machine-readable data into Web pages using the well-known schema.org language, allowing for better search engine display or automatic processing through third-party apps. utilizing external datasets to enhance one's own. For example, a dataset on paintings may be enhanced to provide users access to a wealth of information on the relevant artists and associated websites by linking it to the Wikidata entries for those artists. The connection of Application Programming Interface (API) feeds makes it simple for consumers to learn how to obtain more data. building data aggregations around certain subjects using the datasets that have already been made available as linked data. Constructing sporadic social networks by connecting RDF descriptions of individuals from many sources. provide a standard-compliant database data sharing method. Interconnecting disparate datasets within a specific organization to enable SPARQL cross-dataset searches [29].

The RDF data model is based on three disjoint pairs called RDF terms. The set that has been referred to as International Resource Identifiers (IRIs) is used for resource recognition. While the group set L refers to literals, it also refers to strings and datatypes that may be language-tagged. The group set B of what are called blank nodes has been interpreted as local existential variables. RDF is a triple $(s, p, o) \in IB \text{ X } I \text{ X } IBL$. The RDF graph consists of a combination of RDF terms, where each triple $\in G$ represents a directed labeled edge. The letters (s, p, o) refer to subject, predicate, and object. The graph has been denoted by s(G), P(G), and O(G) so that $(G)^{=}S(G) \cup O(G)$, the group set of nodes in G. Figure 11 shows the RDF graph as an example.

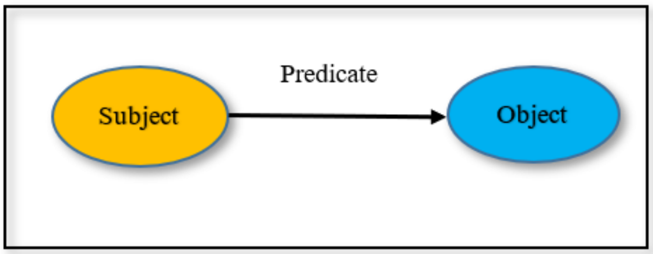


Figure11. RDF example

Vocabularies are used in the Semantic Web. The script illustrated in this section is a very essential document describing an individual person using RDF and OWL. This example explains the format of using FAOF vocabulary, which is very common in the Semantic Web [30].

```
<foaf: person>
   <foaf:name> Dan Brickley </foaf:name>
< foaf:mbox_sh1sum>241021fb0e6289f92815f
C210f9e9137262c252e</ foaf:mbox_sha1sum>
    < foaf: homepage
rdf:resource=http://rdfweb.org/people/danbri/”/>
    < foaf:img
rdf:resource=http://rdfweb.org/people/danbri/mugshot/
danbri-small.jpeg/>
   </ foaf:person>
```

6.1 Semantic annotation process

As we mentioned before, in semantic annotation, there are two main functions. The first function maps the documents into formal representation, called annotation; the second function is from formal representation to document, called indexing. The formal representation is modeled after ontologies because it consists of the description of the type of object and the concepts with their relationships and properties. On this occasion, the goal of semantic annotation in textual input is to identify the concepts with the support of the domain ontology because the ontology is domain-specific. Semantic annotation is the operation of adding metadata to text, images, videos, or other digital content to describe its meaning and context. This information helps search engines and other applications comprehend the content and provide more accurate search results or recommendations [31]. Semantic annotations can be created using various standards, such as RDF, Microdata, or JSON-LD, and can include information such as the author, date of publication, title, and relevant topics or entities. The goal of semantic annotation is to enable better organization, discovery, and analysis of digital content [32-35].

6.2 Semantic Web

It is a combination of criterion technologies to understand a web of data. The Semantic Web builds upon the traditional World Wide Web (WWW) by adding a layer of meaning to the published information on the web. This meaning is represented as metadata, or data about data, which describes the relationships and context of the information. The idea is that by making information more machine-readable, it can be easily processed, analyzed, and combined with other data sources to provide new insights and knowledge.

The Semantic Web uses technologies such as RDF, which is a flexible model of data that describes resources and their relationships, and OWL, which is a language for expressing more complex relationships and rules between concepts. Additionally, SPARQL, a query language that has been used to retrieve information from RDF data sources, is used to access and manipulate the data stored on the Semantic Web.

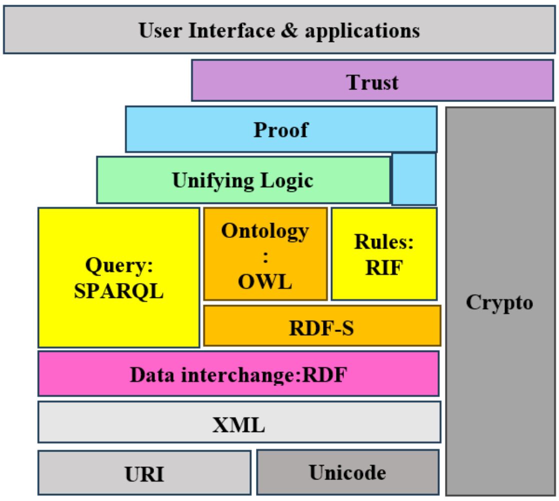


Figure 12. Semantic Web Layers

The Semantic Web has the potential to transform a wide range of applications, including search engines, e-commerce, and knowledge management systems, among others, by enabling machines to comprehend the meaning of material on the web. It is seen as a way to bring structure and meaning to the enormous amount of unstructured data available on the web, making it easier to find, analyze, and use [36].

Figure 12 illustrates what is so called the Semantic Web layers. In this figure, we can find out how the evolution of the Semantic Web starts with the URI and XML reaching the trust stage, which means the last stage depends on the previous stages, and the result will be trusted.

6.3 Deep Neural Network (DNN)

The role of DNN in semantic annotation is crucial; it is considered a powerful tool to predict and assign meaningful tags to the text. A lot of labeled data is frequently needed for NLP jobs in order to train machine learning models. Annotation gives us the labeled data we need to train models for many NLP tasks, such as sentiment analysis, named entity identification, machine translation, and part-of-speech tagging. The usefulness and performance of these models are directly impacted by the precision and quality of the annotations [37]. Supervised learning, dataset creation, and knowledge acquisition are also important roles. DNN has the ability to learn very complex patterns from tremendous datasets that allow it to capture the semantic meanings of words in hierarchical form. DNN consists of multiple layers designed to process data in hierarchical form; each layer learns from a higher level of input data and then builds a deep understanding of the low-level layers gradually. DNN is a class of ML models that are inspired by the human brain in terms of functions and structure. In NLP, deep neural networks, such as transformer models and recurrent neural networks (RNNs), have the ability to be trained on a huge number of textual datasets for the purpose of learning semantic representations of words or sentences [38]. These representations can then be used to automatically annotate text data with semantic information, such as NER to identify entities like people, organizations, or locations, or sentiment analysis to determine the sentiment of a piece of text [39]. DNN is capable of significantly improving the accuracy and efficiency of semantic annotation tasks by automatically learning complex patterns and representations from the data [40]. However, it is important to remember that deep neural networks are not infallible and may still have limitations, such as bias in their training data, which should be carefully considered and addressed in the annotation process to ensure accurate and unbiased results [41]. In the context of NLP and comprehension, deep neural networks (DNNs) and semantic annotation are connected. Labeling or marking text with extra semantic information, entities, connections, meanings, etc. in order to improve the text's comprehension is known as semantic annotation. Semantic annotation is one of the many NLP tasks in which DNNs, more precisely deep learning models, have found widespread use. DNNs are well-suited for jobs requiring the comprehension and extraction of meaning from text since they are adept at learning intricate patterns and representations from vast volumes of data [42].

7. METHODS OF ANNOTATION

Semantic annotation is the operation of assigning entities to the input text [43]. Semantic annotation is suitable for any type of text data, such as web pages, documents, social media posts, educational content, etc. The result of annotation is a document or web page with machine-interpretable markup to create annotations with semantics that are well defined.

7.1 Manual annotation

The key advantage of manual annotation is the ability of an expert human to understand the context and nuances of the data. A human annotator has enough knowledge and sense to interpret the input data. For instance, in NLP, human annotators can distinguish scoff, irony, or text ambiguity, which are often difficult for automated methods to understand. This contextual comprehension and domain expertise of the annotators contribute to the accuracy and quality of the annotated data, leading to better performance of machine learning models. Another advantage of manual annotation is its flexibility and adaptability to changing requirements, which ensure that the annotation process remains up-to-date and relevant. Furthermore, manual annotation plays an important role in utilizing complex or rare cases. In different domains, the data that requires annotation may not be suitable for standard patterns or may be outliers. For example, human annotators, who have the ability to think, can grip various cases effectively by designing or planning on their expertise and judgment to provide accurate annotations. Manual annotation could also detect potential biases in the input data and alleviate them, keeping fairness and ethical considerations in mind in the annotation process.

In spite of the advantages of manual annotation, it also has some challenges. First is the possibility of human bias. The human annotator might have inherent biases, aware or unaware, that can impact the annotations that they provide. Bias in the annotation process can lead to biased machine learning models, perpetuating unfairness and differentiation. However, this challenge can be alleviated through appropriate guidelines, training, and regular quality control checks to ensure the accuracy and consistency of annotations [44]. Real-world examples that have been employed are image and video annotation, text annotation, speech and audio annotation, medical annotation, social media and user-generated content, autonomous vehicles, and e-commerce and recommendation systems.

7.2 Semi-Automatic annotation

It is considered a bridge between the manual and automatic annotation gaps. It is a hybrid method that combines the strengths of manual and automatic methods, in other words, bridging the gap between human annotation expertise and the efficacy of the machine. The process begins with human annotation that provides the initial state of data labeling, and this process is considered a training set for the machine learning algorithm. The algorithms then use these training sets to annotate the remaining data automatically. The annotation process is iteratively repeated until we achieve the desired level of annotation accuracy. Semi-automatic annotation is efficient; the reason is that this process can rapidly annotate a large amount of data. This is beneficial in applications that require frequent updates of data annotation, such as customer sentiment analysis, social media monitoring, etc. The semi-automatic method can improve annotation accuracy. When combining human expertise with machine learning efficiency, semi-automatic annotation can achieve high accuracy, making it appropriate for applications that require high-quality annotations, such as legal document analysis and medical diagnosis [45].

Despite the advantages of semi-automatic annotation, it also has some challenges. One challenge is that this method sometimes includes errors, which can be propagated by the machine learning algorithms, leading to inaccurate automated annotations. Regularly checking the quality and feedback loops between human and machine learning algorithms is significant to characterize and validate errors in the annotation process. Automatic annotation is the revolution of data annotation in artificial intelligence (AI) and machine learning (ML). There are real-world examples of this method: active learning, pre-annotation or seed annotations, weak supervision, transfer learning, and crowdsourcing.

7.3 Automatic annotation

Table 2. The differences between the three methods of annotation

Method	Pros	Cons
Manual annotation	Accuracy and Quality: High-quality data annotations.	Time and Cost: Manual annotation may be costly and time-consuming.
	Flexibility	Inter-annotator Variability: Subjectivity and Bias
	Complex Labeling.	
	Iterative Improvement.	Scalability: It is difficult to scale manual annotation.
	Domain Expertise.	Training and Retention: It can be challenging to find and keep qualified annotators.
Semi-automatic annotation	Efficiency: faster annotation	Dependency on Automated Components.
	Cost Savings	Restricted Applicability.
	Consistency: reduce inter-annotator variability.	Requirement for Manual Review.
	Scalability: better suited to manage bigger datasets.	Technical Difficulties.
	Time savings	First Setup and Training.
Automatic annotation	Speed and Efficiency: handle large data rapidly.	Accuracy and Quality: sometimes inaccurate.
	Cost Savings.	Absence of Subjectivity and Context.
	Consistency: guarantees consistent labeling judgments.	Limited Adaptability.
	Scalability: very scalable and capable of handling datasets of any size.	Depend on Training Data.
	Minimization of Human Errors and Biases	Challenges in Managing Ambiguity

It is a very fast approach that uses machine learning algorithms to annotate the data without human intervention. Most recent research has also paid attention to this method due to its ability to annotate data by improving scalability, efficiency, and cost effectiveness [46]. Methods of automatic annotation can handle a huge dataset with ease, making them well-convenient for applications that involve big data, such as speech recognition, image recognition, and NLP. Automatic annotation also alleviates human bias, which can impact the quality and fairness of data annotation. Furthermore, automatic annotation depends on data-driven algorithms that are not affected by subjective biases. This can lead to more consistent and objective annotation, especially in sensitive areas of research such as finance, healthcare, and legal applications where the accuracy of annotated data is crucial. The limitation of using automatic annotation is that ML algorithms do not always accurately capture complex data, especially when the data are context-dependent, rare, and ambiguous. These errors can lead to inaccurate annotation, so regular validation, feedback loops, and quality checks are significant to ensure the accuracy and reliability of the annotation process. Another limitation is the lack of interpretability. ML algorithms, especially DNNs, can be mysterious, fuzzy, complex, and opaque, leading to difficulty in understanding and interpreting the input data [47]. The real examples deployed in this method are object detection, text classification, speech recognition, NER, image captioning, sentiment analysis, semantic segmentation, and document layout analysis. Table 2 illustrates the pros and cons and differences between the three methods of annotation manual, semiautomatic, and automatic.

Pre-trained models and transfer learning are a recent development in annotation that involves optimizing models trained on extensive datasets for particular tasks. In order to decrease the quantity of human annotation needed and increase annotation efficiency, future initiatives can incorporate pre-trained models and transfer learning strategies. Integrating AI with human expertise and bringing together the best aspects of both fields can provide annotation procedures that are more precise and effective in the future.

8. BACKGROUND

Albukhitan et al. [48] explored the functionality of word embedding usage from the algorithms of utilizing deep learning for semantic annotation of Arabic documents. Food, neutrinos, and health ontologies have been used to evaluate the performance of the proposed model. In this research paper, the author mentioned that it is not feasible to make semantic annotations on web documents manually due to the huge amount of web content. Since semantic annotation is considered the process of adding content that is machine-readable to the NLP in the format of RDF and ontology, the RDF is for information extraction, and the ontology is for the concepts of representation, relationships, and rules of semantics that are applied to the knowledge. The limitations of this research are that the semantic annotation has been done for Arabic, while most research papers have been done for Latin languages. By using deep learning, we have the ability to get word embedding by applying NLP models such as CBOW and skip gram. The semantic annotation framework of the research includes the stages of data acquisition, data preprocessing, word embedding using Word2Vec, instance and candidate matrix generators, and training neural networks for vector weight calculations.

The research paper by Campos-Rebelo et al. [49] refers to the fact that using semantic annotation creates interoperability between systems that are heterogeneous, so the researcher proposes a group of semantic annotation rules in the XML schemas. SAWSDL (Semantic Annotation Web Description Language): in this method, the semantic annotation is added to the XML schema definition XML Schemas (XSD) files and the metadata that describes the XML message exchange operation. Semantic annotation XML schema with SAWSDL: each element of XML schema (XSD) has the ability to be annotated with an ontology concept; for instance, the XML element" indoorTemp" has been annotated.

The XML element indoor is annotated with the concept indoor temperature for the ontology, as shown in Figure 13. The annotation path is capable of generating annotations that are more expressive, and it also has the ability to annotate the XML Schemas (XSD) elements with ontology concepts and properties, not only with concepts like Semantic Annotations for Web Service Description Language (SAWSD).

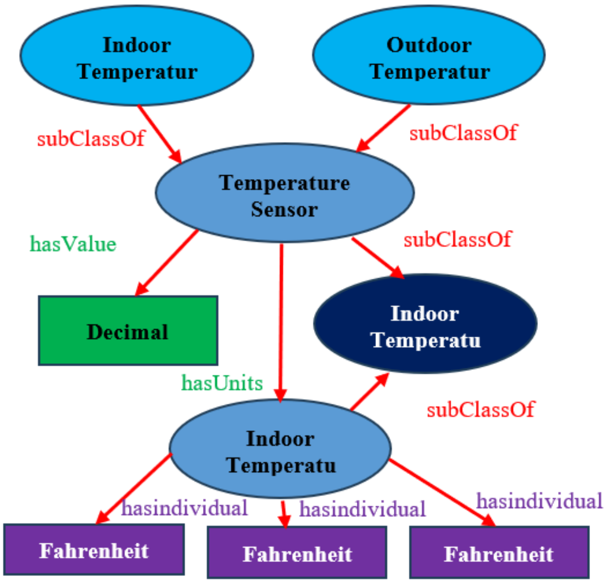


Figure 13. Example of ontology representation for temperature sensors

The XML schema definition XML Schemas (XSD) schema elements have been annotated with an annotation path, whereas every path is a series of steps, and every step is a property or concept of the ontology reference in the path of annotation. We have two classification paths in which they are: the odd and even steps, the even our property, while the odd are concepts. The object property is incapable of being the eventual step, and the datatype property is not the middle step. Conceptual steps may have some restrictions. The XSD element annotated in (1) uses Semantic Annotations for Web Service Description Language (SAWSDL), which has been annotated in (2) with an annotation path. In step (2), the first step (an even step) is considered a concept step, and the second step (an odd step and the final one) is considered a data type property. Another example of an annotation path is presented in (3). In this current example, the first and third steps are concepts, and the second is an object property.

<x s: element type="xs:float" name="indoorTemp"
sawsdl:modelReference="=IndoorTemperature"=> (1)

$$< xs : elementtype = "xs : \ float" name = "indoorTemp" \\ sawsdl : modelReference = " = IndoorTemperature = hasValue" => \quad (2)$$

<x s: element type="xs: string" name="unitsa" sawsdl: modelReference
=="Temperatu reSensor=hasUnits=TemperatureUnits"=> (3)

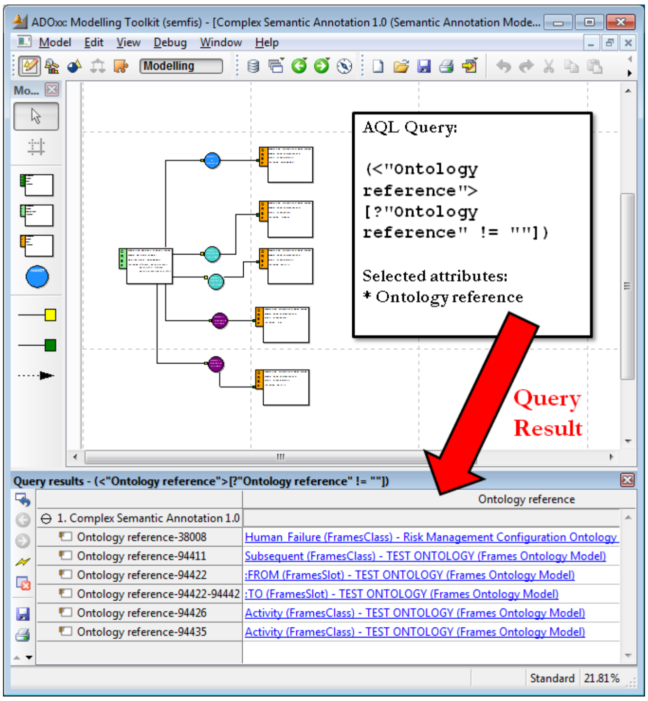
Here is an example of a message of XML that has been issued in Figure 14, and the associated XML schema XSD with semantic annotations, using paths of annotation, has been illustrated in Figure 8. In Figure 15 an example of an XML message has been illustrated.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:complexType>
 <xs:sequence>
  <xs:element type="xs: float" name="sensorTempl"
 sawsdl: modelReference="/TemperatureSensor/hasValue"/>
   <xs:element type="xs:string" name="unitssa"
        sawsdl: model Reference="/TemperatureSensor/
   hasUnits/TemperatureUnits" />
<xs:element type="xs: float" name="sensorTemp2"
     sawsdl: model Reference-"/TemperatureSensor/hasValue"/>
      <xs:element type-"xs:string" name="unitsb"
        sawsdl: model Reference="/Temperature Sensor/
      hasUnits/TemperatureUnits" />
```

Figure 14. XSD of the XML from Figure 5 with paths of annotation refer to the ontology

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
<sensorTemp1>72.3</sensorTempl>
<unitsa>F</unitsa>
<sensorTemp2>25.3</sensorTemp2>
<unitsb>C</unitsb>
</data>
```

Figure 15. Example of an XML message


ADOxx: Modelling Toolkit (semfis) - [Complex Semantic Annotation 1.0 (Semantic Annotation Mode...

AQL Query:

(<"Ontology reference">[?"Ontology reference" != ""])

Selected attributes:
* Ontology reference

Query results - (<"Ontology reference">[?"Ontology reference" != ""])

	Ontology reference
⊖ 1. Complex Semantic Annotation 1.0	
Ontology reference-38008	Human_Failure (FramesClass) - Risk Management Configuration Ontology
Ontology reference-94411	Subsequent (FramesClass) - TEST ONTOLOGY (Frames Ontology Model)
Ontology reference-94422	:FROM (FramesSlot) - TEST ONTOLOGY (Frames Ontology Model)
Ontology reference-94422-94442	:TO (FramesSlot) - TEST ONTOLOGY (Frames Ontology Model)
Ontology reference-94426	Activity (FramesClass) - TEST ONTOLOGY (Frames Ontology Model)
Ontology reference-94435	Activity (FramesClass) - TEST ONTOLOGY (Frames Ontology Model)

Figure 16. Using AQl in SeMFIS [50]

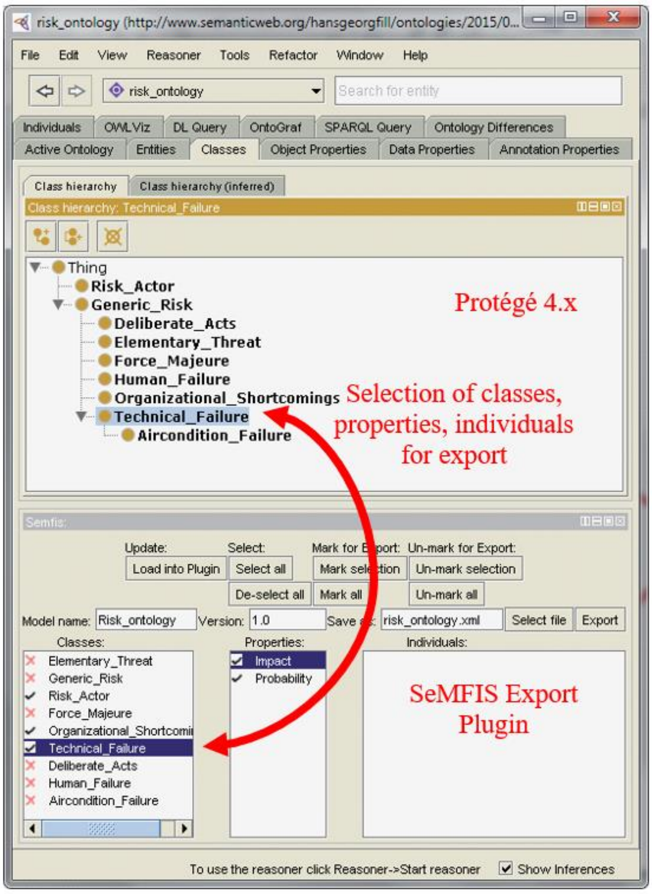
SeMFIS is considered a pliable engineering platform for conceptual semantic annotation models [50]. This platform provides a link between the two fields of ontologies and conceptual modeling; the benefit will be from both sides. One is that ontologies provide formal information to enrich conceptual models. Second: the visual editors and semiformal use of conceptual models to ease the interaction for non-technical users. SeMFIS can be appended to the existing models without affecting the available structures.

Queries in SeMFIS are expressed in AQL (ADOxx Query Language). This language is not as powerful as SQL, but it is similar in that it gathers information by providing query definitions targeting reference elements in semantic annotation and ontology. The output of AQL is either rtf, csv, or HTML.

It can be interpreted by the user for post processing. Figure 16 illustrates the example of using AQL in SeMFIS.

The Protégé ontology management tool has been used in SeMFIS because this plugin is widely used in different scopes of science. The protégé has been adopted in order to provide import and export interfaces for the reason of exchanging information of the model in different file formats.

The protégé plugin allows for loading properties, classes, and instances from ontologies (OWL) in the related environment, and then it can decide which one of the referred elements could be exported to the SeMFIS. Finally, the picked elements are stored as XML or RDF files that compatible with SeMFIS and then visualized in according to OWL model. Protégé is a software that has a user interface with the ability to create ontology classes and concepts, and then it could map the created ontologies with datasets from research projects. Figure 17 shows the protégé example with SeMFIS export plugin with the Selection of classes, properties, individuals for export.



Model name: Risk_ontology Version: 1.0 Save as: risk_ontology.xml

Classes:
- ✗ Elementary_Threat
- ✗ Generic_Risk
- ✓ Risk_Actor
- ✗ Force_Majeure
- ✓ Organizational_Shortcomi
- ☑ Technical_Failure
- ✗ Deliberate_Acts
- ✗ Human_Failure
- ✗ Aircondition_Failure

Properties:
- ☑ Impact
- ✓ Probability

☑ Show Inferences

Figure 17. Protégé example [43]

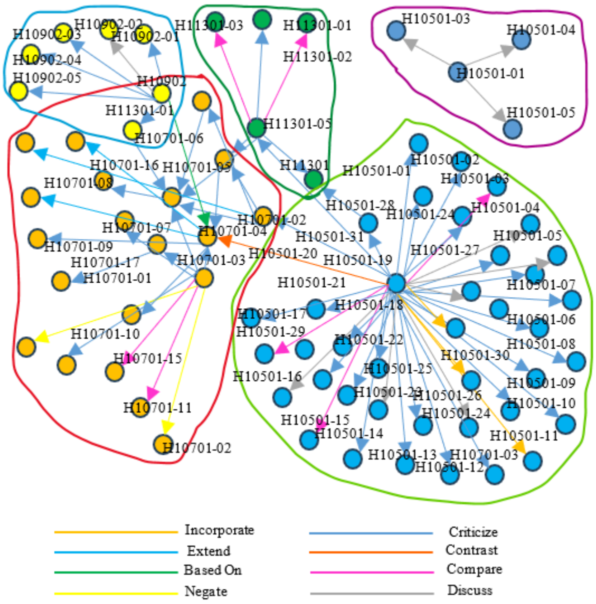


Figure 18. CCRO Based Semantic Citation Graph

According to the study by Han et al. [5]. Relation Extraction (RE) approaches are still used in simplified situations, and these methods focus on training models with a high number of human annotations to categorize entities with one phrase into relations. However, in the actual world, (1) gathering high-quality human annotations is costly and time-consuming; (2) many long-tail interactions cannot give vast quantities of training instances; and (3) most facts are stated in an extended context consisting of numerous phrases. (4) It is difficult to use a predetermined set to cover such relationships with open-ended expansion. As a result, create an effective and resilient RE system for real-world deployment. The researcher provides a comprehensive and detailed review of relation extraction model development, generalizes four promising directions leading to more powerful RE systems (using more data, performing more efficient learning, handling more complicated contexts, and orienting more open domains), and investigates two key challenges faced by existing RE models.

Semantic tags can enrich the citation graph by interconnecting papers with citation reasons. The writer noted that the research is one of the best sources of information for determining the purpose of citation [52]. In this research, the writers developed a system called CCRO (Citation, Context, and Reasons Ontology). This system has been used to semantically tag the citation in Latex documents to find out the relation between articles. They also examined a variety of automatic and human authoring systems for the integration of citation reasons, as shown in Figure 18. In order to facilitate the annotation and organization of citation contexts and justifications, the CCRO ontology offers a systematic framework that helps scholars examine and comprehend the connections among various academic publications. It enables the display of several citation contexts, including direct quotes, paraphrases, or summaries, as well as the justifications for referencing particular sources, like supporting data, opposing views, or similar works. Applications for the ontology include recommendation systems, information retrieval, citation analysis, and automated literature reviews.

The typesetting program LaTeX is widely used in academic and scientific publications. It enables writers to produce texts with excellent typography, especially when writing about science and technology. Authors may organize their writings, add mathematical equations, handle references, and produce output that looks professional with LaTeX's markup language and macros. The motivation in this study illustrates why the researchers chose Latex. They found from the history that there are so many authors and different disciplines used in Latex, which makes it an interesting amount of data to do research, as shown in Table 3.

Table 3. Different disciplines used in Latex

Discipline	Cited Documents	Rate	Documents
Mathematics	4,190,427	96,9	4,002,810
Physics and Astronomy	8,262,894	60,0	4,957,736
Computer Science	6,556,510	45,8	2,950,430
Total	19,009,831		11,930,976

Loreggia et al. [53] used the SenTag web-based tool that is used for semantic annotation of documents in textual format. In this paper, there are a group of people who are experts employed to annotate the documents manually in order to identify the important details to train the AI models, so the researcher presents SenTag, a web-based application that has the ability to provide an intuitive interface for document annotation. Finally, the output of the model will be an extensible markup language (XML). The dataset that has been manually annotated plays an important role in defining the standards. ML systems provide better performance, but these systems strongly depend on human annotation. The NER annotator is a web-based annotation tool that provides a GUI, a graphical user interface, that assists any user in creating document annotations and also generates training data. The team of SenTag annotators concludes from admin, editor, and annotator. The admin role is significant in that it has the highest level by creating other users and giving them privileges. The editor's role is to upload the XML schema and assign the text to the schema, and he can also assign the text to the annotator. Finally, the annotator has the right to tag the documents and validate the work of the XML schema.

The study by Chen et al. [54] mentioned that matching table elements with knowledge graphs is called semantic annotation with tabular data. It is significant to semantically annotate tabular data for downstream applications such as data management and analysis. Because of insufficient tabular data descriptions, heterogeneity, and vocabulary issues, semantic annotation is considered a challenging task. This work presents MTab4D, an automated semantic annotation method for generating annotations using DBpedia. In order to address issues with schema heterogeneity, data ambiguity, and noise, MTab4D is a table annotation system that integrates diverse matching signals from various table components. This paper also provides relevant research and further resources for knowledge graph-based semantic annotation measurement. The researcher also creates MTab4D APIs and graphical user interfaces for repeatability. Their approach excels at all three matching tasks, according to the findings of their trials using the initial and revised datasets of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019).

The suggested system's approach consisted of three tasks: (1) matching the cells of the table with entities; (2) matching the columns to entities; and (3) matching the pair of columns to attributes. The author suggests a pipeline annotation that merges multiple matchings from distinct table parts to handle schema heterogeneity and then data ambiguity. Figure 19 depicts the annotation workflow.

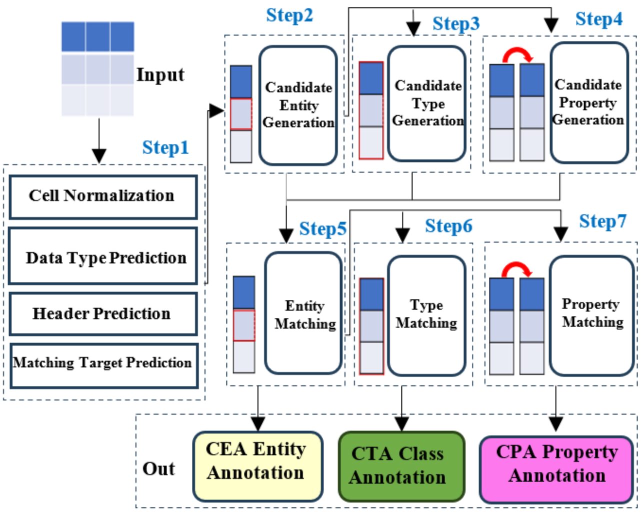


Figure 19. MTab4D framework for tabular data annotation

The research by Di Martino et al. [55] uses a technology named SemPRNN, which is a web-based tool that provides semantic annotation of business process model notation (BPMN) with OWL ontology concepts. SemPRNN uses the domain ontology to provide unambiguity to identify the concepts; this tool is used to upload the ontology, BPMN notation, and prolog to provide the annotations required for the purpose of business process models. Figure 20 shows the semantic annotation in BPMN.

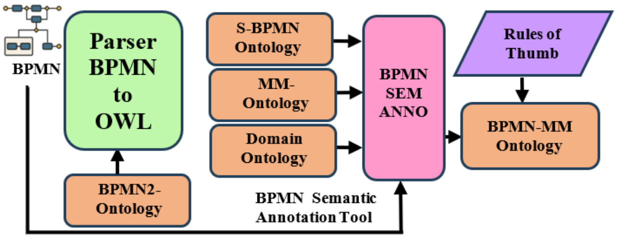


Figure 20. semantic annotation in BPMN

9. CONCLUSION

In this article, a review of previous works has been presented about semantic annotation and the tools and methods that are used to annotate the input text data in different formats. Various systems and technologies have been explained from the beginning of using semantic annotation methods to the present. The role of ontology and how it is mapped to the dataset. Additionally, this article illustrates the use of LOD, RDF, and SPARQL, in which these methods and tools are fully compatible with the structure of semantic annotation. The benefits of interoperability, flexibility, and querying capabilities may be used by semantic annotation through the combination of RDF as a data model, LOD principles, and SPARQL as a query language. They can be searched for using SPARQL, linked to other RDF datasets, and shown as RDF data so that semantic annotations can be taken out and changed to fit different needs or use cases. Semantic annotation systems are more interoperable and scalable as a result of their integration and compatibility with LOD, RDF, and SPARQL. The research mentions the use of deep learning methods with semantic annotation and how they are utilized to solve problems mentioned in the literature. First, using semantic annotation for Arabic corpus and how to enrich this data with ontology to convert the data into machine-readable format to be processed with NLP tasks. Then, using semantic annotation in XML schemas, it is aided by the XML schema definition, and then the metadata describes the XML message exchange. The SeMFIS method is also used to provide interconnection between two fields of ontologies and conceptual modeling. The MTab4D web-based tool is also used to solve the problem of tabular data annotation tasks. This research explains that protégé is a significant tool to create and manage ontologies that can be used to map them with the entities extracted from dataset used. The evolution of semantic annotation methods is starting from simple tasks to contemporary tasks in which deep learning methods are used. Because ontology offers a formal, organized representation of domain knowledge, it is essential to semantic annotation. The contribution of ontology to semantic annotation in terms of conceptual framework. It offers a conceptual framework and standard terminology that annotators can employ to precisely and consistently represent the material. Semantic annotation uses ontologies to guarantee that meanings are clear and consistent across many applications and systems. In terms of meaning representation, through the specification of classes, subclasses, properties, and links between concepts, ontologies provide an organized representation of knowledge. Annotators can provide a more accurate and insightful description of the content by utilizing these pre-established ideas and relationships. From a future viewpoint, natural language comprehension tasks have demonstrated impressive advancements in the use of deep learning NLP techniques, such as neural networks and deep language models. Subsequent investigations may examine the utilization of these methodologies to enhance the precision and effectiveness of semantic annotation. To handle large-scale annotation activities, automate the annotation process, and extract more complex interpretations, for example, deep learning models specialized particularly for semantic annotation might be developed. The contribution of this article gives an overview of the contemporary topic by summarizing and surveying the body of work that has been done on semantic annotation. They pinpoint important research trends, approaches, obstacles, and trending themes. This contributes to a greater knowledge of the field's evolution for scholars and practitioners.

REFERENCES

[1] Yin, X., Huang, Y., Zhou, B., Li, A., Lan, L., Jia, Y. (2019). Deep entity linking via eliminating semantic ambiguity with BERT. IEEE Access, 7: 169434-169445. https://doi.org/10.1109/ACCESS.2019.2955498

[2] Press, O., Smith, N.A., Levy, O. (2019). Improving transformer models by reordering their sublayers. arXiv preprint arXiv:1911.03864. https://doi.org/10.48550/arXiv.1911.03864

[3] Pamart, A., Roussel, R., Hubert, E., Colombini, A., Saleri, R., Mouaddib, E. M., Castro, Y., Goic, G.L., Mansouri, A. (2022). A semantically enriched multimodal imaging approach dedicated to conservation and restoration studies. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 46: 415-420. https://doi.org/10.5194/isprs-archives-XLVI-2-W1-2022-415-2022

[4] Hinze, A., Heese, R., Schlegel, A., Paschke, A. (2019). Manual semantic annotations: User evaluation of interface and interaction designs. Journal of Web Semantics, 58: 100516. https://doi.org/10.1016/j.websem.2019.100516

[5] Croce, V., Caroti, G., De Luca, L., Piemonte, A., Véron, P. (2020). Semantic annotations on heritage models: 2D/3D approaches and future research challenges. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43(B2): 829-836. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-829-2020

[6] Balakrishna, S., Thirumaran, M., Solanki, V.K., Núñez-Valdez, E.R. (2020). Incremental hierarchical clustering driven automatic annotations for unifying IoT streaming data. International Journal of Interactive Multimedia and Artificial Intelligence, 6(2): 56-70. https://doi.org/10.9781/ijimai.2020.03.001

[7] Scerri, S., Handschuh, S., Sintek, M., Oren, E., Möller, K.H. (2006). What are semantic annotations? http://w3.org/RDF.

[8] Kügler, P., Marian, M., Dorsch, R., Schleich, B., Wartzack, S. (2022). A Semantic annotation pipeline towards the generation of knowledge graphs in tribology. Lubricants, 10(2): 18., https://doi.org/10.3390/lubricants10020018

[9] Fan, Y., Fan, L., Yang, J. (2018). Improving semantic annotation using semantic modeling of knowledge embedding. In International Conference on Cloud Computing and Security, Springer, Cham, pp. 575-585. https://doi.org/10.1007/978-3-030-00021-9_51

[10] What is Semantic Annotation? https://www.ontotext.com/knowledgehub/fundamentals/semantic-annotation/.

[11] Bouhissi, H.E., Salem, A.B.M., Tari, A. (2019). Semantic enrichment of web services using linked open data. International Journal of Web Engineering and Technology, 14(4): 383-416. https://doi.org/10.1504/IJWET.2019.105594

[12] Shamoug, A., Cranefield, S., Dick, G. (2018). Information retrieval via a semantically classified word embedding. Information Retrieval via a Semantically Classified Word Embedding.

[13] Soavi, M., Zeni, N., Mylopoulos, J., Mich, L. (2022). Semantic annotation of legal contracts with contrattoA. Informatics, 9(4): 72. https://doi.org/10.3390/informatics9040072

[14] Muhie Yimam, S., Eckart de Castilho, R., Gurevych, I., Biemann, C. (2014). Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno, pp. 91–96. http://www.{lt,ukp}.tu-darmstadt.de.

[15] Haghgoo, M., Nazary Aghche Mazary, A., Monti, A. (2022). SiSEG-auto semantic annotation service to integrate smart energy data. Energies, 15(4): 1428. https://doi.org/10.3390/en15041428

[16] Stork, L., Weber, A., Miracle, E.G., Verbeek, F., Plaat, A., van den Herik, J., Wolstencroft, K. (2019). Semantic annotation of natural history collections. Journal of Web

Semantics, 59: 100462. https://doi.org/10.1016/j.websem.2018.06.002

[17] Neal, M.L., König, M., Nickerson, D., Mısırlı, G., Kalbasi, R., Dräger, A., Atalag, K., Chelliah, V., Cooling, M.T., Waltemath, D. (2019). Harmonizing semantic annotations for computational models in biology. Briefings in Bioinformatics, 20(2): 540-550. https://doi.org/10.1093/bib/bby087

[18] Kesäniemi, J., Suominen, T., Mankinen, K. (2022). Challenges in managing semantic annotations in harvested research objects in a national CRIS context. Procedia Computer Science, 211: 251–256. https://doi.org/10.1016/j.procs.2022.10.199

[19] Dash, N.S. (2021). Language corpora annotation and processing. Springer Singapore. https://doi.org/10.1007/978-981-16-2960-0

[20] Haider, N., Hossain, F. (2018). CSV2RDF: Generating RDF data from CSV file using Semantic Web technologies. Journal of Theoretical and Applied Information Technology, 96(20): 6889-6902.

[21] Mihindukulasooriya, N. (2020). A framework for linked data quality based on data profiling and rdf shape induction. Doctoral dissertation, ETSI_Informatica. https://doi.org/10.20868/UPM.thesis.62935

[22] Tomaszuk, D., Hyland-Wood, D. (2020). RDF 1.1: Knowledge representation and data integration language for the web. Symmetry, 12(1): 84. https://doi.org/10.3390/sym12010084

[23] Zhong, X., Wang, Y., Wen, X., Liao, J. (2022). An ontology-based automation system. International Journal of Semantic Web and Information Systems, 18(1): 1-22. https://doi.org/10.4018/ijswis.295946

[24] Beek, W., Rietveld, L., Bazoobandi, H.R., Wielemaker, J., Schlobach, S. (2014). LNCS 8796 - LOD laundromat: A uniform way of publishing other people's dirty data, 8796: 213-228. https://doi.org/10.1007/978-3-319-11964-9_14

[25] Xu, H., Sun, H. (2022). Application of rough concept lattice model in construction of ontology and semantic annotation in Semantic Web of things. Scientific Programming. https://doi.org/10.1155/2022/7207372

[26] Kang, X., Zhao, Y., Yuan, P., Jin, H. (2021). Grace: An efficient parallel SPARQL query system over large-scale RDF data. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2021), Dalian, China, pp. 769-774. https://doi.org/10.1109/CSCWD49262.2021.9437674

[27] Göppert, A., Grahn, L., Rachner, J., Grunert, D., Hort, S., Schmitt, R.H. (2023). Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems. Journal of Intelligent Manufacturing, 34(5): 2133-2152. https://doi.org/10.1007/s10845-021-01860-6

[28] Bonifati, A., Martens, W., Timm, T. (2020). An analytical study of large SPARQL query logs. VLDB Journal, 29(3): 655-679. https://doi.org/10.1007/s00778-019-00558-9

[29] Ali, W., Saleem, M., Yao, B., Hogan, A., Ngomo, A.C. N. (2022). A survey of RDF stores & SPARQL engines for querying knowledge graphs. The VLDB Journal, 1-26. https://doi.org/10.1007/s00778-021-00711-3

[30] Hadi, A.S., Fergus, P., Dobbins, C., Al-Bakry, A.M. (2013). A machine learning algorithm for searching vectorised RDF data. In 2013 27th International Conference on Advanced Information Networking and Applications Workshops, Barcelona, Spain, pp. 613-618 https://doi.org/10.1109/WAINA.2013.204

[31] Neal, M.L., Thompson, C.T., Kim, K.G., James, R.C., Cook, D.L., Carlson, B.E., Gennari, J.H. (2019). SemGen: A tool for semantics-based annotation and composition of biosimulation models. Bioinformatics, 35(9): 1600-1602. https://doi.org/10.1093/bioinformatics/bty829

[32] Tsvigun, A., Lysenko, I., Sedashov, D., Lazichny, I., Damirov, E., Karlov, V., Belousov, A., Sanochkin, L., Panov, M., Panchenko A., Burtsev M., Shelmanov, A. (2023). Active learning for abstractive text summarization. arXiv preprint arXiv:23 Schreiber 01.03252. https://doi.org/10.48550/arXiv.2301.03252

[33] Amjad, A., Azam, F., Anwar, M.W., Butt, W.H., Rashid, M. (2018). Event-driven process chain for modeling and verification of business requirements–A systematic literature review. IEEE Access, 6: 9027-9048. https://doi.org/10.1109/ACCESS.2018.2791666

[34] Yu, P., Zhang, Z., Voss, C., May, J., Ji, H. (2022). Building an event extractor with only a few examples. In Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, Hybrid, pp. 102-109. https://aclanthology.org/2022.deeplo-1.11

[35] Mutlu, O. (2022). Utilizing coarse-grained data in low-data settings for event extraction. arXiv preprint arXiv:2205.05468. https://doi.org/10.48550/arXiv.2205.05468

[36] Hendler, D.A.J. (2020). Semantic Web for the Working Ontologist. Effective Modeling in RDFS and OWL. https://books.google.iq/books?hl=en&lr=&id=_qGKPOlB1DgC&oi=fnd&pg=PP1&dq=Semantic+Web+for+the+Working+Ontologist&ots=-0ekJYwRhS&sig=NT9FvtBX6UgJVZCgdK9-2rHVGsE&redir_esc=y#v=onepage&q=Semantic%20Web%20for%20the%20Working%20Ontologist&f=false, accessed on Feb. 19, 2024

[37] Morris, J.X., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y. (2020). Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. arXiv preprint arXiv:2005.05909. https://doi.org/10.48550/arXiv.2005.05909

[38] Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE, 109(3): 247-278. https://doi.org/10.1109/JPROC.2021.3060483

[39] Yang, Y., Kim, S., Joo, J. (2022). Explaining deep convolutional neural networks via latent visual-semantic filter attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8333-8343.

[40] Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B, Shahrzad, H., Navruzyan, A., Duffy, N., Hodjat, B. (2024). Evolving deep neural networks. In Artificial intelligence in the age of neural networks and brain computing, pp. 269-287. https://doi.org/10.1016/B978-0-323-96104-2.00002-6

[41] Adnan, M.M., Rahim, M.S.M., Khan, A.R., Saba, T., Fati, S.M., Bahaj, S.A. (2022). An improved automatic image annotation approach using convolutional neural network-slantlet transform. IEEE Access, 10: 7520-7532.

https://doi.org/10.1109/ACCESS.2022.3140861

[42] Wang, H., Qin, K., Zakari, R. Y., Lu, G., Yin, J. (2022). Deep neural network-based relation extraction: An overview. Neural Computing and Applications, 1-21. https://doi.org/10.1007/s00521-021-06667-3

[43] Kherwa, P. (2022). A survey on semantic annotation tools for knowledge management. International Journal of Engineering Applied Sciences and Technology, 6(12): 217-219.

[44] Neves, M., Ševa, J. (2021). An extensive review of tools for manual annotation of documents. Briefings in bioinformatics, 22(1): 146-163. https://doi.org/10.1093/bib/bbz130

[45] Tosi, D., Morasca, S. (2015). Supporting the semi-automatic semantic annotation of web services: A systematic literature review. Information and Software Technology, 61: 16-32. https://doi.org/10.1016/j.infsof.2015.01.007

[46] Andrews-Todd, J., Steinberg, J., Pugh, S. L., D'Mello, S.K. (2022). Comparing collaborative problem solving profiles derived from human and semi-automated annotation. In Proceedings of the 15th International Conference on Computer-Supported Collaborative Learning-CSCL 2022, pp. 363-366. International Society of the Learning Sciences. https://doi.dx.org/10.22318/cscl2022.363

[47] Bouscarrat, L., Bonnefoy, A., Capponi, C., Ramisch, C. (2021). AMU-EURANOVA at CASE 2021 Task 1: Assessing the stability of multilingual BERT. arXiv preprint arXiv:2106.14625. https://doi.org/10.48550/arXiv.2106.14625

[48] Albukhitan, S., Alnazer, A., Helmy, T. (2018). Semantic annotation of Arabic web documents using deep learning. Procedia Computer Science, 130: 589-596. https://doi.org/10.1016/j.procs.2018.04.108

[49] Campos-Rebelo, R., Moutinho, F., Paiva, L., Maló, P. (2019). Annotation rules for xml schemas with grouped semantic annotations. In IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, 1: 5469-5474.

[50] Fill, H.G. (2017). SeMFIS: A flexible engineering platform for semantic annotations of conceptual models. Semantic Web, 8(5): 747-763. https://doi.org/10.3233/SW-160235

[51] Han, X., Gao, T., Lin, Y., Peng, H., Yang, Y., Xiao, C., Liu, Z.Y., Li, P., Sun, M., Zhou, J. (2020). More data, more relations, more context and more openness: A review and outlook for relation extraction. arXiv preprint arXiv:2004.03186. https://doi.org/10.48550/arXiv.2004.03186

[52] Ihsan, I., Ullah, M., Khan, R.U., Uddin, M.I., Alharbi, A., Alosaimi, W. (2022). SEAL: Semantically enriched authoring in latex-a model for scientific discourse. IEEE Access, 10: 13525-13535. https://doi.org/10.1109/ACCESS.2022.3145954

[53] Loreggia, A., Mosco, S., Zerbinati, A. (2022). Sentag: A web-based tool for semantic annotation of textual documents. In Proceedings of the AAAI Conference on Artificial Intelligence, 36(11): 13191-13193. https://doi.org/10.1609/aaai.v36i11.21724

[54] Chen, J., Jiménez-Ruiz, E., Horrocks, I., Sutton, C. (2019). Learning semantic annotations for tabular data. arXiv preprint arXiv:1906.00781. https://doi.org/10.48550/arXiv.1906.00781

[55] Di Martino, B., Colucci Cante, L., Esposito, A., Graziano, M. (2023). A tool for the semantic annotation, validation and optimization of business process models. Software: Practice and Experience, 53(5): 1174-1195. https://doi.org/10.1002/spe.3184