Table of Contents

1 Introduction

This section sets the stages for the ontology application that can impact healthcare and medical research by creating human disease information. You can find this project on my Github page https://github.com/sarashahin/Huma_Disease_Ontology.

1.1. Abstract, Aims

The main human diseases for this ontology are chronic respiratory disease, infectious disease, and acute disease, and these diseases are causing death in the world. One of the problems is risk factors, which are associated with chronic diseases and correlated with people's lifestyles. Finding disease in the early stages could prevent some chronic disease deaths.

A possible solution for this is to recognise behaviours related to risk factors, such as drinking alcohol, smoking, and weight gain, which were mapped by the World Health Organisation[14].

The proposed ontology project can track risk factors associated with diseases. Based on the behaviour risk factors, the App can make good recommendations, for example, a healthy diet or quitting smoking.

However, this ontology app performs as a knowledge-based for information systems; the primary goal is to search for the behaviour related to risk factors for chronic, infectious, and acute diseases.

The method I applied for this ontology that has axioms and semantic principles to create queries and inferences, I defined five question queries that enabled inferences to evaluate the human disease ontology.

I have partially imported a dataset from the Bioportal Biotechnology website and classified it into the expected classes and properties. Additionally, I have used SPARQL queries to answer those competence questions for this project.

As a second aim of this project, I gathered and arranged this information about various diseases, causes, symptoms, and treatments into a simple method(App). This will also assist doctors and researchers to easily find and understand information about various diseases. By doing this I hope to make it easier for people who learn more about diseases, causes, risk factors, and treatments, help researchers find new treatments or cures, and make this useful in improving healthcare and medical research.

2. Related works/ Literature review

Chronic diseases are one of the most dangerous diseases in the world leading to death, among 7 of 10.

This finding emphasises is necessary to prevent and treat Chronic diseases, such as diabetes, chronic respiratory and heart diseases[1].

By early changing of a healthy lifestyle can lower risk of chronic diseases and minimise the death rates.
Reducing tobacco and alcohol consumption, sport activity, and maintaining a balanced diet are a few examples of these adjustments.Thus, encouraging healthy habits need an understanding of human behaviour related to lifestyle choices[2].
Recent research has found that the pandemic of COVID-19 has had an impact on the people's lifestyle and healthcare, mainly with increased intake of alcohol, tanbacoo, and fast foods and also decreased sport activity. The recent research by Malta et al. showed that around 58% of young ages with noncommunicable chronic diseases and decreased physical activity while during the pandemic, increased the consumption of frozen food by 53% [7].
The paper describes how to create a consistent OWL ontology for vitamin A and its impact on humans by using Protege Tool. It also provides information about vitamin A effect on human health, risk factors affecting these impacts, and diet control. Additionally, it indicates a group of people at risk of Vitamin A deficiency[11].

The Human Disease Ontology significantly has grown with its user-base and disease content developing greatly since 2018. Use of Disease Ontology terms has been increased over five years, and now linked to more than 1.5 million biomedical data sources. The latest research updates added around 1793 disease terms and improved the structure of Disease ontology and disease classification with thousands of new links between diseases[12].

This paper illustrates a system using an Ontology-based approach that improves diagnosis in human disease as well as the research. The proposed Ontology system features Generic Human Disease Ontology, that contains broad information about organised human disease into four categories: 1- disease type, 2- symptoms, 3- causes including both genetics and environmental factors, 4- treatments such as medicine, chemotherapy and surgery. The mentioned Ontology is especially useful for studying complex disorders caused by multiple factors like manic depressive disorders. Moreover, this is the creation of specific Human disease Ontology to help physician and medical research[13].

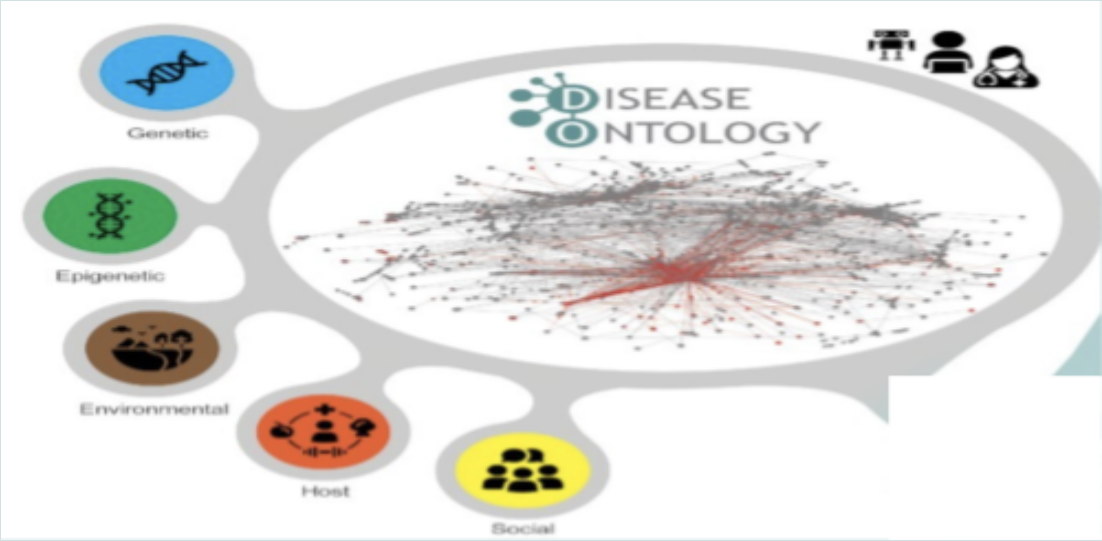
3. Design

3.1. Semantic Data Ontology:

An ontology is a formal and shareable representation of a domain knowledge which is made up of well defined concepts and rules. A list of competency questions(SPQRL) can be used to specify the Ontology scope. Therefore, reasoners can help in the search for answers to questions [3].

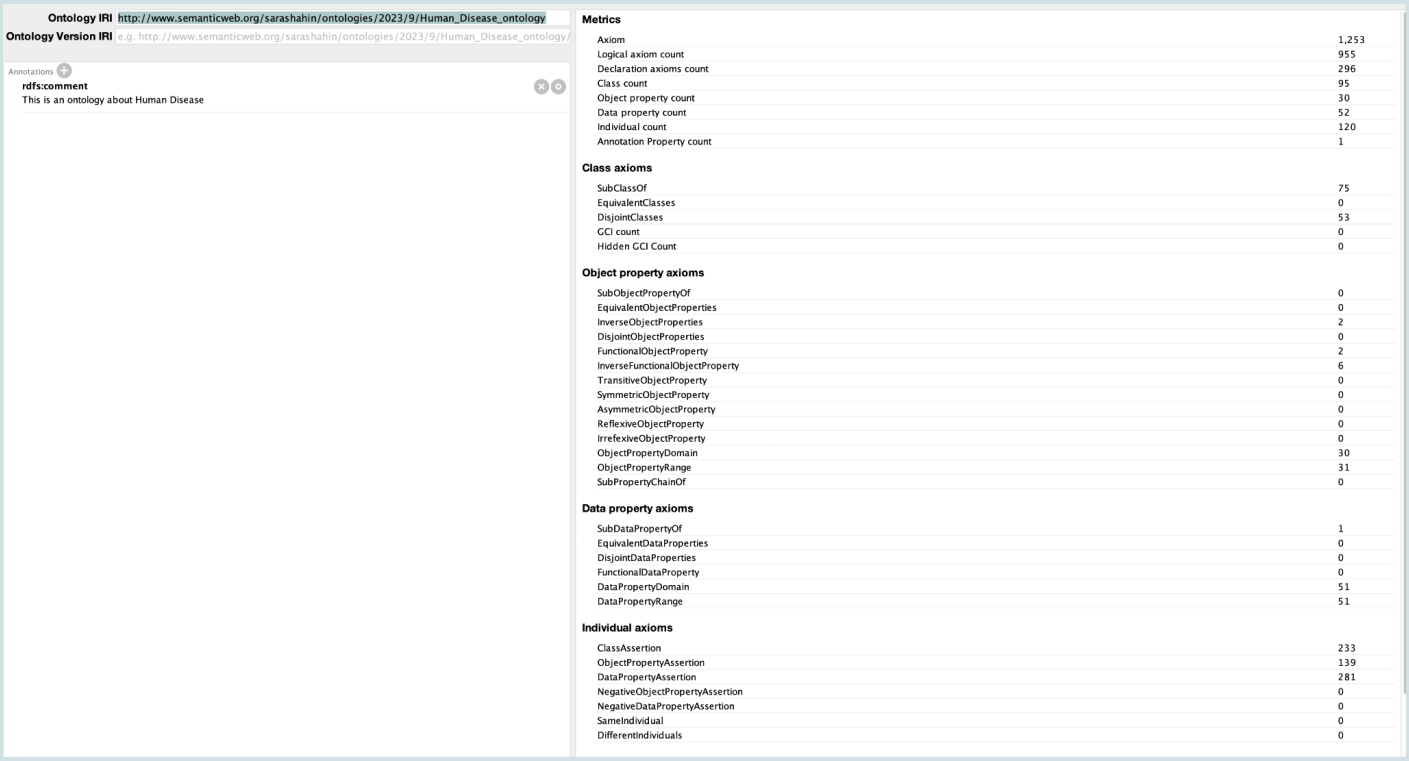
Ontology, as illustrated via Ontology Web Language(OWL), is a standard language of the World Wide Web Consortium (W3C) (https://www.w3.org).
An OWL ontology is made up of four components: instances, concepts, properties, and restrictions.

In the ontology definition, concepts are used to define the objects. The individuals of classes defined by instances. Individuals can be associated with other individuals among properties.
Constraints belong to a class where they are used to identify boundaries for the individuals.
SPARQL query language being used for an evaluation of the contents of an ontology. Inferences are related to classes and individuals can be used to identify Semantic Web Rule Language (SWRL)[4].
Properties are a binary relationship which is a one-to-one relation between two entities or among itself[6].
As a result, from construction methodologies, ontologies are used showing infer knowledge and domains, in various fields, for example the Web of Things (WoT) and Decision Support Systems (DSS), compared with others[5]. Goal of this project to expand an ontology that represents the knowledge for tracking behaviour and providing educational support.



Graphical of Disease[8]

3.2. Human Disease Ontology



Ontology IRI http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology
Ontology Version IRI e.g. http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology/
Annotations
rdfs:comment
This is an ontology about Human Disease

Metrics

Axiom	1,253
Logical axiom count	955
Declaration axioms count	296
Class count	95
Object property count	30
Data property count	52
Individual count	120
Annotation Property count	1

Class axioms

SubClassOf	75
EquivalentClasses	0
DisjointClasses	53
GCI count	0
Hidden GCI Count	0

Object property axioms

SubObjectPropertyOf	0
EquivalentObjectProperties	0
InverseObjectProperties	2
DisjointObjectProperties	0
FunctionalObjectProperty	2
InverseFunctionalObjectProperty	6
TransitiveObjectProperty	0
SymmetricObjectProperty	0
AsymmetricObjectProperty	0
ReflexiveObjectProperty	0
IrrefexiveObjectProperty	0
ObjectPropertyDomain	30
ObjectPropertyRange	31
SubPropertyChainOf	0

Data property axioms

SubDataPropertyOf	1
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	0
DataPropertyDomain	51
DataPropertyRange	51

Individual axioms

ClassAssertion	233
ObjectPropertyAssertion	139
DataPropertyAssertion	281
NegativeObjectPropertyAssertion	0
NegativeDataPropertyAssertion	0
SameIndividual	0
DifferentIndividuals	0

Figure 1. Ontology metrics

The ontology is implemented by Protege version 5.6.1 in Resource Description Framework (RDF) format. The current version of ontology includes 1253 Axiom, 95 classes, 30 Object properties, 52 Data properties, and 120 Individuals. Axioms are logical expressions that define a concept. The metrics of "Total classes" and "Total subclasses" show the number of classifications in the ontology. Object properties indicate the relationships between instances of two classes. Class and property partially imported from bioportal bio ontology website, initially, I collected data as RDF and imported into Protege then made some modifications(Figure 2)[19].
(https://bioportal.bioontology.org/ontologies/DOID/?p=summary)

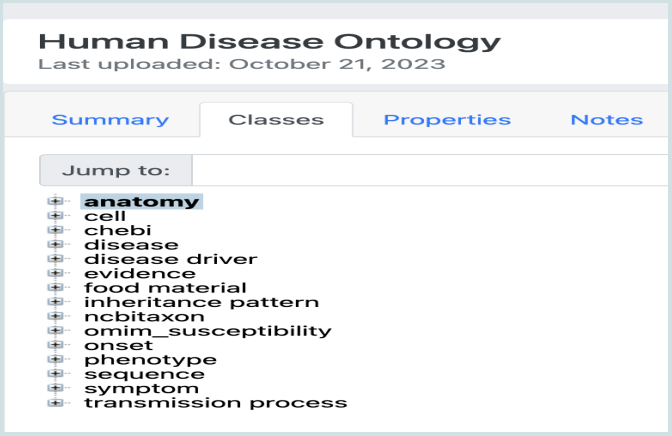


Figure 2. Importing data

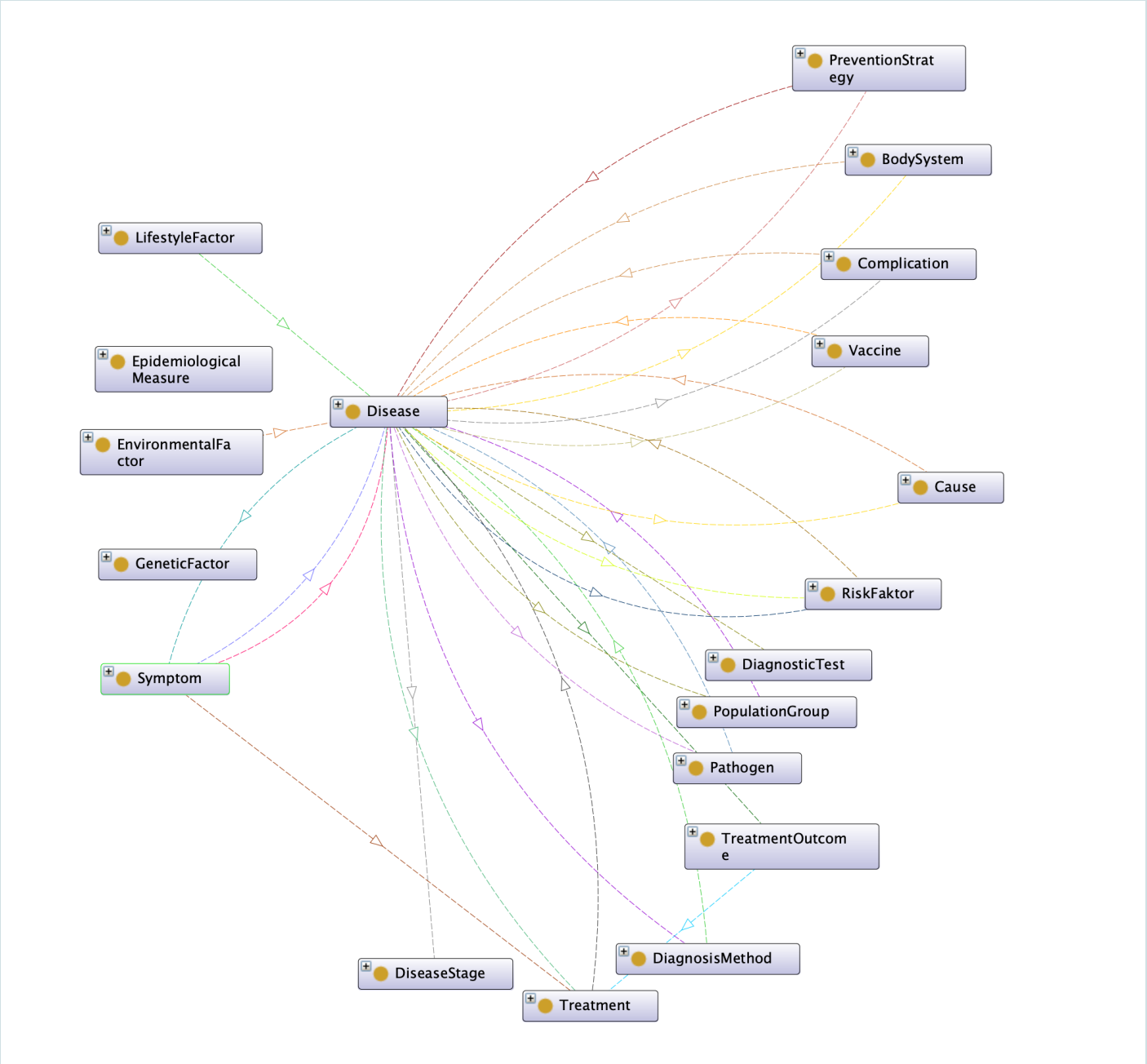
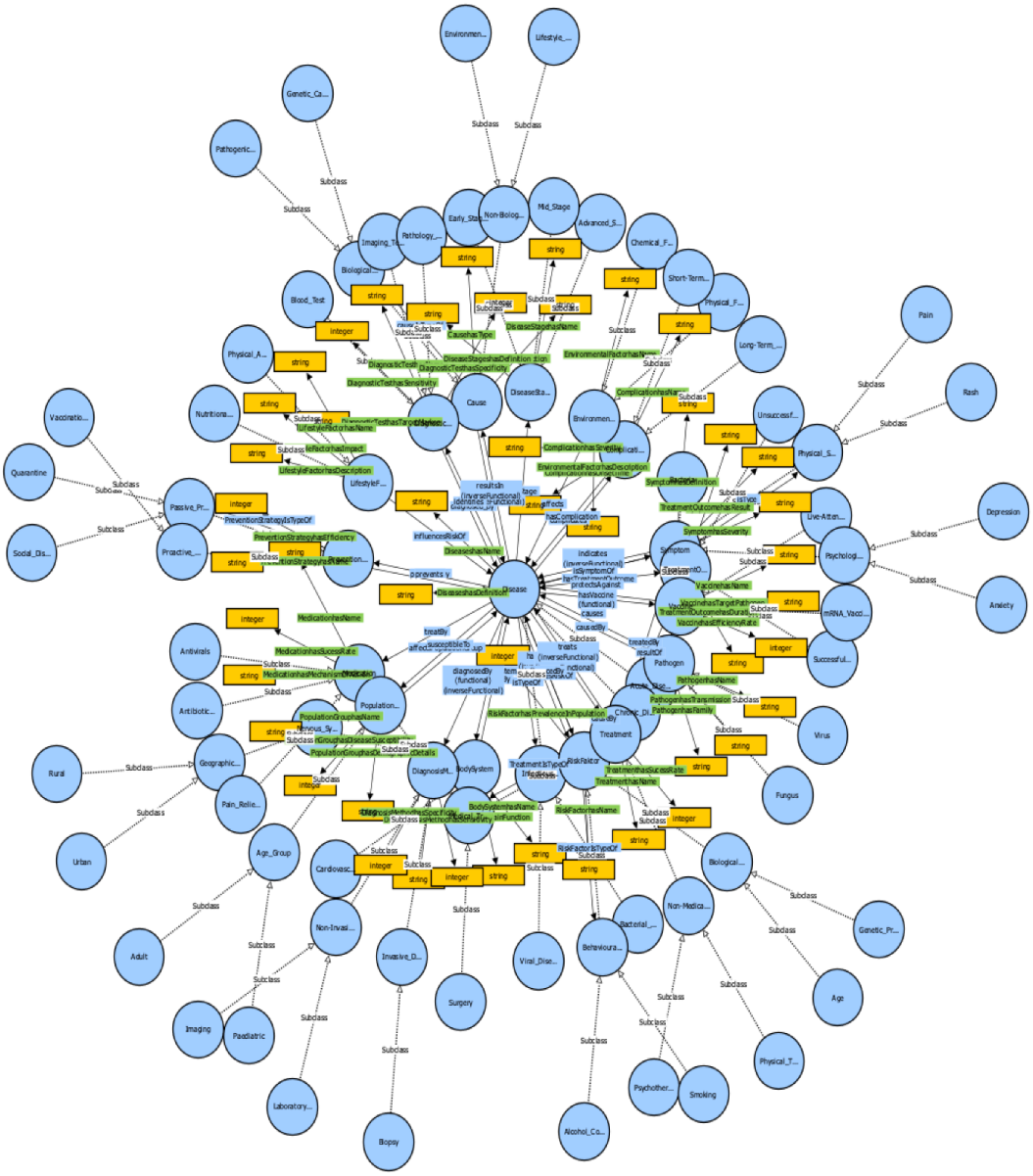


Figure3. Hierarchical of ontology classes.
The classes are described in the particular form, and each class can indicate single instance or multi instances. Additionally, this behaviour risk factor is related to chronic diseases. Table 1 describes the main classes.



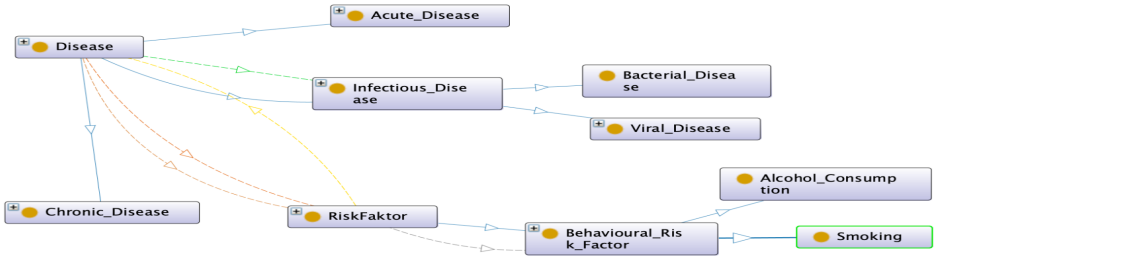
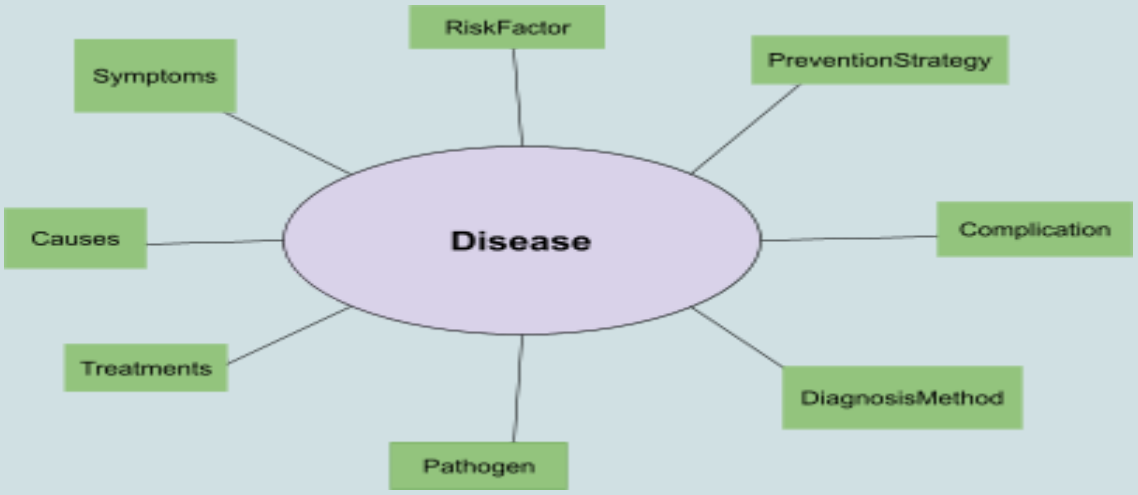


Table 1. Main classes of ontology.

Class	Description/ Subclasses
Disease	Class consisting of subclasses such as Infectious Disease, Chronic Disease and, Acute Disease.
BodySystem	Class represents the body, like nerves or heart.
Cause	Class that includes subclasses like Biological Cause and Non-Biological Cause.
DiagnosisMethod	Classes contain subclasses such as Invasive Diagnosis and Non-Invasive Diagnosis.
DiagnosisTest	Class represents blood tests, such as image tests and pathology tests.
Pathogen	This class consists of bacteria, fungus and viruses.
PreventionStrategy	Class includes subclasses such as Proactive Prevention and Passive Prevention.
RiskFaktor	Class consists of Biological Risk Factor and Behavioural Risk Factor.
Symptom	Class contains subclasses like Physical Symptoms and Psychological Symptoms.
Treatment	Class includes subclasses such as Medical Treatment and Non-Medical Treatment.
LifestyleFactor	Class consists of nutritional choices and physical activity.
Complication	Class includes long term complication or short term complication.

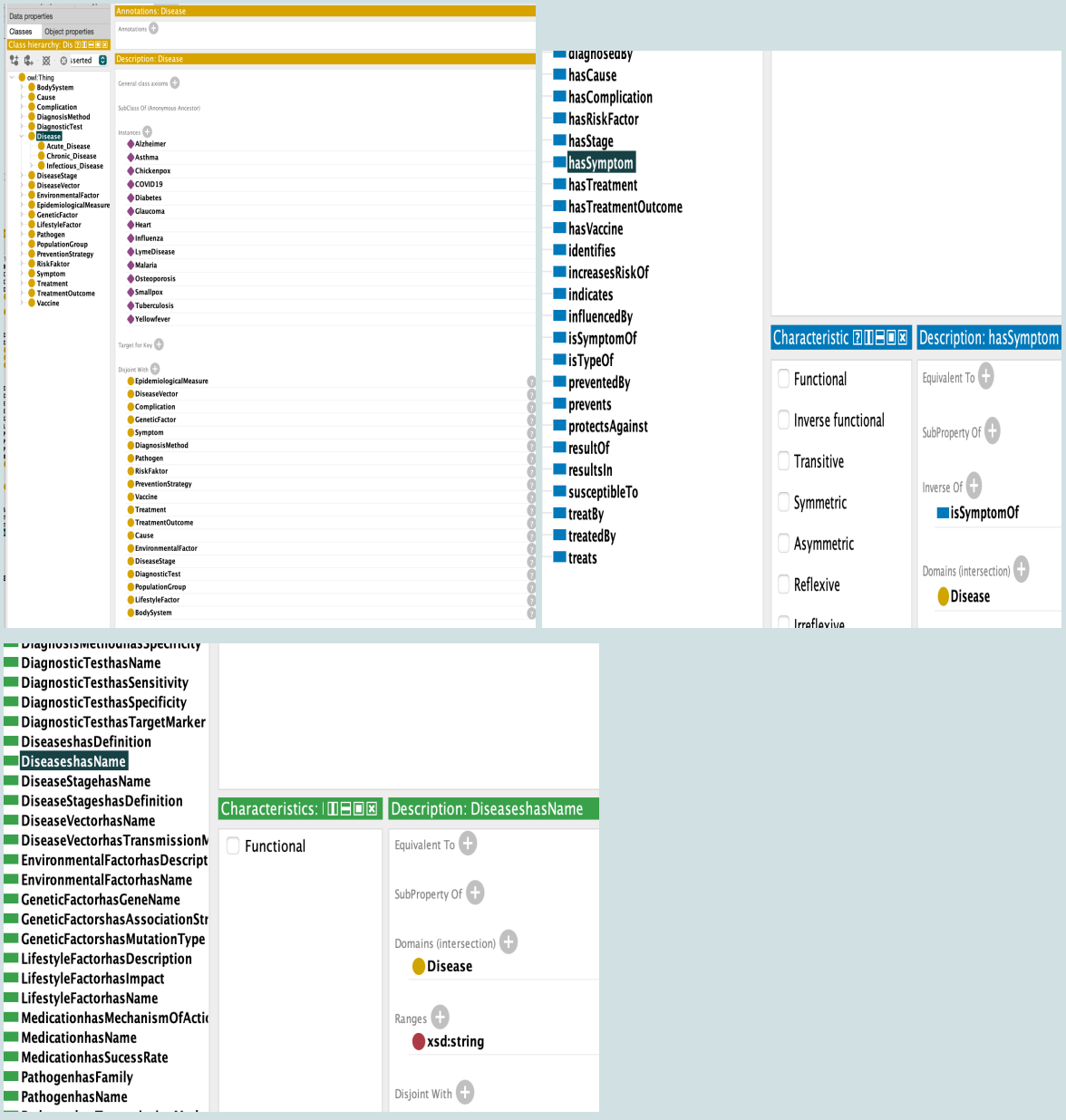


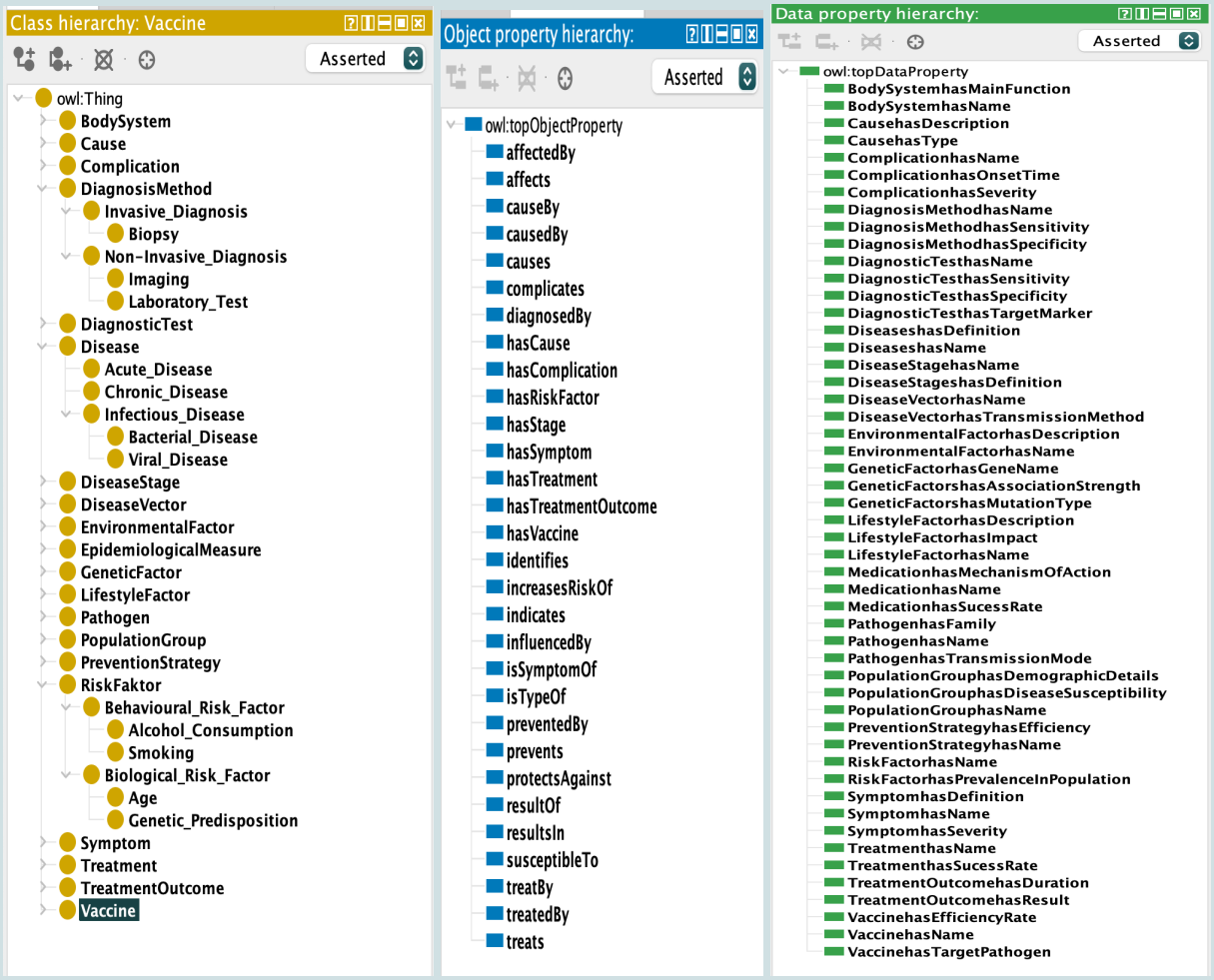
3.2.1. Define Relationships and Class Properties:

There are five steps to identify the relationships and properties of the classes.
Figure 3. displays the structure of ontology classes in hierarchical yellow colour. The connections between these classes are as object property and marked in blue, while the attributes of each class are indicated as data property and marked in green. All of this detailed information is available in Protege.

Based on my small project, I have decided to use this format for the data properties, they are clear and follow a pattern, making it easy to understand what each property represents, for example, DiseasehasDifinition or SymptomhasName.

Figure 3 displays the structured layers of the ontology's classes in yellow, illustrating a clear hierarchy. The connections between these classes are marked in blue (object properties), while the specific characteristics of each class are indicated in green (data properties). All of this detailed information can be accessed through the Protege software.





RiskFactorhasName(Figure 4.):

- This is a data property in the human disease ontology and assigned to a RiskFaktor entity.
- This represents that I have 30 individual risk factors, and each one of these with a name property set.
- Each risk factor is listed with a triangle which means as individuals.
- The data property RiskFactorhasName is showing that it is the type of xsd:string, and values are expected to be strings.
- This indicate that the domain of RiskFactorhasName is RiskFaktor class.

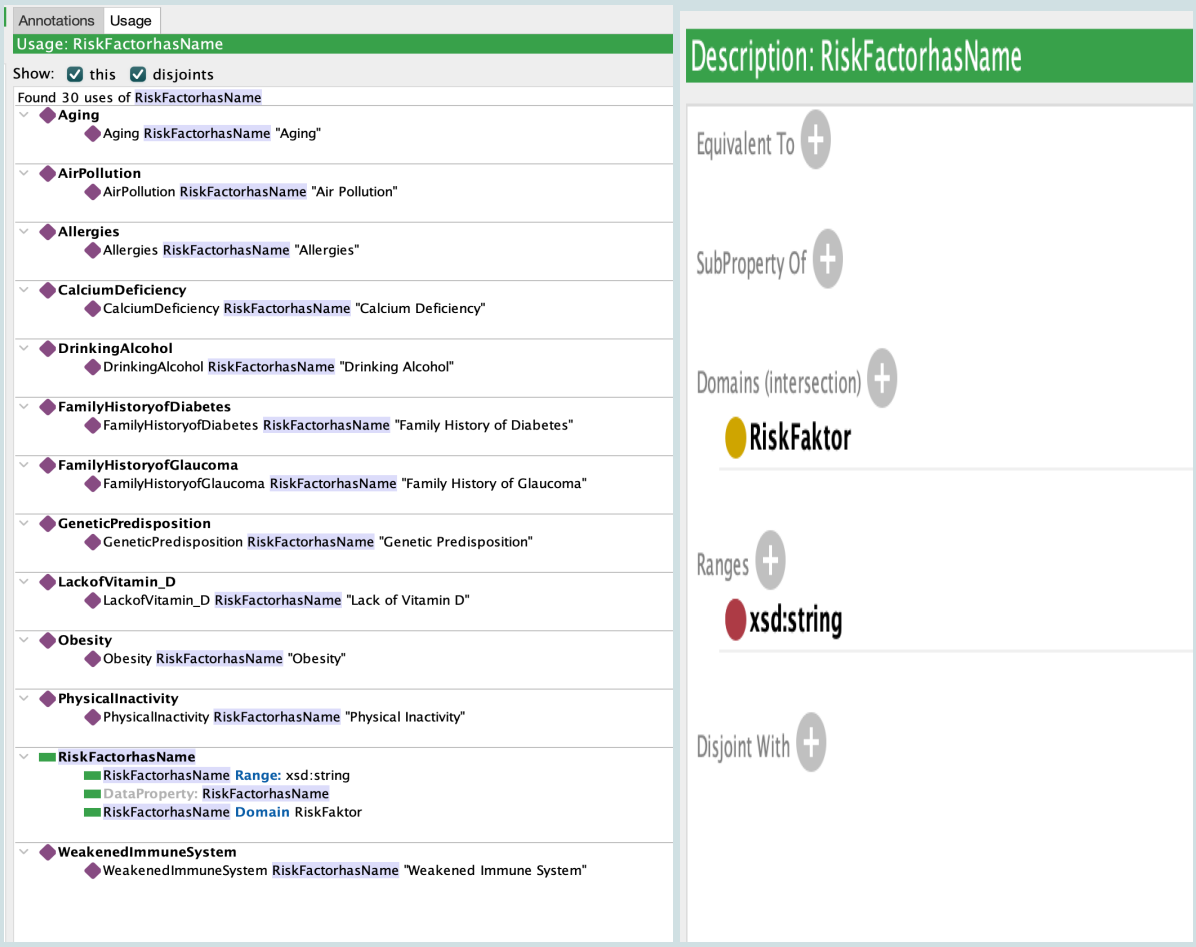


Figure 4.

hasRiskFactor property(Figure 5.):

- This is an object property in the human disease ontology. It is used to link diseases with their associated risk factors.
- There are 28 instances being used.
- For instance, Diabetes is related to FamilyHistoryofDiabetes, DrinkingAlcohol, and Obesity. This represents that having a family history of diabetes, alcohol consumption, and obesity are defined as risk factors for diabetes.
- The domain of hasRiskfactor is Disease, which means that this property is used to describe attributes of disease entities.
- Inverse Functional indicates that for any given risk factor, there can be only one disease related to it.
- The Inverse of shows that hasRiskFactor and increaseRiskOf are semantically related but in opposite directions such as if Diabetes hasRiskFactor Obesity, then Obesity increaseRiskOf Diabetes.

- The Range of hasRiskFactor is RiskFaktor, which indicates that the values of this property are expected to be individuals of the class RiskFaktor.

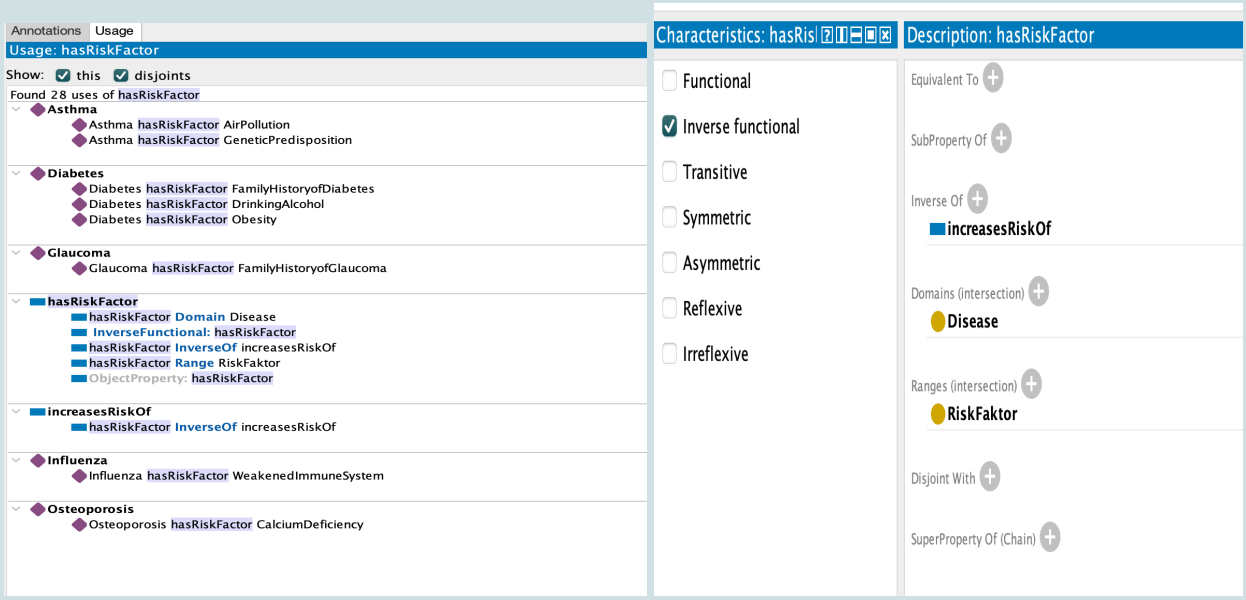


Figure 5.

DrikingAlcohol(Figure 6.):

- Is listed as an individual instance of the Behavioural_Risk_Factor subclass, which means that drinking alcohol is considered a specific behaviour that can increase blood sugar.
- increaseRiskOf Diabetes, this object property(Assertions) links DrinkingAlcohol directed to Diabetes is suggesting that alcohol consumption is a behaviour that can increase the risk of developing diabetes.
- RiskFactorhasName ‘Drinking Alcohol’, string,: this is data property(Assertions), the name of the risk factor.
- RiskFactorhasPrevalenceInPopulation, 85, integer: this indicates the prevalence of drinking alcohol as a percentage.

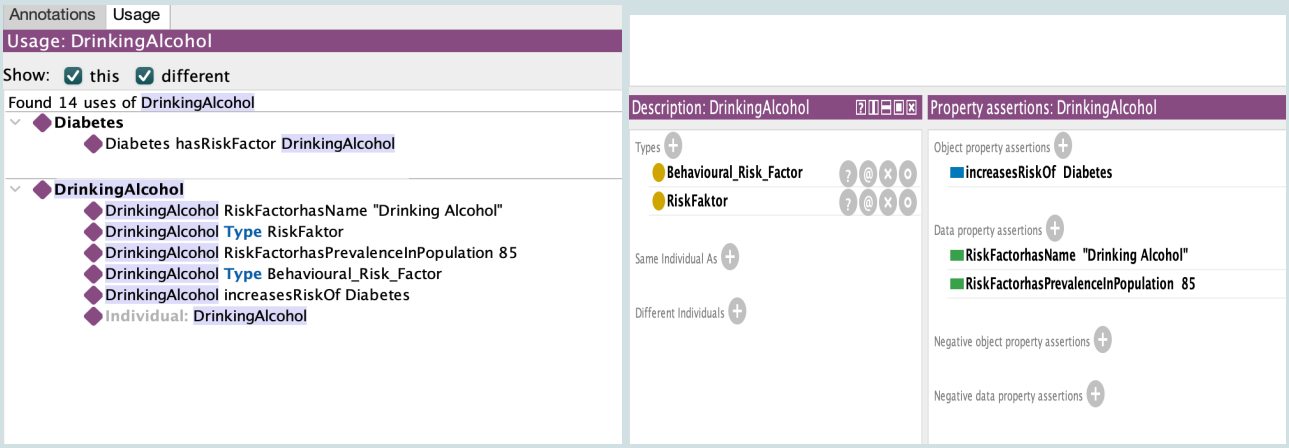


Figure 6.

treats (Figure 7.):

- This Object property has been used 18 times to connect treatments to the diseases they are intended to manage.
- Antibiotics are indicated as a treatment for both Influenza and LymeDisease.
- InsulinTherapy is used to treat Diabetes.
- Is characterised as Inverse Functional, showing that each disease is related to a unique treatment.
- The Domain of the treats property is Treatment, which describes attributes of the treatment entities.
- The Range of the 'treats' property is Disease, which indicates that the values of this property are expected to be instances of the disease class.

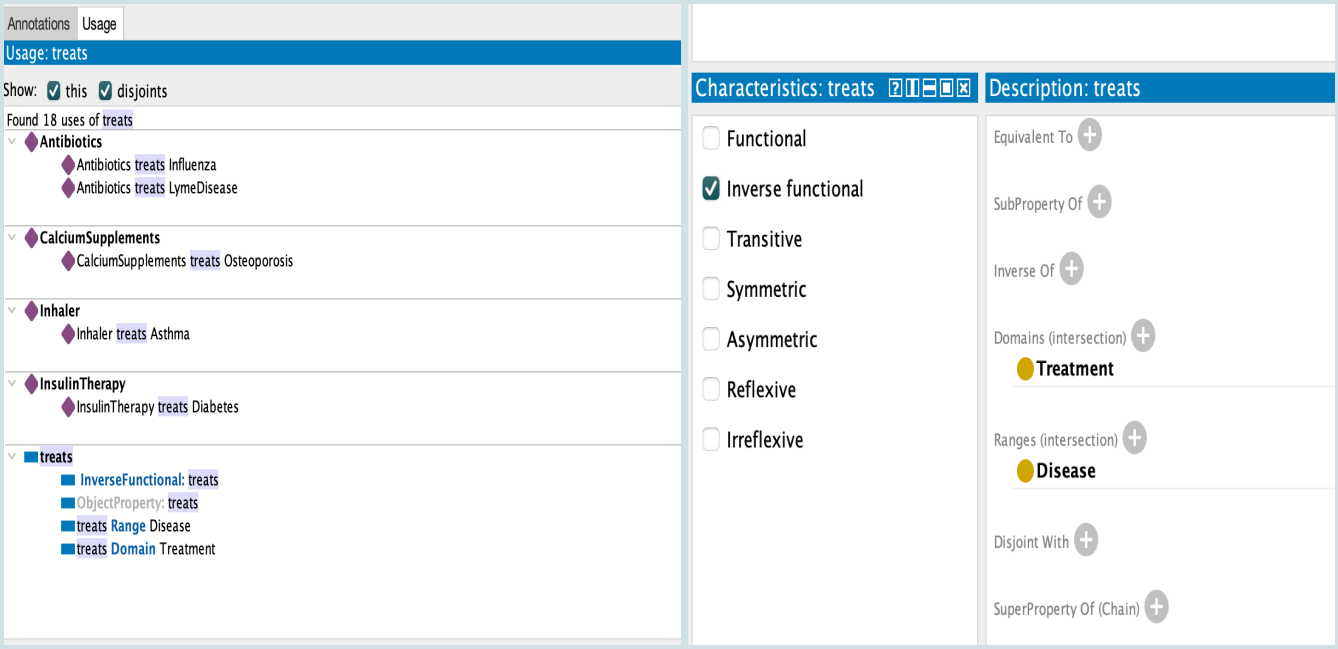
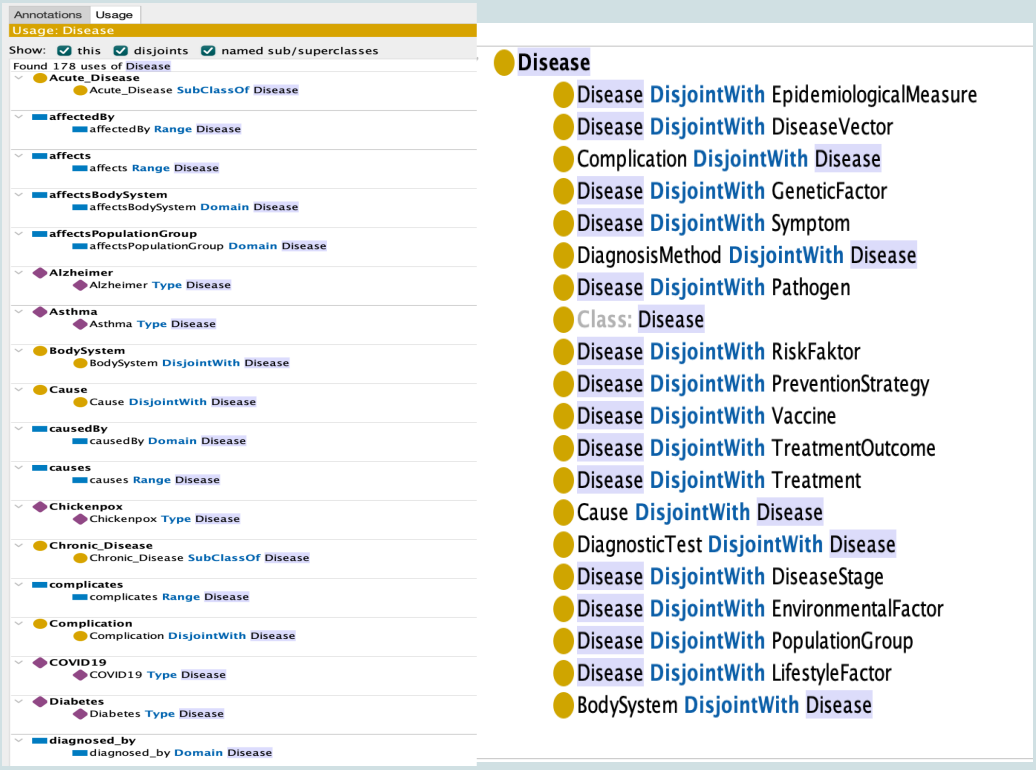


Figure 7.



Figure(8)

4. Implementation

4.1. Identify the Domain, Range, and Competency Questions

I defined the structure of the knowledge domain of ontology which is for educational support.

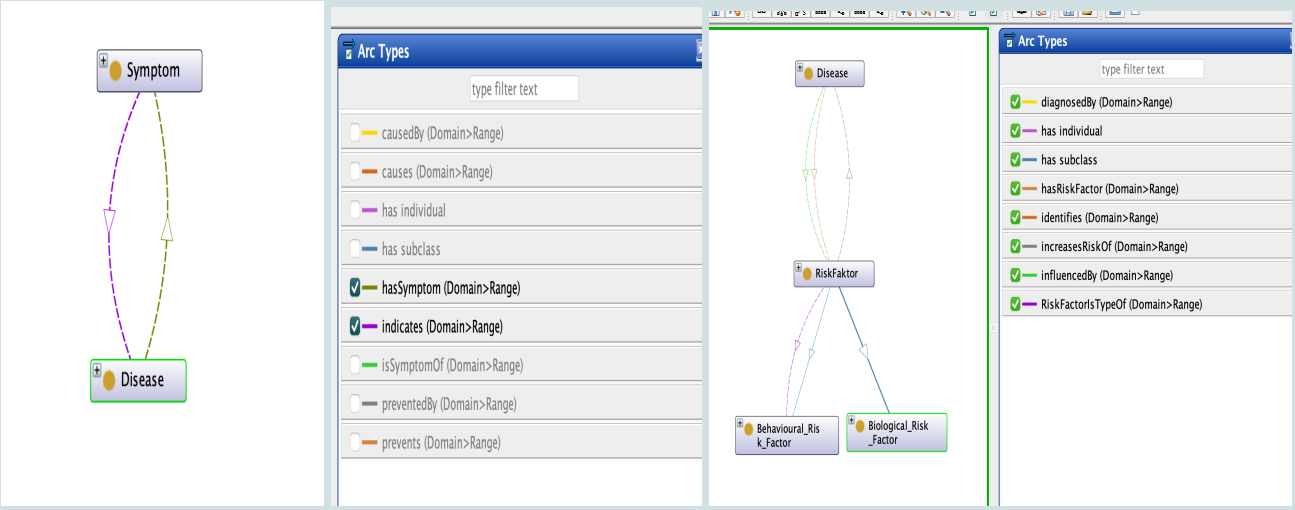
I defined the key concepts such as Disease, Symptom, Treatment, RiskFactor, … ect, and relations(predicate) between them like a Disease hasSymptom or indicates(Figure 9).

The scope of this ontology is identified, prevent and follow-up of chronic disease.

There are five competency questions that the ontology must answer.

These kinds of questions, apart from proving the ontology, also perform for the evaluation of the model.

(Figure 9)

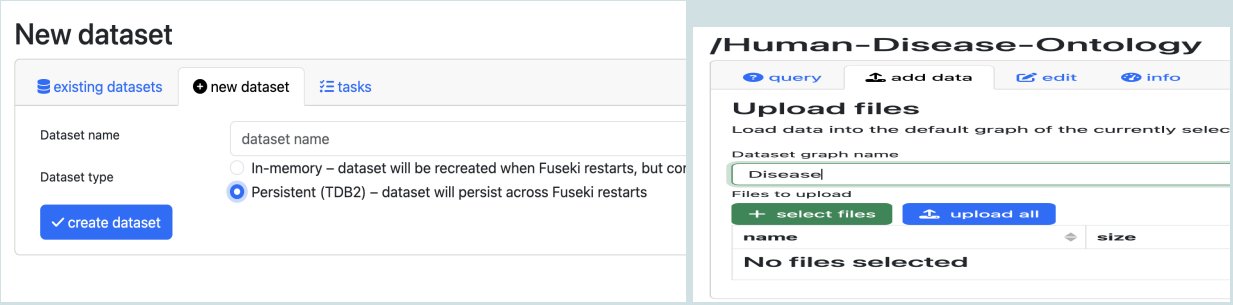


Figure(10)

4.2. The competency questions are:

1- What are the names and counts of symptoms for each disease which has more than 3 symptoms along with the names of those symptoms?
2- What are the names of the symptoms, diagnostic tests and treatments associated with LymeDisease?
3- list of the diseases and their definitions to handle diseases without a definition provided?
4- What are the names and counts of pathogens in each pathogen family along with a list of the names of these pathogens and associated with disease names?
5- What are the names and prevalence rates of risk factors for diabetes that the prevalence is more than > 15 percent with the names and efficiencies of prevention strategies used against diabetes?

4.3. Setup SPARQL Endpoint: I used a server such as Apache Jena Fuseki that is a SPARQL server, to host the SPARQL. I have loaded the RDF ontology into Fuseki and then allowed me to execute SPARQL queries from the data.



Figure(11)

4.4. Queries: With ensuring that the server is running and the RDF ontology loaded, then can run the SPARQL queries to get information from the RDF ontology[18].

https://www.w3.org/TR/turtle/

4.5. Prefixes: to execute the RDF queries in Apache Jena Fuseki environment , require to have these prefixes.

1. RDF, PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>: is the prefix for RDF syntax namespace. It describes the basic structure of my data, such as Disease, Symptom and Treatment.
2. RDFs, PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>: is for RDF schema and a semantic extension of RDF.
3. XSD, PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>: is for XML schema definition, to define data types in RDF like strings, dates, or decimals.
4. hd, PREFIX hd:<http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>: is a custom prefix which stands for the specific Human Disease Ontology namespace.

4.6. **Natural Language Processing, Parsing queries:**

This python script used spaCy library and a natural language processing library to retrieve information from an ontology about the Human_Disease by interpreting user queries and formulated a SPARQL query Figure(12). Explain each part individually.

```
In [41]: from SPARQLWrapper import SPARQLWrapper, JSON
import spacy
from spacy.matcher import Matcher

# load the spaCy model for english language
nlp = spacy.load("en_core_web_sm")

# initialize a Matcher with the shared vocab
matcher = Matcher(nlp.vocab)


# define patterns for the Matcher
patternsOntology = {
    "risk factor": [[{"LOWER": "risk"}, {"LOWER": "factor"}, {"ENT_TYPE": "DISEASE", "OP": "?"}]],
    "diagnosis method": [[{"LOWER": "diagnosis"}, {"LOWER": "method"}, {"ENT_TYPE": "DISEASE", "OP": "?"}]],
    "disease": [[{"LOWER": "disease"}]],
    "symptom": [[{"LOWER": "symptom"}]],
    "treatment": [[{"LOWER": "treatment"}]],
    "cause": [[{"LOWER": "cause"}]]
}
for key, pattern in patternsOntology.items():
    matcher.add(key, pattern)

# process text with spaCy and Matcher
def processText(text):
    doc = nlp(text)
    matches = matcher(doc)

    foundTerms = set()
    specificDisease = None
    for matchId, start, end in matches:
        span = doc[start:end]
        # check if the span contains a disease entity
        for token in span:
            if token.ent_type_ == "DISEASE":
                specificDisease = token.text
                break
        foundTerms.add(span.text.lower())

    return foundTerms, specificDisease


# create a dynamic SPARQL query
def createSparql_query(foundTerms, specificDisease):
    baseQuery = """
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    """

    termMapping = {
        "disease": "hd:Disease",
        "symptom": "hd:Symptom",
        "treatment": "hd:Treatment",
        "cause": "hd:Cause",
        "risk factor": "hd:RiskFaktor",
        "diagnosis method": "hd:DiagnosisMethod"
    }

    queryConditions = []
    for term in foundTerms:
        if term in termMapping:
            queryConditions.append(f"?item rdf:type {termMapping[term]}")

    if specificDisease:
        queryConditions.append(f'?item hd:isRelatedTo "{specificDisease}"')

    if not queryConditions:  # default condition if no specific terms are identified
        queryConditions.append("?item rdf:type hd:Disease")

    whereClause = " . ".join(queryConditions)
    query = f"{baseQuery}SELECT DISTINCT ?item WHERE {{ {whereClause} }} LIMIT 10"

    return query

# run SPARQL query and return results
def executeSparql_query(query):
    sparql = SPARQLWrapper("http://localhost:3030/Human-Disease-Ontology/sparql")
    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    try:
        results = sparql.query().convert()
        return results
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

# main function to handle user queries
def main():
    userQuery = input("Enter your query in natural language: ")
    foundTerms, specificDisease = processText(userQuery)
    sparqlQuery = createSparql_query(foundTerms, specificDisease)
    print(f"SPARQL query to execute:\n{sparqlQuery}")
    results = executeSparql_query(sparqlQuery)
    if results and results["results"]["bindings"]:
        print("Results found:")
        for result in results["results"]["bindings"]:
            print(result["item"]["value"])
    else:
        print("No results returned or an error occurred.")

if __name__ == "__main__":
    main()



# What are the risk factor for Asthma?
# Give me diagnosis method for Diabetes
# What are some diseases?
# disease has names?
# List all symptom.
# List diagnosis method
# Show treatment.
# Give me cause.
# What are the risk factor?
# Give me diagnosis method

Enter your query in natural language: What are some diseases?
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:Disease } LIMIT 10
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Diabetes
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Alzheimer
```

Figure(12)

1. Importing necessary libraries:

```
from SPARQLWrapper import SPARQLWrapper, JSON
import spacy
from spacy.matcher import Matcher
```

Figure(13)

SPARQLWrapper: this is for querying SPARQL endpoints from Python wrapper. It is used for databases to store the data in RDF format such as Human Disease Ontology.

Spacy and Matcher:

spaCy is used for Natural Language Processing(NLP). Matcher is one of the spaCy tools, it is used for matching phrases or word sequences such as Treatment, Symptom and Disease, and is more suited for linguistic context. It acts like a Bag of Words but more advanced than a simple Bag of Words. It allows me to detect multi-words phrases. For example, it checks each word in a sentence if it matches with a pattern I have identified, such as finding a word that is a noun followed by a verb. It is good for grammatical structure. In summary, Matcher tool performs as a filter that it is looking for the patterns which I have defined, can contain base form of words like lemmas or kind of phrase, including named entities. [9] https://spacy.io/api/matcher.

```
# load the spaCy model for english language
nlp = spacy.load("en_core_web_sm")

# initialize a Matcher with the shared vocab
matcher = Matcher(nlp.vocab)


# define patterns for the Matcher
patternsOntology = {
    "risk factor": [[{"LOWER": "risk"}, {"LOWER": "factor"}, {"ENT_TYPE": "DISEASE", "OP": "?"}]],
    "diagnosis method": [[{"LOWER": "diagnosis"}, {"LOWER": "method"}, {"ENT_TYPE": "DISEASE", "OP": "?"}]],
    "disease": [[{"LOWER": "disease"}]],
    "symptom": [[{"LOWER": "symptom"}]],
    "treatment": [[{"LOWER": "treatment"}]],
    "cause": [[{"LOWER": "cause"}]]
}
for key, pattern in patternsOntology.items():
    matcher.add(key, pattern)
```

Figure(14)

2- Implement Matcher:

First of all, I have loaded a small english model that includes data and algorithms for english text processing.

After that I have initialised Matcher with the vocabulary of the loaded spaCy model.

The reason I have chosen Matcher over regex, using Matcher for text analysing, understanding the relation between words in a phrase and combining named entity recognition[17]. https://spacy.io/api/phrasematcher, https://spacy.io/api/matcher

Then, I have defined patterns to match specific words such as ‘risk factor’ or ‘symptom’ and these patterns are added to the Matcher tool. In summary, I have setted up the rules in this code for the Matcher to identify specific phrases.

```
# process text with spaCy and Matcher
def processText(text):
    doc = nlp(text)
    matches = matcher(doc)

    foundTerms = set()
    specificDisease = None
    for matchId, start, end in matches:
        span = doc[start:end]
        # check if the span contains a disease entity
        for token in span:
            if token.ent_type_ == "DISEASE":
                specificDisease = token.text
                break
        foundTerms.add(span.text.lower())

    return foundTerms, specificDisease
```

Figure(15)

This function, processText takes text as input to find specific patterns such as disease name or symptom name and defines a specific disease as mentioned. It has used spaCy library for Natural Language Processing and used Matcher tool to find the pattern which matches in the text.

```python
# create a dynamic SPARQL query
def createSparql_query(foundTerms, specificDisease):
    baseQuery = """
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    """

    termMapping = {
        "disease": "hd:Disease",
        "symptom": "hd:Symptom",
        "treatment": "hd:Treatment",
        "cause": "hd:Cause",
        "risk factor": "hd:RiskFaktor",
        "diagnosis method": "hd:DiagnosisMethod"
    }

    queryConditions = []
    for term in foundTerms:
        if term in termMapping:
            queryConditions.append(f"?item rdf:type {termMapping[term]}")

    if specificDisease:
        queryConditions.append(f'?item hd:diagnosedBy "{specificDisease}"')

    if not queryConditions:  # default condition if no specific terms are identified
        queryConditions.append("?item rdf:type hd:Disease")

    whereClause = " . ".join(queryConditions)
    query = f"{baseQuery}SELECT DISTINCT ?item WHERE {{ {whereClause} }} LIMIT 10"

    return query
```

Figure(16)

The creatSparql_Query function has created dynamically a SPARQL query to retrieve data from Human_Disease_Ontology based on user input. termMapping is a dictionary that translates Natural Language terms into the SPARQL endpoint that can be understood like mapping disease.

```python
# run SPARQL query and return results
def executeSparql_query(query):
    sparql = SPARQLWrapper("http://localhost:3030/Human-Disease-Ontology/sparql")
    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    try:
        results = sparql.query().convert()
        return results
    except Exception as e:
        print(f"An error occurred: {e}")
        return None
```

Figure(17)

Function excuteSparql_query sends a SPARQL query to the endpoint, fetching the results in JSON format. During the process, handles any potential errors.

```python
# main function to handle user queries
def main():
    userQuery = input("Enter your query in natural language: ")
    foundTerms, specificDisease = processText(userQuery)
    sparqlQuery = createSparql_query(foundTerms, specificDisease)
    print(f"SPARQL query to execute:\n{sparqlQuery}")
    results = executeSparql_query(sparqlQuery)
    if results and results["results"]["bindings"]:
        print("Results found:")
        for result in results["results"]["bindings"]:
            print(result["item"]["value"])
    else:
        print("No results returned or an error occurred.")

if __name__ == "__main__":
    main()
```

Figure(18)

The centre of this script of code is the main function. It takes a natural language (NLP) query from the user input, by processing it, extracting the related terms and diseases, then creates and runs a SPARQL query based on this information, and at the end displays the results.

The question I have provided for Natural Language query:

1. Give me a diagnosis method?
2. What are the risk factors?
3. Give me causes ?
4. Show treatment?
5. List all symptoms?
6. What are some diseases?

```
# What are the risk factor for Asthma?
# Give me diagnosis method for Diabetes
# What are some diseases?
# disease has names?
# List all symptom.
# List diagnosis method
# Show treatment.
# Give me cause.
# What are the risk factor?
# Give me diagnosis method
```

Figure(19)

Results of this queries:

```
# Give me diagnosis method

Enter your query in natural language: Give me diagnosis method
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:DiagnosisMethod } LIMIT 10
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#CognitiveAssessment
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BloodGlucoseTest
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BloodSugarTest
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BloodTest
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BoneDensityScan
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BoneDensityTest
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#EyeExam
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Spirometry
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#TuberculinSkinTest
```

Figure(20)

Results of the query is a list of diagnosis methods.

```
Enter your query in natural language: What are the risk factor?
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:RiskFaktor } LIMIT 10
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Aging
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#AirPollution
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Allergies
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#GeneticPredisposition
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#CalciumDeficiency
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#DrinkingAlcohol
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#FamilyHistoryofDiabetes
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Obesity
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#FamilyHistoryofGlaucoma
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#WeakenedImmuneSystem
```

Figure(21)

Results of the query is a list of risk factors.

```
Enter your query in natural language: Give me cause
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:Cause } LIMIT 10
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#InsulinResistance
```

Figure(22)

Results of the query is a list of causes.

```
Enter your query in natural language: Show treatment
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:Treatment } LIMIT 10
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Antibiotics
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Acetaminophen
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#CognitiveTherapy
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#AntimalarialDrugs
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Antipyretics
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#AntituberculosisAntibiotics
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Inhaler
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BoneDensityMedication
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#CalciumSupplements
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Rest
```

Figure(23)

Results of the query is a list of treatments.

```
Enter your query in natural language: List all symptom
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:Symptom } LIMIT 10
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#MemoryLoss
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Cough
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#ShortnessofBreath
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Wheezing
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BlurryVision
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#BodyAches
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Chills
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Fatigue
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Fever
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Headache
```

Figure(24)

Results of the query is a list of symptoms.

```
Enter your query in natural language: What are some diseases?
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:Disease } LIMIT 10
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Diabetes
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Alzheimer
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Osteoporosis
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Asthma
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Influenza
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#LymeDisease
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#COVID19
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Heart
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Chickenpox
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Malaria
```

Figure(25)

Results of the query is a list of disease names.

```
# What are the risk factor for Asthma?
# Give me diagnosis method for Diabetes
# What are some diseases?
# disease has names?
# List all symptom.
# List diagnosis method
# Show treatment.
# Give me cause.
# What are the risk factor?
# Give me diagnosis method
```

```
Enter your query in natural language: What are some diseases?
userQuery:  What are some diseases?
foundTerms : set()
whereClause:  ?item rdf:type hd:Disease
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:Disease } LIMIT 10
results:  {'head': {'vars': ['item']}, 'results': {'bindings': [{'item': {'type': 'uri', 'value': 'http://www.seman
ticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Diabetes'}}, {'item': {'type': 'uri', 'value': 'htt
p://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Alzheimer'}}, {'item': {'type': 'uri',
'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Osteoporosis'}}, {'item':
{'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Asthma'}},
{'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#I
nfluenza'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disea
se_ontology#LymeDisease'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/202
3/9/Human_Disease_ontology#COVID19'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ont
ologies/2023/9/Human_Disease_ontology#Heart'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/saras
hahin/ontologies/2023/9/Human_Disease_ontology#Chickenpox'}}, {'item': {'type': 'uri', 'value': 'http://www.semanti
cweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Malaria'}}]}}
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Diabetes
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Alzheimer
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Osteoporosis
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Asthma
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Influenza
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#LymeDisease
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#COVID19
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Heart
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Chickenpox
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Malaria
```

Figure(32)

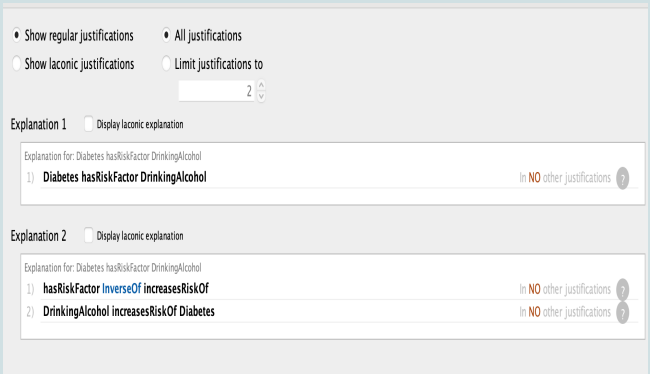
More detailed.

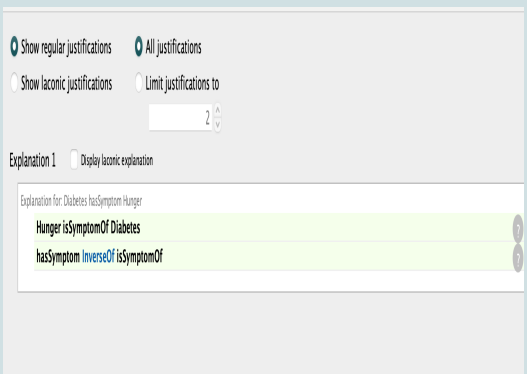
5. Evaluation and Use

5.1. Reasoning Task

First, I have validated the ontology model by using an automated reasoning process which was installed in Protege [10].

The reasoner used the ontology's class, object property, data property hierarchies, class assertions, and object property assertions to find all inferences that help to validate the ontology's structure and content.





Figure(26)

5.2. Queries: This section is about the SPARQL queries utilised by the application that validate my ontology [16]. Questions have been mentioned in section 4.2.

https://www.w3.org/TR/2012/PR-sparql11-query-20121108/,
https://www.openresearch.org/wiki/Sparql_endpoint/Examples

Q1:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>

SELECT ?diseaseName (GROUP_CONCAT(DISTINCT ?symptomName; separator=", ") AS ?symptomNames) (COUNT(DISTINCT ?symptom) AS ?symptomCount)
WHERE {
  ?disease rdf:type hd:Disease ;
           hd:hasSymptom ?symptom ;
           hd:DiseaseshasName ?diseaseName .
  ?symptom hd:SymptomhasName ?symptomName .
}
GROUP BY ?diseaseName
HAVING (COUNT(DISTINCT ?symptom) > 3)
```

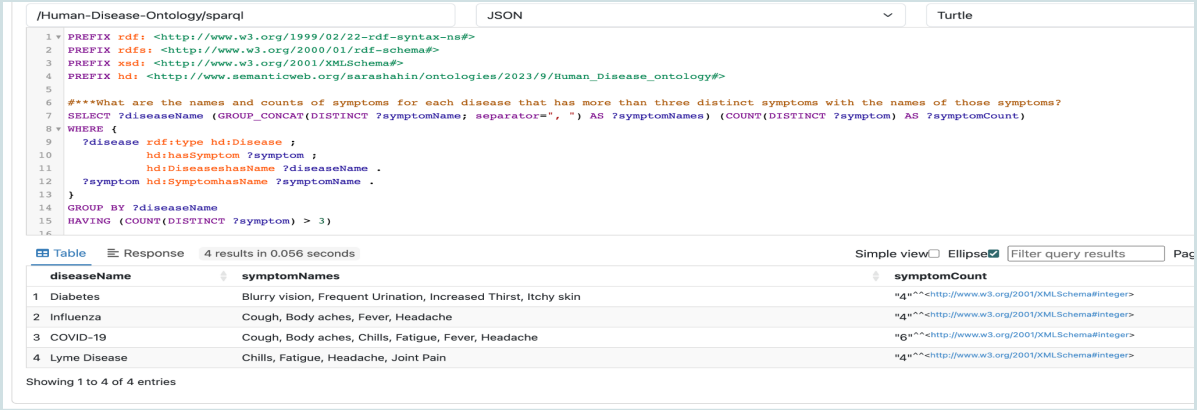


Figure 27. is designed to have a list of each disease that has more than 3 distinct symptoms.
The results show that the disease's name(groupby), a concatenated string of all the distinct symptom names separated by commas, and total count for each symptom.

Q2:

```
SELECT (STR(?diseaseLabel) AS ?diseaseName)
      (GROUP_CONCAT(DISTINCT ?symptomName; separator=", ") AS ?symptomNames)
      (GROUP_CONCAT(DISTINCT ?treatmentName; separator=", ") AS ?treatmentNames)
      (GROUP_CONCAT(DISTINCT ?diagnosticTestName; separator=", ") AS
?diagnosticTestNames)
WHERE {
  ?disease hd:DiseaseshasName ?diseaseLabel .
  FILTER (STR(?diseaseLabel) = "Lyme Disease")
  ?disease hd:hasSymptom ?symptom .
  ?symptom hd:SymptomhasName ?symptomName .
  ?disease hd:hasTreatment ?treatment .
  ?treatment hd:TreatmenthasName ?treatmentName .
  ?disease hd:diagnosed_by ?diagnosticTest .
  ?diagnosticTest hd:DiagnosticTesthasName ?diagnosticTestName .
}
GROUP BY ?diseaseLabel
```

Figure 28. This query returns the name of LymeDisease along with a concatenated string of all related to symptoms, diagnostic test and another concatenated string related treatments.

Q3:

```
SELECT ?diseaseName (COALESCE(?definition, "Definition not provided") AS ?diseaseDefinition)
WHERE {
  ?disease hd:DiseaseshasName ?diseaseName .
```

```
  OPTIONAL { ?disease hd:DiseaseshasDefinition ?definition . }
}
LIMIT 5
```

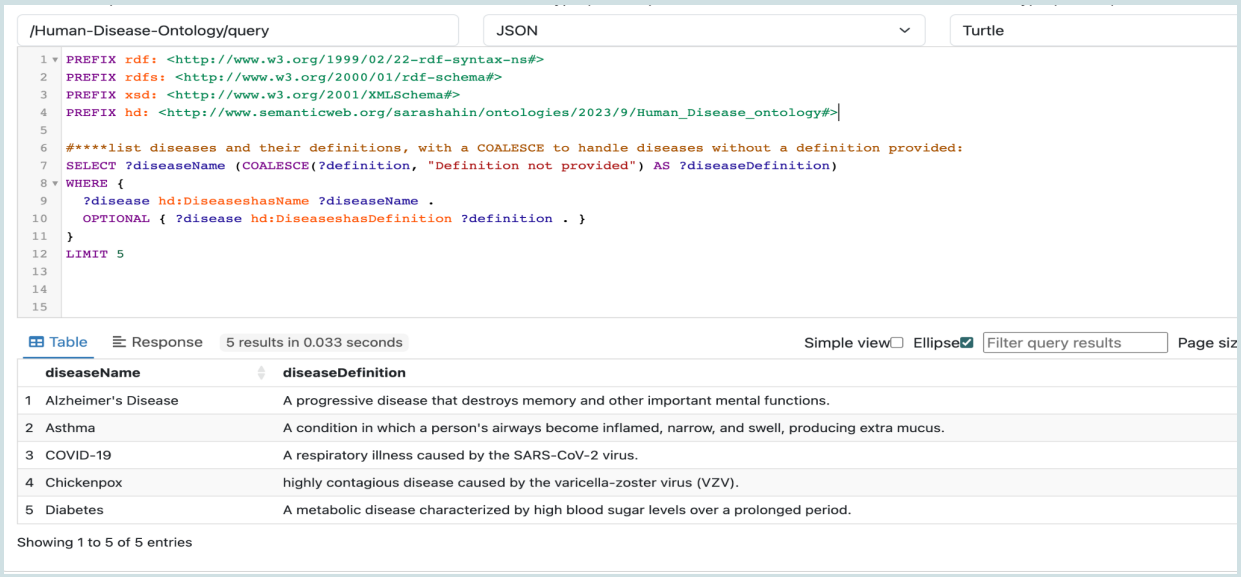


Figure 29. This SPARQL query retrieves the names of diseases and their definitions from RDF data.

Q4:

```
SELECT
  ?pathogenFamily
  (GROUP_CONCAT(DISTINCT ?pathogenName; separator=", ") AS ?pathogenNames)
  (GROUP_CONCAT(DISTINCT ?diseaseName; separator=", ") AS ?diseaseNames)
  (COUNT(?pathogen) AS ?pathogenCount)
WHERE {
  ?pathogen hd:PathogenhasFamily ?pathogenFamily ;
        hd:PathogenhasName ?pathogenName.
  ?pathogen hd:causes ?disease.
  ?disease hd:DiseaseshasName ?diseaseName.
}
GROUP BY ?pathogenFamily
```

/Human-Disease-Ontology/ JSON Turtle

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>

#***What are the names and counts of pathogens within each pathogen family with a list of the names of these pathogens and disease names?
SELECT
  ?pathogenFamily
  (GROUP_CONCAT(DISTINCT ?pathogenName; separator=", ") AS ?pathogenNames)
  (GROUP_CONCAT(DISTINCT ?diseaseName; separator=", ") AS ?diseaseNames)
  (COUNT(?pathogen) AS ?pathogenCount)
WHERE {
  ?pathogen hd:PathogenhasFamily ?pathogenFamily ;
            hd:PathogenhasName ?pathogenName.
  ?pathogen hd:causes ?disease.
  ?disease hd:DiseaseshasName ?diseaseName.
}
GROUP BY ?pathogenFamily
```

Table Response 4 results in 0.027 seconds Simple view Ellipse Filter query results Page size: 50

	pathogenFamily	pathogenNames	diseaseNames	pathogenCount
1	Mycobacteriaceae	Mycobacterium tuberculosis	Tuberculosis	"1"^^<http://www.w3.org/2001/XMLSchema#integer>
2	Coronaviridae	Severe acute respiratory syndrome coronavirus 2	COVID-19	"1"^^<http://www.w3.org/2001/XMLSchema#integer>
3	Orthomyxoviridae	Orthomyxoviridae	Influenza	"1"^^<http://www.w3.org/2001/XMLSchema#integer>
4	Spirochaetaceae	Borrelia	Lyme Disease	"1"^^<http://www.w3.org/2001/XMLSchema#integer>

Showing 1 to 4 of 4 entries

Figure 30.This query retrieves a list of pathogen families with the names of the pathogens in each family, disease caused by these pathogens and the total count of unique pathogens in each family.

Q5:

```
SELECT ?riskFactorName ?prevalenceInPercent ?preventionStrategyName ?efficiency
WHERE {
  ?diabetes rdf:type hd:Disease ;
        hd:DiseaseshasName "Diabetes" ;
        hd:hasRiskFactor ?riskFactor.
  ?riskFactor hd:RiskFactorhasName ?riskFactorName ;
         hd:RiskFactorhasPrevalenceInPopulation ?prevalenceInPercent.
  ?preventionStrategy hd:prevents ?diabetes;
              hd:PreventionStrategyhasName ?preventionStrategyName;
              hd:PreventionStrategyhasEfficiency ?efficiency.
  FILTER(?prevalenceInPercent > 15)
}
```

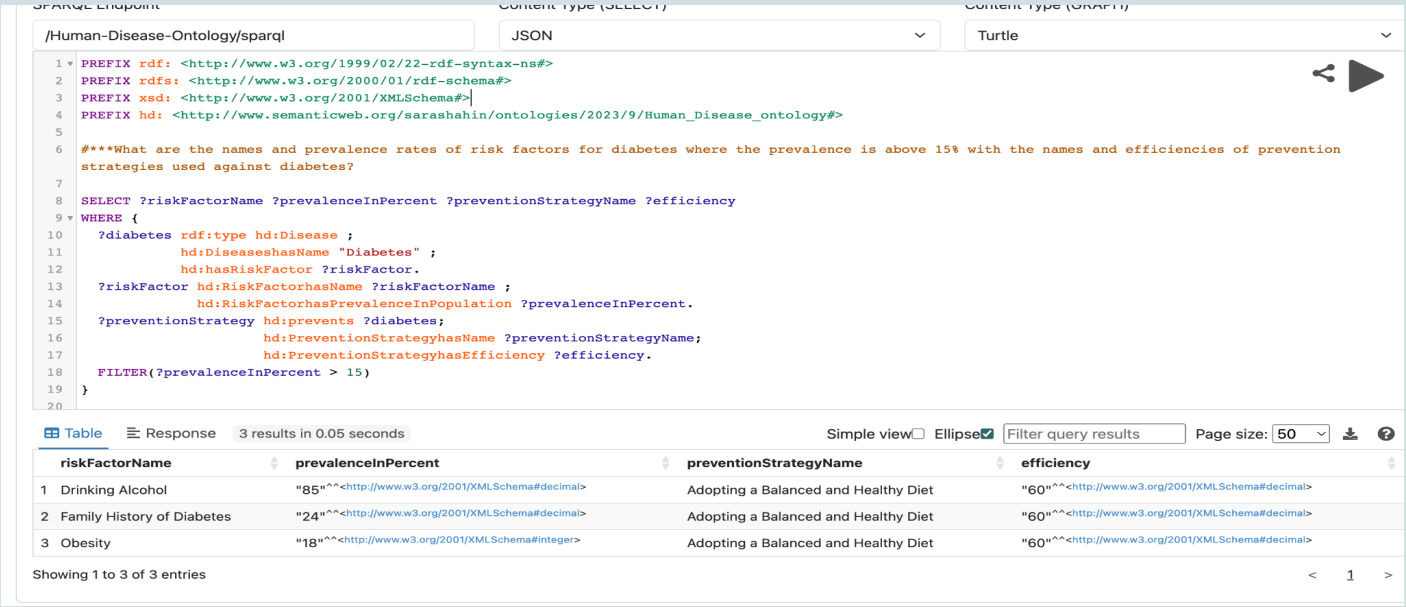


Figure 31. This query is important risk factors for diabetes, and also the different strategies used to prevent diabetes, including how effective these strategies are and used FILTER for prevalence greater than 15.
https://sparqlwrapper.readthedocs.io/en/latest/main.html

6. Research Mapping/ Conclusion

Refer to Related works section 2.

This report is about Human Disease Ontology. I have started with my aim of understanding OWL by expanding a prototype application in the aim and concept section to create a searchable ontology for human disease.

My project represents an ontology which categorised Human Disease into chronic respiratory, infectious, and acute diseases. The Human Disease Ontology focuses on identifying lifestyle-related issues, like smoking, drinking alcohol, and obesity as early as possible to potentially prevent chronic disease as the cause of death. Therefore, based on behaviour risk factors, and leveraging an application that can make recommendations such as diet changing or quitting smoking.

The project has been using the protege software tool to identify relationships and class properties comprehensible and effectively to make the disease ontology easy to navigate.

The ontology can answer complicated questions about disease, symptoms, treatments, and risk factors by using SPARQL queries and its use as an educational tool and for medical research.

Regardless of small scope, the project has successfully illustrated how semantic data and ontology can be powerful tools to manage large databases in medical research.

6.1. Critical Reflection

As my project indicates a well structured approach to build a disease ontology, with significant potential impact on healthcare and medical research. Also by choosing good definitions of properties and classes and the combination of SPARQL queries, that indicates a robust system.
Therefore, the project considers the complication of diseases and their complexity of factorial nature, keeping updates and validation against current medical standards would be crucial.

The aim of this project is to make connections in different biomedical data points, and success will be dependent on its ability to remain up to date and medically accurate.

6.2. Future Scope

Human Disease Ontology, key areas for developments:

- ❖ Expanding ontology with new symptoms, risk factors and disease while medical research develops.
- ❖ Build personalised healthcare recommendations.
- ❖ Create a platform to contribute and use the ontology for researchers.

6.3. Compare with alternative approaches[15]

- ❖ Standard databases like SQL might lack the semantic relationships that ontology provides while my ontology approach has complex relationships between disease, symptoms, and treatments.
- ❖ Statistical models are great for defying trends and make a prediction based on data but might be crucial for understanding disease ontology.
- ❖ Rule based models might be strong for specific tasks but have less flexibility than an ontology model does.

6.4. Discussion of Limitations

- ❖ Ontologies can grow complex, fast, and hard to maintain as new data is continuously integrated. Make sure of the accuracy of disease ontology.

7 References

1. WHO. 2021. Strengthening NCD service delivery through UHC benefit package: technical meeting report, Geneva, Switzerland, 14-15 July 2020.

2. James C. Griffiths, Jan De Vries, Michael I. McBurney, Suzan Wopereis, Samet Serttas, and Daniel S. Marsman. 2020. Measuring health promotion: translating science into policy.European Journal of Nutrition59, 2 (01 Sep 2020).

3. Larentis, A.V.; Neto, E.G.d.A.; Barbosa, J.L.V.; Barbosa, D.N.F.; Leithardt, V.R.Q.; Correia, S.D. Ontology-Based Reasoning for Educational Assistance in Noncommunicable Chronic Diseases. Computers 2021, 10, 128.

4. Larentis, A.V.; Neto, E.G.d.A.; Barbosa, J.L.V.; Barbosa, D.N.F.; Leithardt, V.R.Q.; Correia, S.D. Ontology-Based Reasoning for Educational Assistance in Noncommunicable Chronic Diseases. Computers 2021, 10, 128.

5. 12Blomqvist, E. The Use of Semantic Web Technologies for Decision Support—A Survey. Semant. Web 2014, 5, 177–201.

6. Logical Design, The Morgan Kaufmann Series in Data Management Systems, 2011, Pages 85-108.

7. Malta, D.C.; Gomes, C.S.; Barros, M.; Lima, M.G. Noncommunicable diseases and changes in lifestyles during the COVID-19 pandemic in Brazil. Rev. Bras. Epidemiol. 2021, 24, e210009.

8. Schriml LM, Munro JB, Schor M, et al. The Human Disease Ontology 2022 update. Nucleic Acids Research. 2022 Jan;50(D1):D1255-D1261. DOI: 10.1093/nar/gkab1063. PMID: 34755882; PMCID: PMC8728220.

9. Matcher · spaCy API Documentation

10. Matthew Horridge, Holger Knublauch, Natasha Noy, Alan Rector et al.

https://protege.stanford.edu/conference/2007/slides/owlTutorial-reasoning_dameron.pdf

11. International Journal of Computer Applications (0975 – 8887) Volume 56– No.6, October 2012.

12. Schriml LM, Munro JB, Schor M, Olley D, McCracken C, Felix V, Baron JA, Jackson R, Bello SM, Bearer C, Lichenstein R, Bisordi K, Dialo NC, Giglio M, Greene C. The Human Disease Ontology 2022 update. Nucleic Acids Res. 2022 Jan 7;50(D1):D1255-D1261. doi: 10.1093/nar/gkab1063. PMID: 34755882; PMCID: PMC8728220.

13. M. Hadzic and E. Chang, "Ontology-Based Support for Human Disease Study," Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 2005, pp. 143a-143a, doi: 10.1109/HICSS.2005.472.

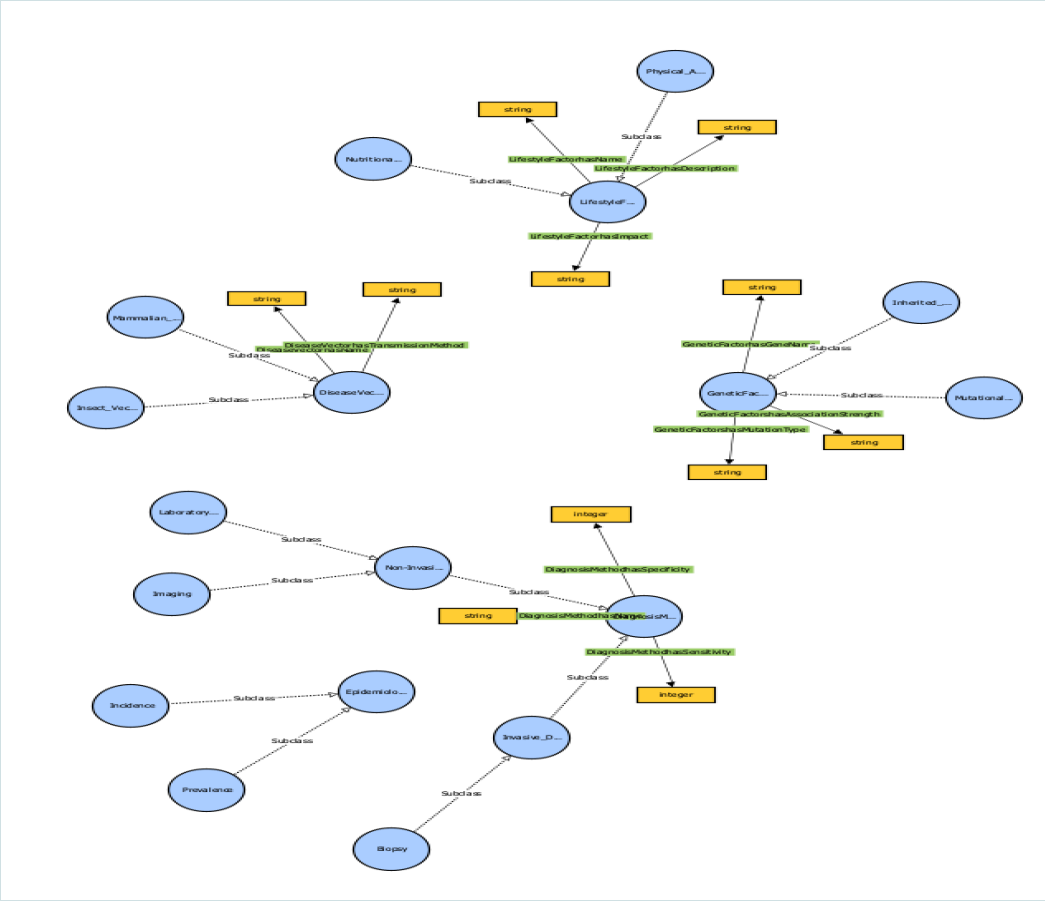
14. Lucas Pfeiffer Salomão Dias, Henrique Damasceno Vianna, Wesllei Heckler, and Jorge Luis Victória Barbosa. 2023. Ontology-Based Reasoning to Classify Behaviors Associated with Chronic Disease Risk Factors. InXIX Brazilian Symposium on Information Systems (SBSI '23), May 29–June 01, 2023, Maceió.

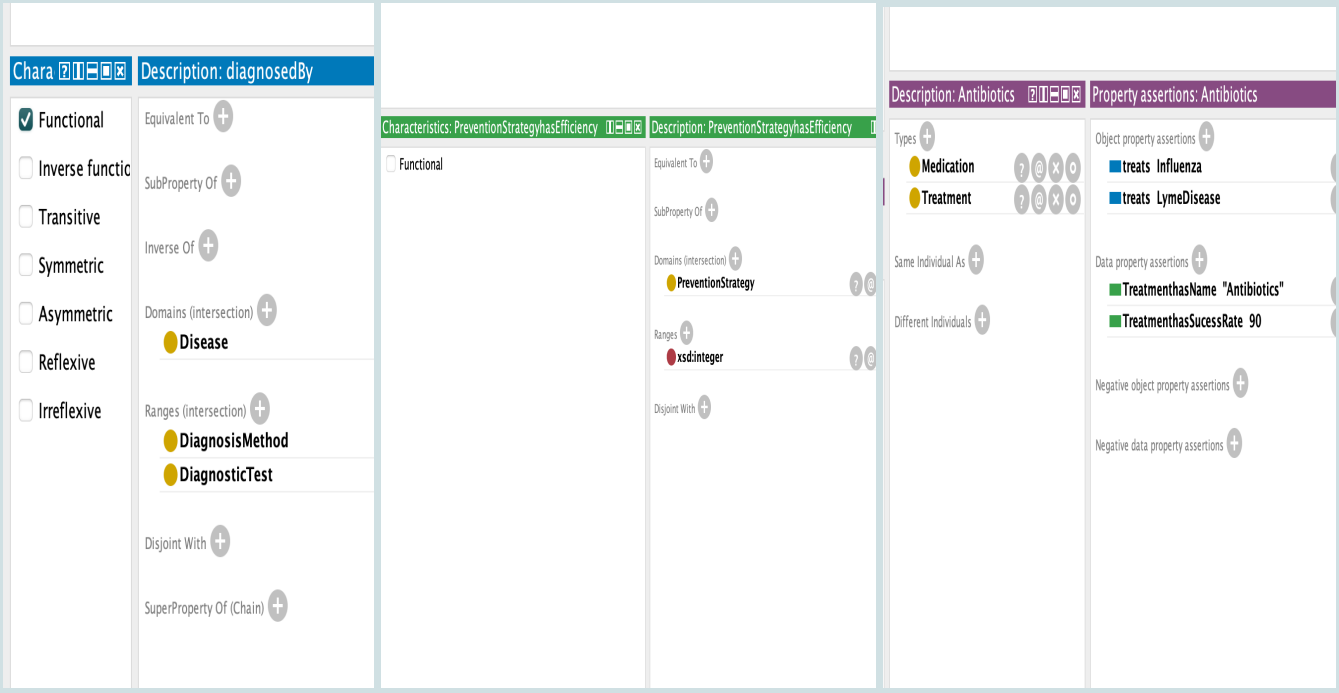
15. Kamran Munir, M. Sheraz Anjum, The use of ontologies for effective knowledge modelling and information retrieval, Applied Computing and Informatics, Volume 14, Issue 2, 2018, Pages 116-126, ISSN 2210-8327, https://doi.org/10.1016/j.aci.2017.07.003.

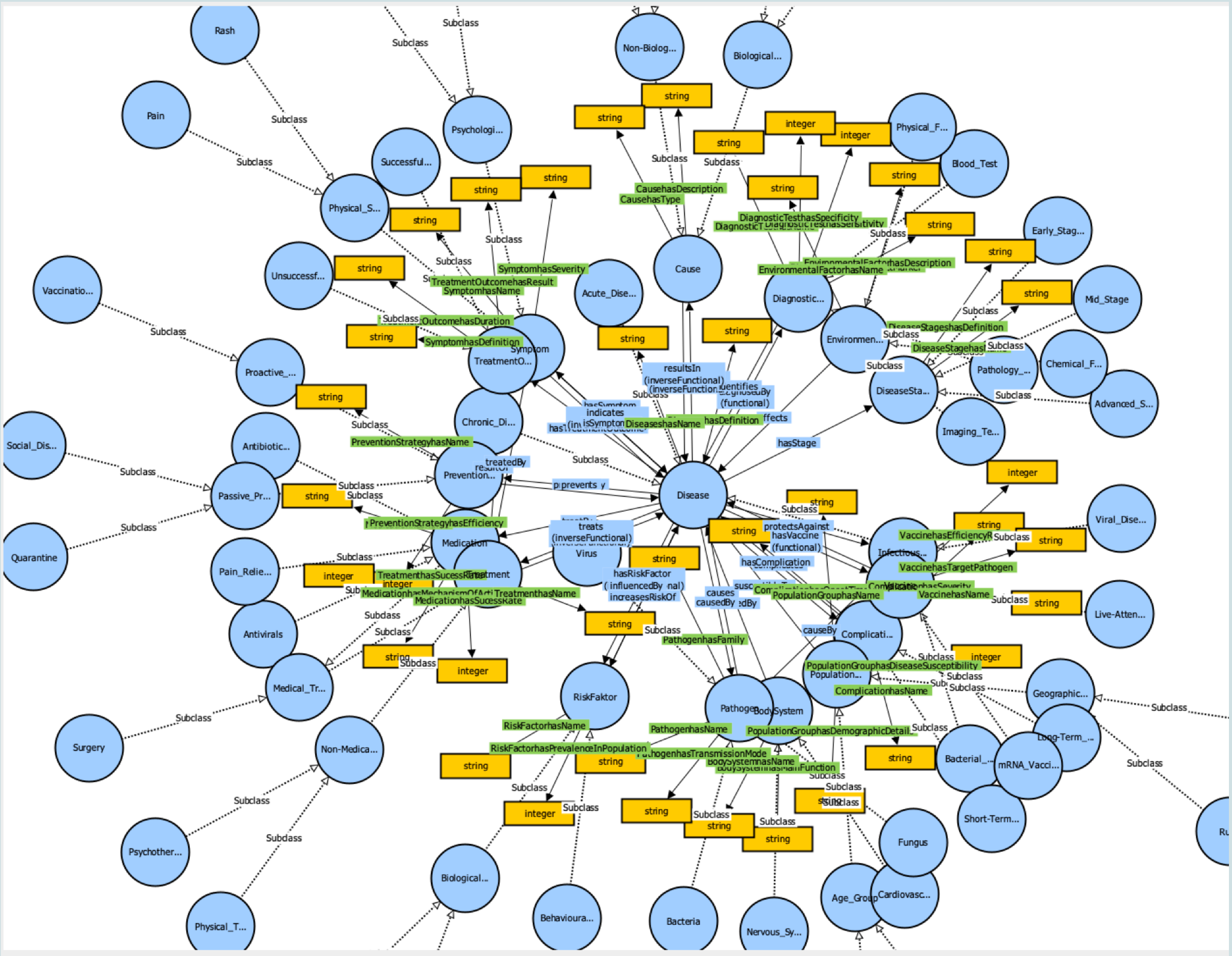
16. https://www.w3.org/TR/2012/PR-sparql11-query-20121108/, https://www.openresearch.org/wiki/Sparql_endpoint/Examples

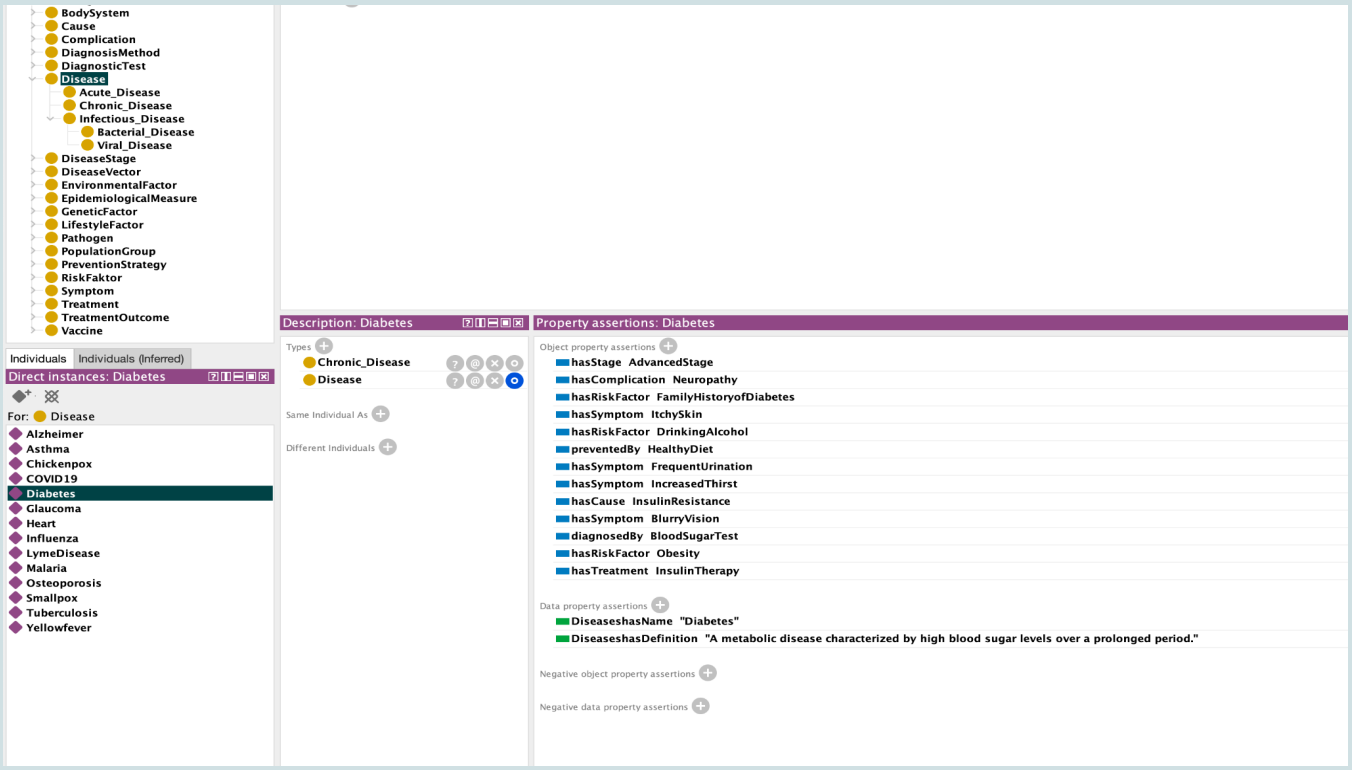
17. https://spacy.io/api/phrasematcher, https://spacy.io/api/matcher
18. https://www.w3.org/TR/turtle/
19. https://bioportal.bioontology.org/ontologies/DOID/?p=summary

8. Appendix


Physical_A...
string
string
Subclass
Nutritiona...
LifestyleFactorhasName
LifestyleFactorhasDescription
Subclass
LifestyleF...
LifestyleFactorhasImpact
string
string
string
Mammalian_...
DiseaseVectorhasTransmissionMethod
Subclass
DiseaseVec...
Insect_Vec...
Subclass
string
Inherited_...
GeneticFactorhasGeneName
Subclass
GeneticFac...
Subclass
Mutational...
GeneticFactorhasAssociationStrength
GeneticFactorhasMutationType
string
string
Laboratory...
Subclass
Non-Invasi...
Imaging
Subclass
integer
DiagnosisMethodhasSpecificity
Subclass
string
DiagnosisMethodhasRange
DiagnosisM...
DiagnosisMethodhasSensitivity
integer
Subclass
Incidence
Subclass
Epidemiolo...
Subclass
Prevalence
Invasive_D...
Subclass
Biopsy


Chara
Description: diagnosedBy
Functional
Inverse functic
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Disease
Ranges (intersection)
DiagnosisMethod
DiagnosticTest
Disjoint With
SuperProperty Of (Chain)
Characteristics: PreventionStrategyhasEfficiency
Functional
Description: PreventionStrategyhasEfficiency
Equivalent To
SubProperty Of
Domains (intersection)
PreventionStrategy
Ranges
xsd:integer
Disjoint With
Description: Antibiotics
Types
Medication
Treatment
Same Individual As
Different Individuals
Property assertions: Antibiotics
Object property assertions
treats Influenza
treats LymeDisease
Data property assertions
TreatmenthasName "Antibiotics"
TreatmenthasSucessRate 90
Negative object property assertions
Negative data property assertions


Rash
Pain
Psychologi...
Non-Biolog...
Biological...
Physical_F...
Blood_Test
Successful...
Physical_S...
Unsuccessf...
Vaccinatio...
Cause
Acute_Dise...
Diagnostic...
Early_Stag...
Mid_Stage
Environmen...
Chemical_F...
Pathology_...
Advanced_S...
DiseaseSta...
Imaging_Te...
Proactive_...
Symptom
TreatmentO...
Chronic_Di...
Social_Dis...
Antibiotic...
Passive_Pr...
Prevention...
Disease
Quarantine
Medication
Treatment
Virus
Pain_Relie...
Antivirals
Medical_Tr...
Infectious...
Viral_Dise...
Live-Atten...
Complicati...
Population...
Geographic...
Long-Term_...
mRNA_Vacci...
Bacterial_...
Short-Term...
Surgery
Non-Medica...
RiskFaktor
Pathogen
BodySystem
Fungus
Psychother...
Biological...
Behavioura...
Bacteria
Nervous_Sy...
Age_Group
Cardiovasc...
Physical_T...
CausehasDescription
CausehasType
DiagnosticTesthasSpecificity
EnvironmentalFactorhasDescription
EnvironmentalFactorhasName
SymptomhasSeverity
TreatmentOutcomehasResult
SymptomhasName
OutcomehasDuration
SymptomhasDefinition
DiseaseStageshasDefinition
PreventionStrategyhasName
PreventionStrategyhasEfficiency
DiseasehasName
hasDefinition
hasStage
resultsIn
(inverseFunctional)
indicates
treatedBy
prevents
treats
protectsAgainst
hasVaccine
(functional)
hasComplication
hasRiskFactor
increasesRiskOf
causes
causedBy
VaccinehasTargetPathogen
VaccinehasName
PopulationGrouphasName
PopulationGrouphasDiseaseSusceptibility
ComplicationhasName
PopulationGrouphasDemographicDetail
PathogenhasFamily
PathogenhasName
PathogenhasTransmissionMode
BodySystemhasName
TreatmenthasName
MedicationhasSucessRate
RiskFactorhasName
RiskFactorhasPrevalenceInPopulation
string
integer
Subclass




```
# What are the risk factor for Asthma?
# Give me diagnosis method for Diabetes
# What are some diseases?
# disease has names?
# List all symptom.
# List diagnosis method
# Show treatment.
# Give me cause.
# What are the risk factor?
# Give me diagnosis method
```

```
Enter your query in natural language: What are the risk factor?
userQuery:  What are the risk factor?
matchId: 13289220008681790430
span:  risk factor
token: risk
token: factor
foundTerms : {'risk factor'}
term: risk factor
whereClause:  ?item rdf:type hd:RiskFaktor
SPARQL query to execute:

    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX hd: <http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#>
    SELECT DISTINCT ?item WHERE { ?item rdf:type hd:RiskFaktor } LIMIT 10
results:  {'head': {'vars': ['item']}, 'results': {'bindings': [{'item': {'type': 'uri', 'value': 'http://www.seman
ticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Aging'}}, {'item': {'type': 'uri', 'value': 'http://
www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#AirPollution'}}, {'item': {'type': 'uri',
'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Allergies'}}, {'item': {'t
ype': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#GeneticPredis
position'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disea
se_ontology#CalciumDeficiency'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologi
es/2023/9/Human_Disease_ontology#DrinkingAlcohol'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/
sarashahin/ontologies/2023/9/Human_Disease_ontology#FamilyHistoryofDiabetes'}}, {'item': {'type': 'uri', 'value':
'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Obesity'}}, {'item': {'type': 'ur
i', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#FamilyHistoryofGlaucom
a'}}, {'item': {'type': 'uri', 'value': 'http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_onto
logy#WeakenedImmuneSystem'}}]}}
Results found:
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Aging
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#AirPollution
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Allergies
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#GeneticPredisposition
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#CalciumDeficiency
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#DrinkingAlcohol
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#FamilyHistoryofDiabetes
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#Obesity
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#FamilyHistoryofGlaucoma
http://www.semanticweb.org/sarashahin/ontologies/2023/9/Human_Disease_ontology#WeakenedImmuneSystem
```