Ontology Specifications to Generate Questions

Steve Wang
wngshu003@myuct.ac.za
University of Cape Town
Cape Town, South Africa

ABSTRACT

Ontology have been used across multiple fields. This paper investigates the use of ontology to generate quality questions. This project aims to define a set of specifications on an ontology for it to generate instances of different question types. In this way, higher quality, in the aspects of syntax, semantics and ontology-answerable questions can be generated. A software program was developed to generate questions from an ontology and a list of question templates. The resulting questions were evaluated for their quality and the feedback was analysed. These results show that only small portion of the generated questions were good quality. The feedback provides several reasons with regards to the poor quality of the generated questions. Some of these reasons are the lack of scenario identified to cover different cases of generation for the same template, the incorrect use of axioms when generating questions and syntax errors when defining the templates.

CCS CONCEPTS

• **Applied computing → Enterprise ontologies, taxonomies and vocabularies**; • **Computing methodologies** → *Natural language generation*;

KEYWORDS

ontology, knowledge representation, question generation, natural language generation

1 INTRODUCTION

Ontology was initially created to provide a common vocabulary at an abstract level above conceptual data models which assists in the integration of multiple data and information systems. However, over the last few decades, the use of ontology has spread into multiple fields to assist in e-learning, navigating and the understanding of scholarly documents, meta-mining of data mining experiments, structuring and reasoning over scientific contents, deep question-answering and ontology-driven conceptual data modelling. [4] Inquire Biology is an example of an intelligent textbook that makes use of ontology to support content based navigation and question answer generation [1].

Inquire Biology also make use of template based NLG to generate instances of various question types. This way of generating natural language requires a set of question templates for each question type. Questions are generated in this style by replacing the "blank spaces" in the template with the appropriate words. In this study, the "blank space" in a question template are represented by a class in the ontology and such "blank space" can be replace with any subclass of the original class. For example: The template "What does a <animal> <objectProperty>?" can generate the question "What does a lion eat?" where lion is a sub-class of animal and eat is a sub-class of the objectProperty in the underlying ontology.

In this paper, high quality questions are defined as questions that conform to the syntax and semantic rules of English and are answerable by the ontology.

Often problematic questions are generated when the ontology does not have sufficient axioms, classes or object properties that is required to generate an instance of the question type. This paper aim to solve this problem by providing a set of prerequisites for an ontology to generate questions.

A list of question types along with their templates are defined. Since ontology is class specific, as opposed to instance specific, the list is refined to a set which contains only class specific questions. This list is then fed into a program along with an ontology to generate questions. This process is then repeated, with varying characteristics and structure of the ontology each time. The resulting questions are evaluated according to the notion of quality questions defined above. From this, a set of specifications of an ontology can be analysed and concluded according to the quality of questions generated.

This paper focuses on answering this question, " What are the characteristics that are required of an ontology to generate different question types?"

2 RELATED WORKS

The foundation of this paper is based across various fields. Ontology is a major building block for this project and within the ontology, the decrption logic (DL) that it make use of is also an essential element in determining whether if a generated question is answerable or not. Protégé allow the ease to manage, edit and create ontologies used in this project. Generating questions using template based NLG is also an essential part of the project.

2.1 Ontology

Maria Keet's book, "An Introduction to Ontology Engineering", mentioned a few attempts to define that ontology is. One of which stats that ontology is equivalent to a description logic knowledge base. Essentially, ontology can be seen as classes or concepts in an subject domain that are link with each other with different relation and the underlying description logic provides more meanings and restrictions to the ontology.

Web Ontology Language (OWL) is based of DL representation formalism. It is most commonly used to format and serialise ontology into a text file. OWL was primarily designed to represent categorised objects and the relationship between these objects.

Protégé is Ontology Development Environment (ODE) which make it easy to design ontology by providing necessary tools to construct domain models and knowledge-based applications. Among all the other API-like functionalities include the function of generating serialised ontology into RDF/XML format.

2.2 Template Based Question Generation

Template based NLG is a form of NLG that maps non-linguistic words into "gaps" in a linguistic structure. Such linguistic structure have "gaps" in them are known as template. In this paper, templates of different question types are identified and the "gaps" in template are replace by the appropriate classes in an ontology.

3 DESIGN OF QUESTION TEMPLATE

The process of designing question templates were done is several iterations. "Inquire Biology" is an smart textbook that makes use of an ontology and template based NLG to generate questions relating to the content. In the first attempt to identify question templates, I have adapted question template used for "Inquire Biology" that is mentioned in the paper "Question generation from a knowledge base" [1] as follows:

- Definition Questions:What is <X>?
- Find a Value Questions (Basic Triple): What is R of <X>? What are Rs of <X>?
- Yes/No Questions: Is it true that <Y> is R of <X>? and Is it true that <Ys> are Rs of <X>?
- Identification Questions: What <X> has the following specific property/Properties? and List all <Xs> that have the following specific property/Properties?
- Relationship Questions: What is the relationship between <X> and <Y>? and What are the relationships between <X> and <Y>?
- Comparison Questions: What is the [similarity/differences] between <X> and <Y>? and What are the [similarities/differences] between <X> and <Y>?
- How Many Questions : How many <Ys> are R of <X>? and How many <Xs> have this specific property?
- Event Relations: What is R of <Event X>?
- Features/Property Questions: What is the property of <X>? and What are the properties of <X>?
- Max Count Questions: Which <X> has most <R>?

However, these templates are domain specific to the field of biology. Hence, there are not abstract enough to accommodate all types of ontologies. In the second attempt to identify the appropriate question templates that are more abstract and well rounded, I have made use of question taxonomy defined by the information system institute from the University of Southern California. The taxonomy classifies question according to the type of answer that the question is asking for [2]. Figure 8 in the appendix shows how the taxonomy classifies different question types.

Although such question taxonomy covers a large variety of question types in a structured manner, but certain questions are not answerable by the ontology. This is due to the fact that ontology are generally used for abstract and class level concept as opposed to instance level question. For example, a person instance, "Steve", is not stored as an subclass of person in the ontology. Neither is any detail such as birth date and age of Steve will be stored as an entity in the ontology. Therefore, questions such as "How old is Steve?" is not answerable in the ontology. As a result, I have remove instance level templates such as "How old is <Person>?" In general, such instance level questions typically involves time, location and names.

For simplicity, the final set of question templates are kept in the singular form. The "gaps" in the templates are classes from the DOLCE foundations ontology and they can be easily changed to other equivalent classes of other foundations ontology. Each of the question templates are mapped to description logic queries. This makes sure that the question template are answerable within the ontology.

3.1 Application of Question templates

An example of a question template is "Does a <particular> <ObjectProperty> <particular>?" The "gaps" in the template is enclosed by '<' and '>' and the word in between is a token that represent a class in the ontology. Such 'gaps' can be replace with any subclass of the class specified in the template. In the above example <particular> is replaced with a subclass of particular and <ObjectProperty> is replace by any object property in the ontology. In the case of replacing a "gap" with a specific type of object property, <ObjectProperty:Verb> is used. This allows only object properties that are verbs to replace the token.

Besides replacing tokens with its subclass, the replacement needs to conform to the underlying DL so that the question is answerable by the ontology. However, this differs across different question types and will be explained for each question type for the rest of section 3.

3.2 Yes/No Questions

These are questions that expects yes or no as an answer. I have defined three different types of Yes/No questions which takes in different sets of parameters. The first type requires two particulars and a single relation and the second takes in two particulars, one relation and a quantifier while the third type only requires one eudurant and one perdurant. The choice of different sets of parameter also depends on the DL of the ontology. Since ontology operates under the open world assumption (OWA), we may only use a set of parameters if the DL of the ontology explicitly says yes or the negation of that for no.

3.2.1 Yes/No Questions with two particulars and one relation. In DOLCE foundational ontology <Thing> is the root node of the ontology and since subclass relations are transitive by definition, any other class is a subclass of <Thing>. Templates for such questions are as follows:

- Does a <Thing> <ObjectProperty:Verb> a <Thing>?
- Did the <Thing> <ObjectProperty:Verb> the <Thing>?
- Is the <Endurant> <ObjectProperty:Nonverb> the <Perdurant>?
- Was the <Endurant> <ObjectProperty:Nonverb> <Perdurant>?

The tokens in the template can be varied to other lower classes so that the template is specific to the subject domain. To explain the underlying DL for this question type, consider the first template in the above list. The generated question "Does a X OP a Y" gives the answer "Yes", if $X \sqsubseteq \exists OP.Y$ and "No" if $X \sqsubseteq \neg \forall OP.Y$

3.2.2 *Yes/No Questions with two particulars, one relation and a quantifier.*
Templates of such questions are as follows:

- Does a <Thing> <ObjectProperty:Verb> <Quantifier> <Thing>?
- Did the <Thing> <ObjectProperty:Verb> <Quantifier> <Thing>?
- Is the <Endurant> <ObjectProperty:Nonverb> <Quantifier> <Perdurant>?
- Was the <Endurant> <ObjectProperty:Nonverb> <Quantifier> <Perdurant>?

Consider the following question "Does a X VERB some Y?" This question results in "Yes" if $X \sqsubseteq \exists$ VERB.Y and "No" if $X \sqsubseteq \neg \exists$ VERB.Y. The question "Does a X VERB all Y?" results in "Yes" if $X \sqsubseteq \forall$ VERB.Y and "No" if $\sqsubseteq \neg \forall$ VERB.Y

3.2.3 *Yes/No Questions with one Endurant and one Perdurant.*
Generating instances of these question type require "participate-in" relation within the ontology. Since DOLCE is the foundational ontology in the project and DOLCE have "participate-in" relations. Template of such questions are as follows:

- Does a <Endurant> <Perdurant>?
- Did a <Endurant> <Perdurant>?
- Is the <Endurant> <Perdurant>?
- Was the <Endurant> <Perdurant>?

Consider the following generated question "Does a X Y?" This question will result in "Yes" if $X \sqsubseteq \exists$ participate-in.Y and "No" if $X \sqsubseteq \neg \forall$ participate-in.Y

3.3 True/False Questions

Similar to questions templates in section 3.2. The expected answer of these question instance are either "True" or "False". There are two different types of true/false questions, the first type uses two particulars and a relation as parameters and the second types has an additional quantifier. Once again, I took in to consideration that the ontology operates under OWA. Hence, a question is false if the ontology explicitly says so.

3.3.1 *True/False Questions with two particulars and one relation.*
The template of such questions are as follows:

- True or false: A <Thing> <ObjectProperty:Verb> a <Thing>.
- True or false: The <Thing> is <ObjectProperty:Nonverb> <Thing>.
- A <Thing> <ObjectProperty:Verb> <Thing>. True or false?
- The <Thing> is <ObjectProperty:Nonverb> <Thing>. True or false?

Consider the question "True or false: A X OP a Y." The answer is "True" if $X \sqsubseteq \exists$ OP.Y and "False" if $X \sqsubseteq \neg \forall$ OP.Y

3.3.2 *True/False Question with additional quantifier.*
The template of such questions are as follows:

- True or false: The <Thing> is <ObjectProperty:Nonverb> <Quantifier> <Thing>.
- True or false: A <Thing> <ObjectProperty:Verb> <Quantifier> a <Thing>.
- A <Thing> <ObjectProperty:Verb> <Quantifier> <Thing>. True or false?
- The <Thing> is <ObjectProperty:Nonverb> <Quantifier> <Thing>. True or false?

Consider the generated question "True or false: The X is OP some Y." The answer is "True" if $X \sqsubseteq \exists$ OP.Y and "False" if $X \sqsubseteq \neg \exists$ OP.Y. The question, "True or false: The X is OP all Y.", is "True" if $X \sqsubseteq \forall$ OP.Y and "False" if $X \sqsubseteq \neg \forall$ OP.Y.

3.4 Equivalence Questions

This is a special case of sections 3.3.2 where the relation between class X and Y is the equivalent-to relation. The template of such question type is as follows:

- Are there any differences between a <Endurant> and a <Endurant>?
- Are there any differences between <Perdurant> and <Perdurant>?
- Are there any differences between <Quality> and <Quality>?

The question "Are there any differences between a X and a Y?" gives answer "Yes" if $X \equiv \neg$ Y and "No" if $X \equiv Y$ under the OWA.

3.5 Subclass Identification Questions

These can also be seen as "What" questions and they are also categorised into three class, one with quantifiers and the other without quantifiers and both of which takes in two particulars and a relation. The third type takes one particular and one relation.

3.5.1 *Subclass identification questions with two particulars and one relation.*
and the templates are as follows:

- What <Thing> <ObjectProperty> <Thing>?

The answer of a generated question "What X OP Y?" is any subclass $Z \sqsubseteq X \sqcap$ OP.Y

3.5.2 *Subclass identification question with additional quantifier.*
and the template is as follows:

- What <Thing> <ObjectProperty> <Quantifier> <Thing>?

The answer of a generated question "What X OP some Y?" is any subclass $Z \sqsubseteq X \sqcap \exists$ OP.Y and the question "What X OP all Y?" would have the answer of any subclass $Z \sqsubseteq X \sqcap \forall$ OP.Y

3.5.3 *Subclass identification questions with one particular and one relation.*
The question templates are as follows:

- What does a <Endurant> <ObjectProperty:Verb>?
- What did a <Endurant> <ObjectProperty:Verb>?
- What is the <Endurant> <ObjectProperty:Nonverb>?
- What was the <Endurant> <ObjectProperty:Nonverb>?

For generated question "What does a X OP?" the answer is any class Y such that $X \sqsubseteq \exists$ OP.Y

3.6 Narrative Questions

These question can also be seen as definition questions. The templates used in this project is the following:

- Define <Thing>.
- What is <Thing>?

A class in an ontology is defined if it satisfies one of the following criteria:

(1) The class is annotated with a definition.
(2) The class has at least one equivalent class.
(3) The class has at least one superclass, at least one subclass or a combination of both

Therefore, for question such as "Define X" can either annotated definition, equivalent class or its relations to its super and subclasses as an answer.

4 QUESTION GENERATOR

The question generator was created using Java with Netbeans IDE. It makes use of the two major dependencies, the OWL api and the Hermit reasoner. Figure 10 shows the graphical user interface (GUI) of the question generator. The generator takes a .txt textfile that contains the template and an .owl file that contains the ontology as inputs.

4.1 Input standards

The question templates text file needs to be in a specific format so that the question generator can recognise the question types and apply algorithms to substitute the set of tokens accordingly. The question template file start with a question type and followed by template of that question type. The following is an illustration of how a question template should look like:
Question type 1
Template 1
Template 2
....
Question type 2
Template 1
Template 2
....

There are in total 10 different question types and templates of each question type is explained in the previous section. The string representation for each question type that is recognised by the generator are as follows:

(1) Yes-No 2 particular 1 relation
(2) Yes-No 2 particular 1 relation + quantifier
(3) Yes-No 1 particular 1 relation
(4) Equivalence
(5) True-False
(6) True-False + quantifier
(7) What 2 particular 1 relation
(8) What 2 particular 1 relation + quantifier
(9) What 1 particular 1 relation
(10) Define

The question generator also defines various result sets. Each result set contain sub-class name(s) and/or sub-object property name(s) that are used to replace tokens in the templates. Some question groups may use the same result set since they expects the same form and combination of sub-classes, sub-object properties and/or quantifiers. Question type 1 and question type 5 from the above list requires the same result set, both requires two sub-classes and an object property, and the result set needs to conform to the same DL, that is:
For OWLClass X, Y and OWLObjectProperty OP
$X \sqsubseteq \exists$ OP.Y
$X \sqsubseteq \neg \forall$ OP.Y
Hence, these two question types are grouped into Question Group 1, and the same algorithm applies to both of them. The same applies to question type 2 and question type 6. Both require result sets in the form of two particulars, one relation and a quantifier and conform one of the following DLs:
For OWLClass X, Y and OWLObjectProperty OP
$X \sqsubseteq \exists$ OP.Y
$X \sqsubseteq \neg \exists$ OP.Y
$X \sqsubseteq \forall$ OP.Y
$X \sqsubseteq \neg \forall$ OP.Y
Since, the same algorithm is applicable to both question types, they are grouped into Question Group 3. The rest of the question types belongs its own question group: yes/no questions with an endurant and a perdurant belongs to question group 2, what questions with two particular and a relation belongs to question group 4, what questions with one endurant and one object property belongs to question group 5, what questions with an additional quantifier belongs to question group 6, definition questions belongs to question group 7 and equivalence questions belongs to group 8.

Figure 1 give a table of how the question types were grouped

Different result sets are defined to present words that replaces tokens in a structured manner. Certain question groups share the same result set because they expect the same entity types as a result.

Figure 2 shows different result sets, the structure of the result set and which question types makes use of it.

Question groups 7 (Definition Questions) and 8 (Equivalence Questions) are not assigned any result sets. This is because there are no decencies between the expected result of these question group.

The question generator assumes that all the question templates have only "a" and "the" as articles. Therefore, the generator is equipped to change "a" to "an" if the substituted word starts with a vowel but it is unable to handle change of articles if "an" appears in the templates.

On the other hand, there are a set of specifications and standards on the input ontology so that the question generator can recognise and work with it. The input question templates use class names of the DOLCE foundational ontology as tokens, as a result the generator may not be able to recognise the tokens if another foundational ontology is used in the input ontology. However, a quick and easy fix is to replace the tokens in the template with the equivalent class names or object property names from the input ontology.

Figure 9 in the appendix gives a set of classes and relational properties and their equivalent names across three different foundational ontology.

Another standard that input ontology should have is that all classes and object property beside the top node (OWL:Thing) and the bottom node (OWL:Nothing) must have the same prefix. The question generator make use of org. semanticweb. owlapi. model. PrefixManager to add prefixes to a String so that it can used in ontology queries as an OWLClass. For an example animal in the African wildlife ontology has the full IRI name, file:/Applications/ Protege_4.1_beta / AfricanWildlifeOntology1.owl#animal. When given the token <animal> from the template as a String, the prefix manager adds the prefix to the String and outputs it to the OWLDataFactory to create the corresponding OWLClass. This become a problem is the classes in the ontology have different prefixes, the PrefixManager does not know which prefix corresponds to which String. Therefore, it is necessary that all class and object properties have the same prefix. However, the OWL:Thing and

Question Groups	Question Types
1	Yes/No and True/False questions with 2 particulars and 1 relation
2	Yes/No questions with 1 endurant and 1 perdurant
3	Yes/No and True/False questions with an additional quantifier
4	What questions with 2 particulars and 1 relation
5	What questions with 1 endurant and 1 relation
6	What questions with an additional quantifier
7	Definition questions
8	Equivalence questions

Figure 1: Table of Question Groups

Result Set Type	Structure
1	OWLClassExpression X, OWLObjectProperty OP, OWLClassExpression Y
2	OWLClassExpression X, OWLClassExpression Y
3	OWLClassExpression X, OWLObjectProperty OP, OWLClassExpression Y, String Quantifier
5	OWLClassExpression X, OWLObjectProperty OP

Figure 2: Table of Result sets

OWL:Nothing are two exceptions, they usually have different prefixes but these two classes can be called by the reasoner using reasoner.getTopClassNode() and reasoner.getBottomClassNode() methods respectively.

In OWL, object properties can be seen as roles or relations between two OWLClasses. It is necessary that these object properties are classified into verbs and non-verbs. This is because the syntax rules of English, certain question template only conforms to the syntax rules of English if the object property is a verb. Likewise, other question templates can only work with non-verb object properties. For example: the result set [“branch”, “a-part-of”, “tree”] where both “branch” and “tree” are OWLClasses and “a-part-of” is an object property that is a non-verb. Here we cannot use the template “Does a <Thing> <ObjectProperty> a <Thing>?” because the resulting sentence “Does a branch a part of a tree?” violates the syntax of an English sentence. However, the same result set works perfectly to a different question template. Consider, the template “Is the <Thing> <ObjectProperty> the <Thing>?” by substituting the same result set gives the sentence “Is the branch a part of the tree?” which conforms to the syntax of an English sentence. Similarly, the result set [“lion”, “eat”, “impala”] would not work the template “Is the <Thing> <ObjectProperty> the <Thing>?” because the resulting sentence “Is the lion eat then impala?” does not make sense but using the same result set with the template, “Does a <Thing> <ObjectPropert> a <Thing>?” give a sensible resulting sentence of “Does a lion eat a impala?”. Therefore, it is important for both the template and the ontology to distinguish between verb and non-verb object property. For the above example, a-part-of should be a sub-object property of non-verb and eat should be a sub-object property of verb in the ontology. As the templates were changed to “Does a <Thing> <ObjectPropert:Verb> a <Thing>?” and “Is the <Thing> <ObjectProperty:Non-verb> the <Thing>?”

The question generator also assumes that the input ontology is consistent. Any inconsistency in the ontology may cause the Hermit reasoner to give unexpected results. This can output unwanted result set and generate poor quality questions.

4.2 Question Generation Algorithms

This subsection explains the algorithms used for each of the question groups. This ensures that the generated question is answerable by the ontology. Although, a major part of the algorithms varies for different question groups, there are several steps in the algorithms that are same across all question groups. There are 3 different type of tokens: quantifier tokens, OWLObjectProperty tokens and OWLClass tokens. All tokens in a template appear in between ’<’ and ’>’. Quantifier tokens are represented as <quantifier> in the templates, it is replaced with either "some" or "all". OWLObjectProperty tokens can be represented in the template in one of two way. When the token appears as <ObjectProperty> then it can be replaced with any sub-object property in the ontology that satisfies the DL of the question type. If it appears as <ObjectProperty:X> where X is an object property in the ontology, then the token is replaced with any sub-object property of X that satisfies the DL of the question type. If a token is either a quantifier token or an OWLObjectProperty

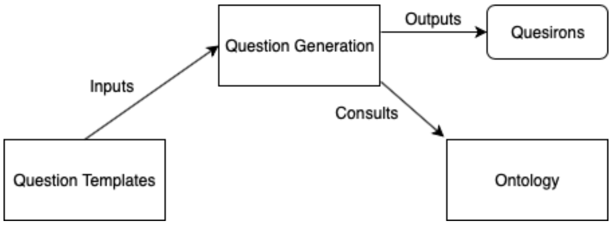

Figure 3: Diagram of the question generation process

token, then it is a OWLClass token, where the word in between ’<’ and ’>’ is the name of a class in the ontology. If <Y> appears in the template, the it can be replaced with any subclass of Y that conforms to the underlying DL of the question type.

The reasoner was used across many question generations. Getting sub-classes of a given class is mostly widely used. The reasoner was also used when getting the domain and range of a OWLObejctProperty. However, these may not always return what you expect. For example: If an ontology have classes ["Land", "Sea", "Sky", ...] as sub-classes of the class ["Endurant"] and an OWLObjectProperty ["Lives-on"] that have range Land $\sqcup$ Sky $\sqcup$ Sea. Then the reasoner query, reasoner.getObjectPropertyDomains (participate-in,false) return "Endurant" rather that the set ["Land", "Sky", "Sea"] and notice that "Endurant" is the superclass of a lot more classes than just "Land", "Sky" and "Sea".

To handle the NLG aspect of the question generation, a java class called Linguistic Handler contains algorithms that handles article checking, word replacement and String manipulations. When a word replaces a token in the template, the linguistic handler checks if the word starts with a vowel and adjust the word "a" to "an" accordingly. Linguistic handler also produces class names from String representations of OWLClass. For example: The toString method of an OWLClass produces "file:/Applications/ Protege_4.1_beta / AfricanWildlifeOntology1.owl#animal", the linguistic handler removes the prefix and produces the class name "animal". The most essential functionality of the linguistic handler is that it maps parts of the result set to the corresponding "gaps" in the templates. For example: consider the template "Does a <Thing> <ObjectProperty:verb> <Thing>?" with type 1 result set ["lion", "eat", "Impala"]. The linguistic handler recognises that type 1 result set is of the form [OWLClass X, OWLObjectProperty OP, OWLClass Y] and it maps to the tokens in the templates as "Does a X OP Y?". Hence, the above example becomes "Does a lion eat a impala?"

Figure 3 show a flow cart diagram of the overview of question generation process.

4.2.1 Group 1 Question Generation. For group 1 questions expects a types 1 result set which contains [OWLClassExpression (X), OWLObjectPropertyExpression (OP), OWLClassExpression (Y)] and the result set need to conform to either one of the two ontology axioms:

(1) $X \sqsubseteq \exists$ OP.Y
(2) $X \sqsubseteq \neg \exists$ OP.Y

In OWL, we can view object property, OP, as a function where the domain is, OWLClass X and the range is OWLClass Y. Therefore, it is clear that both X and Y is dependent on OP. The algorithm starts off by determining all the sub-ObjectProperty of the OP specified in the template. For each sub-ObjectProperty, the algorithm finds a set of domain and a set of range. Keep in mind that a domain in only valid if it is a subclass of the OWLClass specified in between ’<’ and ’>’ in the template and the same applies to ranges resulting in a set of valid ranges. A type 1 result set is then construct with a random element from the set of valid domains, a random element from the set of valid ranges and the sub-objectProperty current at hand. This step is repeated for each sub-objectProperty, this results in a valid result set generated every iteration. The file final result set that are applied to the template are randomly picked from the set of valid result sets.

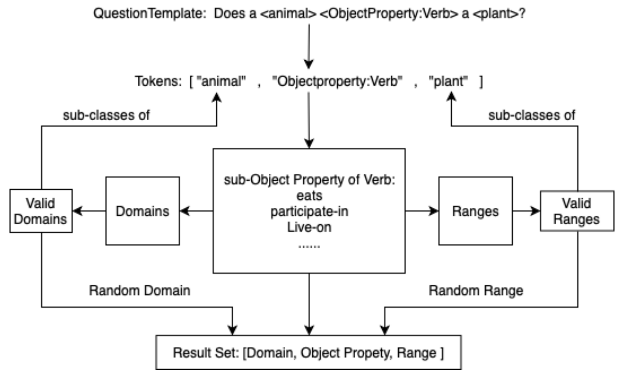

Figure 4: Diagram of group 1 question generation

figure 4 is a flow chart diagram of the generation algorithms explained above.

4.2.2 Group 2 Question Generation. There two different way to view group 2 question template:

(1) Does a <Endurnat> <Predurnat>?
(2) Does a <Endurant> <ObjectRelation:Verb>?

To generate instances of questions from template (1), the question generator have to assume that the ontology makes use of the participate-in object property. In this case, for an OWLClass $X \sqsubseteq$ Endurant and an OWLClass $Y \sqsubseteq$ Perdurant, the pair have to satisfy one of the following axioms:

- $X \sqsubseteq \exists$ participate-in.Y
- $X \sqsubseteq \neg \forall$ participate-in.Y

if the pair (X,Y) satisfies the first axiom then algorithm generates a question with a "Yes" answer and if the second axiom is satisfied then the algorithm generates a question with a "No" answer.

Template (2) can be used for any ontology as long the ontology contains some OWLClass X $\sqsubseteq$Endurant and some OWLObjectProperty OP $\sqsubseteq$Verb that satisifies the one of the following axioms:

- $X \sqsubseteq \exists$ OP.Y, where Y is any class in the range of OP
- $X \sqsubseteq \neg \forall$ OP.Y, where Y is any class in the range of OP

Template (2) can be used for a larger range of ontologies. However, the question generator assumes that the ontology makes use of the DOLCE foundational ontology and the "participate-in" object property is always available in DOLCE. Hence, template (1) is used here to generate questions.

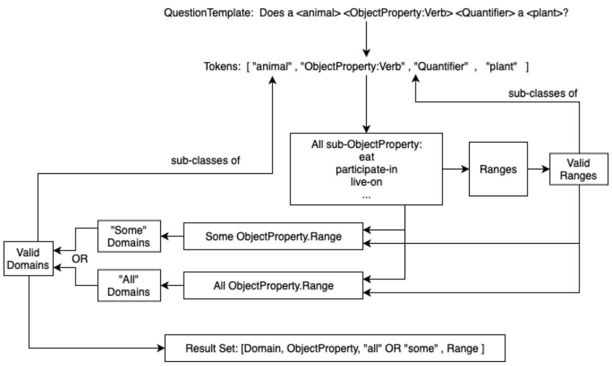

Figure 5: Diagram of group 3 question generation

The algorithm for question group 2 output type 2 result sets. Type 2 result set consists of 2 OWLClassExpressions. The algorithm of group 2 questions are similar to that of group 1 questions. Instead of, going getting domains and ranges for all sub-objectProperties, the algorithm get only the domains and the ranges for participate-in object property only. The output result set consists of a randomly selected domain and a randomly selected range. Once again, keeping in mind that the selected domain and range need to be sub-classes of endurant and perdurant respective according to the template.

Figure 11 shows that the algorithm is the same as group 1 question generation algorithm shown in figure 4 with the object property fix at "participate-in" OWLObjectProperty.

4.2.3 Group 3 Question Generation. The algothem first finds all sub-object properties (subOP) of the given object property (OP) from the template. For each sub-object property, the algorithm finds a set of ranges(R). It then filters out all the invalid ranges that are not sub-classes of the second OWLClass specified in the template. For each sub-objectProperty and each valid range (validR), the algorithm finds all OWLClass X such that it satisfied at least one of the follwing axioms:

(1) $X \sqsubseteq \exists$ subOP.validR
(2) $X \sqsubseteq \neg \exists$ subOP.validR
(3) $X \sqsubseteq \forall$ subOP.validR
(4) $X \sqsubseteq \neg \forall$ subOP.validR

Lastly, the algorithem filters out all invalid domains that are not a subclass of the first OWLClass specified in the template. If x satisfies (1) or (2), the algorithm return as a type 3 result set which contains [X , subOP, Y, "some"]. On the other hand, if X satisfies (3) or (4), the algorithem return as type 3 result set which contains [X, subOP, Y, "all"]. For questions generated based on axiom (1) or (3) should return a "Yes" or "True" answer and questions generated based on axiom (2) or (4) should return a "No" or "False" answer.

Figure 5 depicts the above generation algorithm for question group 3.

4.2.4 Group 4 Question Generation. The algorithm that generate group 4 questions expects results of two OWLClasses and an OWLObjectPropert. Therefore, a type 1 result set is used here. This algorithm works in a similar way as to group 1 question generation: Start off by finding all sub-objectProperty of the given object property. Determine a set of valid ranges and valid domains for each of the sub-objectPropety. An extra step of domain filtering is required in this algorithm, the selected domain cannot be an end node in the ontology. This is because the answer of group 4 question is always a subclass of the domain. Group 4 questions are based on the following axioms: For OWLClass X, OWLClass Y, and OWLObjectProperty OP that replaces tojkens in the template, we have an OWLClass Z such that, $Z \sqsubseteq X$ and $X \sqsubseteq \exists$ OP.Y and Z is the expected answer of the generated question.

Figure 12 in the appendix depicts the above generation algorithm for question group 4.

4.2.5 Group 5 Question Generation. Group 5 questions requires an OWLClass and an OWLObjectProperty to replace the tokens in the templates. Therefore, result set 5 is used to manage the output of the algorithm. This algorithm operates exactly the same as the group 1 question generation, we get a triplet [OWLClass X, OWLObjectProperty OP, OWLClass Y] that satisfies the following axiom:

- $X \sqsubseteq \exists$ OP.Y

The difference is that the template only requires OWLClass X and OWLObjectProperty OP. OWLClass Y is the expected answer of the generated question.

Figure 13 in the appendix depicts the above generation algorithm for question group 5.

4.2.6 Group 6 Question Generation. The algorithm for group 6 questions output a set of quadruplet in the form of [OWLClass X, OWLObjectProperty OP, quantifier, OWLClass Y], hence type 3 result set is used here. The result set is generated similar to group 3 question generation with an extra domain filtering step which removes all domains that are an end node of the ontology. Similar to group 4 question generation, the answer of group 6 question generation is always a sub-class of the domain. For each result set generated there exist a OWLClass Z such that $Z \sqsubseteq X$ and the result set satisfies at least one of the following axioms:

(1) $X \sqsubseteq \exists$ OP.R
(2) $X \sqsubseteq \forall$ OP.R

The corresponding result set for axiom (1) have the format of [X, OP, "some", Y] and [X, OP, "all", Y] corresponds with axiom (2). The expected answer for this question group is class Z mentioned above.

Figure 14 in the appendix depicts the above generation algorithm for question group 6.

4.2.7 Group 7 Question Generation. As explained in section 3.6, a class in an ontology in the following three ways:

(1) The class is annotated with a definition.
(2) The class has at least one equivalent class.
(3) The class has at least one superclass, at least one subclass or a combination of both

The generating algorithm checks each class for the above ways to define a class using the following methods respectively:

(1) Check if the set, OWLClass.getAnnotations(ontology, isDefinedBy) is empty or not. An empty set means that the class is not annotated with a definition.

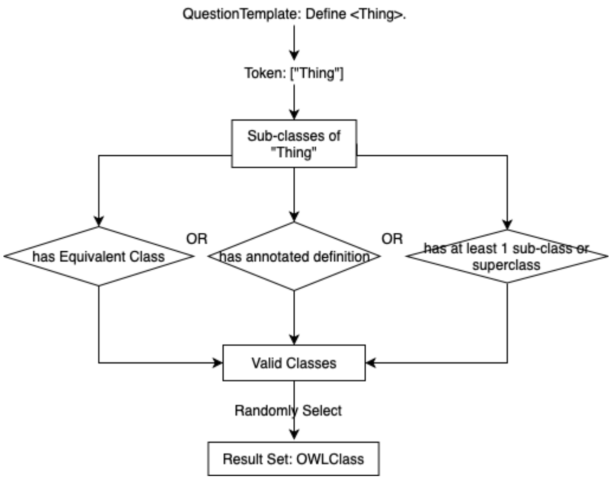

Figure 6: Diagram of group 7 question generation

(2) Check if the set, OWLClass.getEquivalentClasses(ontology) is empty or not. An empty set shows that the class does not have any equivalent classes.
(3) Check if the sets, OWLClass.getSuperClasses(ontology) and OWLClass.getsubClasses(ontology) is empty. The algorithm then removes OWL:Thing from the first set and OWL:Nothing from the second set because these classes are trivial. If both resulting set are empty, then it means that the class is not related to any other class.

If the class does not satisfy any of the three ways of definition, in other words, if the class is not annotated with a definition and is disjoint from the rest of the ontology, then algorithm removes such classes as candidates for possible replacement of the token in the template.

Figure 6 depicts the above generation algorithm for question group 7.

4.2.8 Group 8 Question Generator. Group 8 questions only contains equivalence questions. Although there are two "gaps" in the template, it is intuitive that these two tokens are in one of the following structure:

- The two token are representations of the same class in the ontology. Example: "Are there any differences between a <Endurant> and a <Endurant>?"
- The class representation of one token is a subclass of the class representation of the other class. Example: "Are there any differences between a <Endurant> and a <Animal>?"

A counter intuitive example is "Are there any differences between a <Endurant> and a <Perdurant>?" Unless these is an axiom in the ontology that specifically states that Endurant is not equivalent to Perdurant.

Group 8 question generator results in two OWLClasses. Although it is possible to just replace any sub-classes of the classes specified by the template, but under the OWA class X is not equivalent to class Y only if the ontology specifically say so. Therefore, it cannot be assumed that if Y is not in set, X.getEquivalentClasses(ontology), implies that X is not equivalent to Y. If OWLClass X is equivalent to OWLClass Y ($X \equiv Y$), then by the symmetry of equivalence, OWLClass Y is also equivalent to OWLClass X ($Y \equiv X$). The algorithm checks $X \equiv Y$ by searching for Y in the set, X.getEquivalentClasses(ontology) and $X \equiv \neg Y$ is the set, X.getEquivalentClasses(ontology) contains $\neg$ Y.

Figure 15 in the appendix depicts the above generation algorithm for question group 8.

5 EXPERIMENT DESIGN AND EXECUTION

The main aim of this experiment is to evaluate the quality of the generated questions. Besides the evaluation, another aim is to gather other details through feedback provided by the participants which is discussed in section 5.1. Section 5.2 gives a general overview of what was expected as the outcome of this experiment. There are 2 sets of evaluations in the experiment. Both sets are a list of generated questions from the same ontology but with different templates. The materials used to do so is provided in section 5.3. Section 5.4 provides the process of this experiment and experiment methods used in detail.

5.1 Aim

The aim of this evaluation is to determine the quality of questions generated by the software mentioned in section 4. The quality of questions in this paper consists of three aspects. The syntax of a question refers to rules in the grammar of English for the use of words, punctuation, phrases, clauses and the structure of the sentence. The second aspect of a quality question is the semantics of the question. A question with a clear and unambiguous meaning is known to have good semantics. The last aspect of a quality question is that it must be answerable by the ontology. However, questions used in this experiment should be answerable by the nature of question generation algorithms mentioned in section 4.2.Unless the algorithms defined are incorrect and this can be reflected from the semantics of the generated questions.

Feedback from participant are analysed and compared to given an indication on any problems. Such problem can be in term of syntax, semantic or the question generation algorithms. These identified problems can be used to improve the next version of ontology question generator and for the better understanding of limitations and specifications to generate quality questions.

As mentioned above, there are evaluation of two different sets of questions generated from the same ontology but different templates. The differences for the template is mentioned in section 5.3. By comparing the evaluation results from both sets of questions, it may give indications as to what are the pros and cons of applying template of different levels and how that affects the quality of the questions.

5.2 Hypothesis

The hypothesis of this experiment is that the generated questions should make some sense in term the underlying ontology and the axioms that each question type refers to. Since, the token replacements are randomly generated, typically from a set of domain or range, some unforeseen errors may arise. Any errors with the generation algorithms should have a knock on effect on the semantics of the questions.

The syntax of the generated questions should be correct in a typical scenario where the expected words appears in the correct token slots. Unforeseen cases of question syntax may also arise. Once again, this is due the random selection of replacement word from a set of domain and range. Syntax errors may also arise from the fact that many possible change of syntax depending different factors, such as the replacement words for tokens, may not be covered in the limited time frame of this project.

5.3 Materials

The participants evaluated two set of questions generated from the same ontology but different question templates. The ontology used is an extended version of the African Wildlife ontology, by adding enough classes and object properties to the original African-WildlifeOntology1.owl [3] that allows the question generator to generate all question types investigated in this project. This ontology has 41 classes which are shown in figure 16 and 10 object properties show in figure 17. Besides the addition of DOLCE foundational ontology to the AfricanWildlifeOntology1.owl by classifying all classes into endurants and perdurants, the object properties were also classified into verb and non-verbs. Ranges on the newly added object properties such as "Live-on" and "participate-in" are "Land or sea or sky" and "walk or fly or hibernate", respectively. Domains of such object properties were not specified as it causes inconsistencies of the ontology. An example of such inconsistency occurs when adding "animal" as the domain of the object property "eats" with its existing range from AfricanWildlifeOntology1.owl as this cause a problem since "animal" is disjoint from "plant".

The first set of questions consists of 30 automatically generated questions. This set of questions are generated from the set of template identified in section 3 which are listed is figure 18 in the Appendix. Question templates used in the 1st evaluation are generally abstract and the tokens are usually high level OWL-Classes in the ontology. Such OWLClasses are usually contained in the foundational ontology, in this specific ontology the DOLCE foundational ontology. This allows the template to be used across variety of different ontologies that uses DOLCE foundational ontology ontologies. The tokens can be replace with other equivalent OWLClasses from other foundational ontologies as discussed in previous section 4.1. The resulting questions used for evaluation 1 is listed in figure 20.

In contrast, the question templates that are used to generate evaluation 2 are more subject domain specific. A template in evaluation 1 were replaced with multiple templates based on the different scenarios that this template can be apply to. For example: This template in the first set

- "What <Thing> <ObjectProperty> <Thing>?"

can be replaced with the following templates which are different scenarios of the above template.

- "What <animal> <ObjectProperty:eat> <animal>?"
- "What <animal> <ObjectProperty:eat> <plant>?"
- "What <animal> <ObjectProperty:Live-on> <Habitat>?"
- "What <animal> <ObjectProperty:Participate-In> <Perdurant>?"

In this way 40 question templates were developed form the 1st set of templates. On the down side, this type of template are less diverse and not applicable to other ontology. The template "Does a <animal> <ObjectProperty:eats> <plant>?" is applicable to the African Wildlife ontology, but not for other ontology such as the data mining ontology because the OWLCLASS "animal", "plants" and OWLObjectPeoperty "eats" does not exist in the data mining ontology and hence there are no ontologies to replace these tokens with. Figure 19 in the appendix shows the templates used to generate a set of questions for evaluation 2. The resulting question are list in 21.

Evaluation 1 (Total 30 questions)	Evaluation 2 (Total 40 questions)
0	15
11	10
8	16
12	12
9	15
	14

Figure 7: Number of quality questions that were evaluated

5.4 Methods

A group of University of Cape Town (UCT) students were recruited to complete the evaluations. All participant has at least a matric pass of English and are able to speak fluent English. Each of the participant were given either evaluation 1 or evaluation 2 and in some cases both evaluations depending of the availability of the participants.

An evaluation required participants to read through every generated question and determine whether each question conforms to the syntax and semantic of English. The participants were also given an option to provide feedback to questions that they think violated grammatical rules of English or just simply just didn't make sense. These feedback can be either stating what they think was wrong with the question or provide a question that conform to the rules of English and makes sense to them.

6 RESULTS AND FINDINGS

There were in total 11 evaluations done. 5 of which were evaluation 1 and 6 evaluation 2. Figure 7 shows the number of quality questions in each of the evaluation that was conducted. This show that 26% of the generated questions in the first evaluation were considered quality questions while the second evaluation produce 34% of quality questions. From this one can see that specifying the template to a lower level class token helps to improve the quality of the generated questions.

However, the issue arise while developing template for the second evaluation is that there may be the risk of over specifying the templates. If that was the case, the tokens in the template are located at lower levels of the ontology or even worse at instance level, then it defeats the purpose of creating a template for automated generations.

Analyse and reflection on the feedback provided by the participants gave insights as to why the rate of quality question generation is so low. A list of issues with the current question generation system and potential improvements for the next attempts in an ontology-based question generation are as follows:

- The present of ambiguity in the generated questions may cause some questions not too be answerable by the ontology. For example: "Does the carnivore participate in fly?", the more direct and easier way to ask this question is "Does carnivores fly?". However, ambiguity occurs depending on how the user interprets "participate in"
- Depending on the scenarios and the nature of the replacing word, the articles "a/the" changes. For example: Using the template "Was the <Thing> <ObjectProperty:Nonverb> <Thing>?", the sentences "Was the Impala eat by the lion." and "Was the branch a prat of a tree?" both conformed to the template but the article of the second replacing words differs.
- The need to take into consideration that terms used for the names of the object property may not be suitable to be used for NLG. For example: "Group of" or "Type of" are more sensible replacements for "subclass of".
- Better decision of the use of templates. For example: consider the templates "Does a <animal> <Perdurant>?" and "Did a <animal> <Perdurant>?". Replacement tokens with "bumblebee" and "fly" respectively, works for the 1st template but not the 2nd.
- Same issues as the previous point but applied to was/is. This shows that the choice of words changes the templates that is required to generate a quality question and vise versa the choice of the templates changes the tense and form of the replacing words.
- Some words always uses article "the" instead of a/an. Example: The land, the sea
- Considering the previous 3 points one can see that the articles "a" and "the" cannot to fixed in the templates since it changes according to the replacing word. However, this can be solved by replace one template with multiple scenarios of that template. Like the differences between evacuation 1 and evaluation 2 5.3.
- Avoid repetition of replacing tokens. Example: "The hibernate is proper part of hibernate."
- The use of a-proper-part-of does not work in sentences, rather use a-part-of.
- In some case of group 4 questions ("What" questions), better option is to use "Which" rather than "What". For example: "Which terrestrial eats rock dassie?" makes more sense then the question "What terrestrial eat rock dassie?"
- Words changes singular/plural depending on where they are in the sentence. Example: the word "plant" in the following sentences "what plant is eaten by impala?" "what animal eat plants?"
- The generation of open ended questions such as "What does an impala participate in?". Even though the range of participate-in is eat, walk and hibernate but the reader does not know the underlying ontology well enough to understand the question.
- In some cases the tense change depending on the Genaro. For example: Human is a part of animals , but never: human was a part of animals

Besides the above listed syntax and semantic errors that occurred, another major issue that caused in the low rate of quality questions generation is that use if domain and range of the objectProperties rather than using of axioms that that generate more sensible question.

The insufficiency of using a random domain and range to replace tokens come from the fact that domain cannot be specified for certain object properties which inconsistent ontologies as discussed in section 5.3.

Another down fall of the questions generated for the evaluations in because that the ontology used was too small and causes repeated random generations.

7 CONCLUSION

In this project, a set of question templates was defined. Each template is questions that have "gaps" known as tokens in it. A token represents a class in the ontology and when generating questions these token are replace with sub-classes of the token that conforms to the underlying axiom of the question type.

Given a template we can specify the requirement on the ontology to generate such questions. Besides, the simple fact that the ontology need to be populated with a set classes and object property, more importantly, it requires the axioms that correspond to the question type to be present in the ontology. The question generator used in this experiment implemented the randomly select domain and ranges instead of acquiring domain as a element of the set objectProperty.range by applying the axiom. This caused low quality question generation in several question types.

Through the analyse of the results from the evaluations, it is shows that using templates that contains tokens that are classes at a reasonable level also increase the quality of the questions generated. Typically, ideal classes to use in the templates as tokens are classes below the foundational ontology and but higher up in the domain specific ontology.

However, the downfall of having higher classes in the domain specific ontology as tokens is that the templates are not able to be widely use across multiple ontologies. Defining question template with higher, foundational class token allows the flexibility of to be used across different ontologies with the trade-off for the quality of generated questions.

REFERENCES

[1] Vinay K Chaudhri, Peter E Clark, Adam Overholtzer, and Aaron Spaulding. 2014. Question generation from a knowledge base. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 54–65.
[2] Ulf Hermjakob Michael Junk Chin-Yew Lin Eduard Hovy, Laurie Gerber. 2002. Question Answering in Webclopedia. (2002).
[3] Maria Keet. 2010. African Wildlife ontology tutorial ontologies. (aug 2010). Retrieved September 01, 2019 from https://keet.wordpress.com/2010/08/20/african-wildlife-ontology-tutorial-ontologies/
[4] Maria Keet. 2018. *Introduction to ontology engineering*. Vol. 1.

A POTENTIAL EXTENSION WORK

A.1 Extend Question generation

Since ontology is defined at a high abstract level and focuses on classes as opposed to instances, many instance level question may not be answerable by the ontology. However, there is a possibility that there are instance level question are answerable within the textbook that the ontology is associated with. Therefore, there is a possibility to generate well defined and answerable, instance level questions.

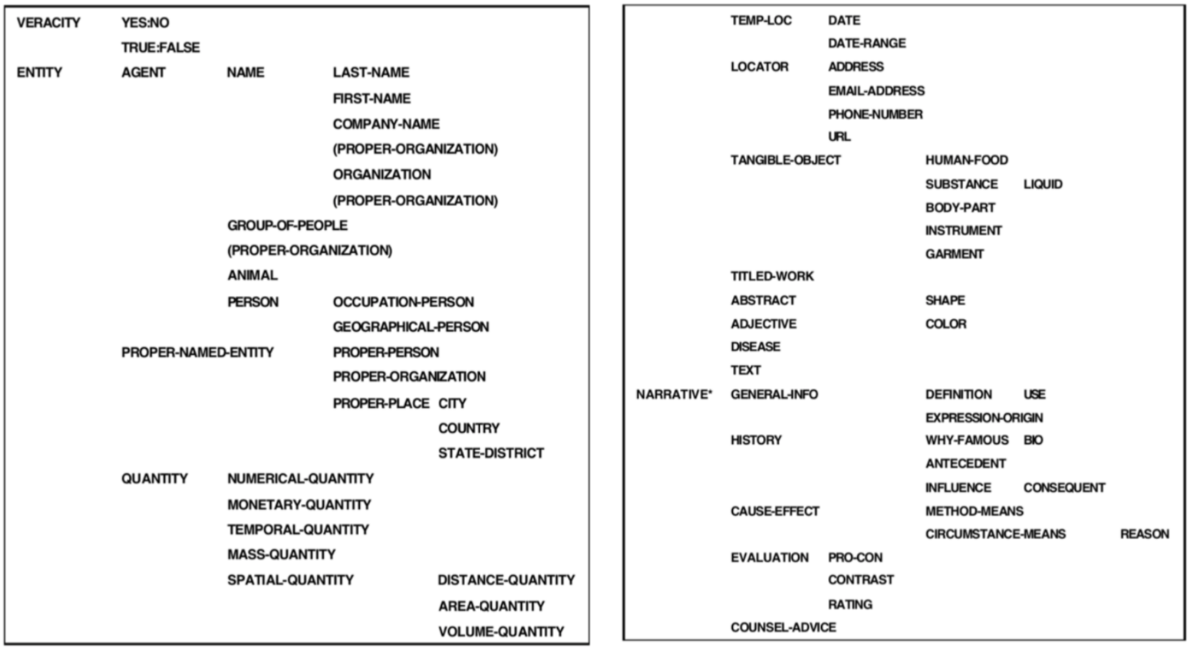

Figure 8: Question Taxonomy by information system institute from the University of Southern California [2]

	DOLCE-Lite	BFORO	GFO
	Class		
1.	endurant	Independent Continuant	Presential
2.	physical-object	Object	Material_object
3.	perdurant	Occurrent	Occurrent
4.	process	Process	Process
5.	quality	Quality	Property
6.	space-region	SpatialRegion	Spatial_region
7.	temporal-region	Temporal-Region	Temporal_region
	Relational property		
1.	proper-part	has_proper_part	has_proper_part
2.	proper-part-of	proper_part_of	proper_part_of
3.	participant	has_participant	has_participant
4.	participant-in	participates_in	participates_in
5.	generic-location	located_in	occupies
6.	generic-location-of	location_of	occupied_by

Figure 9: List of equivalent classes across 3 different foundational ontology

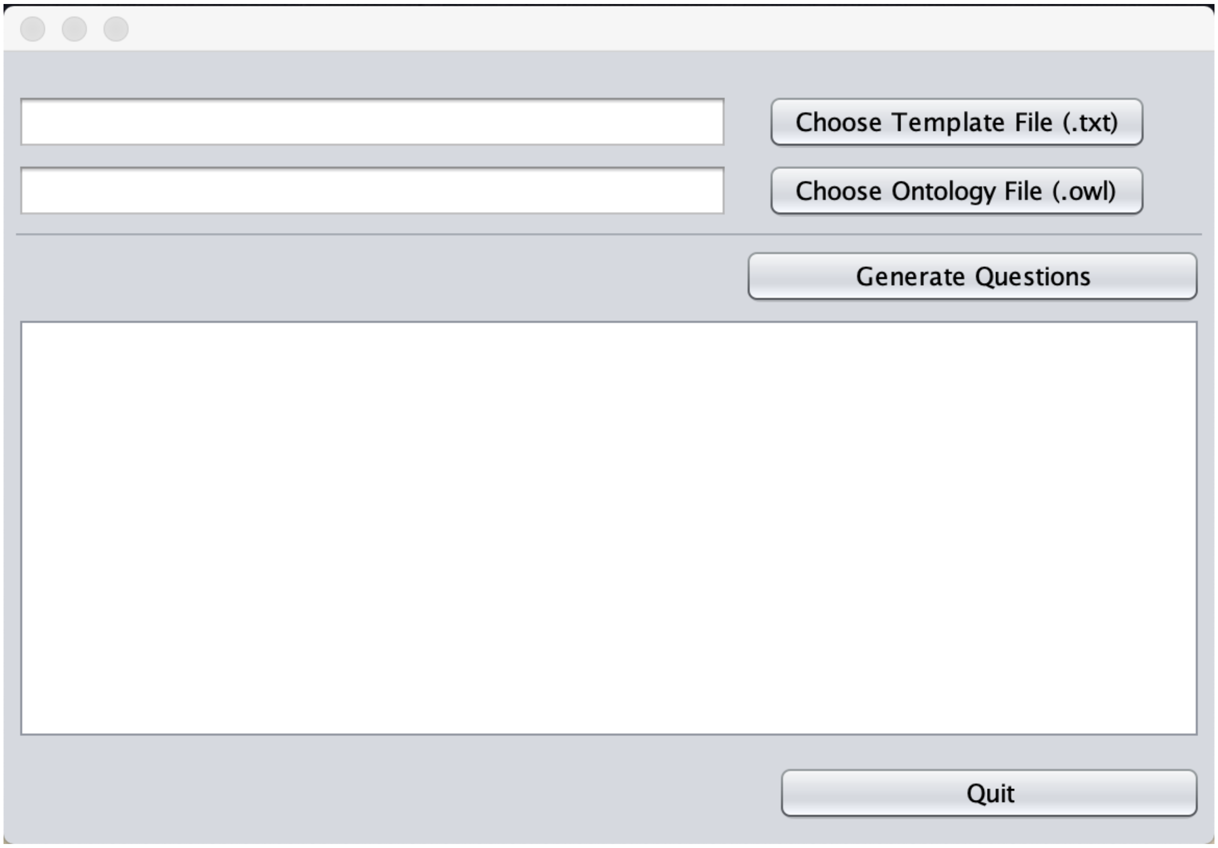

Figure 10: Graphical interface of the question generator

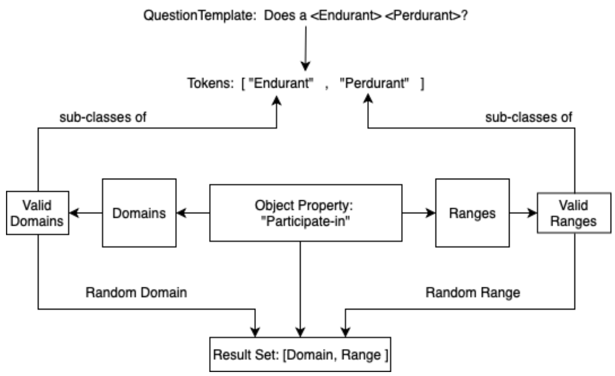

Figure 11: Diagram of group 2 question generation

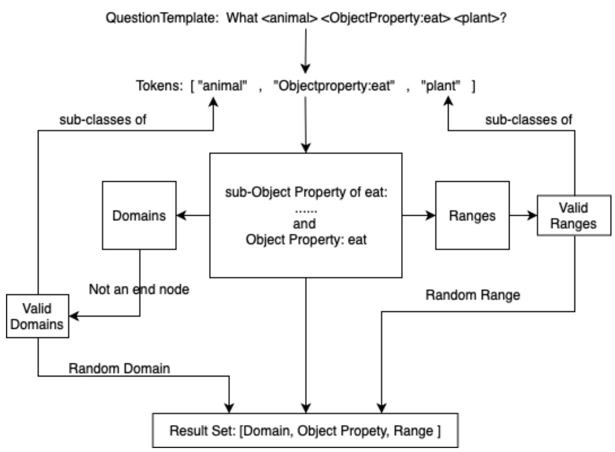

Figure 12: Diagram of group 4 question generation

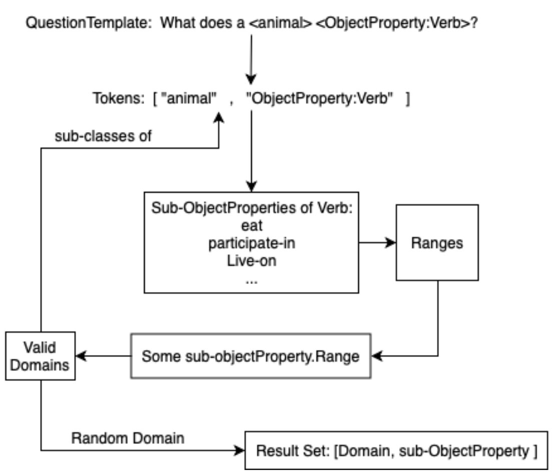

Figure 13: Diagram of group 5 question generation

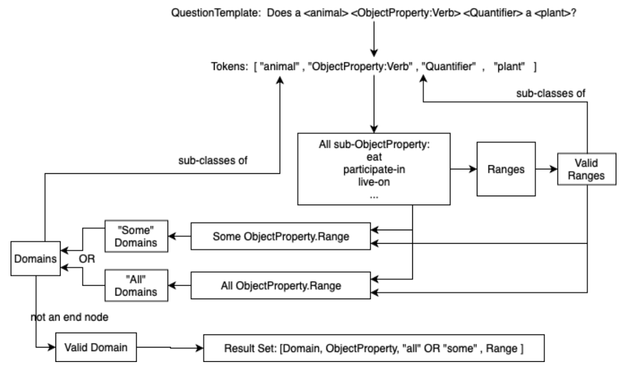

Figure 14: Diagram of group 6 question generation

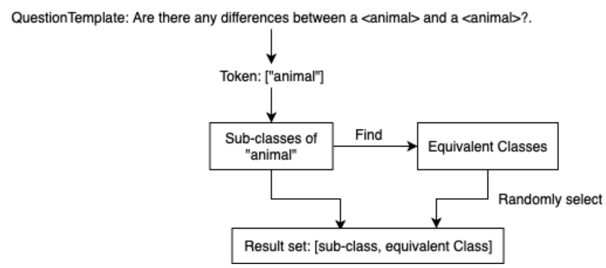

Figure 15: Diagram of group 8 question generation

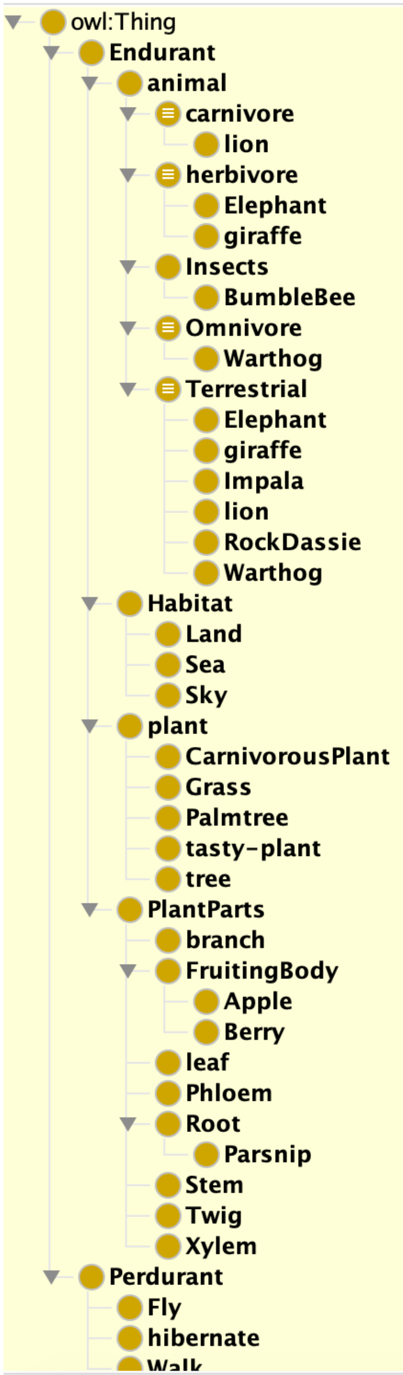

Figure 16: Class hierarchy of the ontology used to generate questions for the experiment

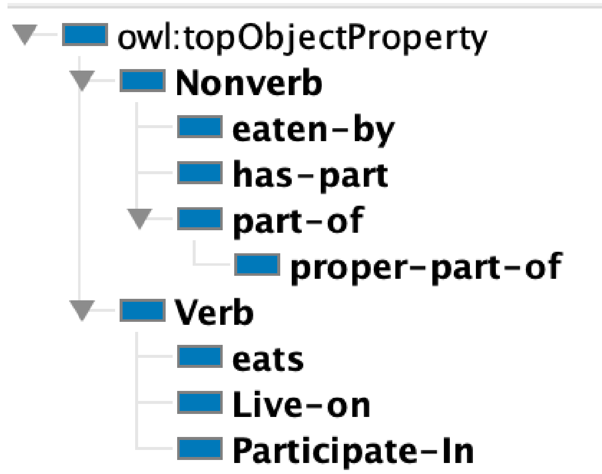

Figure 17: Object Property hierarchy of the ontology used to generate questions for the experiment

```
Yes-No 2 particular 1 relation
1. Does a <Thing> <ObjectProperty:Verb> a <Thing>?
2. Did the <Thing> <ObjectProperty:Verb> the <Thing>?
3. Is the <Endurant> <ObjectProperty:Nonverb> the <Perdurant>?
4. Was the <Endurant> <ObjectProperty:Nonverb> <Perdurant>?
Yes-No 2 particular 1 relation + quantifier
5. Does a <Thing> <ObjectProperty:Verb> <Quantifier> a <Thing>?
6. Did the <Thing> <ObjectProperty:Verb> <Quantifier> of the <Thing>?
7. Is the <Thing> <ObjectProperty:Nonverb> <Quantifier> <Thing>?
8. Was the <Thing> <ObjectProperty:Nonverb> <Quantifier> <Thing>?
Yes-No 1 particular 1 relation
9. Does a <Endurant> <Perdurant>?
10. Did a <Endurant> <Perdurant>?
11. Is the <Endurant> <Perdurant>?
12. Was the <Endurant> <Perdurant>?
Equivalence
13. Are there any differences between a <Endurant> and a <Endurant>?
14. Are there any differences between <Perdurant> and <Perdurant>?
True-False
16. True or false: A <Thing> <ObjectProperty:Verb> a <Thing>.
17. True or false: The <Thing> is <ObjectProperty:Nonverb> <Thing>.
18. A <Thing> <ObjectProperty:Verb> <Thing>. True or false?
19. The <Thing> is <ObjectProperty:Nonverb> <Thing>. True or false?
True-False + quantifier
20. True or false: The <Thing> is <ObjectProperty:Nonverb> <Quantifier> <Thing>.
21. True or false: A <Thing> <ObjectProperty:Verb> <Quantifier> a <Thing>.
22. A <Thing> <ObjectProperty:Verb> <Quantifier> <Thing>. True or false?
23. The <Thing> is <ObjectProperty:Nonverb> <Quantifier> <Thing>. True or false?
What 2 particular 1 relation
24. What <Thing> <ObjectProperty> <Thing>?
What 2 particular 1 relation + quantifier
25. What <Thing> <ObjectProperty> <Quantifier> <Thing>?
What 1 particular 1 relation
26. What does a <Endurant> <ObjectProperty:Verb>?
27. What did a <Endurant> <ObjectProperty:Verb>?
28. What is the <Endurant> <ObjectProperty:Nonverb>?
29. What was the <Endurant> <ObjectProperty:Nonverb>?
Define
30. Define <Thing>.
31. What is <Thing>?
```

Figure 18: Template used to generate the first set of questions in the ontology

```
Yes-No 2 particular 1 relation
1. Does a <animal> <ObjectProperty:Verb> a <Thing>?
2. Did the <animal> <ObjectProperty:Verb> the <Thing>?
3. Is the <Endurant> <ObjectProperty:Nonverb> the <Endurant>?
3. Is the <Perdurant> <ObjectProperty:Nonverb> the <Perdurant>?
4. Was the <Endurant> <ObjectProperty:Nonverb> <Endurant>?
4. Was the <Perdurant> <ObjectProperty:Nonverb> <Perdurant>?
Yes-No 2 particular 1 relation + quantifier
5. Does a <animal> <ObjectProperty:Verb> <Quantifier> a <Thing>?
6. Did the <animal> <ObjectProperty:Verb> <Quantifier> the <Thing>?
7. Is the <Endurant> <ObjectProperty:Nonverb> <Quantifier> <Endurant>?
8. Was the <Endurant> <ObjectProperty:Nonverb> <Quantifier> <Endurant>?
Yes-No 1 particular 1 relation
9. Does a <animal> <Perdurant>?
10. Did a <animal> <Perdurant>?
11. Is the <animal> <Perdurant>?
12. Was the <animal> <Perdurant>?
Equivalence
13. Are there any differences between a <animal> and a <animal>?
13. Are there any differences between a <plant> and a <plant>?
13. Are there any differences between a <PlantParts> and a <PlantParts>?
13. Are there any differences between a <Habitat> and a <Habitat>?
14. Are there any differences between <Perdurant> and <Perdurant>?
True-False
16. True or false: A <animal> <ObjectProperty:Verb> a <Thing>.
17. True or false: The <Endurant> is <ObjectProperty:Nonverb> <Endurant>.
17. True or false: The <Perdurant> is <ObjectProperty:Nonverb> <Perdurant>.
18. A <animal> <ObjectProperty:Verb> <Thing>. True or false?
19. The <Endurant> is <ObjectProperty:Nonverb> <Endurant>. True or false?
19. The <Perdurant> is <ObjectProperty:Nonverb> <Perdurant>. True or false?
True-False + quantifier
20. True or false: The <Endurant> is <ObjectProperty:Nonverb> <Quantifier> <Endurant>.
20. True or false: The <Perdurant> is <ObjectProperty:Nonverb> <Quantifier> <Perdurant>.
21. True or false: A <animal> <ObjectProperty:Verb> <Quantifier> a <Thing>.
22. A <animal> <ObjectProperty:Verb> <Quantifier> <Thing>. True or false?
23. The <Endurant> is <ObjectProperty:Nonverb> <Quantifier> <Endurant>. True or false?
23. The <Perdurant> is <ObjectProperty:Nonverb> <Quantifier> <Perdurant>. True or false?
What 2 particular 1 relation
24. What <animal> <ObjectProperty:eat> <animal>?
24. What <animal> <ObjectProperty:eat> <plant>?
24. What <animal> <ObjectProperty:Live-on> <Habitat>?
24. What <animal> <ObjectProperty:Participate-In> <Perdurant>?
What 2 particular 1 relation + quantifier
25. What <Thing> <ObjectProperty> <Quantifier> <Thing>?
What 1 particular 1 relation
26. What does a <animal> <ObjectProperty:Verb>?
27. What did a <animal> <ObjectProperty:Verb>?
28. What is the <Endurant> <ObjectProperty:Nonverb>?
29. What was the <Endurant> <ObjectProperty:Nonverb>?
Define
30. Define <Thing>.
31. What is <Thing>?
```

Figure 19: Template used to generate the second set of questions in the ontology

```
1.      Does an insect live on a lion?
2.      Did the rock dassie eats the grass?
3.      Is the carnivore eaten by the walk?
4.      Was the fruiting body eaten by hibernate?
5.      Does a rock dassie live on some a terrestrial?
6.      Did the sky live on some of the endurant?
7.      Is the leaf eaten by some plant parts?
8.      Was the apple proper part of all plant parts?
9.      Does a herbivore walk?
10.     Did an apple hibernate?
11.     Is the elephant walk?
12.     Was the lion walk?
13.     Are there any differences between a palm tree and a giraffe?
14.     Are there any differences between hibernate and walk?
15.     True or false: an animal eats a xylem.
16.     True or false: The bumblebee is eaten by herbivore.
17.     A twig participates in walk. True or false?
18.     The branch is proper part of lion. True or false?
19.     True or false: The impala is proper part of all endurant.
20.     True or false: A habitat live on some an endurant.
21.     A parsnip participates in some endurant. True or false?
22.     The parsnip is proper part of some endurant. True or false?
23.     What animal nonverb plant?
24.     What carnivore part of some animal?
25.     What does a xylem live on?
26.     What did a carnivore eat?
27.     What is the stem eaten by?
28.     What was the terrestrial proper part of?
29.     Define walk.
30.     What is bumblebee?
```

Figure 20: Questions used in evaluation 1

```
1.      Does a carnivore eat a terrestrial?
2.      Did the carnivore participate in the fly?
3.      Is the tasty plant eaten by the omnivore?
4.      Is the fly eaten by the walk?
5.      Was the xylem eaten by carnivore?
6.       Was the hibernate proper part of fly?
7.       Does an insect live on some an animal?
8.      Did the terrestrial participate in all the hibinate?
9.      Is the carnivorous plant proper part of all plant?
10.     Was the bumblebee eaten by some insects?
11.     Does an impala walk?
12.     Did a bumblebee fly?
13.     Is the terrestrial walk?
14.     Was the carnivore fly?
15.     Are there any differences between an elephant and a rock dassie?
16.      Are there any differences between a tasty plant and a palm tree?
17.     Are there any differences between a leaf and a phloem?
18.      Are there any differences between a sea and a land?
19.     Are there any differences between fly and walk?
20.     True or false: an elephant eats an impala.
21.     True or false: The berry is eaten by herbivore.
22.     True or false: The hibernate is proper part of hibernate.
23.     A warthog lives on sea. True or false?
24.     The omnivore is eaten by plant. True or false?
25.      The hibernate is eaten by hibernate. True or false?
26.     True or false: The leaf is proper part of some plant parts.
27.     True or false: A warthog eats some omnivore.
28.     A warthog participates in some animal. True or false?
29.     The omnivore is eaten by all animal. True or false?
30.     What terrestrial eat rock dassie?
31.      What herbivore eat palm tree?
32.      What omnivore live on land?
33.      What herbivore participate in fly?
34.     What insects live on all animal?
35.     What does an impala participate in?
36.     What does an impala participate in?
37.     What is the parsnip eaten by?
38.     What was the animal proper part of?
39.     Define xylem.
40.     What is land?
```

Figure 21: Questions used in evaluation 2