Text Generation Starting From an Ontology

Dragoş Alexandru Cojocaru
Politehnica University of Bucharest
Bucharest, Romania
dragos.cojocaru@cti.pub.ro

Ştefan Trăuşan-Matu
Politehnica University of Bucharest
and
Research Institute for Artificial Intelligence
Bucharest, Romania
stefan.trausan@cs.pub.ro

ABSTRACT

The subject of this paper is the development of an application which generates natural language text, starting from an OWL ontology. The Natural Language Generation, in the context of Semantic Web, represents a relatively new field of research, but due to the capabilities of the ontologies (central element of the Semantic Web) of being dynamically modified and completed with new information, the theme of the application is of great importance. The project employs Rhetorical Structure Theory to structure hierarchically the ontological content, resulting in a human-like discourse structure. Since the Semantic Web is continuously adding machine-readable content, the user can take advantage of this impressive database of knowledge transformed into coherent texts for human with the aid of our application.

Author Keywords

Semantic Web; ontology; Rhetorical Structure Theory; natural language generation

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

The Semantic Web aims at enriching the current Web with an additional layer of data which can be read and analyzed by intelligent agents [2]. Inside the Semantic Web, the central element is represented by the ontology. This element describes a domain of knowledge by introducing axioms which define the properties of a certain concept and the relations with other fields . Ontologies may be completed by adding new information regarding a notion and new knowledge can be deduced with the help of inference rules in the model of description logics [1].

The application presented in this paper has the objective to transform knowledge and information from an ontology into natural language text, in a manner as concise and expressive as possible, text dedicated to the human user. These characteristics of the resulting text are obtained with the help of the Rhetorical Structure Theory, which uses discourse markers to emphasize the relations between clauses.

Applications in the same field of text generation from an ontology are usually “verbalisers” which translate each axiom from the ontology into a sentence. The main disadvantages of this kind of applications are represented by the monotony, lack of expressivity of the resulting texts and the presentation of a great amount of information which might be either redundant or difficult to follow [2, 3].

A more advanced system in this field is “NaturalOWL”, which generates coherent, fluent texts due to algorithms based on the “Centering Theory” [5]. In a similar way, the current application uses referring expressions for subjects and objects, sentence aggregation and techniques to prevent information repetition. On the other hand, the current application intends to obtain a more structured text with the aid of Rhetorical Structure Theory.

The application is functional for any ontology, but in order to obtain the final text, domain dependent data is required. Therefore, at present, only information in the context of the Wine ontology is constructed, which defines the way to treat each property in order to build sentences.

In the next sections there will be presented the application architecture, the process of generating text, suggestions for future work and the conclusions of this paper.

SYSTEM ARCHITECTURE

The text generation is developed in several steps (see Figure 1), where the result of a stage represents the input for the next stage. We will now discuss the process behind each stage separately.

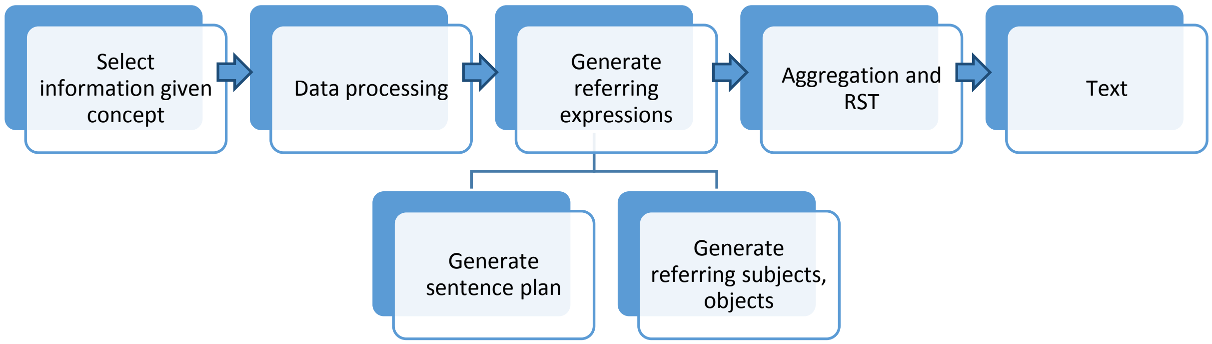

Figure 1.Application architecture describing the main stages of development towards reaching the final natural language text

Information selection

The application receives the name of a class or individual from a specified ontology. Using the Jena framework, the ontology is loaded and consequently we have all the available knowledge from the chosen ontology, as well as from imported ontologies.

Considering the name of the class or individual, we select the axioms which refer this concept. For instance, assuming the input is the name of a class, we begin by extracting the superclasses, subclassses and equivalent classes from the graph of the ontology.

Next, we obtain the other RDF triples which indicate the properties of the class, whether it is an intersection, reunion class or not and its constituent elements. The following step represents the extraction of data related to objects connected to the initial class through properties, in order to provide their description, too. If we present the subject S and we have the RDF triple <S, P, O>, where O is also a class, we would also describe the object O in a similar manner as we would do for S. The stage of axioms selection does not include the description of property classes, or further links in the graph, because they either bring uninteresting information for the user (presenting property characteristics such as: transitive, functional), or they result in a text difficult to follow [5].

Data processing

The selected axioms are transformed into their own form of triplets <Subject, Predicate, Object>. Because the Object in a RDF triple may be an unnamed class representing an union or an equivalent class, we break the object, building new triplets in which the object is a named class, individual, or data value of a property.

The object O of a triplet has to be recursively transformed as we further explain, until we reach a named class. If subject S is an intersection class of objects O_i we produce multiple triplets <S, isA, O_i>. Similarly, if S is a union of other classes, we obtain a disjunction of triplets <S, isA, O_i>. If we encounter an object which is a complement of a class the resulted triplet will be <S, isNotA, O>. Considering the following fragment of ontology:

```
<owl:Class rdf:ID="Meursault">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty rdf:resource="#hasBody"
/>
     <owl:hasValue rdf:resource="#Full" />
   </owl:Restriction>
<owl:Restriction>
   <owl:onProperty rdf:resource="#hasFlavor"
/>
     <owl:hasValue rdf:resource="#Moderate" />
   </owl:Restriction>
  </rdfs:subClassOf>
  <owl:intersectionOf
rdf:parseType="Collection">
   <owl:Class rdf:about="#WhiteBurgundy" />
   <owl:Restriction>
   <owl:onProperty rdf:resource="#locatedIn"
/>
     <owl:hasValue
rdf:resource="#MeursaultRegion" />
   </owl:Restriction>
  </owl:intersectionOf>
 </owl:Class>
```

we obtain the following triplets:

<Mersault, hasBody, Full>

<Mersault, hasFlavor, Moderate>

<Mersault, isA, WhiteBurgundy>

<Mersault, locatedIn, MersaultRegion>

A special case is represented by the constraints over the properties: restriction value, cardinality. As we can observe from Table 1, if we have a cardinality restriction on a property we transform in the triplet <S, Property , n>. In the case of a “AllValuesFrom” or “SomeValuesFrom” restriction we obtain an intersection and union respectively, of triplets, <S, P, O_i>, where O_i belongs to the set of restriction values.

Property constraints	Resulted triplet
<owl: Class A> <owl:onProperty rdf:resource=#Prop /> <owl:maxCardinality> n </owl:maxCardinality>	<A, maxCard(Prop), n>
<owl: Class A> <owl:onProperty rdf:resource=#Prop /> <owl:allValuesFrom> <owl:oneOf Collection <owl : Thing rdf:about=#B> <owl: Thing rdf:about=#C>	<A, Prop, AND(B, C)>
<owl: Class A> <owl:onProperty rdf:resource=#Prop /> <owl:SomeValuesFrom> <owl:oneOf Collection> <owl : Thing rdf:about=#B> <owl: Thing rdf:about=#C>	<A, Prop, OR(B, C)>

Table 1. The resulting triplets for property restrictions: cardinality and value

Generate sentence plan

The obtained triplets represent the base for generating the sentences, using a template for each property. This template specifies certain patterns and parts used in the specific sentence for each property(verb, prepositions), the sentence structure and allows the filling of fields with information received from the triplets [5]. A model of sentence plan is the following:

Subject_Expression(article, number) – Verb(voice) – Nouns / Adjectives(optional) – Object_Expression.

Property: locatedIn

Template: Subject_Expression – “to locate”(passive voice) – in(preposition) – Object_Expression.

In the case of properties which also hold constraints, it is generated a plan according to the type of restriction. For instance, using the same property “locatedIn”, but also having the maximum cardinality restriction, the result will be:

Subject_Expression – “to_locate”(passive voice) – in(preposition) – “at most” – n – “regions: “ – Object_Expression. (“LaneTannerPinotNoir is located in at most 2 regions: SantaBarbara and Sonoma.”)

Generate referring expressions

The subject and the object of a triplet can have different forms inside a sentence. Initially, they are obtained through the tokenization of the subject or object of the triplet or using the ontology field rdfs:label, a string which represents the natural language form of that concept. Then, in a similar manner to sentence plan we can provide a template for the subject and the object. This plan indicates what kind of article should be used with this noun(definite, indefinite), its number and optional adjectives. Moreover, to avoid the repetition of the subject, the noun can be replaced in the following sentences with a pronoun, a determiner or a demonstrative pronoun:

<Zinfandel, hasColor, Red>

<Zinfandel, hasFlavor, Strong>

We obtain: “A zinfandel has red color. It/ This wine has strong flavor.”

Sentence aggregation

An important step towards reaching the final text is represented by the text aggregation and the use of Rhetorical Structure Theory. The text aggregation is utilized to obtain concise text and can be applied at the identification of certain patterns [1]. The aggregation is possible due to the order of the resulted triples. We can group sentences which express axioms at the same level of relation compared to the main class. Therefore, this stage can only occur if the triplets are sorted before, according to the same subject, or property. In the case of triplets of the form: <S, P, O_1>, <S, P, O_2> , they are replaced with the triplet <S, P, O_1 and O_2>.

If we have the triplets <S, P_1, O_1> and <S, P_2, O_2> and P_1 and P_2 are equivalent, they are replaced with <S, P_1, $O_1$and O_2>.

Aggregation can also occur in the situation in which we have triplets with different properties and objects, but with the same subject: <S, P_1, O_1>, <S, P_2, O_2>, <S, P_3, O_3>:

<Muscadet, hasBody, Light>

<Muscadet, hasFlavor, Delicate>

<Muscadet, hasSugar, Dry>

We obtain: “Muscadet is light, it has delicate flavor and it is dry.”

In the case we have 2 triplets which refer one property and then a retriction of the same property, the 2 triplets combine in the manner suggested by the following example:

<StEmilion, exactCard(madeFromGrape), 1>

<StEmilion,madeFromGrape, CabernetSauvignon>

Result : “StEmilion is made from exactly 1 grape: Cabernet Sauvignon.”

Rhetorical Structure Theory

The Rhetorical Structure Theory is employed to create a tree hierarchy of sentences or part of sentences linked with rhetorical relations to ensure the coherence of the resulted text [4]. The Rhetorical Structure Theory helps to emphasize the rhetorical relations between clauses or sentences, using “discourse markers”. These relations describe the intention of the clauses on the reader and can be: Condition, Antithesis, Justification, List, Contrast [8, 9]. However, considering the content of an ontology and its informational purpose, only part of these relations can be applied on the ontological content.

The “list” relation can be used to suggest equality in the text hierarchy between different members or axioms. The “elaboration” relation may indicate additional information about a class, whereas the “concession” relation, indicating a contradiction between statements, may be used to link with the description of related classes. Finally, the “condition” relation is applied in the case of restricted properties [6]. Considering a given concept, this will become the root of the tree structure and the main nucleus of the Rhetorical Structure Theory schema. Next, we discover the subclasses and superclasses of this concept, which will be placed in a relation of “elaboration”. Since superclasses and subclasses occupy the same level in the hierarchy, they will be grouped in a rhetorical relation of “list”, highlighted by the markers: “,”, “and”. Then, we describe the properties of that class and in the case of restriction on properties, we place them in a relation of condition, using the demonstrative pronoun “that” or the pronoun “which”.

The elements of the “allValuesFrom” restriction on a property are in a relation of “elaboration” to the main sentence, using the marker “:”. Lastly, to add information about a linked class, the following markers are used: “Additionally”, “Furthermore”, placing these statements in a relation of “elaboration” with the main nucleus [6].

SUGGESTIONS FOR FUTURE WORK

The main drawback of this project is represented by the static adding of domain dependent information for each ontology we want to obtain natural language text. This operation depends on the number of properties in the ontology and has to be updated manually if the ontology is modified or improved with new knowledge. A future improvement could be the integration of a global database called lexicon to obtain different forms of verbs, nouns [5]. Additionally, each template for a property could be saved in a new ontology which can be used in similar projects in the same field.

Because there is no precise technique to evaluate the output obtained, we should show samples of outputs to different users to rate the resulted texts, according to how useful and easy to follow they consider the text.

Additionally, we can produce better outputs if we would consider trials with different methods involved, and based on the feedback received from users we would choose the proper combination of techniques.

CONCLUSIONS

We have provided a description for a natural language generation application which aims at producing fluent, coherent text describing a class or individual from a selected OWL ontology. In order to achieve accessible texts our application uses different techniques such as: Rhetorical Structure Theory, domain dependent information, or sentence aggregation.

The field of Natural Language Generation is a relatively new research topic, but of great perspective in the context in which the Semantic Web is in a continuous expansion, and the user could benefit from new-created or modified ontologies to receive knowledge in a easy to assimilate manner.

REFERENCES

1. Kalina Bontcheva. 2004. Automatic Report Generation from Ontologies: the MIAKT approach. *9th International Conference on Applications of Natural Language to Information Systems.*
2. Nadjet Bouayad-Agha. 2012. From ontology to NL: generation of multilingual user-oriented environmental reports. *Proceedings of the 17th international conference of Applications of Natural Language Processing and Information Systems.*

3. Roger Evans, Paul Piwek and Lynne Cahill. 2002. What is NLG?. *Proceedings of Second International Natural Language Generation.*
4. Eva Forsbom. 2005. *Rhetorical Structure Theory in Natural Language Generation.* GSLT:Natural Language Generation Lecture.
5. Gerasimos Lampouras, Dimitrios Galanis and Ion Androutsopolous. 2013. Generating Natural Language Descriptions from OWL Ontologies. *Journal of Artificial Intelligence Research*.
6. Shao Fen Liang, Donia Scott, Robert Stevens and Alan Rector. 2011. *Unlocking Medical Ontologies for Non-Ontology Experts*. Proceedings of BioNLP Workshop.
7. William C. Mannn and Sandra Thompson. 1987. *Rhetorical Structure Theory: Toward a functional theory of text organization*. ISI: Information Sciences Institute
8. Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. MIT Press.
9. Nick Nicholas. 1994. *Problems in the Application of Rhetorical Structure Theory to Text Generation.* M.Eng.Sc.Degree *Thesis, University of Melbourne*