Entendre: Interactive Semantic Feedback for Ontology Authoring

Ronald Denaux, Dhaval Thakker, Vania Dimitrova, and Anthony G. Cohn

School of Computing,
University of Leeds,
Woodhouse Lane, Leeds, LS2 9JT, UK
{r.denaux,d.thakker,v.dimitrova,a.g.cohn}@leeds.ac.uk

Abstract. This demonstration presents Entendre, a framework to analyse ontology authors' inputs and provide meaningful feedback at a semantic level. The feedback aims to make ontology authors aware of potential issues such as inconsistency, class unsatisfiability, unexpected logical implications, redundancy and isolated entities. The implementation of Entendre that will be demonstrated, extends a CNL-based ontology authoring environment, allowing users without prior knowledge engineering experience to build ontologies while becoming aware of the implications of the formal semantics of OWL. An initial evaluation shows that the feedback is helpful to both novice and experienced ontology authors.

1 Introduction

The Semantic Web relies on formal ontologies to structure data for comprehensive and transportable machine understanding [5]. Hence, ontologies are vital for the Semantic Web's success and wider adoption in a range of domains. In practice, ontology creation is a challenging task requiring specialism in knowledge engineering, logical background and domain expertise. However, domain experts usually require extensive training (or tool support) in order to use ontology authoring tools; and when they are able to enter syntactically correct ontology constructs, the resultant ontologies may still contain logical defects.

It has been argued that ontology authoring tools should provide intuitive interfaces for entering ontology constructs, which has lead to a stream of research in using Controlled Natural Languages (CNL) for ontology authoring. Our previous work[1] pointed out that CNL interfaces *still require additional support* in order to reduce the cognitive complexity of ontology authoring and improve efficiency. We developed a tool called ROO which enabled the creation of OWL ontologies using a CNL, and provided syntactic support for editing ontology constructs. Our recent practical experience, where a team of domain experts and knowledge engineers developed ontologies using ROO[1], has confirmed that the semantic aspect of ontology authoring (overseeing the logical consequences

[1] this happened in the context of two European projects: ImREAL http://www.imreal-project.eu and Dicode http://www.dicode-project.eu

of adding OWL axioms to an ontology) is still a major burden. The knowledge engineers had to spend considerable time rectifying the ontologies produced by domain experts, mainly relating to logical implications such as concept satisfiability, inconsistency, redundancies and unintended entailments. The rectification of such issues is often difficult, due to the logical dependencies between ontological constructs.

Entendre, the sytem to be demonstrated, aims to increase the efficiency and effectiveness of the ontology authoring process by providing interactive, semantic feedback that prompts ontology authors to consider logical consequences of the entered facts. Such functionality becomes even more important now, as there is a growing interest in linked data [3] and a push for iterative, collaborative ontology development that favours reusability [2,4]. In this development style, it is especially important to be aware of the logical implications while contributing or expanding existing facts.

2 The Entendre Framework

Entendre is a framework that defines how to *understand* input axioms in order to provide appropriate *semantic feedback*. Entendre defines *axiom categories* depending on the effect of an input axiom on the ontology being constructed. Each axiom category specifies the semantic feedback (and potential semantic issues) that can be associated with the input axiom. Thus, given a consistent ontology $\mathcal{O}$ and an axiom α in the ontology language, Entendre diagnoses the impact of adding α to $\mathcal{O}$. We now semiformally[2] describe the main axiom categories and state how we detect them; we also describe their related semantic feedback[3].

Axioms to be added to an ontology can be either known or novel: α is **known** by $\mathcal{O}$ when α is entailed by $\mathcal{O}$, otherwise α is **novel**. Known axioms can be split into two categories – **asserted** and **inferred**.

(A) Asserted Axiom
Detection: $\alpha \in \mathcal{O}$ **Feedback:** α is already in $\mathcal{O}$. **Defect warning:** Adding α to the ontology $\mathcal{O}$ is not needed.

(R) Inferred Axiom
Detection: $\alpha \notin \mathcal{O}$, but α can be inferred from $\mathcal{O}$ **Feedback:** α is redundant as it can be inferred from $\mathcal{O}$. A set of axioms in $\mathcal{O}$ that implies α is the justification $\mathcal{J}(\alpha, \mathcal{O})$. **Defect warning:** Adding α to $\mathcal{O}$ causes redundancy. Check the justification.

Adding a *novel axiom* to $\mathcal{O}$ can make the ontology **inconsistent** or can **introduce an unsatisfiable concept**. Even when this is not the case, adding

[2] Formal definitions of the categories are out of scope for this demonstration paper.

[3] Note that the feedback is intended to *inform* authors about potential issues (it may not provide all the information necessary to *resolve* the issues). Because of this we often show one (randomly selected) justification of an entailment in the feedback, even if there are multiple justifications for an entailment.

the novel axiom always leads to an infinite number of **new implications**: axioms that could not be inferred from $\mathcal{O}$ or from α alone but that can be inferred from the combination of $\mathcal{O}$ and α. We define a *finite* subset of these new implications that we consider **relevant**, $\Lambda^{\text{newRelevantImplications}}$; we do this by restricting ourselves to axioms that only contain concept expressions and individuals that already appear in $\mathcal{O}$ or in α. This leads to the following categories:

(I) Axiom Leading to Inconsistency
Detection: $\mathcal{O} \cup \{\alpha\}$ is inconsistent.
Feedback: α is novel to $\mathcal{O}$. Adding α to $\mathcal{O}$ leads to an inconsistent ontology. The set of axioms in $\mathcal{O}$ that implies $\neg\alpha$ is the justification $\mathcal{J}(\neg\alpha, \mathcal{O})$.
Defect warning: Check justification in feedback.

(U) Axiom Introducing Unsatisfiable Concept
Detection: The set of concepts that are unsatisfiable in $\mathcal{O} \cup \alpha$ minus the unatisfiable concepts in $\mathcal{O}$ (denoted by $\Theta^{\text{newUnsatisfiable}}$) is not empty.
Feedback: α is novel to $\mathcal{O}$. Adding α to $\mathcal{O}$ makes the concepts $\Theta^{\text{newUnsatisfiable}}$ unsatisfiable. For each concept $C \in \Theta^{\text{newUnsatisfiable}}$, the set of axioms that makes C unsatisfiable is the justification $\mathcal{J}(C \equiv \bot, \mathcal{O} \cup \{\alpha\})$.
Defect warning: Check justification in feedback.

(N) Novel Axiom without new Relevant Implications
Detection: $\Lambda^{\text{newRelevantImplications}} = \emptyset$
Feedback: α is novel to $\mathcal{O}$ but does not bring new relevant implications.
Defect warning: If any new entailments were expected, α should be reviewed or $\mathcal{O}$ may have to be extended.

(N+) Novel Axiom with new Relevant Implications
Detection: $\mathcal{O} \cup \{\alpha\}$ is consistent and $\Lambda^{\text{newRelevantImplications}} \neq \emptyset$
Feedback: α is novel to $\mathcal{O}$. Adding α to $\mathcal{O}$ brings the set of new relevant implications $\Lambda^{\text{newRelevantImplications}}$.
Defect warning: Check that there are no missing or unexpected implications.

3 Entendre in an Ontology Authoring Tool

During the demonstration session, we will show an implementation of Entendre that has been integrated in ROO (Rabbit to OWL Ontology authoring) [1]. ROO allows authors to edit ontologies using the Rabbit controlled natural language (a restricted subset of English that can be converted into OWL) [1]. When authors add knowledge in ROO, they write Rabbit sentences using a *Rabbit editor*, which provides syntactic feedback to help users compose valid Rabbit sentences. Once the sentence is parsed correctly, the author can accept the sentence, which is converted to OWL axioms and added to the ontology.

Entendre extends the Rabbit editor in ROO by providing *semantic feedback*. Before the author adds a Rabbit sentence, it is converted to an OWL axiom, categorised by Entendre and appropriate semantic feedback is generated. ROO then shows textual explanations based on pre-defined templates.[4] For the demonstration, we will prepare an ontology and example inputs to showcase the feedback

[4] ROO uses the feedback to *advise* on a course of action, not to *enforce* it.

for all of the axiom categories. Below, we present an example of the semantic feedback provided by ROO based on an ontology about points of interest in Leeds; in this case the input axiom introduces an unsatisfiable concept.

Rabbit Input: *Every Student Union is contained within a University(Institution).*

ROO Feedback: This sentence makes concept `Student Union` **unsatisfiable!** This means that nothing can be a `Student Union` anymore. **Advice** You should not add an unsatisfiable concept to an ontology because this concept becomes practically unusable. This is especially true if you make a concept unsatisfiable and that concept was defined by somebody else, as you are probably not using the concept in the way it was intended.

Check the list of contradicting sentences:
- *Organisation and POI are mutually exclusive.*
- *Every Student Union is contained within a University (Institution).*
- *Every University (Institution) is a kind of Organisation.*
- *The relationship contains must have subject POI.*
- *The relationship is contained within is the inverse of contains.*

A recent evaluation of Entendre to investigate ontology authors' opinions about the semantic feedback had very encouraging results. The study materials, including ROO and used ontologies, are available online [5] and can be used to demonstrate the system. We are working on a stable release of the tool and may use a different ontology (e.g. points of interest in Bonn) during the demo session.

It should be noted that the current implementation relies heavily on ontology reasoners and is dependent on their scalability (currently, working well only for small or medium size ontologies). As future work, modularisation strategies will be investigated to address possible scalability problems, e.g. providing reasoning based on a relevant subset of the whole ontology.

Acknowledgements The PhD studies of the first author are funded by the University of Leeds and Ordnance Survey. The work is partially supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no ICT 257831 (ImREAL project) and grant agreement no ICT 257184 (DICODE project).

References

1. Denaux, R., Dolbear, C., Hart, G., Dimitrova, V., Cohn, A.G.: Supporting domain experts to construct conceptual ontologies: A holistic approach. Web Semantics: Science, Services and Agents on the World Wide Web (Feb 2011)
2. Grau, C., Horrocks, B., Kazakov, I., Sattler, Y.: Modular reuse of ontologies: Theory and practice. J. of Artificial Intelligence Research 31(1) (2008)
3. Hendler, J.: Why Semantic Web will never work (2011), `http://videolectures.net/eswc2011_hendler_work/`
4. Holsapple, C.W., Joshi, K.D.: A collaborative approach to ontology design. Communications of the ACM 45(2), 42–47 (Feb 2002)
5. Maedche, A., Staab, S.: Ontology learning for the Semantic Web. IEEE Intelligent Systems 16(2), 72–79 (Mar 2001)

[5] `http://www.comp.leeds.ac.uk/confluence/Entendre-Study`