Simplifying a General Ontology Using Instances

Farhad Mostowfi[1], Farshad Fotouhi[1], and William Grosky[2]

[1] Department of Computer Science, Wayne State University, Detroit, Michigan
{fmostowfi, fotouhi} @wayne.edu
[2] Department of Computer and Information Science, University of Michigan-Dearborn, Dearborn, Michigan
wgrosky @umich.edu

Abstract. In this paper, we propose a method to simplify a general morphosyntactic ontology based on its current set of instances for a specific language. The set of instances are collected through ontology-based annotation. After annotation, a simplification process removes unused classes, relations and other artifacts from the ontology. The benefit of this approach is that we can quickly create a morphosyntactic ontology for the focus language that can be used for other purposes including audio, image and video annotation.

1 Introduction

Linguists mark up documents to preserve their linguistic information and content. Instead of using plain text to describe items of interest in a text, they can use concepts from a general morphosyntactic[1] ontology such as GOLD[2] to describe a paragraph, a phrase, a word or a morpheme. This is called ontology-based annotation. A regular annotation is a plain text that is collected based on a fixed structure [2], while ontology-based annotation is a set of instances of classes based on the domain ontology. In ontology-based annotation, user assigns the annotated text to a concept in the ontology (instantiating a class) or to a data type or relates it to another annotated text (instantiating a relation) [1, 2].

Recently there has been interest toward marking up documents in languages that are in the brink of extinction using new Web technologies. Languages that are in danger of disappearing make up half of all the 6500 languages spoken around the world. These are languages with less than 10000 speakers that economical and geopolitical realities make their survival difficult[3]. When a language dies, its cultural heritage and scientific achievements would be lost as well. It is important to implement digital archiving infrastructure that helps linguists to document the language

[1] Morphosyntax covers relation between morphology and syntax. Morphology is the study of word structure and rules that apply in each language to relate a word to other words like relating cat to cats in plural formation. Syntax is the study of rules (such as word arrangement) that make a clause and then a sentence.

[2] Linguistic Ontologies and Data Categories for Language Resources. http://emeld.org/workshop/2005/ontology.html

[3] Foundation for Endangered Languages. http://www.ogmios.org/manifesto.htm

before it vanishes. We regard the morphosyntactic ontology of the language as the core of the digital repository for the language. Using ontology in annotating archived text gives the linguist freedom in describing data while adhering to a formalized definition of concepts. Another application of this ontology is in annotating sounds, images, audio and video material related to the language. This ontology also provides the interoperability that is required to compare different languages. It provides compatibility among datasets collected from different languages and makes integration between different sources of information possible.

As an example, consider number classification system in different languages. Some languages have nullar number (zero instance of referent); others have dual number or trial number (for three instances of referent). Some languages have paucal number (a few instances of the referent) and some have collective number (many referents viewed as a single collection)[4]. We envision a system that can answer queries such as "find all languages that have trial numbers?" or "Which language have a distinction between dual and paucal number"

Currently there is no system available to answer these kinds of cross-linguistic comparisons since there is no way to integrate different language datasets. There is also no semantic mapping between markup systems. Our position is that we can create specific morphosyntactic ontology for each language using the general morphosyntactic ontology. The general ontology is being developed by a large community of linguists and has a wide scope that covers most if not all the languages[5]. We propose using this ontology to annotate documents in any language. The annotation is done using an annotation tool like OntoGloss[3] that we have developed to annotated text with concepts from ontology. A simplification process applies to the ontology to simplify it based the current repository of instances. The result is the morphosyntactic ontology for the specific language. This simplified ontology that closely represent the focus language can be used for other purposes including annotating audio and video material. Fig. 1 shows the process of simplifying the general ontology for a language.

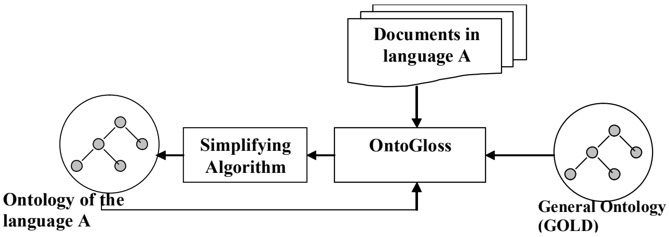

Fig. 1. Simplifying the general ontology for a language

Linguist uses OntoGloss to annotate sample text documents in the focus language. OntoGloss is a stand-off[6] annotator that annotates documents at every granu-

[4] Wikipedia (http://en.wikipedia.org/wiki/Grammatical_number)

[5] Markup and the GOLD Ontology (http://emeld.org/workshop/2003/paper-terry.html)

[6] Stand-off annotator keeps annotation separate from the annotated document.

larity level, from the document level down to the morpheme level. User annotates textual documents with classes and relations from the general ontology. Annotated data are instances of the ontology. In the simplification process, these instances determine which class should stay and which one could be removed. The simplifying algorithm makes sure to keep all the relevant artifacts of the original ontology intact while retiring those that are not needed. In the following sections, we first explain about the annotation process using OntoGloss and then the simplification process through the simplifying algorithm.

1 Annotation Process

User loads the general ontology into OntoGloss and marks up documents, paragraphs, sentences, words and morphemes. Annotation is done through the drag and drop operations. A color-coding scheme is used to give visual clue to the linguist on the type of each markup. OntoGloss can automatically annotate new documents based on the previously annotated documents. During auto annotation process, it compares each word in the document with all the annotated text in the database and assigns the same type of annotation to words or morphemes. This will serve as an initial suggestion and can change by the linguist if needed. OntoGloss is able to use lexical reference systems like WordNet[7] as a resource during annotation process to provide synonymy, hyponymy and different senses for individual words. For languages other than English, this lexical reference system can be built and added gradually within the OntoGloss. OntoGloss exports annotation information in triple format. These triples can be loaded into an RDF repository with querying and reasoning capabilities. Using RDF as the main storage and exchange method makes knowledge in the field portable to other applications and readable by machine as well as by human. Each annotated document is linked to a language code, so that one can extract all material on a particular language.

2 Simplifying Algorithm

Usually languages do not use all the concepts, relations or other artifacts in the general ontology. Therefore, ontology of each language is a subset of the general ontology. These specific ontologies would share the same artifacts but each one has its own unique set of them. As linguist continues to annotate documents in that language, a pattern would emerge on which of the classes are not needed and can be removed from the general ontology for that language. For example, if a language does not have feminine or masculine in referring to the third person, these concepts do not have any instance and would be removed. In other words, if a class does not have an instance, it means it does not apply to the language or an instance has not been found yet.

[7] WordNet: A lexical database for English. http://www.cogsci.princeton.edu/~wn/

Following are few simple rules that apply in removing classes, relations and other constructs. This list is not exhaustive and other rules could be added. Constructs that are not referred by any instance would be removed in presentation, although they stay in the ontology in case new instance refer to them later.

Rule 1 (Removing a Class). If a class does not have any instance of its own but has children with instances, and it is not the domain or the range of an instance, it can be removed. Its children will have the parent(s) of the removed class as their parent(s). This will change the rdfs:subClassOf statement for the children and rdfs:domain and rdfs:range for properties that were referring to the removed class. Fig. 2 shows the removal of class B, which does not have any instance of its own. In this figure, classes are represented in a circle and instances in a small rectangle. Children of the removed class (C and D) have a new parent (A) which used to be the parent of the removed class.

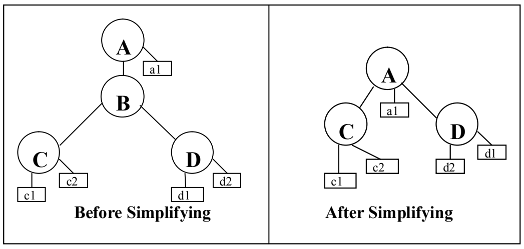

Fig. 2. Removing a class

Rule 2 (Removing a Property). A property, either data-type property or object property, transitive or symmetrical, can be removed when it does not have domain or range reference by any instance and also does not have any sub-property referring to it. Some properties might be generalized or specialized. When a property's domain or range is removed and one of the parents replaces it, the property is generalized. If one of the children replaces the removed domain or range, it is specialized. Fig. 3 shows property $\Re$ being generalized as its new range is the parent of the old range class.

Rule 3 (Removing Cardinality Constraints). Restricted cardinalities such as owl:minCardinality and owl:maxCardinality are defined as properties of the particular classes and therefore are removed with the class.

Rule 4 (Removing Intersection and Union Constructs). owl:intersectionOf and owl:unioinOf constructs are removed when only one class in the intersection or union remains in the ontology.

Rule 5 (Removing Equality Constructs). Equality constructs create synonymous classes or properties. owl:equivalentClass and owl:equivalentProperty are removed when one of the classes and properties in the equality relation is removed.

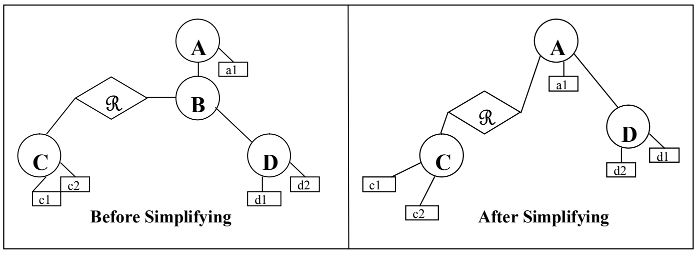

Fig. 3. Removing a class and generalizing a relation

3 Conclusion and Future Works

In this introductory paper, we informally presented a few rules to simplify general morphosyntactic ontology for a specific language based on the available instances. The simplification process is based on two observations. First, the morphosyntactic ontology of any language is a sub-set of a general morphosyntactic ontology like the GOLD. Second, experimental annotated text in different languages has shown that most of them use only a fraction of classes, properties and other constructs in the general ontology. The simplification process helps us rapidly develop ontology for each of the more than 6500 spoken languages. These ontologies are the core of the digital archive library for these languages. They can be readily used to annotate documents in these languages or can be promoted for applications in audio and video annotation. In the future, we intend to formalize the simplification algorithm with mathematically sound rules that can be implemented in a reasoner. We also need to experiment with a bigger sample of annotated data and check the suitability of the simplified ontology for other applications.

References

[1] Cimiano, P., and Handschuh, S., Ontology-based Linguistic Annotation, Proceedings of the ACL Workshop on Linguistic Annotation, 2003, Sapporo, Japan.

[2] Handschuh, S., Staab, S., and Maedche, A., CREAM- Creating relational metadata with a component-based, ontology driven annotation framework, Proceedings of the International Semantic Web Working Symposium, 2001, California, USA.

[3] Lewis, W., Farrar, S. & Langendoen, T., Building a Knowledge Base of Morphosyntactic Terminology, Proceedings of the IRCS Workshop on Linguistic Databases, 150-156, 2001, Philadelphia, PA.

[4] Mostowfi, F., Fotouhi, F., Aristar, A., OntoGloss: An Ontology-based Annotation Tool, http://emeld.org/workshop/2005/papers/mostowfi-paper.doc.