9.2: Ontology Verbalisation

The second topic of ontologies & natural language—ontology verbalisation—can build upon the previous, but need not if you happen to be interested in a grammatically ‘simple’ language, such as English. The core notion is to generate (pseudo-)

![Figure 9.2.1: Annotated small section of part of the FOAF in Chichewa. (Source: http://www.meteck.org/files/ontologies/foaf.ttl, discussed in [CK14]).](http://www.meteck.org/files/ontologies/foaf.ttl)

natural language sentences from the axioms in the ontology. The introduction to Block III already noted that ontology verbalization may be useful for, mainly:

- Ameliorating the knowledge acquisition bottleneck; it:
  - helps the domain experts understand what exactly has been represented in the ontology, and thereby
provides a means to validate that what has been represented is what was intended;
– can be used to write axioms in pseudo-natural language, rather than the logic itself, to develop the ontology.

• Some ontology-driven information system purposes; e.g., e-learning (question generation, textbook search), readable medical information from medical terminologies in electronic health record systems.

The term ‘ontology verbalisation’ is rather specific and restricted, and it falls within the scope of Controlled Natural Language (CNL) and Natural Language Generation fields of research (NLG). While most papers on ontology verbalization will not elaborate on the methodological approach specifically, there are clear parallels with the activities of the traditional NLG pipeline alike proposed in [RD97]. The pipeline is summarized in Figure 9.2.2, top section, and the corresponding answers for ontology verbalization are described in the bottom section of the figure.

Template-Based Approach

There are three principal approaches to generate the sentences: canned text, templates, and a grammar engine. The most commonly used approach for ontology verbalization is templates, as it generally takes less effort to start with and therewith one may reap the low-hanging fruit. Let’s look at two examples (analyzed afterward):

(S1) \texttt{Giraffe} \sqsubseteq \texttt{Animal}

\textcolor{blue}{\underline{Each}}: \textcolor{red}{\texttt{Giraffe}}: \textcolor{green}{\underline{is an}}: \textcolor{red}{\texttt{Animal}}

(S2) \texttt{Herb} \sqsubseteq \texttt{Plant}

\textcolor{blue}{\underline{Each}}: \textcolor{red}{\texttt{Herb}}: \textcolor{green}{\underline{is an}}: \textcolor{red}{\texttt{Plant}}

The underlined text ‘Each’ is the natural language rendering of the silent ‘\(\forall\)’ at the start of the axioms, and ‘is
a(n)' for “\(\sqsubseteq\)”, which remain the same, and the vocabulary from the axiom is inserted on-the-fly by reading it in from the ontology file. One can construct a template for an axiom type with those pieces of text that remain the same for each construct, interspersed with variables that will take the appropriate entity from the ontology. For instance, for this named class subsumption axiom type \((C\sqsubseteq D)\), one could declare the following template:

\[
(\texttt{Each}: \texttt{class}_{1}: \texttt{is a(n)}: \texttt{class}_{2})
\]

and for the basic all-some axiom type \((C\sqsubseteq \exists R.D)\) that we have come across in Chapter 7, either one of the following ones may be selected:

\[
(\texttt{Each}: \texttt{class}_{1}: \texttt{op}: \texttt{at least one}: \texttt{class}_{2})
\]

\[
(\texttt{Each}: \texttt{class}_{1}: \texttt{op}: \texttt{some}: \texttt{class}_{2})
\]

\[
(\texttt{All}: \texttt{class}_{1pl}: \texttt{op}_{inf}: \texttt{at least one}: \texttt{class}_{2})
\]

\[
(\texttt{All}: \texttt{class}_{1pl}: \texttt{op}_{inf}: \texttt{some}: \texttt{class}_{2})
\]

The subscript "\(_{pl}\)" indicates that the first noun has to be pluralized and "\(_{inf}\)" denotes that the object property has to be rendered in the infinitive. The algorithms tend to be based on the assumption that the ontology adheres to conventions of naming classes with nouns in the singular and object properties in 3rd person singular. It is an implementation detail whether a plural or the infinitive is generated on-the-fly from grammar rules or fetched from a Lemon file or another annotation or precomputed lexicon.

One could declare templates of each axiom type, but because it is ‘arbitrary’ within the syntax constraints of the logic, a modular approach makes sense. Essentially, each symbol in the language has one or more ways of putting it in natural language, and based on that, one can create a controlled natural language that can generate only those sentences following the specified template. For instance, a template for \((C\sqcap D)\), i.e., “\((\texttt{class}_{1}): (\texttt{and}): (\texttt{class}_{2})\)” can be fetched as needed when it appears in a class expression such as \((E\sqsubseteq \exists R.(C\sqcap D))\). In this way, then a whole ontology can be verbalized and presented as (pseudo-)natural language sentences that hide the logic. Of course, it takes for granted one already has an ontology in one’s preferred language.

Two recent review papers describe the current state of the art with respect to NLG and the Semantic Web [BACW14] and CNLs [SD17]. Both focus principally on English, largely because there are only few efforts for other languages. Some 13 years ago we tried to create a verbalizer for multiple languages, which worked to some extent [JKD06]; that is, the technology did, but not all sentences were fluent. The main lessons learned then, and which hold just as much now, were: 1) it is easier to generate a new set of templates in a language based on an existing set of templates for a similar language, and 2) the template-based approach is infeasible for grammatically richer languages. Regarding the second aspect, it may be that a few additional rules suffice, which could be added to the same template file, even, or one could perhaps repurpose SimpleNLG [GR09], which has been adapted from English to French, Spanish, and Portuguese. For a language such as isiZulu, however, there are, thus far, no known templates, but only ‘patterns’; that is, for the patterns designed for verbalizing ontologies (roughly within ALC), there is no single canned word in the ‘template’ as each word in the sentence requires some processing with language tags and grammar rules [KK17a].
Reusing the Results for Related Activities

Having the route from axiom to (pseudo-)natural language sentence, one could wonder about the process in the reverse, i.e., using the CNL to generate the axioms. The most successful results for ontologies have been obtained with Attempto Controlled English (ACE) [FKK10] for OWL DL. A related line of work within the scope of from-text-to-axiom, is natural language-based query formulation (e.g., [FGT10]), which also uses a CNL. In this scenario, however, partial completions of an iteratively constructed sentence to query the ontology are based on the step-wise selected vocabulary and the (partial) axioms in which it appears.

One might take another step further: use machine translation in the verbalization process. Such a system has been developed by [GB11]: it uses ACE for ontology verbalization in English and Grammatical Framework for translations between English and Latvian. Which option is the ‘better’ one is not fully clear: a) make templates in one’s language and translate the ontology, or b) skip the template specification by reusing an existing English one, develop a resource grammar for Grammatical Framework, and translate the generated English into the target language. This depends on one’s aims and availability of resources. In theory, the Grammatical Framework should work better as a reusable solid solution but with (very) high start-up costs, whereas templates are easier to specify but the knowledge that goes into it is not easily reused elsewhere. Of course, the latter may not be useful if the ontology has its vocabulary elements already in the target language.

Ontology verbalization can also be used for, e.g., automatically generating documentation of one’s ontology, in a similar fashion as automated code documentation generation. A step toward interesting applications is, e.g., the ontology-enhanced museum guide (in Greek) [ALG13], question generation for textbooks in biology that were annotated with an ontology [CCO+13], and language learning exercises [GK18].