Two Protégé plug-ins for supporting document-based ontology engineering and ontological annotation at document-level

Viktoria Pammer and Peter Scheir and Stefanie Lindstaedt

May 1, 2007

Abstract

We present two plug-ins for Protégé OWL. The first, Discovery Tab, supports document-based ontology engineering with relevant term extraction, clustering and related functionality. The second, Annotation Tab, provides a facility to annotate documents manually and automatically (based on a training set).

1 Introduction and motivation

Knowledge is available in the minds of human experts. One way of externalizing this knowledge is in the form of documents written in natural language. Document-based ontology engineering aims at using information available in documents to create formal knowledge representations. Done manually, it saves time of the domain experts but not of the ontology engineers. Automatically done it also saves time for the ontology engineers.

Among many things, an ontology can be used as a kind of controlled vocabulary for annotation of entities. Annotation is again a time-consuming task. It can be partly automated however, except for the creation of a training set.

2 Supporting document-based ontology engineering

The Discovery Tab offers functionality to discover knowledge from text-documents. Documents can be grouped to form clusters of different topics, which gives an overview of the domain covered by the documents. Clusters of documents can also be labelled. Relevant terms can be extracted from a set of documents using statistical and natural language processing methods. Terms can be grouped by synonymity. A number of other functionalities like phrase or relation extraction and hierarchical clustering are possible but are not currently implemented.

Within Discovery Tab, standard text mining methods have been used. For relevant term extraction, various pre-processing techniques, such as stop-word lists and stemming, are employed. Extracted terms are weighted by TFIDF. Grouping according to synonymity is currently available only for English and

uses WordNet [1] and Apache's Wordnet package [2]. The grouping of documents is a mixture of standard clustering algorithms like Hierarchical Agglomerative Clustering (for seeding) and k-means.

One possible workflow for creating ontological elements from a set of documents could be: Group documents into topic-clusters. This gives an overview over topically different subsets of the domain. Within each cluster, extract relevant terms. To arrange results more clearly, group these terms according to synonymity. Finally, from each term, create either classes or properties.

3 Ontological annotation at document-level

The Annotation Tab offers two functionalities: First, to annotate documents with concepts (actually classes and instances). Second, to automatically suggest concepts for a document if a training set of annotated documents is given. At its backend the Annotation Tab uses a text classification algorithm (k-Nearest Neighbor). This means, that it suggests a set of concepts for a document based on a training set of previously annotated documents. The plug-in presented here shares the backend and a common approach to annotation with the work presented in [2]. However, the frontend as well as the modeling of annotation has been largely revised.

In our approach to semantic annotation we do not aim at identifying concepts at word level. Instead, we follow the tagging metaphor and annotate whole documents with the concepts they deal with. This pragmatic approach follows a motto of which Hendler reminds the Semantic Web community in [1]: "A little semantic goes a long way". In this case, we think that for lots of applications it is not only sufficient but totally sensible to annotate at document level.

4 Conclusion

The interest of our work for a wider scientific community lies first in having grouped together a number of standard text-processing methodologies within one tool. Second it lies in the fact that this tool is embedded in a standard ontology engineering environment, namely Protégé, for which a number of complementary plug-ins (e.g. for rules, querying, visualization etc.) are available. Altogether, this is one further step towards a more and more complete environment in which to implement ontologies.

References

[1] James Hendler. The dark side of the semantic web. *IEEE Intelligent Systems*, 22(1):2–4, 2007.

[2] Peter Scheir, Philip Hofmair, Michael Granitzer, and Stefanie N. Lindstaedt. The ontologymapper plug-in: Supporting semantic annotation of text-documents by classification. In *Proceedings of the SEMANTICS 2006*, 2006.

[1] `http://wordnet.princeton.edu/`

[2] `http://lucene.apache.org/java/2_0_0/api/org/apache/lucene/wordnet`

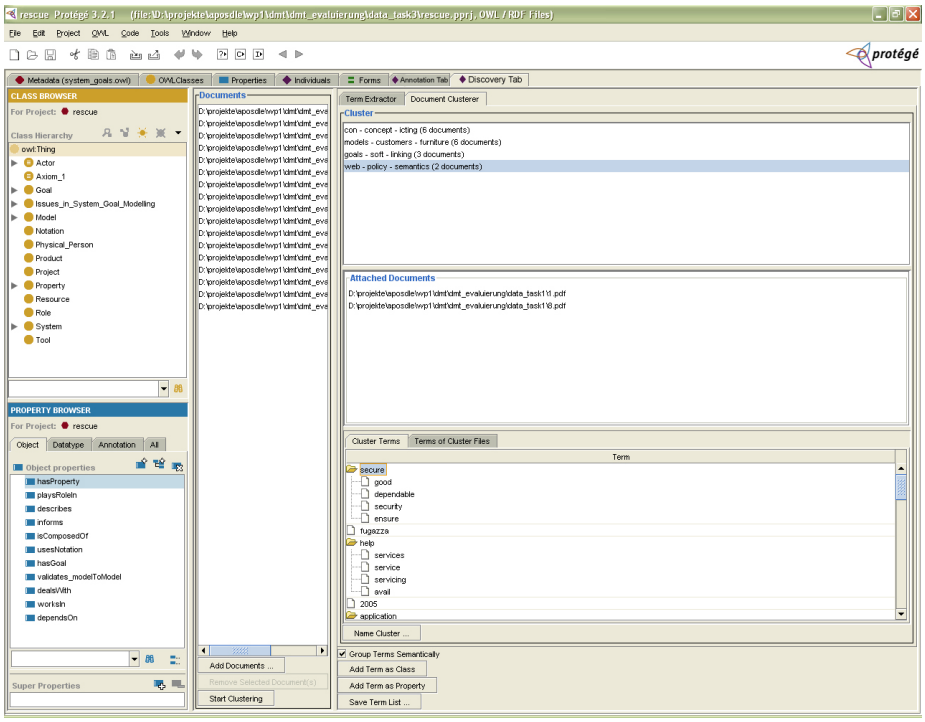

Figure 1: Screenshot of clustering functionality embedded in Discovery Tab

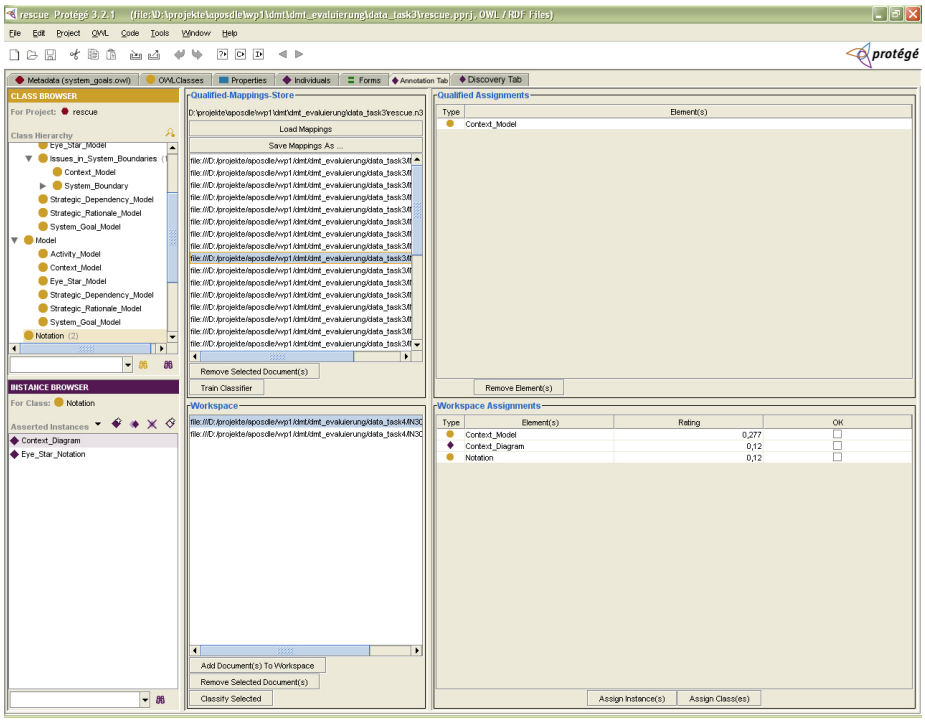

Figure 2: Screenshot of annotation functionality embedded in Annotation Tab