OBA: Supporting Ontology-Based Annotation of Natural Language Resources

Nadeschda Nikitina

Institute AIFB, Karlsruhe Institute of Technology, DE

Abstract. In this paper, we introduce OBA – an application for NLP-based annotation of natural language texts with ontology classes and relations. OBA provides support for different tasks required for semi-automatic semantic annotation. Among other things, it supports creating manual semantic annotations in order to enrich the set of lexical patterns, automatically annotating large corpora based on specified lexical patterns, and evaluating the results of semantic annotation.

1 Introduction

In the last decade, semantic annotation of unstructured data has significantly gained in popularity due to its usefulness for semantic search. A well-known example is Semantic Media Wiki, an extension of Media Wiki allowing for relation annotations, i.e., annotations specifying the meaning of relations between wiki articles, and attribute annotations, i.e. annotations specifying the meaning of a particular text within a wiki article. These annotations enable the editors to make certain facts within wiki articles accessible to programmes, which in turn makes it easier for users to find or reuse the information. However, since human resources are usually sparse, an extensive manual annotation of large information resources is not feasible. In most such cases, semi-automatic annotation is a more efficient alternative potentially allowing to significantly increase the number of semantic annotations that can be obtained for a given information resource with the same effort.

The ontology-based annotation tool (OBA) has been developed as a plugin for GATE[1] – a popular open source framework for analysis and processing of natural language texts – and, therefore, provides access to a wide range of features integrated in the aforementioned framework. OBA is one of the results of the project NanOn[1] aiming at ontology-supported literature search. Within the NanOn project, a hand-crafted ontology specified in the Web Ontology Language OWL 2 DL[3] modeling the scientific domain of nano technology has been used as the core resource to automatically analyze scientific documents for the occurrence of ontology classes and relations based on a set of lexical patterns. In this project, OBA was the main tool, on the one hand, supporting the acquisition of lexical patterns and extension of the ontology structure, and, on the other hand, supporting an automatic annotation of a large text corpus based on the obtained lexical patterns. In general, OBA can be used to support the following tasks relevant to semantic annotation:

[1] http://www.aifb.kit.edu/web/NanOn

– Manual annotation of text with classes and relations of an ontology in order to enrich the amount of lexical patterns or to extend the ontology itself with new classes and relations;
– Automatic annotation of a text or a corpus according to a given set of NLP-based patterns and the corresponding domain and range restrictions resulting in a population of the ontology with instances and relations represented in various formats, for instance, as RDF triples or OWL 2 DL instances and relations;
– Evaluation of automatically or manually created annotations (for instance, in order to estimate precision and recall).

The complete GATE package with OBA plugin including the source code is available at http://people.aifb.kit.edu/nni/GATE.zip.

2 Annotation Properties

OBA comes with a set of pre-defined general annotation types and general settings determining the scope, in which the user can define different types of annotations called *annotation properties* (for instance, those representing types of lexical patterns). First of all, we distinguish between patterns for classes and patterns for relations due to the possibility of additional information about the domain and range annotations of the corresponding relation. In case of class patterns, in addition to the possibility of a part-of-speech restriction, the possible settings include different ways to deal with plural forms, case deviations and non-matching word boundaries. In case of relation patterns, we further distinguish between

– string-based patterns, i.e., patterns containing an expression that must appear in the text between domain and range annotations (for instance, a preposition such as *by, of, as*), and
– patterns based only on the information about the domain and range annotations (for instance, two nouns within a single noun group such as *surface defects* or *ITO nanoparticles*).

In both cases, the settings include the allowed annotation properties for the domain and range annotations and a distance in characters between the subject and object annotations.

Example 1 *Within the project NanOn, we used, among others, the following annotation properties:*

– *patternNomen (class annotation, plural forms, no case-sensitivity, word boundaries, POS:NN)*
– *patternAcronym (class annotation, no plural forms, case-sensitivity, word boundaries, no POS)*
– *patternNomenNomen (relation annotation, POS domain: NN, POS range: NN, no string match required)*
– *patternsModifierNomen (relation annotation, POS domain: JJ|VBG|VBN, POS range: NN, no string match required)*
– *patternNomenVerbP (relation annotation, POS domain: NN, POS range: VBG, no string match required)*

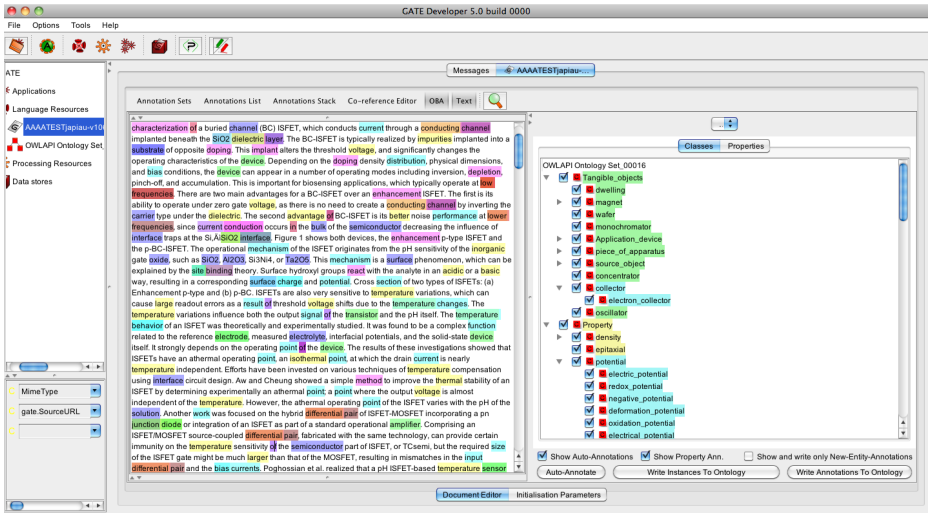

Fig. 1. General user interface of OBA

3 User Interface and Selected Features

Figure 1 shows the general UI of OBA. The plugin extends OCAT [2], which provides basic functionality for ontology-based annotation. Based on the OWL API[2], OBA can load an ontology in all supported formats[3]. The annotations (in this case created mostly automatically) are shown in the color of the corresponding ontology entities.[4] In order to simplify the evaluation of annotations, several general filters (for instance, hiding all automatically generated annotations or all relation annotations) were defined in addition to the check-boxes in front of each ontology entity. In addition to the possibility to save the state of an annotated document as XML, OBA allows the user to convert annotations in the following two ways:

1. Annotations can be converted into new ontology entities and lexical patterns. The latter are stored within the ontology itself as values of OWL 2 annotation properties (rdfs:label is one of such properties). It is up to the user to define OWL 2 annotation properties representing a particular type of a lexical pattern. Given a mapping of pattern types to OWL 2 annotation properties specified by the user, the annotation values are stored in a particular format also representing the metadata of each annotation. Using the context menu, the user can explore for each ontology entity the values of OWL 2 annotation properties stored in the ontology including the values of rdfs:label or any user-defined annotation property such as those repre-

[2] http://gate.ac.uk/sale/tao/splitch14.html#sec:ontologies:OCAT
[3] http://owlapi.sourceforge.net/index.html
[4] Due to the large number of classes, by default colors are only distinguished for direct subclasses of owl:Thing. However, the color of each entity can also be selected manually using the context menu.

senting lexical patterns. The corresponding annotation property viewer also shows the available metadata of lexical patterns.

2. Annotations can be converted into actual *semantic annotations* in form of class and relation instances that can be stored either as OWL 2 instances, RDF triples or quadruples, additionally containing a reference to the source document and a small excerpt .

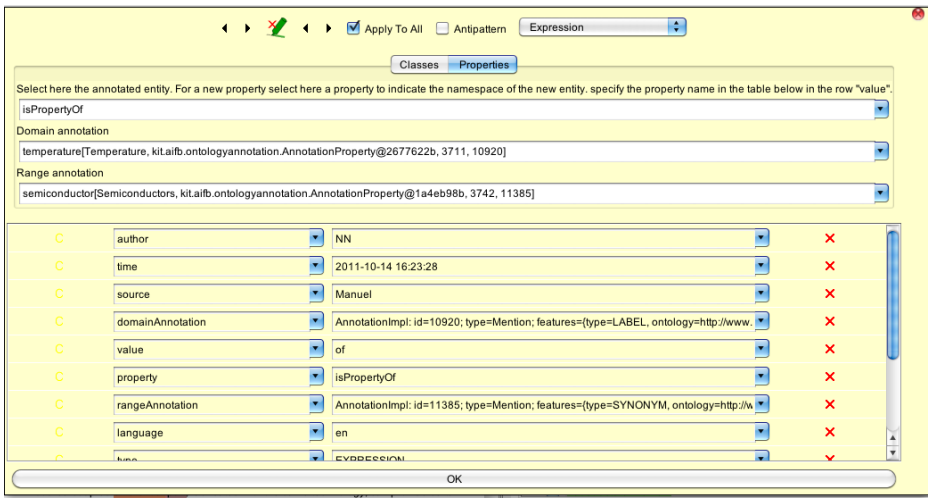

Fig. 2. Creating new pattern annotations with OBA

Figure 2 shows the annotation editor, which is activated by selecting a part of the text in order to create a new annotation. While, in case of class annotations, the selection of the corresponding annotation property and a class is sufficient to create the annotation, in case of relation annotations, the user additionally has to specify the domain and range annotations from the list of existing class annotations.

In addition to the required information such as the corresponding ontology entity, offset within the document and the annotation property, the annotation editor allows the user to edit various metadata entries determined by default every time a pattern annotation is created. Among other things, they include the author of the pattern, document in which the pattern has been annotated, time and date as well as the summary of the corresponding domain and range annotations in case of relation annotations.

In order to create a new class or relation (Figure 3), the pre-defined annotation property “New Class/Property” must be selected. Subsequently, the corresponding superclass or superrelation must be selected along with the domain and range classes.

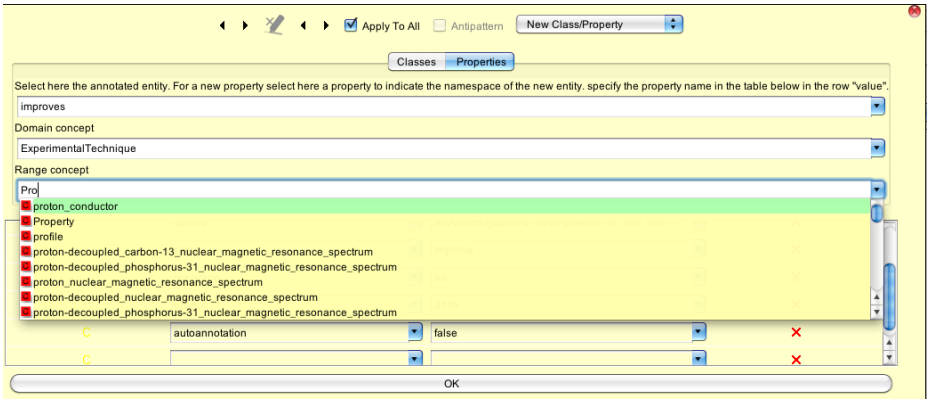

Fig. 3. Creating new classes and properties with OBA

4 Demonstration

The goal of the demonstration is to show how, given an ontology and a set of texts, the various features of OBA are employed to define a set of lexical patterns and automatically annotate a corpus with the pattern set. The demonstration will be split in four main scenarios:

- Definition of annotation properties based on the requirements of the application scenario: Here, we explain the pre-defined settings and the usage of NLP information in order to fine-tune the performance of the annotation.
- Manual annotation using defined annotation properties and classes/relations: We show how new lexical patterns in form of annotations can be created, edited or deleted and explain the available annotation metadata.
- Extension of the ontology with new classes and relations: Here, we demonstrate, how, based on the information occurring in a text, the ontology can be extended on the fly during the annotation.
- Automatic annotation of corpora: Finally, we show how the created lexical patterns can be used to automatically annotate documents.

The visitors will also be shown how to download and install the tool for their own use.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6) (2011), http://tinyurl.com/gatebook
2. Horridge, M., Bechhofer, S.: The owl api: A java api for working with owl 2 ontologies. In: OWLED (2009)
3. Motik, B., Patel-Schneider, P.F., Cuenca Grau, B.: OWL 2 Web Ontology Language: Direct Semantics. W3C Recommendation (October 2009), available at `http://www.w3.org/TR/owl2-direct-semantics/`