OntoLT Version 1.0: Middleware for Ontology Extraction from Text

Paul Buitelaar[1], Michael Sintek[2]

[1] DFKI GmbH, Language Technology, Stuhlsatzenhausweg 3,
66123 Saarbrücken, Germany
[2] DFKI GmbH, Knowledge Management, Erwin-Schrödinger-Straße,
67608 Kaiserslautern, Germany
{paulb,sintek}@dfki.de

Introduction

As human language is a primary mode of knowledge transfer, ontology learning from relevant text collections seems a viable option A typical approach in ontology learning from text first involves term extraction from a domain-specific corpus through a statistical process that determines their relevance for the domain corpus at hand. These are then clustered into groups with the purpose of identifying a taxonomy of potential classes. Other relations can be identified subsequently by computing a statistical measure of 'connectedness' between identified clusters.

The OntoLT approach described here follows a similar procedure, but we aim also at more directly connecting ontology engineering with linguistic analysis. Through the use of mapping rules between linguistic structure and ontological knowledge, linguistic knowledge (context words, morphological and syntactic structure, etc.) remains associated with the constructed ontology and may be used subsequently in its application and maintenance, e.g. in knowledge markup, ontology mapping and ontology evolution.

The OntoLT Plug-In

We describe OntoLT, a plug-in for Protégé[1] with which concepts (Protégé classes) and relations (Protégé slots) can be extracted automatically from linguistically annotated text collections – OntoLT was introduced in (Buitelaar et al., 2003; 2004). OntoLT provides mapping rules, defined by use of a precondition language that allow for a mapping between linguistic entities in text and class/slot candidates in Protégé.

Preconditions are implemented as Xpath-expressions over XML-based linguistic annotation. Predefined preconditions select for instance the predicate of a sentence, its linguistic subject or direct object. Preconditions can also be used to check certain conditions on these linguistic entities, for instance if the subject in a sentence corresponds to a particular lemma (the morphological stem of a word).

Selected linguistic entities may be used in constructing or extending an ontology. For this purpose, OntoLT provides operators to create classes or slots. According to the preconditions that were satisfied, corresponding operators will be activated to create a set of candidate classes and slots. Validated candidates can be integrated into a new or existing ontology.

OntoLT works with the following linguistic information: part-of-speech, morphological analysis, phrases (including head-modifier analysis) and predicate-argument structure (including grammatical functions like subject, direct object). The annotation allows for the automatic extraction of linguistic entities according to predefined preconditions: heads of nominal phrases, predicates and their subjects, etc. Linguistic objects that can be extracted in this way can be used to construct a shallow ontology of domain relevant concepts, sub-concepts and relations between concepts.

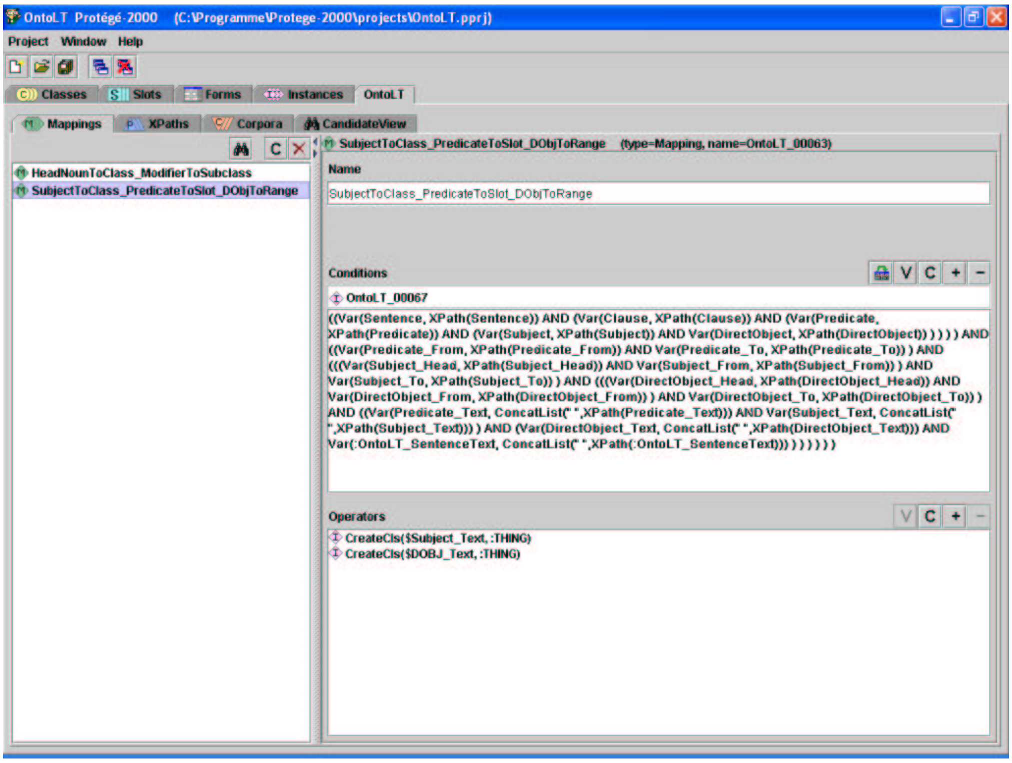
[1] http://protege.stanford.edu/

When installed[2], the OntoLT plug-in will appear as the right-most tab in the Protégé interface – as shown in Figure 1 below. OntoLT consists again of tabs with the following functionality:

- *Mappings* To define a new mapping rule or refine an existing mapping rule
- *XPaths* To (re)define Xpath-expressions (i.e. preconditions)
- *Corpora* To upload one or more annotated document collections
- *CandidateView* To extract concepts and/or relations from the annotated documents

Figure 1 shows the *Mappings* tab of OntoLT with two rules for mapping information from the linguistic annotation to potential Protégé classes and slots:

- **HeadNounToClass_ModToSubClass** maps a head-noun to a class and in combination with its modifier(s) to one or more sub-class(es)
- **SubjToClass_PredToSlot_DObjToRange** maps a linguistic subject to a class, its predicate to a corresponding slot for this class and the direct object to the "range" of this slot.



HeadNounToClass_ModifierToSubclass
SubjectToClass_PredicateToSlot_DObjToRange

SubjectToClass_PredicateToSlot_DObjToRange (type=Mapping, name=OntoLT_00063)

Name: SubjectToClass_PredicateToSlot_DObjToRange

Conditions: OntoLT_00067

```
((Var(Sentence, XPath(Sentence)) AND (Var(Clause, XPath(Clause)) AND (Var(Predicate,
XPath(Predicate)) AND (Var(Subject, XPath(Subject)) AND Var(DirectObject, XPath(DirectObject)) ) ) ) ) AND
((Var(Predicate_From, XPath(Predicate_From)) AND Var(Predicate_To, XPath(Predicate_To)) ) AND
(((Var(Subject_Head, XPath(Subject_Head)) AND Var(Subject_From, XPath(Subject_From)) ) AND
Var(Subject_To, XPath(Subject_To)) ) AND (((Var(DirectObject_Head, XPath(DirectObject_Head)) AND
Var(DirectObject_From, XPath(DirectObject_From)) ) AND Var(DirectObject_To, XPath(DirectObject_To)) )
AND ((Var(Predicate_Text, ConcatList(" ",XPath(Predicate_Text))) AND Var(Subject_Text, ConcatList("
",XPath(Subject_Text))) ) AND (Var(DirectObject_Text, ConcatList(" ",XPath(DirectObject_Text))) AND
Var(:OntoLT_SentenceText, ConcatList(" ",XPath(:OntoLT_SentenceText))) ) ) ) ) ) )
```

Operators:

```
CreateCls($Subject_Text, :THING)
CreateCls($DOBJ_Text, :THING)
```

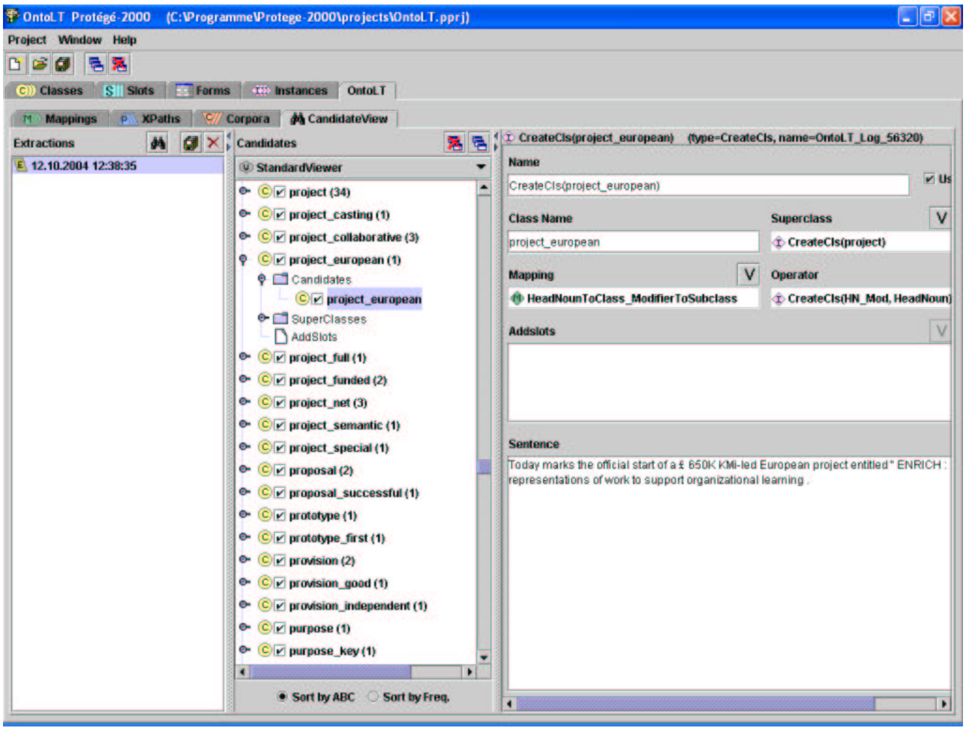
Figure 1: The *Mappings* Tab in the OntoLT Plug-In

Extract Candidates

To extract concept and/or relation candidates, the user uploads the domain-specific text collection that they want to use for the ontology extraction task. For this purpose, a linguistically annotated version of this document collection should be uploaded in the *Corpora* tab[3].

[2] OntoLT is available as a plug-in for Protégé version 1.8 (http://olp.dfki.de/OntoLT/OntoLT.htm)

By running the mapping rules as defined in the *Mappings* tab over the annotated document collection, corresponding candidates will be extracted and displayed in the *CandidateView* tab – as shown in Figure 2 below. Here, the user can inspect extracted candidates for inclusion in the ontology. For this purpose, the user is provided with the linguistic context (sentence) from which the candidate was extracted, the mapping rule that was applied and the results of this. For further ease of use, extracted candidates can be organized according to frequency or in alphabetical order.



12.10.2004 12:38:35

- project (34)
- project_casting (1)
- project_collaborative (3)
- project_european (1)
  - Candidates
    - project_european
  - SuperClasses
  - AddSlots
- project_full (1)
- project_funded (2)
- project_net (3)
- project_semantic (1)
- project_special (1)
- proposal (2)
- proposal_successful (1)
- prototype (1)
- prototype_first (1)
- provision (2)
- provision_good (1)
- provision_independent (1)
- purpose (1)
- purpose_key (1)

CreateCls(project_european) (type=CreateCls, name=OntoLT_Log_56320)

Name: CreateCls(project_european) ☑ Us

Class Name: project_european

Superclass: CreateCls(project)

Mapping: HeadNounToClass_ModifierToSubclass

Operator: CreateCls(HN_Mod, HeadNoun)

Addslots

Sentence: Today marks the official start of a £ 650K KMi-led European project entitled " ENRICH : representations of work to support organizational learning .

Figure 2: Extracted Candidates in the *CandidateView* tab

Statistical Relevance

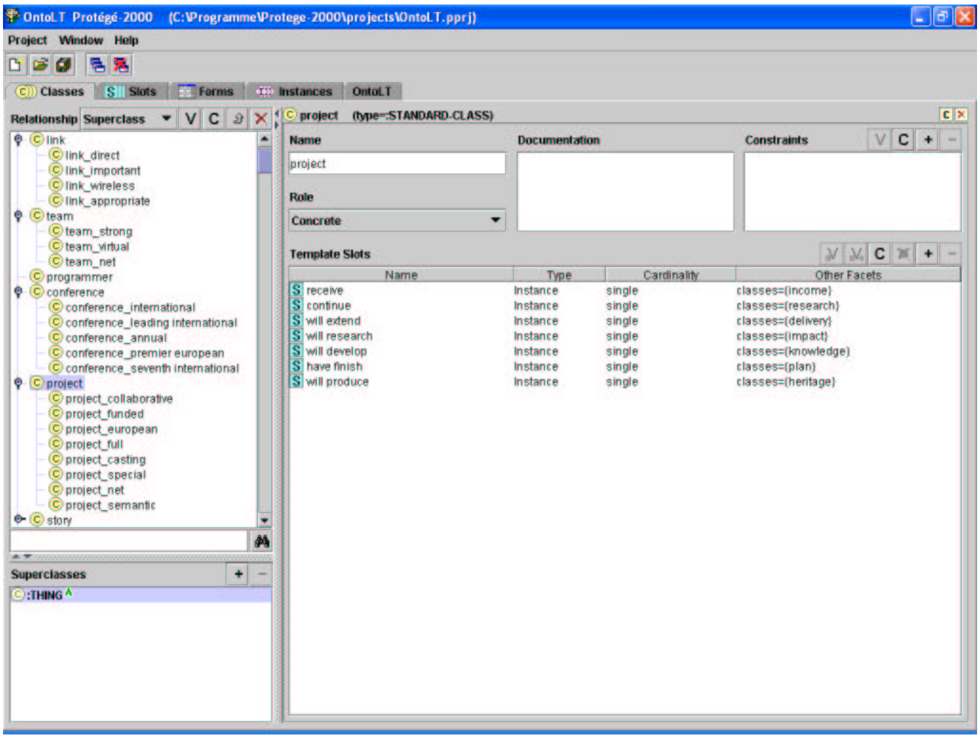
In ontology development for particular tasks and/or domains there is a need to focus only on those concepts and relations that are (highly) relevant for the task or domain in question. For this purpose, OntoLT includes a functionality for the computation of a statistical relevance score for extracted words by comparison of their frequencies in a (domain-specific) corpus with frequencies in a reference corpus. In this way, word use in a particular domain is contrasted with that of more general word use.

The user can use this functionality in the *Mappings* tab by selecting: the specific mapping they want to refine; the XPath element for which frequencies should be compared; the corpus to be examined; the reference corpus to be considered. Results of the statistical relevance measure are displayed in a choice box with the topmost 1% of the words pre-selected. The user can further refine this list by (de-) selection of words. After validation, selected words are included in the mapping rule as a further precondition.

[3] Linguistic annotation is currently provided by SCHUG, a rule-based system for German and English analysis (Declerck, 2002). SCHUG is not an integrated part of OntoLT, but could be accessed through a web service.

Ontology Extraction

After refinement of the mapping rules the user can run the extraction again, extract candidate classes and slots, validate them (select or deselect them) and generate ontology fragments accordingly by pushing the right-most button in the middle pane. The extracted classes and slots will be displayed in the *Classes* tab of Protégé – as shown in Figure 3 below. Using standard Protégé functionalities, the extracted ontology fragments can be further edited and exported in one or more formats.



- link
  - link_direct
  - link_important
  - link_wireless
  - link_appropriate
- team
  - team_strong
  - team_virtual
  - team_net
- programmer
- conference
  - conference_international
  - conference_leading international
  - conference_annual
  - conference_premier european
  - conference_seventh international
- project
  - project_collaborative
  - project_funded
  - project_european
  - project_full
  - project_casting
  - project_special
  - project_net
  - project_semantic
- story

Superclasses: :THING

project (type=:STANDARD-CLASS)

Name: project

Role: Concrete

Documentation

Constraints

Template Slots

Name	Type	Cardinality	Other Facets
receive	Instance	single	classes={income}
continue	Instance	single	classes={research}
will extend	Instance	single	classes={delivery}
will research	Instance	single	classes={impact}
will develop	Instance	single	classes={knowledge}
have finish	Instance	single	classes={plan}
will produce	Instance	single	classes={heritage}

Figure 3: Extracted Ontology Fragment

References

Buitelaar P., Olejnik D. and Sintek M. *OntoLT: A Protégé Plug-In for Ontology Extraction from Text* In: Proceedings of the Demo Session of ISWC-2003, Sanibel Island, Florida, October 2003.

Buitelaar P., Olejnik D. and Sintek M. *A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis* In: Proceedings of the 1st European Semantic Web Symposium (ESWS), Heraklion, Greece, May 2004.

Declerck Th. *A set of tools for integrating linguistic and non-linguistic information.* Proceedings of the SAAKM workshop at ECAI, Lyon, 2002.

Acknowledgements

Previous versions of OntoLT have been implemented by Daniel Olejnik and were developed under grants IST-2000-29243 for the OntoWeb project and IST-2000-25045 for the MEMPHIS project. Current maintenance and extension work by Yasir Iqbal is supported by grants for the projects VIeWs (by the Saarland Ministry of Economic Affairs) and SmartWeb (by the German Ministry of Education and Research: 01 IMD01 A).