A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis

Paul Buitelaar[1], Daniel Olejnik[1], Michael Sintek[2]

[1] DFKI GmbH, Language Technology, Stuhlsatzenhausweg 3,
66123 Saarbruecken, Germany
{paulb,olejnik}@dfki.de

[2] DFKI GmbH, Knowledge Management, Erwin-Schrödinger-Straße,
67608 Kaiserslautern, Germany
sintek@dfki.de

In this paper we describe a plug-in (OntoLT) for the widely used Protégé ontology development tool that supports the interactive extraction and/or extension of ontologies from text. The OntoLT approach provides an environment for the integration of linguistic analysis in ontology engineering through the definition of mapping rules that map linguistic entities in annotated text collections to concept and attribute candidates (i.e. Protégé classes and slots). The paper explains this approach in more detail and discusses some initial experiments on deriving a shallow ontology for the neurology domain from a corresponding collection of neurological scientific abstracts.

1 Introduction

With a recent increase in developments towards knowledge-based applications such as Intelligent Question-Answering, Semantic Web Services and Semantic-Level Multimedia Search, the interest in large-scale ontologies has increased. Additionally, as ontologies are domain descriptions that tend to evolve rapidly over time and between different applications (see e.g. Noy and Klein, 2002) there has been an increasing development in recent years towards learning or adapting ontologies dynamically, e.g. by analysis of a corresponding knowledge base (Deitel et al., 2001, Suryanto and Compton, 2001) or document collection.

Most of the work in ontology learning has been directed towards learning ontologies from text[1]. As human language is a primary mode of knowledge transfer, ontology learning from relevant text collections seems indeed a viable option as illustrated by a number of systems that are based on this principle, e.g. ASIUM (Faure et al., 1998), TextToOnto (Maedche and Staab, 2000; Maedche) and Ontolearn (Navigli et

[1] See for instance the overview of ontology learning systems and approaches in OntoWeb deliverable 1.5 (Gomez-Perez et al., 2003).

al., 2003). All of these combine a certain level of linguistic analysis with machine learning algorithms to find potentially interesting concepts and relations between them (see also Maedche, 2003).

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. Subsequently also relations can be identified by computing a statistical measure of 'connectedness' between identified clusters.

The OntoLT approach 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.

2 OntoLT

The OntoLT approach (introduced in Buitelaar et al., 2003) is available as a plug-in for the widely used Protégé ontology development tool[2], which enables the definition of mapping rules with which concepts (Protégé classes) and attributes (Protégé slots) can be extracted automatically from linguistically annotated text collections. A number of mapping rules are included with the plug-in, but alternatively the user can define additional rules.

The ontology extraction process is implemented as follows. OntoLT provides a precondition language, with which the user can define mapping rules. Preconditions are implemented as XPATH expressions over the XML-based linguistic annotation. If all constraints are satisfied, the mapping rule activates one or more operators that describe in which way the ontology should be extended if a candidate is found.

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). The precondition language consists of Terms and Functions, to be discussed in more detail in section 4.2.

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

---

[2] http://protégé.stanford.edu

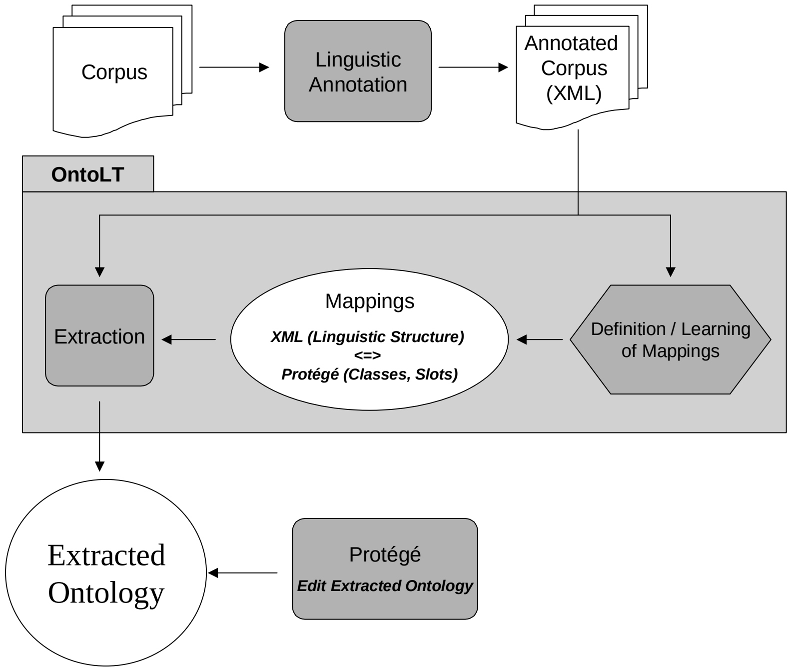

Figure 1: Overview of the OntoLT Approach

3 Linguistic Annotation

Linguistic annotation is not integrated with OntoLT, but is accessed via an XML-based exchange format, which integrates multiple levels of linguistic and semantic analysis in a multi-layered DTD with each analysis level (e.g. morphological, syntactic and dependency structure) organized as a separate track with options of reference between them via indices[3].

Linguistic annotation is currently provided by SCHUG, a rule-based system for German and English analysis (Declerck, 2002) that implements a cascade of increasingly complex linguistic fragment recognition processes. SCHUG provides annotation of part-of-speech (through integration of TnT: Brants, 2000), morphological inflection and decomposition (based on Mmorph: Petitpierre and Russell, 1995), phrase and dependency structure (head-complement, head-modifier and grammatical functions).

In Figure 2, we present a section of the linguistic annotation for the following sentence (German with corresponding sentence from the English abstract):

---

[3] The format presented here is based on proposals and implementations described in (Buitelaar et al., 2003) and (Buitelaar and Declerck, 2003).

An 40 Kniegelenkpräparaten wurden mittlere Patellarsehnendrittel mit einer neuen Knochenverblockungstechnik in einem zweistufigen Bohrkanal bzw. mit konventioneller Interferenzschraubentechnik femoral fixiert.

(In 40 human cadaver knees, either a mid patellar ligament third with a trapezoid bone block on one side was fixed on the femoral side in a 2-diameter drill hole, or a conventional interference screw fixation was applied.)

The linguistic annotation for this sentence consists of part-of-speech and lemmatization information in the <text> level, phrase structure (including head-modifier analysis) in the <phrases> level and grammatical function analysis in the <clauses> level (in this sentence there is only one clause, but more than one clause per sentence is possible).

Part-of-speech information consists of the correct syntactic class (e.g. noun, verb) for a particular word given its current context. For instance, the word *works* will be either a verb *(working the whole day)* or a noun (*all his works have been sold).*

Morphological information consists of inflectional, derivational or compound information of a word. In many languages other than English the morphological system is very rich and enables the construction of semantically complex compound words. For instance the German word *Kreuzbandverletzung* corresponds in English with three words: *cruciate ligament injury*.

Phrase structure information consists of an analysis of the syntactic structure of a sentence into constituents that are headed by an adjective, a noun or a preposition. Additionally, the internal structure of the phrase will be analyzed and represented, which includes information on modifiers that further specify the head. For instance, in the nominal phrase *neue Technik (new technology)* the modifier *neu* further specifies the head *Technik*.

Clause structure information consists of an analysis of the core semantic units (clauses) in a sentence with each clause consisting of a predicate (mostly a verb) with its arguments and adjuncts. Arguments are expressed by grammatical functions such as the subject or direct object of a verb. Adjuncts are mostly prepositional phrases, which further specify the clause. For instance, in *John played football in the garden* the prepositional phrase *in the garden* further specifies the clause "*play (John, football)"*.

All such information is provided by the annotation format that is illustrated in Figure 2 below. For instance, the direct object (**DOBJ**) in the sentence above (or rather in clause **cl1**) covers the nominal phrase **p2**, which in turn corresponds to tokens **t5** to **t10** (*mittlere Patellarsehnendrittel mit einer neuen Knochenverblockungstechnik*). As token **t6** is a German compound word, a morphological analysis is included that corresponds to lemmas **t6.l1, t6.l2, t6.l3**.

```
<sentence id="s3" stype="decl" corresp=" ">

  <clauses>
    <clause id="cl1" from="p1" to="p5" pred="p5" type="pass">
      <arg id="a1" type="SUBJ" phrase="none" />
      <arg id="a2" type="IOBJ" phrase="p1"/>
      <arg id="a3" type="DOBJ" phrase="p2" />
      <arg id="a4" type="PP_ADJ" phrase="p3"/>
    </clause>
  </clauses>

  <phrases>
     …
    <phrase id="p2" from="t5" to="t10" type="NP">
       <mod from="t5" to="t5" />
       <head from="t6" to="t6" />
       <mod_post from="t7" to="t10" />
    </phrase>
     …
  </phrases>

  <text>
    <token id="t1" pos="APPR" str="An">
      <lemma id="t1.l1">an</lemma>
    </token>
    <token id="t2" pos="CARD" str="40" />
    <token id="t3" pos="NN" str="Kniegelenkpraeparaten">
      <lemma id="t3.l1">Kniegelenk</lemma>
      <lemma id="t3.l2">Praeparat</lemma>
    </token>
    <token id="t4" pos="VAFIN" str="wurden">
      <lemma id="t4.l1">werden</lemma>
    </token>
    <token id="t5" pos="ADJA" str="mittlere">
      <lemma id="t5.l1">mittler</lemma>
    </token>
    <token id="t6" pos="NN" str="Patellarsehnendrittel">
      <lemma id="t6.l1">patellar</lemma>
      <lemma id="t6.l2">Sehne</lemma>
      <lemma id="t6.l3">Drittel</lemma>
    </token>
     …
    <token id="t19" pos="ADJD" str="femoral" />
    <token id="t20" pos="VVPP" str="fixiert">
      <lemma id="t6.l1">fixieren</lemma>
    </token>
    <token id="t21" pos="PUNCT" str="." />
  </text>
</sentence>
```


Figure 2: Linguistic Annotation Example

4 Ontology Extraction from Text with OntoLT

The ontology extraction process is implemented as follows. OntoLT provides a precondition language with which the user can define mapping rules. Preconditions are implemented as XPATH expressions over the linguistic annotation. If the precondition is satisfied, the mapping rule activates one or more operators that describe in which way the ontology should be extended if a candidate is found.

4.1 Mapping Rules

A number of mapping rules are predefined and included with the OntoLT plug-in, but alternatively the user may define additional mapping rules, either manually or by the integration of a machine learning process. In Figure 3, two rules are defined 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.

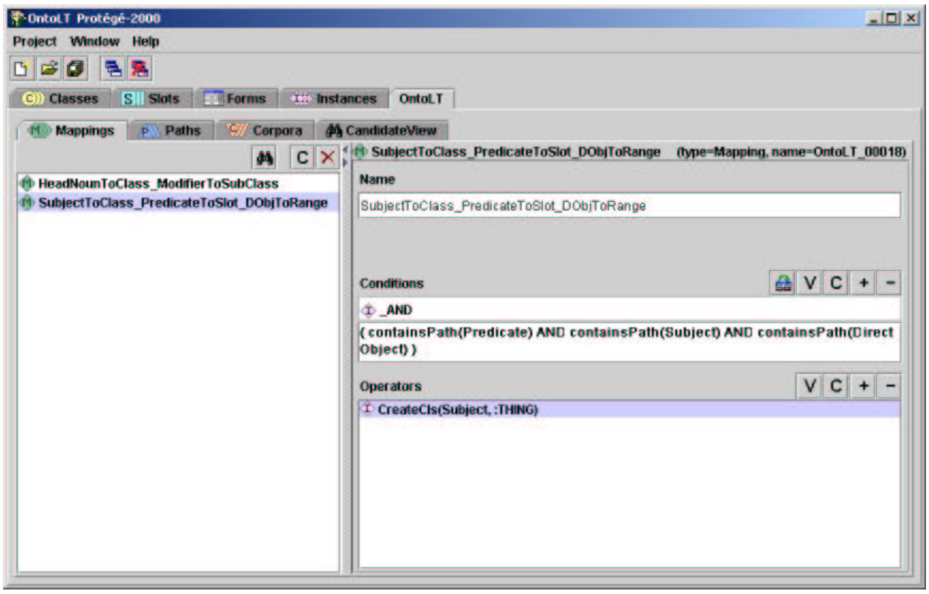

Figure 3: Example Mappings in OntoLT

4.2 Precondition Language

OntoLT provides a precondition language for defining mapping rules, which allows for the selection of particular linguistic entities in the annotated documents. Predefined predicates[4] of the precondition language select, e.g., the pred (linguistic predicate) of a sentence, its subject or object. Preconditions can also be used to check certain conditions on these linguistic entities, for instance if the subject corresponds to a certain semantic class[5]. Correspondingly, the precondition language consists of terms (constants and functions) and predicates.

Predicates can be one of[6]:

containsPath	:	returns true/false if a certain path is contained in the source or not
HasValue	:	returns true/false if a certain path has a specified value or not
HasConcept	:	returns true/false if a certain path corresponds to a specified concept code (e.g. a EuroWordNet sense)
AND	:	Boolean 'and'
OR	:	Boolean 'or'
NOT	:	Boolean 'not'
EQUAL	:	tests if two terms are equal

Currently, the only supported function is:

ID	:	returns the ID of a node of the XML-source

Selection of the pred, object and subject of a sentence can then be implemented by the definition of the precondition that was shown in Figure 3., which checks if there are any valid XPATH expressions for subject, object and pred.

```
(containsPath(Pred) AND
 containsPath(Subject) AND
 containsPath(Object))
```

---

[4] Unfortunately, we need to use the word "predicate" in two different meanings, either as: 1. a predicate of the precondition language; 2. a predicate in the linguistic analysis. To distinguish the two meanings, we will write pred for the linguistic use of "predicate".

[5] Semantic class information may be provided by a lexical semantic resource, such as WordNet (Miller, 1995) for English or EuroWordNet (Vossen, 1997) for various other languages, or by a domain-specific thesaurus or ontology, such as MeSH (Medical Subject Headings) for the biomedical domain: http://www.nlm.nih.gov/mesh/meshhome.html

[6] In the current implementation, with more predicates (and functions) to be added upon need.

4.3 Operators

Selected linguistic entities may be used in constructing or extending an ontology. For this purpose, OntoLT provides operators to create classes, slots and instances:

CreateCls	:	create a new class
AddSlot	:	add a slot to a class or create it if non-existing
CreateInstance	:	introduce a new instance for an existing or new class
FillSlot	:	set the value of a slot of an instance

OntoLT executes all mapping rules collectively. Therefore, according to which preconditions are satisfied, all corresponding operators will be activated to create a set of candidate classes and slots that are to be validated by the user. According to this interactive process, classes and slots will be automatically generated into a new ontology or integrated into an existing ontology.

4.4 Statistical Preprocessing

In order to use only extracted linguistic information that is relevant for the domain, the approach includes a statistical preprocessing step. Here we base our approach on the use of the "chi-square" function in (Agirre et al., 2001) for determining domain relevance[7]. This function computes a relevance score by comparison of frequencies in a domain corpus under consideration with that of frequencies in a reference corpus. In this way, word use in a particular domain is contrasted with that of more general word use.

4.5 Semi-Automatic Generation of Mapping Rules

The statistical preprocessing step also allows for a semi-automatic generation of mapping rules. For this purpose, we can simply generate mapping rules for all possible XML-elements in the linguistic annotation (e.g. pred, mod, head) constrained to only those words that were selected by the chi-square measure. User interaction will however still be needed to specify the operators associated with these generated conditions for the mapping rules to be defined. For instance, it would need to be decided if the lemma of a pred should be generated as a class, or rather as a slot for a class that should be generated for the lemma of a particular head, or if a class should be generated for the lemma of the modifier (mod), etc. In future work, also this aspect could be further supported by inclusion of a machine-learning component based on active learning (see e.g. Finn and Kushmerick, 2003) that would enable the automatic generation of operators given a training process on previous specifications by the user.

---

[7] The chi-square function gives a good indication of relevance, but experiments showed that also absolute frequency is an important indication of relevance. We therefore additionally multiply the chi-square score by absolute frequency to obtain a combined measure of frequency and relevance.

5 Experiment: Extracting an Ontology for Neurology

In order to test our approach in a realistic setting, we defined the following experiment. Given a corpus of medical texts in the neurology domain, we applied OntoLT in combination with linguistic annotation as described above to extract a shallow ontology for this domain.

The neurology corpus that we used in the experiment is a section of the bilingual (English-German) medical corpus that was constructed within the MuchMore project on cross-lingual information retrieval in the medical domain. The MuchMore corpus includes around 9000 scientific abstracts in various medical sub-domains[8] with around 1 million tokens for each language (see Buitelaar et al., 2004). The neurology section of the MuchMore corpus consists of 493 abstracts.

As a first step, the neurology corpus is linguistically analyzed and annotated with SCHUG, according to the XML-format presented in section 3 above. In all further steps, this linguistically annotated version of the corpus is used rather than the original text version.

5.1 Statistical Preprocessing of the Neurology Corpus

To extract only relevant linguistic entities from the neurology corpus, we applied the chi-square measure as discussed above. The rest of the MuchMore corpus was used in this process as a contrasting reference corpus (representing the medical domain in general) that allowed for the identification of those linguistic entities that are specific to neurology.

In the following tables, a selection of extracted 10 topmost relevant linguistic entities (head, mod, pred) are given for the neurology corpus (German with English translations):

head	*Dysgenesie (dysgenesia)*
	Denkstörung (thought disorder)
	Epilepsie (epilepsia)
	Psychiater (psychiatrist)
	Aura (aura)
	Tremor (tremor)
	Asystolie (asystole)
	Dopaminfreisetzung (dopamine release)
	Obdachlose (homeless)
	Aphasie (aphasia)

Table 1: 10 topmost relevant Heads in the Neurology corpus

[8] The MuchMore corpus and related evaluation resources and interactive demos are publicly available from the project website: http://muchmore.dfki.de

mod	*schizophren (schizophrenic)*
	epileptisch (epileptic)
	transkraniel
	paranoid (paranoid)
	neuroleptisch (neuroleptic)
	neuropsychriatisch (neuro psychiatric)
	serotonerg
	impulsiv (impulsive)
	intraventrikulär (intra ventricular)
	neuropsychologisch (neuro psychological)

Table 2: 10 topmost relevant Modifiers in the Neurology corpus

pred	*zuerkennen (to adjudicate, award)*
	staerken (to boost, encourage, strengthen)
	sparen (to conserve, save)
	betreten (to enter)
	hervorbringen (to create, produce)
	befuerworten (to support, advocate)
	gebrauchen (to employ, use)
	begreifen (to apprehend, understand)
	ueben (to exercise, practice)
	imitieren (to copy, imitate, mimic)

Table 3: 10 topmost relevant Predicates in the Neurology corpus

5.2 Definition of Mapping Rules for Neurology

The results of the statistical processing are now used to generate one or more mappings between selected elements in the linguistic annotation (e.g. head, mod, pred) and Protégé classes and or slots. Here we present two examples.

HeadNounToClass_ModToSubClass

This mapping generates classes for all head-nouns (head) that were determined to be statistically relevant for the domain. For instance, classes are generated for the head-nouns *Dysgenesie (dysgenesia)* and *Epilepsie (epilepsia)*. Further, for each of these, sub-classes are generated for corresponding modifiers (mod). For the two classes just mentioned, the following sub-classes are generated:

Dysgenesie	:	Dysgenesie_kortikal	*(cortical)*
Epilepsie	:	Epilepsie_myoklonisch	*(myoclonic)*
		Epilepsie_idiopathisch	*(idiopathic)*
		Epilepsie_fokal	*(focal)*

SubjToClass_PredToSlot_DObjToRange

This mapping generates for all statistically relevant predicates (pred) a class for the head of the subject, a slot for the pred and a corresponding slot-range for the head of the object. For instance, consider the sentence:

Transitorische ischaemische Attacken imitieren in seltenen Fällen einfache fokale motorische Anfälle.

("Transient ischemic attacks mimicking in some cases simple partial motor seizures.")

In this case, a class is generated for the head of the subject *Attacke (attack*) and for the head of the object *Anfall (seizure*). Further, a slot imitieren *(to mimic*) is generated for the new class attacke with the new class anfall as its range (i.e. the class of possible fillers for this slot).

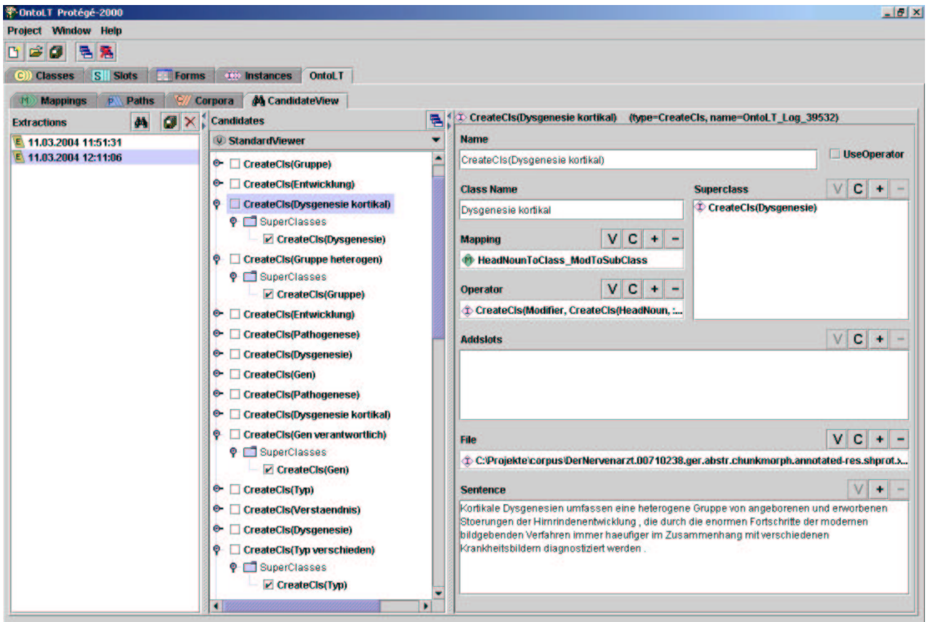

Figure 4: Class Candidates Generated by HeadNounToClass_ModToSubClass

6 Future Work

OntoLT allows for the semi-automatic extraction of shallow ontologies from German and English text collections. Future work will be concerned with providing further automatic support in the definition of mapping rules by refining and extending the statistical preprocessing step and by including an active learning approach (see also section 4.5).

Other aspects that will be worked on include: integration of linguistic annotation over a web service; integration of an information extraction approach for ontology population (identifying class instances); definition and implementation of an evaluation platform to evaluate extracted ontologies in a quantitative (technical performance) and qualitative (user satisfaction) way.

As discussed before, a number of different methods for text-based ontology extraction and learning have developed over recent years. However, in order to compare the performance of OntoLT with these and other methods, a proper evaluation framework needs to be set up. Currently it is very hard to compare methods and approaches, due to the lack of a shared understanding of the task at hand. Future work will therefore be concerned also with a contribution towards the development of such a shared understanding and an appropriate evaluation framework accordingly.

We expect that the general problem of ontology extraction and learning can be decomposed into a set of simpler tasks, which can be addressed with well-established evaluation methodologies (i.e. *precision* and *recall*). The assessment of the system results will be based on a comparison with respect to a benchmark ontology, which has to be manually produced by domain experts taking into consideration the content that is implicitly available in a corresponding text collection. Evaluation measures will be defined on the basis of the benchmark, according to the experience of related evaluation efforts in information retrieval (TREC[9], CLEF[10]) and natural language processing (SENSEVAL[11]).

7 Conclusions

OntoLT provides a middleware solution in ontology development that enables the ontology engineer to bootstrap a domain-specific ontology from a relevant text corpus (document collection). A sequence of automatic and interactive steps are involved in this process:

- automatic linguistic analysis and annotation
- automatic statistical preprocessing of extracted linguistic entities
- interactive definition of mapping rules between extracted linguistic entities and Protégé class and slot candidates
- interactive user validation of generated Protégé class and slot candidates
- automatic integration of validated class and slot candidates into an existing or new ontology

[9] http://trec.nist.gov/
[10] http://clef.iei.pi.cnr.it:2002/
[11] http://www.senseval.org/

Acknowledgements

This research has in part been supported by EC grants IST-2000-29243 for the OntoWeb project, IST-2000-25045 for the MEMPHIS project and IST-2001-34373 for the ESPERONTO project.

Special thanks also to our colleagues at Saarland University (Thierry Declerck, Mihaela Hutanu, Alexander Schutz) for making available the SCHUG linguistic analysis tool and for their cooperation in the neurology experiment.

References

Agirre E., Ansa O., Martinez D., Hovy E. Enriching WordNet concepts with topic signatures. In: Proceedings NAACL WordNet Workshop, 2001.

Brants, T. TnT - A Statistical Part-of-Speech Tagger. In: Proceedings of 6th ANLP Conference, Seattle, 2000.

Buitelaar P., Declerck Th., Sacaleanu B., Vintar Š., Raileanu D., Crispi C. A Multi-Layered, XML-Based Approach to the Integration of Linguistic and Semantic Annotations. In: Proceedings of EACL 2003 Workshop on Language Technology and the Semantic Web (NLPXML'03), Budapest, Hungary, April 2003.

Buitelaar P. and Declerck Th. Linguistic Annotation for the Semantic Web. In: Handschuh S., Staab S. (eds.) Annotation for the Semantic Web, IOS Press, 2003.

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

Buitelaar P., Steffen D., Volk M., Widdows D., Sacaleanu B., Vintar Š., Peters S. and Uszkoreit H. Evaluation Resources for Concept-based Cross-Lingual Information Retrieval in the Medical Domain. In: Proceedings of LREC2004.

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

Deitel A., Faron C. and Dieng R. Learning Ontologies from RDF Annotations In: Proceedings of the IJCAI Workshop on Ontology Learning, Seattle, Washington, 2001.

Faure D., Nédellec C. and Rouveirol C. Acquisition of Semantic Knowledge using Machine learning methods: The System ASIUM. Technical report number ICS-TR-88-16, 1998.

Finn A. and Kushmerick N. Active Learning Strategies for Information Extraction In: Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining (ATEM), Cavtat-Dubrovnik, Croatia, September 22nd, 2003.

Gomez-Perez A. and Manzano-Macho D. A Survey of Ontology Learning Methods and Techniques. Deliverable 1.5, OntoWeb Project, 2003.

Gruber T. Towards principles for the design of ontologies used for knowledge sharing. Int. Journal of Human and Computer Studies 43(5/6), 1994, 907-928.

Maedche A. Ontology Learning for the Semantic Web. The Kluwer International Series in Engineering and Computer Science, Volume 665, 2003.

Maedche, A., Staab, S. Semi-automatic Engineering of Ontologies from Text. In: Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, 2000.

Miller, G.A. WordNet: A Lexical Database for English. Communications of the ACM 11. 1995.

Navigli R., Velardi P., Gangemi A. Ontology Learning and its application to automated terminology translation. IEEE Intelligent Systems, vol. 18:1, January/February 2003.

Noy N. F. and Klein M. Ontology Evolution: Not the Same as Schema Evolution In: Knowledge and Information Systems, in press. Available as technical report SMI-2002-0926, 2002.

Petitpierre, D. and Russell, G. MMORPH - The Multext Morphology Program. Multext deliverable report for the task 2.3.1, ISSCO, University of Geneva. 1995.

Skut W. and Brants T. A Maximum Entropy partial parser for unrestricted text. In: Proceedings of the 6th ACL Workshop on Very Large Corpora (WVLC), Montreal. 1998.

Suryanto H. and Compton P. Discovery of Ontologies from Knowledge Bases In: Proceedings of the First International Conference on Knowledge Capture, Victoria, BC, Canada, October 2001.

Vossen P. EuroWordNet: a multilingual database for information retrieval. In: Proc. of the DELOS workshop on Cross-language Information Retrieval, March 5-7, Zürich, Switzerland, 1997.