Learning Knowledge Roles for Populating Lightweight Application Ontologies

Eni Mustafaraj[1], Martin Hoof[2] and Bernd Freisleben[1]
[1]Department of Mathematics & Computer Science, University of Marburg, Germany
[2]Department of Electrical Engineering, Fachhochschule Kaiserslautern, Germany
eni@informatik.uni-marburg.de
freisleb@informatik.uni-marburg.de
m.hoof@et.fh-kl.de

Abstract: We present a framework for semi-automatically acquiring domain knowledge necessary for building lightweight, application ontologies. The approach adopts active learning for semantic annotation of knowledge roles that have been derived from the CommonKADS methodology. We discuss the framework advantages by implementing a lightweight, application ontology for a knowledge management application in a technical domain.

Keywords: knowledge acquisition, CommonKADS, knowledge roles, lightweight ontologies, active learning, predictive maintenance.

1. Introduction

Knowledge management processes support sharing and reusing of all forms of organizational knowledge. However, to enable such processes, domain and task knowledge need first to be acquired, modelled, and made available to knowledge-based systems. Ontologies—as knowledge modelling constructs—are being increasingly promoted as facilitators of knowledge management activities (Abecker & van Elst 2004) and their inclusion in the Semantic Web vision (Berners-Lee et al. 2001) has contributed in raising the general interest in them.

Nevertheless, the engineering of ontologies has always been a bottleneck in the implementation of knowledge-based systems. The commonly acknowledged reason for this lies in the difficulties of knowledge acquisition from humans and its appropriate formalization. Currently, in the framework of Semantic Web activities, many research efforts are being dedicated to the task of automatically extracting ontologies from existing resources like text, databases, lexical resources, domain vocabularies, etc. In this context, several techniques relying on machine learning, text mining, or information extraction are being tested and evaluated.

Whereas for some Semantic Web applications such automatically extracted ontologies could be good enough, more demanding applications (as it is often the case with organizational knowledge management) would always require some degree of human knowledge engineering efforts. Therefore, an important research challenge consists in trying to decrease the amount of such efforts by combining principled modelling techniques with automatic knowledge acquisition methods.

A valuable source of principled knowledge modelling techniques is found in the comprehensive CommonKADS methodology (Schreiber et al. 2000), especially in its detailed catalogue of template knowledge models. In addition, the notion of domain independent knowledge roles facilitates the reuse of such templates for specific domain dependent tasks.

In the approach presented in this paper, we try to learn to assign knowledge roles—derived by the task descriptions of CommonKADS methodology—to domain specific knowledge sources (for example, text documents). Then, from these learned pairs of {*knowledge roles, domain text phrases*} we populate a lightweight, application ontology. Ontologies are characterized as lightweight when constraints for the representation formalization are relaxed, as well as when capabilities for automatic reasoning are only partly or even not at all implemented.

For demonstration purposes, we describe the process of building such a lightweight, application ontology for a knowledge management application in the domain of predictive maintenance for rotating electrical machines. We first use the terminology of CommonKADS to define a knowledge model for the task of predictive maintenance by combining and reusing the diagnosis and monitoring knowledge templates expressed in terms of several knowledge roles. Then, we adopt an active learning strategy that will annotate a corpus of text documents with knowledge roles, while keeping the number of instances to be labelled by the user low. The annotated expressions are then extracted and clustered to build the lightweight ontology, which can be used to retrieve knowledge in problem solving situations.

Reference this paper as: Mustafaraj, E, Hoof, M and Freisleben, B (2006) "Learning Knowledge Roles for Populating Lightweight Application Ontologies" *The Electronic Journal of Knowledge Management* Volume 4 Issue 1, pp 75-82, available online at www.ejkm.com

The paper is organized as follows. In Section 2 the knowledge model of predictive maintenance derived from CommonKADS templates is presented. Ontology engineering methods and lightweight ontologies are discussed in Section 3. We continue in Section 4 with the description of the active learning framework for semantic annotation and extraction of ontology concepts/relations. In Section 5, we conclude the paper and outline areas for future research.

2. Knowledge modelling for predictive maintenance

2.1 Knowledge modelling in CommonKADS

As previously mentioned, CommonKADS is a comprehensive methodology that encompasses all the steps for the design and implementation of knowledge intensive systems. In the context of this paper, we are only interested in the knowledge modelling aspect of the methodology and will not consider the organization, agent, or communication models. The knowledge model of the methodology consists of three components: domain knowledge, inference knowledge, and task knowledge. Each of these components is composed of a series of knowledge constructs. For example, the domain knowledge component consists of the domain schema(s) and the knowledge base(s). A domain schema itself includes *concepts*, *relations*, and *rule types* (commonly known as domain knowledge types); a knowledge base will then contain instances of these knowledge types.

The second component, inference knowledge, contains *inferences*, *knowledge roles*, and *transfer functions*. The construct of knowledge roles is particularly interesting, because it is the link connecting domain knowledge constructs (like concepts and rules) to the inferences (primitive functions that perform reasoning tasks on the data mapped to the knowledge roles).

The last component, task knowledge, consists of the *task* – also known as the "what" view (what needs to be done) and the *task method* – the "how" view (how is it done) on the reasoning task. A task is understood as a knowledge-intensive, reasoning process that usually is iteratively decomposed in smaller tasks (until primitive functions like inferences are encountered), whereas the task method defines how this decomposition is realized and carried out.

To summarize, the CommonKADS schematic knowledge model, composed of the three previously described components, is presented in Figure 1.

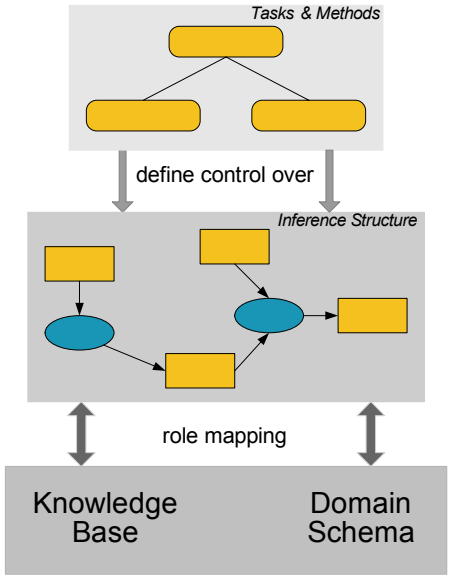

Figure 1: Knowledge model specification in CommonKADS

In many applications, the same tasks appear again and again, therefore it makes sense that within the CommonKADS methodology such tasks were recognized and described in the form of task templates, in order to be reused in different applications. A task template is a partial knowledge model in which inference and task knowledge are specified. Two groups of tasks are distinguished: *analysis tasks* (classification, diagnosis, assessment, monitoring, prediction, etc.) and *synthesis tasks* (design, planning, scheduling, assignment, etc.). For each of these tasks, a template is designed. Table 1 shows a summary for the template of the diagnosis task. Our primary aim for the moment is the creation of the application ontology, thus, since we are not interested in implementing a reasoning system, we concentrate only on the terminology of knowledge roles in the task template and not on the inference structure or task method definitions, which are found in (Schreiber et al. 2000). In the following section, we combine some of these template tasks together, to create a knowledge model for the task of predictive maintenance.

Table 1: General characterization for the *Diagnosis* task template

Diagnosis

Goal	Find the fault that causes a system to malfunction
Typical Example	Diagnosis of a technical device, such as a copier.
Terminology	**Complaint/symptom**: the data that initiate a diagnostic process **Hypothesis**: a potential solution (thus a fault)

	Differential: the active set of hypotheses **Finding(s)/evidence**: additional data about the system being diagnosed **Fault**: the solution found by the diagnostic reasoning process.
Input	Symptoms and or/ complaints
Output	Fault(s) plus the evidence gathered for the fault(s)
Features	In principle, a diagnosis task should always have some model of the behaviour of the system being diagnosed. An example could be a casual model of system behaviour.

2.2 A knowledge model for the predictive maintenance task

Predictive maintenance (for example, of complex industrial systems, large transport vehicles, etc.) is a knowledge intensive task, usually performed or supervised by human experts. Its goal is predicting when and what maintenance actions are due in order to avoid an unexpected breakdown of the system. By reasoning about this goal and how it can be realized in practice, the task can be decomposed as shown in Figure 2.

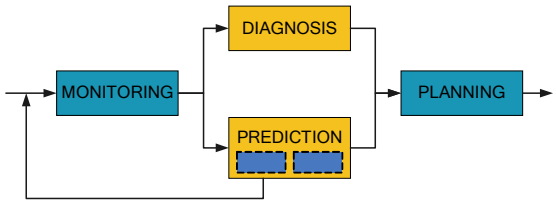

Figure 2: Task decomposition of the predictive maintenance task.

Indeed, the results of monitoring can serve: a) as a natural input for the diagnosis task and b) as input to the prediction task. The prediction task is represented as divided in two parts, because it is necessary to also predict in which intervals the monitoring should be performed (in those cases when monitoring is human-supervised). Both outputs of diagnosis and prediction serve as input to the planning task. This decomposition has the advantage that for the new subtasks, task templates exist in the catalogue of CommonKADS, which can be used as the basis for constructing the model.

- If we establish a list of the knowledge roles that serve as input/output in the CommonKADS task templates and apply it to the predictive maintenance model we created, the most important would be:
- Parameter (a measured or calculated quantity whose value can detect abnormal behaviour)
- finding / evidence (something that can be observed or detected)
- symptom / complaint (a negative finding)
- norm (expected values of a parameter for normal condition)
- discrepancy (a quantified difference to the norm)
- fault (cause of a symptom)
- location (where a symptom or fault is found)
- action (an activity to eliminate a fault or to improve a situation)

As already explained in Section 2.1, these roles can be taken by different knowledge types (domain concepts, relations, or rules). In the next subsection we describe a real scenario of predictive maintenance, giving examples of different mappings between knowledge roles and domain knowledge types.

A real world scenario for predictive maintenance

The selected scenario is drawn from the domain of power engineering, precisely: the predictive maintenance of the insulation system in high-voltage rotating electrical machines (Mustafaraj et al. 2004). It suffices to mention here that the insulation system condition is of vital importance for the operation of an electrical machine, therefore, its ageing process is monitored in order to predict when maintenance is necessary, avoiding a breakdown.

Due to the nature of the measurements and tests to be carried out, the monitoring needs to be performed by human experts. Initially, two activities are performed:

- the monitoring of the invisible state (by performing dedicated measurements); result → parameter values
- the control of the visible state (by visual checks); result → findings (observations)

The human expert then interprets these results and decides to perform other activities if necessary. For example, if the parameter values show a discrepancy from some expected ones or a finding is negative, a search for the cause is started. After the task has been completed, the human expert writes down the interpretations for the operator of the machine. An example of such an interpretation, annotated with the knowledge roles identified in Section 2.2, is given in Figure 3. (The text is a translation of the original German text.)

The knowledge roles annotations show mappings to different domain knowledge types: parameters are domain concepts, discrepancies are attribute values of concepts, symptoms or faults are relations between concepts and so on. If these mappings could be automatically extracted from such texts, they could then be used to create a domain

knowledge model based on empirical evidence. Actually, during the domain analysis for this scenario of predictive maintenance we found out that knowledge of the domain is heuristic, experience-based, incomplete, and uncertain.

The [parameter lost factor characteristic values] indicate [finding a stable behavior], although the [parameter loss factor rising values] are [discrepancy relatively high]. At the [parameter operation voltage], [symptom, fault intensive corona discharges] take place, [symptom which strongly strain the stator winding insulation] at the [location phase lead region]. This will weaken the winding insulation. Therefore, this circumstance should be followed with attention in the future, and in the long run (ca. 5 years) [action a transition to a neutral point] or a [action new winding] should be planned. The [discrepancy negative rising values] indicate [fault coupling problems] in the [location region insulation / stator iron], which nevertheless are not yet threatening for the operation.

Figure 3 : A predictive maintenance evaluation text with annotated knowledge roles

Furthermore, knowledge is found either in the mind of the experts or in written reports containing evaluations such the one in Figure 3. Due to the mentioned characteristics of domain knowledge, it makes sense to try to extract domain knowledge by processing knowledge in these text documents, instead of directly eliciting knowledge from the experts. Nevertheless, human experts will collaborate in the initial annotation with knowledge roles, as well as in the evaluation of the achieved results and the organization of knowledge in ontologies. We will describe such an approach in Section 4 after discussing ontologies and ontology engineering in the following.

3. Lightweight ontologies and ontology engineering

3.1 Ontology definitions

In computer science, ontologies are seen as engineering artefacts that can tell what kind of things can exist in the domain of a system, how these things are related, and what they mean (Welty 2003). A formalization of such a description is given by (Mädche 2002), defining the structure of an ontology as a 5-tuple:

$$O := \{C, R, H^C, rel, A^O\} \qquad [1]$$

where:

- C, R disjoint sets of concepts and relations, respectively
- H^C concept hierarchy (a taxonomy)
- rel function that relates concepts non-taxonomically
- A^O a set of axioms, expressed in a logical language

If we compare this formal definition of an ontology with the informal description of the domain knowledge model in the CommonKADS methodology, it is clear that they refer to the same thing, although in CommonKADS the term ontology was not explicitly mentioned.

Based on the level of generality expressed by an ontology, (Guarino 1998) distinguished among the following types of ontologies: a) top-level ontologies (foundational ontology), b) domain ontologies, c) task ontologies, and d) application ontologies. Application ontologies can be seen as specialization of domain and task ontologies, since concepts defined here correspond to *roles* played by domain entities while performing a certain activity. The example we gave for the predictive maintenance task agrees with this description (or view) of application ontologies.

3.2 Lightweight ontologies

A fully-fledged ontology should contain all the components in the definition [1] and should be expressed in a formal language (e.g. description logic). Indeed, completeness and formality are the two parameters that distinguish between different flavours of ontologies. Assuming that the common core of an ontology would usually contain: a) a vocabulary of terms that refer to the things of interest in the domain, and b) some specification of meaning for the terms, (Uschold & Gruninger 2004) presents a continuum of kinds of ontologies, starting with lightweight ones like glossaries and data dictionaries, followed by more formalized structures like metadata and XML Schemas, up to formal ontologies that support automated reasoning. In principle, every kind of a so-called lightweight ontology could then be upgraded in a more formal ontology when the need arises. However, it is important to notice that even the construction of a formal ontology will usually start with the creation of an informal (or lightweight) ontology. This is also the approach that we are following.

3.3 Ontology engineering approaches

There are several approaches for engineering ontologies, the most usual being: a) building an ontology from scratch, b) building an ontology by using existing resources, c) merging existing ontologies.

Uschold & Gruninger (1996) presents a methodology for developing ontologies from scratch with the participation of domain experts. On the modelling level, it is similar to the CommonKADS approach for domain knowledge modelling, but

stresses more the aspect of formalization, by using a language like KIF (Knowledge Interchange Format). Its balanced middle-out approach for the conceptual hierarchy creation is the most commonly used in practice.

Goméz-Peréz & Manzano-Macho (2003) surveys the most relevant methods, techniques and tools used for building ontologies from existing sources, like text, machine readable dictionaries, knowledge bases, structured and semi-structured data, etc. For the task of building ontologies from text, 18 approaches found in the literature are described. The majority of them makes extensive use of WordNet (Miller 1995), and is intended to enrich existing ontologies with new concepts and relations. Some of the main used techniques mentioned in the survey were: statistical approaches, clustering techniques, natural language processing, term-extraction, linguistic patterns, machine learning, etc. In our framework we use some of these techniques and others, as the discussion in the Section 4 will demonstrate.

3.4 Ontologies for the predictive maintenance application

For the scenario presented in Section 2.3, predictive maintenance of an electrical machine's insulation system, several domain knowledge models (i.e., ontologies) can be constructed. One would be the model of the electrical machine itself, another one that of the electro-magnetic processes and physical quantities related to the operation of the machine, and so on. These ontologies would be the so-called domain ontologies, which represent knowledge of the domain independently of their use. However, we will construct an application ontology for the predictive maintenance. This ontology will have as meta-concepts the knowledge roles defined in the Section 2.2 and will express the relation: task – domain (predictive maintenance – electrical machine). The reason for choosing to build an application ontology instead of domain ontologies is the possession of an internal corpus of documents (in German language) related to the application. Then, in the spirit of reusing existing ontologies, we could use the application ontology to derive concepts and relations for a domain ontology that models, for example, the electrical machine.

4. An active learning framework for annotating knowledge roles

If a corpus of domain documents has been annotated with knowledge roles as in the previously shown example of Figure 4, several usage scenarios can be envisioned: querying, retrieval of context-based information, knowledge extraction, and others. For example, expressions annotated with the same knowledge role could be extracted from the text and processed to recognize and group together different lexical terms expressing concepts, attributes of concepts, or relations between concepts. However, we do not possess such an annotated corpus for our domain and in general neither do other domains. Thus, the principal aspect of the derived research problem is to combine different techniques from natural language processing, machine learning, and information extraction in a learning framework that would semi-automatically annotate such corpora with knowledge roles. The approach cannot be fully automatic, because the knowledge roles will always depend on the domain, and thus, annotated examples for the learning process have to be provided by a domain user. A related aspect of the research problem would be then to alleviate the annotation burden by keeping the number of instances to be manually annotated low. This process is known as active learning (Jones et al. 2003), because the learning framework will present to the user only instances that have a highly anticipated value for the learning process. Before discussing the learning framework in Section 4.2, we shortly discuss the learning of semantic roles for natural language understanding that served as a model for our approach.

4.1 Semantic roles versus knowledge roles

Recent research efforts in the natural language learning domain concentrate on the task of semantic role labelling (SRL) (Carreras & Marquez 2004). For this task, two large corpora have been manually annotated with semantic roles: PropBank and FrameNet. Several machine learning strategies are being trained on these corpora with the aim to build statistically-based semantic taggers and parsers, similar to the existing syntactic taggers and parsers. In the context of SRL, semantic roles are seen as arguments of the verb in a sentence, for example, the agent, the theme, the patient, the location, etc. An example is:

[$_{agent}$ She] clapped [$_{body\_part}$ her hands] [$_{cause}$ in inspiration].

As it can be noticed, the similarity to knowledge roles for knowledge modelling is obvious. Nevertheless, since SRL is intended to cover domain independent, naturally occurring language, the labelled corpora need to be large in order to contain examples of a large number of possible semantic roles. In contrast, knowledge roles are fewer in number and the domain language restricted in its variability and ambiguity. These reasons make the task of learning knowledge roles

more feasible. In our framework we make use of research results from the field of SRL by following (Gildea & Jurafsky 2002) in the construction of learning features and (Fleichman et al. 2003) in the use of maximum entropy as the learning approach. More importantly, we follow the principle that the possible roles to be learned are related to the meaning of the sentence verb.

4.2 Components of the learning framework

4.2.1 Natural Language Processing (NLP)

Since our approach is inspired by the semantic role labelling in natural language, NLP is the first task of the framework. A good sentence tokenizer is necessary, because every text sentence is handled separately. The next step is the tagging of sentences with part of speech (PoS) tags. We used TreeTagger (Schmid 1995), a probabilistic tagger that does also lemmatisation for the majority of the words; this is useful for language with rich morphology, such as German, the language of our domain corpus. A chunking or a parsing step that can detect noun phrases (NP), verbal phrases (VP), or prepositional phrases (PP) is necessary to get the syntactic structure of the sentence. We used a statistical parser for the German language (Dubey 2003) which produces relatively flat tree-structures as the one shown in Figure 4. Then the tagged, stemmed, and parsed data are collected together to create a tree data structure where the leaves are the words of a sentence and the internal nodes syntactic phrases. Each element of the tree is referred to as a constituent.

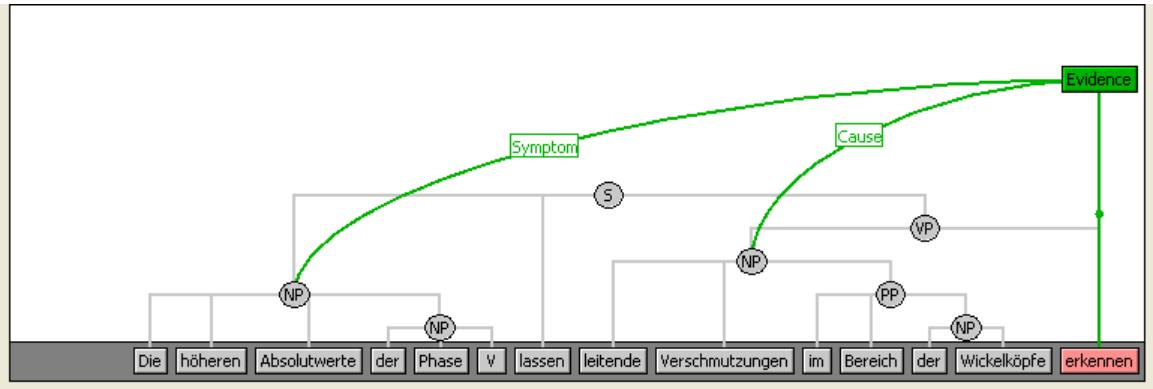

Figure 4: Annotation of a parsed sentence with the knowledge roles “Symptom” and “Cause”.

4.2.2 Feature selection and learning algorithm

Learning knowledge roles is seen as a classification problem. Consider the following German sentence in Figure 4 (translation: “The higher absolute values of phase V reveal conducting dirtiness in the area of the winding.”). Here two constituents (NP) are labelled with the knowledge roles *Symptom* and *Cause*, all the other constituents will have a *None* label. The image on Figure 4 is produced with the Salsa annotation tool (Erk et al. 2003) that was used for the manual annotation of sentences with knowledge roles.

The task is now to learn a classifier that is able to classify constituents of the tree in one of the classes. In order for a classifier to learn a class, it needs features extracted from the training data. The features in our case come from the tree representation of a sentence, containing in this way syntactic and semantic information for each constituent. The problem with features for text classification tasks is that the sets of their values tend to explode, evincing in this way the data sparsity problem. Thus, it is necessary to invest some time in the creation and selection of the right features. The literature mentioned in Section 4.1 is a good information source in this issue.

After the feature values are encoded as numeric values, every known machine learning algorithm could be used for the classification: Naïve Bayes, EM (Expectation Maximization), SVM (Support Vector Machines), or maximum entropy. Currently, we have chosen the maximum entropy classifier (Berger et al. 1996), because its probabilistic confidence in the class labels is used during the active learning selection strategy. Mallet (McCallum 2002) is the tool containing an implementation of the MaxEnt algorithm that we use.

4.2.3 Active learning selection strategy

The previously listed classifiers are based on supervised learning. The classifier learns during a training session with manually labelled data and uses the learned model to classify new, unseen data. Usually, training data are very costly to acquire, therefore, it is desirable to find learning approaches that makes the most from the training data we could get. One could expect from a do-

main user to label a few hundred sentences, while the corpus could contain several thousands, thus selecting the most informative examples becomes a priority. The selected examples should serve to the task, in our case to the labelling of knowledge roles, thus, we need to select sentences were the knowledge roles appear. Our assumption is that the meaning of the principal verb of a sentence (or clause) will evoke most of the knowledge roles. For example, verbs like '*to reveal'* (German: erkennen, see Figure 4) or *'to trace'* (German: zurückführen), evoke the meaning of *Evidence* where the knowledge roles *Symptom* and *Cause* will participate. Based on the frequency of such verbs on the text, we connect to each of them the corresponding set of roles, annotate some sentences with these roles, and learn the classifier by performing iterations to improve the accuracy of labelling. This process is described in more details in (Mustafaraj et al. 2005).

4.3 Populating the application ontology

As the result of the learning process described above, sentences on the text are annotated with knowledge roles like: finding, symptom, cause, condition, or location. Since both the text documents as well as the annotation of the sentences are represented in the XML language, a lightweight ontology containing pairs of *{knowledge roles, domain text phrases}*, queryable by XPath is dynamically available. For a more fine-grained ontology, further processing of the annotated text phrases would be necessary. During this post-processing, words tagged as nouns are candidates for the concepts of the ontology, words tagged as adjectives are candidates for attributes of the related concepts. We also use the document context where the annotations appear for further grouping. In our domain corpus, diagnostic evaluations have always the title of the performed measurement; therefore, we group together annotated terms arising in the same context. For example, terms in the "location" annotation for the stator diagnostic evaluations describe components of the machine's stator.

4.4 Evaluation

In order to evaluate the results of the learning framework, we manually annotated a set of 570 sentences with 8 different knowledge roles (e.g. finding, symptom, etc.). The processing of the sentences as described in Section 4.2.1 created 26296 constituents, each of them represented with a feature vector of length 23. During the annotation, 1852 constituents were labelled with one of the roles, the rest of the constituents received the label "none". Averaging 10 trials of 2-fold validation with a split of 0.7(training): 0.3 (testing), we received a recall of 89.96% and a precision of 92.46%.

By inspecting the errors, the cause was frequently found to be the erroneous parsing of the sentences, especially for very large sentences (up to 30 words), which had forced us to split some roles across several constituents. Since at the current state of research, the accuracy of natural language processing tools cannot be expected to be optimal, we could view the achieved recall/precision in the knowledge roles learning task with optimism.

5. Conclusion

The paper presented an approach of using knowledge role annotations for acquiring domain knowledge to build a lightweight, application ontology. The advantages of annotating knowledge roles directly in the corpus are manifold. We will not consider here its uses for different knowledge related tasks, like question answering, textual case base reasoning, or text mining, but will restrict only on its advantages for acquiring knowledge for ontology construction. One of the most important advantages is the preservation of the context where roles appear, because this permits the identification of relations between role instances, and as a consequence of the relations for the ontology. Not only relations between different roles within context, but also the syntactic relations within the role instance itself are preserved. This is the source where attributes for the concepts of the ontology will be derived. Furthermore, several lexical forms expressing the same concept or attribute are revealed, which can be stored as synonyms in the lexicon connected to the ontology.

Whereas the advantages of such an annotation cannot be praised enough, the difficulties related to the practical implementation of the knowledge roles annotation process should not be left without attention and thus represent several areas for future work. Very high quality natural language processing tools like POS taggers and parsers mostly exist for the English language only. Semantic role labelling tools that could be easily adapted to knowledge roles learning have not yet made it out of the research labs of universities and no clear guidelines for feature creation and selection for the learning algorithms are available. Furthermore, in order for the approach to be useful for real-world tasks, the components need to be integrated in a user-friendly framework that can perform many of the tasks: annotation, extraction, ontology creation, or reasoning. An example of such an integrated framework is Protégé (http://protege.stanford.edu), where new compo-

nents for separate tasks can be added as plug-ins. Finally, introducing appropriate refinements into the active learning approach is another interesting area for future research.

Acknowledgments

We are very grateful to Katrin Erk und Sebastian Padó from the Salsa Project, University of Saarland, for making available their Salsa annotation tool, code for semantic role labelling, as well as answering many questions on linguistic questions. We thank Amit Dubey for his parsing tool and helpful insights on the topic.

References

Abecker, A. and van Elst, L. (2004) "Ontologies for Knowledge Management", in *Handbook of Ontologies* , Springer, Berlin, 435-455

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., and Wielinga, B. (2000) *Knowledge Engineering and Management: The CommonKADS Methodology*, The MIT Press, Cambridge, MA

Berners-Lee, T., Hendler, J., Lassila, O. (2001) „The Semantic Web" in *Scientific American*, May 17, 2001.

Mustafaraj, E., Peters, M., Freisleben, B. (2004) "Issues in Developing an Experience Management System for a Preventive Maintenance Application in Electrical Power Engineering" in *Proc. Of ECKM'04*, Academics Publishers, Paris, France, 641-649

Welty, C. (2003) „Ontology Research", *AI Magazine*, Vol. 24, No.3, 11-12

Mädsche, A. (2002) *Ontology Learning for the Semantic Web*, Kluwer Academics Publishers

Guarino, N (1998) "Formal ontology and information systems" in *Proc. of Formal Ontologies in Information System (FOIS'98)*, IOS Press, 3-15

Goméz-Peréz, A., Manzano-Macho, D. (2003) "A Survey of Ontology Learning Methods and Techniques" *OntoWeb Deliverable 1.5*, http://ontoweb.org/Members/ruben/Deliverable%201.5/view

Miller, A. G. "WordNet: A Lexical Database for English" in *Communications of the ACM,* Vol. 38, No.11, 39-41

Uschold, M., Gruninger, M. (1996) "Ontologies: Principles, Methods, and Applications" in *Knowledge Engineering Review*, Vol.11, No. 2, 93-155

Uschold, M., Gruninger, M. (2004) "Ontologies and Semantics for Seamless Connectivity" in *SIGMOD Record*, Vol 33, No. 4, 58-64

Jones, R., Ghani, R., Mitchell, T., Riloff, E. (2003) "Active Learning for Information Extraction with Multiple View Features Sets" in *Proc. of Adaptive Text Extraction and Mining, EMCL/PKDD-03,* Cavtat-Dubrovnik, Croatia, 26-34

Carreras, X., Marquez , L. (2004) "Introduction to the CoNLL Shared Task: Semantic Role Labeling" in *Proc. of 8th Conference of Natural Language Learning,* Boston, MA, 89-97

Gildea, D., Jurafsky, D. (2002) "Automatic Labeling of Semantic Roles" in *Computational Linguistics*, Vol 28, No 3, 245-288

Fleischman, M., Kwon, N., Hovy, E. (2003) "Maximum Entropy Models for FrameNet Classification" in *Proceedings. of EMLNP-2003*, Saporo, Japan, 49-56

Schmid, H. (1995) "Improvement in Part-of-Speech Tagging with an Application to German" in *Proceedings of the ACL SIGDAT-Workshop*, Dublin, Ireland, 47-50

Dubey, A. (2003) "Statistical Parsing for German", PhD Thesis, University of Saarland, Germany

Berger, A., Della Pietra, S., Della Pietra, V. (1996) „A Maximum Entropy Approach to Natural Language Processing" in *Computational Linguistics*, Vol. 22, No. 1, 39-71

McKallum, A. K. (2002) "MALLET: A Machine Learning for Language Toolkit", http://mallet.cs.umass.edu

Erk, K., Kowalski, A., Padó, S. (2003) "The Salsa Annotation Tool - Demo Description" *in Proceedings of the 6th Lorraine-Saarland Workshop*, Nancy, France, 111-113

Mustafaraj, E., Hoof, M., Freisleben, B. (2005) "Learning Semantic Annotations for Textual Cases" *in Workshop Proceedings of ICCBR '05*, Chicago, IL, 99-109

