LARC: Learning to Assign Knowledge Roles to Textual Cases

Eni Mustafaraj[1], **Martin Hoof**[2], **and Bernd Freisleben**[1]

[1]Dept. of Math. and Computer Science, University of Marburg, Hans-Meerwein-Str 3, D-35032 Marburg, Germany
eni,freisleb@informatik.uni-marburg.de

[2]Dept. of Electrical Engineering, FH Kaiserslautern, Morlauterer Strasse 31, D-67657 Kaiserslautern, Germany
m.hoof@et.fh-kl.de

Abstract

In this paper, we present a learning framework for the semantic annotation of text documents that can be used as textual cases in case-based reasoning applications. The annotations are known as knowledge roles and are task-dependent. The framework relies on deep natural language processing techniques and does not require the existence of any domain-dependent resources. Several experiments are presented to demonstrate the feasibility of the proposed approach. The results show that the framework allows to robustly label cases with features which can be used for case representation, contributing to the retrieval of and the reasoning with textual cases.

Introduction

An important problem in case-based reasoning (CBR) is how to represent cases by a set of features that capture their meaning appropriately. While in some occasions the cases already have an inherent structure of easily identifiable features, in others, like e.g. in Textual or Conversational CBR, this is frequently not true. In the latter applications, the features are typically provided manually by domain experts—a costly approach, especially when the number or the size of cases is large, or when there are no simple rules for creating a semantic mapping between features and textual phrases. Thus, approaches that are able to learn to extract features and perform such mappings are an interesting area of research.

In previous work (Mustafaraj, Hoof, & Freisleben 2005), we introduced such a learning approach for semantically annotating textual cases with features that we refer to as *knowledge roles.* In the meantime, we have continually improved our framework, and in this paper we present new and more extensive results, as well as discuss the benefits and costs of our learning framework called LARC (Learning to Assign Roles to Cases).

Recently, in Conversational CBR, an approach known as FACIT (Gupta & Aha 2004), (Gupta, Aha, & Moore 2004) for extracting and organizing features from text has been introduced. While the underlying purpose in this as in our approach is the same, the implementation philosophies differ largely. Actually, both approaches rely on deep natural language processing, but FACIT uses domain-dependent sublanguage ontologies which need initially be implemented for a given domain; we, instead, do not assume the existence of any domain-dependent resources, although we recognize their advantanges. The reality is that usually such ontologies are not available.

The work on Textual CBR in the legal domain (Brüninghaus & Ashley 1999), (Brüninghaus & Ashley 2001), (Brüninghaus & Ashley 2005) has been an inspiration and we share many foundational ideas with it. Nevertheless, that work had the advantage that the textual cases were manually annotated with abstract features (factors) beforehand, a fact that has made all the explorations on the topic of learning and reasoning with textual cases published during these years possible. Our philosophy, on the other hand, is to investigate how to approach the representation of and reasoning with textual cases when one has to start from scratch and grow incrementally.

The paper is organized as follows. Initially, we discuss knowledge roles as features for representing textual cases. Then, we motivate the necessity of having natural language semantic resources that could support Textual CBR. Further, our learning framework LARC is described, followed by experiments and evaluation of results. Finally, we briefly discuss what we plan to do further with the annotated cases.

Features for Representing Textual Cases

In the knowledge engineering (KE) literature, a clear distinction between the notions of domain and task is drawn. For example, the CommonKADS methodology (Schreiber 2000) models knowledge in three separate layers: domain knowledge, inference knowledge, and task knowledge. The rationale behind such a distinction is the observation that while the number of domains or subdomains could be large, the number of essential tasks that can be performed remains small, and new tasks can be always represented as a combination of more basic tasks. In general, two large groups of tasks are identified: analytic tasks (such as diagnosis, classification, monitoring, etc.) and synthetic tasks (such as planning, design, etc.).

The CommonKADS methodology goes a step further and defines a library of generic task templates for the identified basic tasks, which can be applied to different domains. To illustrate, consider the following sentences, be-

Medicine:
A chest X-ray revealed right-sided lung nodules.
Flight Accidents:
Radar data revealed two transponder-equipped airplanes.
Power Engineering:
The plotted current curves revealed a homogenous insulating structure.

longing to three different domains. Despite their different domain-specific vocabulary, when it comes to the underlying task (diagnosis) to which these propositions refer, all sentences can be represented as: `[Observed_Object] revealed [Finding]`.

The annotations Observed_Object and Finding are referred to as knowledge roles in the CommonKADS methodology. They are simply abstract names indicating a role in the reasoning process and are used as an interface between domain knowledge types and inference functions. For example, in the diagnosis task, the knowledge roles that serve as input/output in the reasoning process are: complaint (or symptom), finding (or evidence), hypothesis, fault (or condition), etc. Indeed, independently of the domain, it is intuitively assumed that the underlying representation of the problem solving situation will undoubtedly include some of these roles. Consequently, the question is how do these roles map to domain knowledge types, because such a mapping could serve as an initial representation for cases in CBR, too.

Considering the last question within the context of Textual CBR, where problem solving situations are expressed in free or semi-structured natural language, we are concerned with answering the following questions:

- Is it possible to automatically annotate textual cases with such domain-independent, task-related knowledge roles?
- Does the use of such annotations as case representation features improve the case-retrieval step compared to more easily implementable indexing features, like e.g., domain keywords?
- What kind of domain and case knowledge could be extracted from the mappings {`knowledge roles`, `text phrases`} that would support reasoning?

In this paper, we try to extensively answer the first question by describing the learning framework that we have developed for such a task. Due to space reasons, we only sketch the answers for the last two questions, with the intention to elaborate on them in further publications.

As a starting point for answering the first question, we use a collection of text documents (further referred to as corpus) from a technical domain (predictive maintenance of the insulation systems of high-voltage rotating electrical machines) described in detail in (Mustafaraj, Hoof, & Freisleben 2005). While the corpus itself is domain-specific, we do not assume the existence of any domain related knowledge for the annotation step:

- The corpus is not a collection of cases in a restricted sense. It contains textual descriptions of problem solving situations, which can be repetitive. That is, no human expert has previously selected interesting cases for annotation. Thus, with the annotation process we also try to handle the situation of iteratively creating a base of interesting cases for reasoning.
- No background knowledge in the form of lexica, thesauri, or ontologies is available.
- Only publicly available natural language processing (NLP) tools are used for processing, or only publicly available sources like FrameNet or VerbNet are consulted.

Our only constraints for a corpus that will be annotated are:

- The corpus language should be grammatically correct, i.e., complete sentences compounded of noun phrases and verbs should be available.
- The corpus language should be related to a knowledge task (e.g. diagnosis, configuration, or assessment) within a given domain, i.e. the text follows an underlying structure that can be represented by a set of repeated knowledge roles.

There are several reasons for these constraints. To start with, complete sentences are needed because the annotation process is based on a tree data structure created from a fully parsed sentence. Then, verbs are the so called *target feature* for evoking a semantic frame where the knowledge roles are embedded. Finally, a corpus with semantically repetitive structure could permit a probabilistic learning without the need of being extensively large.

Before describing the learning framework, we discuss in the next section how current research on natural language understanding can contribute to the needs of Textual CBR.

On Linguistic Resources for Annotating Cases

Currently, one of the new research developments in natural language understanding is the task of semantic role labeling (SRL) (Carreras & Màrquez 2004), (Carreras & Màrquez 2005). This research direction, first introduced in (Gildea & Jurafsky 2002), draws on the Frame Semantics of Charles Fillmore (Fillmore 1976). At present, the corpora of FrameNet and PropBank are used for training classifiers on this task. The improved results of 2005 (Carreras & Màrquez 2005) compared to those of 2004 (Carreras & Màrquez 2004) are based on the use of full syntactic parsed trees instead of the shallow parsing techniques used one year before.

The rationale behind trying to improve on the SRL task is the hope of building so called semantic parsers, which could then be used at tackling such hard tasks like question answering or language generation.

While such envisioned semantic parsers for general, unrestricted natural language still need some time to become available, we regard their working philosophy and underlying assumptions as an interesting opportunity to advance the state-of-the-art of Textual CBR, in particular with respect to the following areas:

- Creation of publicly available resources of frame semantics for frequent tasks handled in Textual CBR, containing semantic and lexical information as in FrameNet.

- Distribution of domain-independent code that automates the process of creating learning features for the annotation process.

The fact that the existence of such resources could be beneficiary to some of the problems treated in Textual CBR, can be demonstrated by an example taken from the legal domain. FrameNet has a large portion of sentences drawn from legal processes, resulting in many legally concerned frames with their related roles and lexical items. Had such resources been available at the time when CATO (Aleven & Ashley 1995) was developed, they could probably have largely supported the manual process of constructing the factors for indexing the cases. For example, in (Brüninghaus & Ashley 1999) the following sentence, annotated with the factor F1 `Disclosure-In-Negotations`, is found:

[f1 Plaintiff's president sent a letter to defendant which conveyed plaintiff's manufacturing formula.]

In FrameNet, the verb *convey* evokes the frame *Statement*, which inherits from frame *Communication* and is inherited by *Reveal_Secret* that together can be seen as components of the `Disclosure-In-Negation` factor.

Actually, the creation of such resources is not a trivial task. FrameNet or WordNet are the results of the work of many people during the course of many years. But they are also far too exhaustive in their efforts to capture the usage of the entire (English) language. On the other hand, semantic language constructs used to convey the meaning of knowledge tasks relevant to problem solving (thus, Textual CBR, too) are more restricted and not very prone to sense ambiguity, characteristics that make the undertaking of creating such resources more feasible. Once created, they have the advantage of being reusable across domains where the same tasks are performed, and can be incrementally extended in a hierarchical fashion to capture specialized meanings.

The LARC Framework

Figure 1 represents a changed view of the learning framework, previously introduced in (Mustafaraj, Hoof, & Freisleben 2005). One of the changes is the replacement of the chunker with a statistical parser (Dubey 2003), since the learning results with the chunker were found to be unsatisfactory. Then, due to the fact that the output of the parser can be represented by a tree structure, several complex features for the learning process can be created, something that was not possible or easy implementable with the chunker.

We experimented with three different statistical parsers: the Stanford parser (Klein 2005), available for English, German, and Chinese; BitPar (Schiehlen 2004) and Sleepy (Dubey 2003)—the last two specific for the German language. We then selected Sleepy, due to its speed and its more specific output (it labels every constituent of the tree with its grammatical function, like: SB (subject), OC (clausal object), NG (negation), etc.

The execution flow in the framework is as follows:

1. *Tagging*: Text is tagged by a probabilistic tagger (Schmid 1995), with the purpose of gaining stemming information. For words where this information could not be found, we created a list with pairs of word-stem.

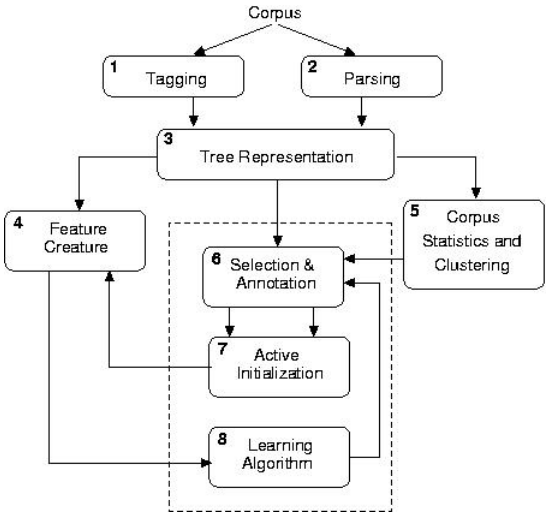

Figure 1: The architecture of LARC.

2. *Parsing*: Text is parsed by the statistical parser (Dubey 2003), which outputs a bracketed data structure.
3. *Tree representation*: The output of the tagger and the parser are combined together to create for every sentence a tree data structure. For export purposes the tree is stored in a XML format, something that makes then visualizations and operations (like subcorpora creation, etc.) with the TigerSearch tool[1] (freely available for research purposes) possible.
4. *Feature Creation*: Out of the tree data structure, a set of syntactical and semantic features is created. For this step, no software frameworks are publicly available yet, although the Salsa Project[2] is developing an integrated framework for semantic role labeling, expected to be released soon. In the meantime, we have implemented a first prototype that was used to construct some of the most important features described in the literature on the SRL task.
5. *Corpus Statistics and Clustering*: The frequency of verbs is calculated and sentences with the same verbs are grouped together. Then, within each group, clusters of sentences with the same sub-tree structure containing the target verb are created.
6. *Selection and Annotation*: Some sentences from the biggest clusters created in step 5 are presented to a human annotator to be annotated with roles. For the annotation we use the Salsa Tool (Erk, Kowalski, & Pado 2003)(freely available for research purposes). The Salsa tool is very intuitive in its use, offering a click-and-point annotation interface. Figure 2 shows the annotation of a sentence.
7. *Active Initialization*: The frames and roles assigned during the manual annotation are automatically spread to sen-

[1] http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/
[2] http://www.coli.uni-saarland.de/projects/salsa/

tences of the clusters having the same sub-tree structure with the annotated sentences. In this way, a first pool of labeled data for training is created with a small effort. Understandably, as labels are considered the roles assigned to constituents of a sentence during annotation. The constituents without a role receive the label None automatically .

8. *Active Learning*: For learning, we use a maximum entropy classifier, freely available in the MALLET package (McCallum 2002). The rationale behind choosing this classifier is the fact that it assigns a probability to every label, making the instances with less discriminatory power for learning evident. Sentences with such instances are presented to the user for manual annotations in step 6, and the steps 6–8 are repeated a few times.

What distinguishes our learning approach from the SRL task is the use of an active learning approach. As previously mentioned, SRL is performed on the PropBank corpus which has around 55.000 manually annotated sentences from the Wall Street Journal or the FrameNet corpus with 135.000 annotated sentences. For specific domains, as it will be the case in Textual CBR, manual and indistinctive annotation in a large proportion is prohibitive. Therefore, we claim that the use of an active learning strategy, as the one described, could make the task feasible and successful, besides making it possible to choose for annotation the desired knowledge roles, as discussed in the following.

Selecting Frames and Roles

The active learning approach we propose is completely domain independent. Important is only that the corpus complies with the two constraints identified previously. Nevertheless, the roles to be learned and the frames they are related to will be task and domain dependent. Exactly in this issue we previously proposed a research community effort for the creation of a repository of resources organized by tasks and domains, where roles, frames, lexical items evoking these frames, as well as annotated sentences are made available. In this way, this information can be reused when new applications of Textual CBR or larger projects of knowledge engineering need to be implemented.

While in the FrameNet project several word categories are defined as frame evokers (noun, verbs, adjectives), in our work we have initially considered only verbs, because they are usually more independent of the domain as, for example, nouns.

Consulting FrameNet, VerbNet, and (Schulte im Walde 2003) for German verbs, we have grouped the most frequent verbs appearing in our corpus as shown in Table 1. For space reasons, only some verbs for each frame are shown.

Identifying roles to be associated with each frame is a more subtle issue. FrameNet could be used as a starting point, but one should be aware of its linguistic bias. For example, there is a frame `Evidence` in FrameNet with the two most important roles being `Proposition` and `Support`. Someone performing a diagnosis task will hardly think in such terms, but simply in terms of Cause (for Proposition) and Symptom (for Support).

We started our annotation work with only two frames (Observation and Evidence) and two roles per frame (Observed_Object and Finding; Symptom and Cause), and in the process of exploring and annotating the corpus, added other roles and discovered other frames when some semantic structures appeared frequently. Currently, the more elaborated frame is Observation (the roles and their abbreviated form are shown in Figure 3).

Frame *Observation*

ObservedObject	ObsObj
Finding	Fin
ReferencePoint	Ref
Location	Loc
Reason	Reason
Risk	Risk
Manner	Manner

Figure 3: Knowledge Roles for the Observation Frame.

From all these roles, only Manner is purely linguistical, since it serves to capture among others negation.

Experimental Evaluation

For evaluation purposes, we created two subcorpora, one containing descriptions from the category of numerical measurements (the isolation current on three phases of the stator), the other containing descriptions from the category of visual controls (visual control of the wedging system of the electrical machine). We refer to each text description as an episode.

$Subcorpus$	Ep	Se	$UnSe$	$Roles$
IsolationCurrent	491	1134	585	1847
WedgeSystem	453	775	602	1751

Table 2: Statistics for the manually annotated subcorpora.
Ep - no. of episodes, *Se* - no. of all sentences,
UnSe - no. of unique sentences, *Roles* - no. of roles.

During the parsing, each episode is divided in sentences and only unique sentences are selected for further processing. Using our framework incrementally, we annotated these sentences and then manually controlled each, in order to receive a gold version for the evaluation process of active learning. Some statistics are summarized in Table 2.

Experiment 1

In the first experiment, we measured the upper bound of our classification scheme. For that, we performed 10 trials of cross-validation with a split of 0.70:0.30 of the gold set instances. The measures of recall and precision for the two subcorpora are given in Table 3. The reason for this experiment is to acquire a metric that could indicate how well our active learning approach is doing from one iteration to the other. As we can see from the values, recall is between 83–90%. This is often due to errors made by the parser, so that we were forced to split some roles across several constituents. Furthermore, some of the roles like Risk or Location do appear rarely, making the usual sparsity problem of learning with text data visible.

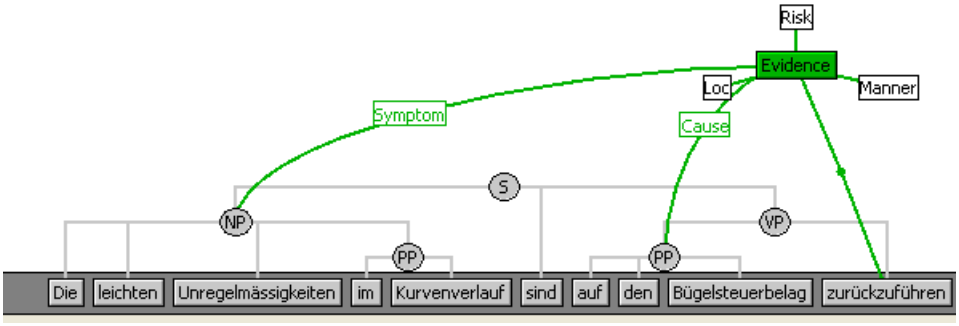

Figure 2: Annotation with the Salsa Tool. English Translation: "The slight anomalies in the curve shape are traced back to the stress control coating."

Frame	*Verbs*
Observation	feststellen, zeigen, aufweisen, erkennen lassen, liegen, ermitteln, ergeben detect, show, exhibit, recognize, lay, determine, result in
Change	verändern, ansteigen, erhöhen, sinken, verbessern, verschlechtern, übergehen change, raise, increase, drop, improve, deteriorate, turn into
Evidence	zurückführen, hindeuten, hinweisen, ableiten, schliessen lassen, hervorgehen trace back to, point to, indicate, derive, conclude, follow
Activity	erfolgen, durchführen, reparieren, abbrechen, untersuchen, beseitigen, weiterführen take place, perform, repair, break off, examine, eliminate, continue

Table 1: Some frequent verbs (in German and English) and their parent frames.

Experiment 2

In this experiment, we measured in how many iterations our active learning strategy can achieve results comparable to those of Experiment 1. In each iteration, 10 new sentences were annotated. The results in the Table 4 belong to the subcorpus IsolationCurrent.

Experiment 3

Finally, we measured what results are possible if we use the classifier trained with one subcorpus to label the other subcorpus. In Table 5, the results for the roles common to both subcorpora are presented.

Subcorpus	*Recall*	*Precision*
IsolationCurrent	89.96%	92.46%
WedgingSystem	82.91%	88.22%

Table 3: Results of Experiment 1.

Iteration	*Sentences*	*Recall*	*Precision*
0	10	67.37%	76.48%
1	20	77.25%	84.35%
2	30	78.10%	85.89%
3	40	77.91%	85.74%
4	50	79.59%	86.62%

Table 4: Results of Experiment 2.

TrainingFile	*TestFile*	*Recall*	*Precision*
IsolCurr	WedgeSys	76.48%	85.86%
WedgeSys	IsolCurr	64.23%	73.67%

Table 5: Results of Experiment 3.

Discussion of Results

The active learning approach in Experiment 2 achieved a F-Measure $F_{\beta=1} = 80$ with only 50 annotated sentences, which is 10 points below that achieved with approximatly 500 annotated sentences in Experiment 1. We regard these initial results as very promising, although more tests could be necessary for a more reliable analysis. Especially interesting are the results of Experiment 3. Although the two subcorpora contain textual descriptions of different types of diagnostic measurements, they could successfully bootstrap the learning process on the other subcorpus, based on the fact that they share knowledge roles.

Uses of Annotated Cases

In this phase of our work, we are investigating the following uses for the annotated episodes of problem solving situations.

Creating a Case Base

We are interested in dividing the episodes in groups of *interesting* and *non-interesting*, from their exptected contribution in solving new diagnostic problems. For this, we represent each episode with a fingerprint of its frames and roles. In

this way, we get the following representation, independent of the syntactic surface of the textual episodes:

Observation(ObsObj, Find), Observation(Find, Loc), Evidence(Symptom, Manner, Cause)

Our intention is then to use these fingerprints for clustering, in order to find interesting cases for the case base. Still, an automatization for this procedure has yet to be implemented.

Collecting Domain Knowledge

Our cases, as well as their annotations with frames and roles are in XML format. With simple XPath queries, we are able to identify for each type of diagnostic procedure the list of phrases annotated with roles such as Symptom and Cause. Nevertheless, here we detected a problem similar to that of anaphora in NLP, which needs some further thought. More precisely, in some sentences the role Symptom is matched to a pronoun, which usually refers to a constituent annotated in the preceding clause or sentence with a Finding or Observed_Object role. Thus, a heuristic for automatically resolving such relations will be needed.

Retrieving Cases via Roles

Similarly to the previously discussed issue, the XML representation and XPath are used to retrieve cases described by the desired roles any time. Furthermore, since stems of constituents are also stored in the XML format, queries combining roles and words can be executed.

Conclusions

We have presented the LARC learning framework for the semantic annotation of text documents with task-related knowledge roles. We regard such an annotation as a first important step for identifying and representing textual cases for case-based reasoning applications. In the future, apart from trying to improve the accuracy of learning, we will concentrate on the issues presented in the last section, in order to enable reasoning with textual cases annotated with knowledge roles.

Acknowledgments

We are grateful to Katrin Erk, Sebastian Pado, Amit Dubey, and Helmut Schmid for making their linguistic tools available as well as offering invaluable advice. We thank the anonymous reviewers for their helpful comments.

References

Aleven, V., and Ashley, K. 1995. Doing things with factors. In *Proc. of the 5th Intl. Conf. on Artificial Intelligence and Law, ICAIL'95*, 31–41.

Brüninghaus, S., and Ashley, K. 1999. Bootstrapping case base development with annotated case summaries. In *Proc. of 3d ICCBR*, volume 1650 of *Lecture Notes in Artificial Intelligence*, 59–73. Springer-Verlag.

Brüninghaus, S., and Ashley, K. 2001. The role of information extraction for textual CBR. In *Proc. of 4th ICCBR*, volume 2080 of *Lecture Notes in Computer Science*, 74–89. Springer-Verlag.

Brüninghaus, S., and Ashley, K. 2005. Reasoning with textual cases. In Muñoz Avila, H., and Ricci, F., eds., *Proc. of ICCBR 2005*, volume 3620 of *Lecture Notes in Artificial Intelligence*, 137–151. Springer-Verlag.

Carreras, X., and Màrquez, L. 2004. Introduction to the coNLL shared task: Semantic role labeling. In *Proc. of 8th Conference of Natural Language Learning*, 89–97.

Carreras, X., and Màrquez, L. 2005. Introduction to the coNLL-2005 shared task: Semantic role labeling. In *Proc. of 9th Conference of Natural Language Learning*, 152–165.

Dubey, A. 2003. *Statistical Parsing for German*. Ph.D. Dissertation, University of Saarland, Germany.

Erk, K.; Kowalski, A.; and Pado, S. 2003. The salsa annotation tool-demo description. In *Proc. of the 6th Lorraine-Saarland Workshop*, 111–113.

Fillmore, C. 1976. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conf. on the Origin and Development of Language and Speech*, volume 280, 20–32.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling of semantic roles. In *Computational Linguistics*, volume 23, 245–288.

Gupta, K., and Aha, D. 2004. Towards acquiring case indexing taxonomies from text. In *Proc. of the 17th FLAIRS Conference, AAAI Press*, 59–73.

Gupta, K.; Aha, D.; and Moore, P. 2004. Learning feature taxonomies for case indexing. In *Proc. of ECCBR 2004*, volume 1809 of *Lecture Notes in Artificial Intelligence*, 211–226. Springer-Verlag.

Klein, D. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. Dissertation, Stanford University.

McCallum, A. 2002. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Mustafaraj, E.; Hoof, M.; and Freisleben, B. 2005. Learning semantic annotations for textual cases. In *Workshop Proc. of ICCBR'05*, 99–109.

Schiehlen, M. 2004. Annotation strategies for probabilistic parsing in german. In *Proc. of the 20th Intl. Conf. on Computational Linguistics*.

Schmid, H. 1995. Improvement in part-of-speech tagging with an application to german. In *Proc. of the ACL SIGDAT-Workshop*, 47–50.

Schreiber, G. e. a. 2000. *Knowledge Engineering and Management: The CommonKADS Methodology*. Cambridge, MA: The MIT Press.

Schulte im Walde, S. 2003. *Experiments on the Automatic Induction of German Semantic Verb Classes*. Ph.D. Dissertation, Universität Stuttgart, Germany.