INTEGRATION OF LINKED OPEN DATA IN CASE-BASED REASONING SYSTEMS

Christian Severin Sauer[1] **Kerstin Bach**[2] **Klaus-Dieter Althoff**[2]

University of Hildesheim, Dep. of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
[1]christiansauer@gmail.com, [2]{lastname}@iis.uni-hildesheim.de

Abstract

This paper discusses the opportunities of integrating Linked Open Data (LOD) resources into Case-Based Reasoning (CBR) systems. Upon the application domain travel medicine, we will exemplify how LOD can be used to fill three out of four knowledge containers a CBR system is based on. The paper also presents the applied techniques for the realization and demonstrates the performance gain of knowledge acquisition by the use of LOD.

1 Introduction

The recent developments in the field of Linked Open Data (LOD) further added value to the already existing vast amount of structured information that is available through the use of LOD [Bizer *et al.*, 2008]. The availability of such an amount of structured information suggests applying LOD to the semi-automatic generation of knowledge. In the context of our work knowledge represents the data that is organized in the knowledge containers of a Case-Based Reasoning (CBR) system. Each CBR system uses the four knowledge containers *vocabulary*, *similarity measures*, *transformational knowledge* and *cases*. The work presented in this paper focuses on filling the vocabulary and similarity measure containers as well as generating cases from LOD. Therefor, we demonstrate some possibilities of filling the aforementioned knowledge containers using simplified examples. These examples are preliminary, thus cannot be used in a real-world CBR system, because they are incomplete.

The linked data is provided for free and contains comprehensive knowledge where each term is assigned to at least one category and thus being represented in a specific context. The latest development in the field of LOD can be seen in the development of such complex knowledge repositories as for example given by the DBpedia ontology[1]. DBpedia contains many terms originating from the on-line encyclopedia Wikipedia. These terms are organized in an ontology and are being enriched with further information such as different labels in various languages. Currently, the DBpedia ontology contains 1,478,000 instances, i.e. Table 1.

Knowledge acquisition is still the bottleneck within the development of CBR systems. Our approach aims at automatization and for this purpose we propose a schema that extracts knowledge from LOD sources and provides and transforms it for CBR systems.

[1] http://dbpedia.org/ontology/

class	instances
Place	413,000
Person	312,000
Work	320,000
Species	146,000
Organisation	140,000
Building	33,000
...	...
Diseases	4,600

Table 1: Instances of DBpedia

The remaining sections of this paper are structured as follows: In section 2 we describe the application domain travel medicine, especially the docQuery project in which the experiments are carried out, followed by section 3 that describes how knowledge can be integrated in CBR systems. Section 4 presents the implementation details of our approach. The following section 5 presents experimental results of our approach before we sum up the paper and give an outlook on future work in the final section 7.

2 Application Domain: Travel Medicine

Travel medicine is an interdisciplinary specialty concerned with the prevention, management and research of health problems associated with travel, and covers all medical aspects a traveler has to take care of before, during and after a journey. For that reason it covers many medical areas and combines them with further information about the destination, the activities planned and additional conditions which also have to be considered when giving medical advice to a traveler. Travel medicine starts when a person moves from one place to another by any mode of transportation and stops after returning home without diseases or infections.

The realization of the travel medicine project docQuery is based on the SEASALT (Sharing Experience using an Agent-based System Architecture LayouT) architecture [Reichle *et al.*, 2009] that is especially suited for the acquisition, handling and provision of experiential knowledge as it is provided by communities of practice and represented within Web 2.0 platforms. It is based on the Collaborative Multi-Expert-Systems (CoMES), approach presented by Althoff et. al. [Althoff *et al.*, 2007], a continuation of combining established artificial intelligence techniques and the application of the product line concept (known from software engineering) creating knowledge lines.

According to the SEASALT architecture, docQuery consists of eight different CBR systems. Each CBR system within docQuery equips a software agent that represents a certain topic. Further the multi-agent-system is aggregating a composed result. However, within this paper we only focus on one CBR systems that contains information about diseases that might somebody can get infected with during a journey.

3 CBR System

CBR, the episodical knowledge, especially made experiences and successfully applied solutions for problems draw a huge potential for the development of an Experience Web [Plaza, 2008], because the experiences of many WWW users can be captured in case bases and provided in CBR systems. Because of the fact that CBR systems are based on experiences it is much easier to use them to handle experiences in comparison to systems that are based on technologies from the areas of information retrieval or semantic web. Experiences in CBR systems occur in all types of knowledge: mostly in cases, but also similarity measures, vocabulary and transformational knowledge can be derived. Huge amounts of raw data for the knowledge extraction is available on the web communities and the challenge is making these experiences available.

According to Richter [Richter, 1998], the knowledge of CBR systems can be provided in all four knowledge containers that include vocabulary, similarity measures, transformational (or adaptation) knowledge and cases. We focus on how the knowledge containers can be filled using the experiences provided in web communities. Therefore we have to extract the knowledge before it can be stored.

Further on, we deal with data sources that are mostly free text what usually requires a symbolic representation of keywords. That is the reason why we currently focus on the extraction of taxonomies that can be used for both, enhancing the vocabulary and assigning the similarity.

4 Implementation

Within this paper we present how knowledge about diseases can be extracted from LOD and integrated in a CBR system. Our CBR system has been developed using the open source tool myCBR [Stahl and Roth-Berghofer, 2008], thus all results produced by our application are compatible with the myCBR case representation and thus can be applied in myCBR-based applications. In case of the generation of taxonomies that contains knowledge about the similarity between objects represented in the taxonomy, we use the Knowledge Extraction Workbench (KEWo) to further refine the initially provided data from LOD. A more detailed description of KEWo can be found in [Bach *et al.*, 2010b] and [Sauer, 2010].

As the first step we retrieved data about diseases from the DBpedia ontology. The result of such a retrieval can be seen in figure 1. In this case we queried for all available diseases in the ontology and their German labels. The figure shows the shortened result containing the URI and the according label.

Furthermore, we queried the DBpedia ontology in order to extend our information retrieval by using the Resource Description Framework Schema (RDFS) to retrieve labels of diseases in different languages. Also, we retrieved the Simple Knowledge Organization System based (SKOS) information about the categories a disease belongs to, e.g. that a "bite" is a member of the category injuries. Analogous to the retrieval of German labels of different diseases we also retrieved the according German labels of these categories. The technique used for these initial retrieval steps was a set of SPARQL queries conducted via the open source Desktop-SPARQL-Query tool "Twinkle"[2]. Listing 4 shows an example for a query retrieving all diseases from the DBpedia ontology together with the categories the are part of filtered in that way, that only the German labels of the diseases and the categories are returned.

Name	label of the disease
Viral_disease	true or false
Infectious_disease	true or false
Bacterial_disease	true or false
Injury	true or false
Foodborne_illness	true or false
Inflammation	true or false
Parasitic_disease	true or false

Table 2: Disease categories (derived from DBPedia)

Listing 1: Example Query

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?disease ?categories ?dislabels ?catlabels WHERE
    {
?disease a <http://dbpedia.org/ontology/Disease> .
?disease rdfs:label ?dislabels .
?disease skos:subject ?categories .
FILTER (lang(?dislabels) = "de")
?categories rdfs:label ?catlabels .
FILTER (lang(?catlabels) = "de")
}
```

This still simple query, compared to the many sophisticated possibilities SPARQL offers to retrieve complex informations about objects and their relations, yielded 2004 category-disease-pairs with their corresponding German labels. Also some of the Items in the disease ontology are of a broader nature like the item Abuse we found enough specific Data to try to incorporate the retrieved Data into our CBR-system.

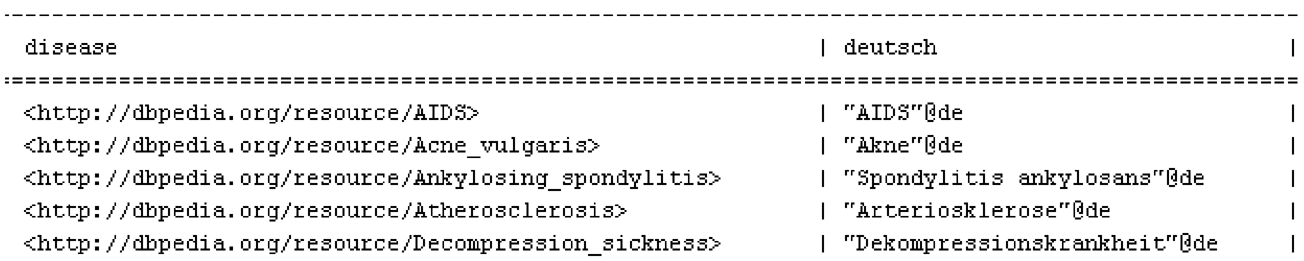
The simplest task was the extraction of data for the knowledge container vocabulary. For this task we simply derived the labels of the diseases from the retrieved data and thus formed a basic vocabulary of diseases for our CBR system. This vocabulary contains 2000 diseases with their German labels.

The next knowledge container we wanted to fill with Data from the LOD retrieval were the structured cases of our structured CBR system. For this purpose we chose six Categories from the English category-disease-pairs we assumed to be of interest for a case as attributes. The attributes that are describing a disease case are the following categories: *Viral_disease*, *Infectious_disease*, *Bacterial_disease*, *Injury*, *Foodborne_illness*, *Inflammation*, and *Parasitic_disease*.

Table 2 describes the structure of a disease case. The cases in our application scenario for this paper that are created based on LOD, of course, cannot serve as full case in a real-life application domain (like docQuery). However, LOD provides the potential data to create such cases.

To derive such cases from the retrieved data we implemented a java-based tool to extract the disease and category labels from the data and check if a disease label was part of a category-disease-pairs in which the category was, or was not one of the categories we chose to be the attributes

[2] http://www.ldodds.com/projects/twinkle/



```
---------------------------------------------------------------------------------------------
 disease                                                        | deutsch                    |
=============================================================================================
 <http://dbpedia.org/resource/AIDS>                             | "AIDS"@de                  |
 <http://dbpedia.org/resource/Acne_vulgaris>                    | "Akne"@de                  |
 <http://dbpedia.org/resource/Ankylosing_spondylitis>           | "Spondylitis ankylosans"@de |
 <http://dbpedia.org/resource/Atherosclerosis>                  | "Arteriosklerose"@de       |
 <http://dbpedia.org/resource/Decompression_sickness>           | "Dekompressionskrankheit"@de |
```

Figure 1: DBpedia retrieval result for disease (incomplete)

of our disease cases. After this extraction and comparison the resulting data was transformed into the csv-format to enable it to be imported as cases by the myCBR plugin we used in our Protégé development environment[3]. The created csv-file then was imported into Protégé resulting into the generation of 612 disease cases.

The third knowledge container we aimed at was the similarity measure. In our current project docQuery [Bach *et al.*, 2010a] we use taxonomies of items e.g. diseases, locations or medicaments to encode the similarity of these items into the distance of two items within the taxonomy [Bergmann, 2002]. So our goal was it to construct the according taxonomy from the retrieved LOD. This taxonomy, representing the similarity of diseases, was built using again a small java-based tool we implemented and one of our recently developed tools known as the Knowledge Extraction Workbench (KEWo). The process of taxonomy generation was thus the following: We extracted the diseases and the categories to which the diseases belong to as described above by deriving this information via a simple Java tool from the raw retrieval data. This process was again carried out on the German data set we retrieved. The resulting data was, due to the techniques used for taxonomy generation by the KEWo, further formated in a special way. The special formating listed the category twice followed by the disease. This resulted into a line describing a category-disease-pair looking as follows: *Mykose* (category) – *Mykose* (category) – *Kokzidioidomykose*(disease). This special formating is owed to the numerical approach the analysis methods of the KEWo employ. More details on the analysis with a brief discussion can be found in [Bach *et al.*, 2010b].

Our tool KEWo was created as an knowledge extraction tool operating on an on-line forum of travel medicine experts. Its goal is, in short, to derive medical knowledge from this forum employing various NLP and Information Extraction techniques on the content of the forum. For further details regarding the exact process model and principle of operation of the KEWo please see [Bach *et al.*, 2010b].

As a workaround to employ our KEWo for taxonomy generation from the retrieved LOD, the generated text,containing all category-disease-pairs was then inserted in the expert forum connected to the KEWo as a posting. Thus the KEWo was able to process the text containing the information from the LOD and so build a taxonomy of diseases.

We took the first 1000 category-disease-pairs and processed them in the described way, receiving a taxonomy describing 116 diseases. Figure 2 shows a snippet from the generated taxonomy. Such a taxonomy can be used in a CBR system as both, a source for vocabulary and the assignment of a similarity value between two concepts.

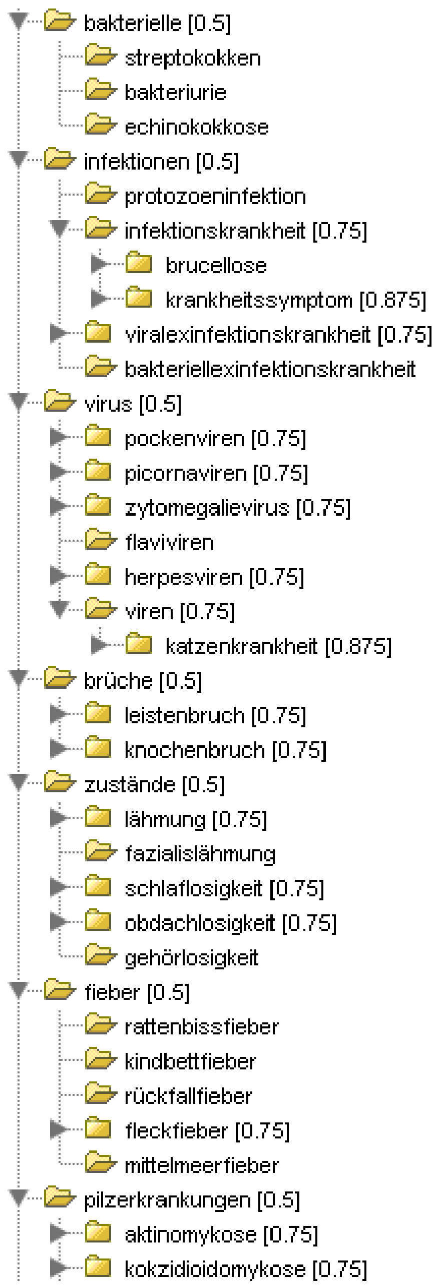


Figure 2: Snippet of the generated taxonomy

For the generation of the taxonomy we used the retrieved data about diseases and the according categories and con-

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

ducted a statistical analysis regarding the occurrences of the different diseases and categories. In a later development it is, of course, desirable to integrate ontological data available in LOD directly into a similarity measure. However, our approach at this point is of a more basic nature in order to point out the possibility of simple data integration from LOD into a similarity measure.

It is obvious to ask the question why there are not more diseases in the taxonomy, given the fact that we used 1000 category-disease-pairs. The comparatively low amount of diseases showing up in the generated taxonomy is partly caused by a certain kind of 'missuse' of our own Tool KEWo which is optimized for analyzing natural language and not such highly structured text as given to it in this experiment, please see Section 5 [ref Section 5] for further details on this 'conflict'. Nevertheless we were in fact able to produce a taxonomy of diseases from the retrieved LOD and in the upper ranges of its structure the linking of the disease items are of satisfying quality.

5 Experimental Results

The goal of our experiments was it to retrieve LOD about diseases and incorporate the retrieved data into 3 out of the 4 knowledge-containers of a CBR-system. These three knowledge containers were the vocabulary, the cases and the similarity measure of a CBR system used in our project docQuery which aims for on-line knowledge provision in the field of travel medicine.

Our SPARQL query to the DBpedia ontology returned 2004 unique English disease-labels. Further queries returned 2000 German disease-labels, 2000 English category-disease-pairs in which a category-disease-pair represents a disease-label and a category it is assigned to in SKOS categories, e.g. *AIDS : viral_diseases* and also 2004 German category-disease-pairs.

For the first knowledge container, vocabulary we were able to extract all of the either English or German disease labels resulting into disease vocabularies consisting of 2004 English respectively German terms for diseases. We conclude that, the small initial amount of time invested into the development of the Java tool for the extraction of terms from the raw retrieved LOD aside, the amount of time for building a vocabulary is drastically reduced, literally to minutes. Once a tool is provided it is a matter of minutes to adapt the tool, run a new query, and extract the resulting terms from the retrieved LOD for any new vocabulary.

For the second knowledge-container we were able to extract 612 cases out of the English data retrieved. Each case consisted of a unique disease-name and six categories of diseases it either belonged to or not, please see Section 4 for the exact structure of the extracted cases. In the process of generating these cases we again used a small Java tool we developed to pre-process the raw retrieved LOD. During the use of this tool we discovered that it is very easy to further adapt this Tool to generate any desired kind of csv-formated files representing cases and thus upon later importing these files via myCBR, easily generate any kind of structural cases from LOD, extracted according to the desired structure of the cases. Similar to the possibility to rapidly built new vocabularies this approach of building structural cases from LOD is also a very rapid approach. Given that a basic tool for the exact extraction from the retrieved LOD is already in place the generation of new cases consists of just a new query to an LOD repository, a slight adjustment of the extraction tool and a new import of the generated csv-file via myCBR. This process can, if the cases are not overly complex, be performed in minutes and yield hundreds of cases only limited in their numbers by the amount of available LOD. Furthermore, using LOD also reduces the costs of knowledge acquisition in terms of time, logistics, effort, money, etc.

For the third knowledge container similarity measure we were able to build a taxonomy, carrying informations about its items similarity in the form of their distance within the taxonomy. The generated taxonomy contains 116 disease items an was built upon the input of 1000 category-disease-pairs. Top level structures in the generated taxonomy show a satisfying quality nevertheless in deeper levels of the taxonomy the quality of disease item links e.g. making 'sense' as father-child pairs of nodes deteriorates quickly. Also the amount of disease-items in the taxonomy is not satisfying. We assume this to be a result of a workaround we employed to generate the taxonomy. We used our recently developed tool KEWo which is specialized in processing natural language and information extraction out of unstructured text, generally discussion posts from an on-line forum. Feeding this tool with the highly structured text we derived from the LOD retrieval surely caused some conflicts with the original goal of the KEWo to extract information from unstructured text. We took this into account as far as we were aware of this 'missuse' and accepted the poorer results but thus getting at least a proof of concept taxonomy that shows that it is indeed possible to derive knowledge for the similarity measures from LOD. Also the process of generating a taxonomy from LOD is a bit more complex than the two other processes described in this paper it still is by far faster than a manual approach of building a taxonomy could not ever be. Despite the lack of quality regarding the linking of items in the deeper levels of the taxonomy, as a first proof of concept run, the generation of a similarity measure in form of a taxonomy can be seen as equally accelerated as the generation of the vocabulary and the cases are by the use of LOD.

6 Discussion

During our work with LOD we found it somewhat hard to identify relevant LOD repositories and the specific names of the attributes respectively predicates of the items in these repositories. Also we noticed there are ongoing efforts to improve the searchability of LOD this still is an issue we found somewhat hampering the use of LOD despite its ease and resource fullness if a correct repository and all of its facets are identified.

Nevertheless the rich and fast growing sources of LOD surely legitimate further research into their use as a source for knowledge to be used in the knowledge-containers of CBR. This is especially true if one takes into account their highly structured nature and the fact that LOD is available for free.

As far as we can comment on our first experimental results the use of LOD, once a working development environment consisting of querying tools, tools for further information refinement out of raw LOD and tools for incorporating this refined data into CBR knowledge containers like myCBR is present, filling the knowledge containers vocabulary, structured cases and similarity measure is an easy and fast process.

The fourth knowledge container of CBR that is containing adaptation knowledge, was not part of our experiments and it is still questionable if there is a method to generate and/or extract knowledge for this container from LOD. We assume that there might be a chance to use the taxonomy we generated as a similarity measure to derive some adaptation knowledge from it, like the model-based adaptation approach presented in [Hanft *et al.*, 2010]. Furthermore, one can imagine that the structuring of much of the LOD as ontologies might yield some further sources for the extraction of adaptation knowledge from these ontologies.

7 Summary and Outlook

In this paper we proposed the idea to use LOD as a source to fill three out of the four knowledge containers a CBR system is based on. The containers we addressed were vocabulary, strucutred cases and the similarity measure. We have described our experimental setup regarding the methods used to acquire LOD in the field of diseases, the process of further refining the acquired data involving the development of some simple java-based tools and the use of more complex tools like our own developed KEWo and the external tools myCBR and Protégé to transform the LOD into fitting the structural requirements of the aforementioned knowledge containers [Bergmann, 2002].

We were able to produce good quantity and quality results for the knowledge containers vocabulary and structural cases as well as the data extracted for assigning similarity measures in form of a taxonomy, build upon the retrieved LOD, using our own tool KEWo is also acceptable. Further on, we were able to proof the fact that the approach of a semi-automatic acquisition of knowledge for the three mentioned knowledge containers from LOD resulted into a very rapid built up of these containers compared to manual or even other existing semi-automatic approaches we recently developed (see [Bach *et al.*, 2010b] for details).

Future goals based upon the work presented in this paper consist of the further refinement and automatization of the extraction process from LOD and the development of more efficient and even more easily customizable Tools for the Refinement of retrieved raw LOD. Also the question of how to extract adaptation knowledge should also be resolved to profit on the benefits of rapid knowledge acquisition for all four of the knowledge containers.

A distant future goal is given by the idea of incorporating all process-steps and tools presented in this paper into a single tool that handles the whole process from finding appropriate LOD resources, querying them, refining the retrieved data and formating it to be further processed by our own tools, like the KEWo, or external tools like myCBR. All of these functionalities should be embedded in a convenient GUI which allows the user to specify the knowledge container for which LOD should be retrieved for and the desired characteristics of this LOD, being most customizable in regard to the many facets each knowledge-containers contents can have.

References

[Althoff *et al.*, 2007] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Regis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative Multi-Expert-Systems – Realizing Knowlegde-Product-Lines with Case Factories and Distributed Learning Systems. In Joachim Baumeister and Dietmar Seipel, editors, *Workshop Proceedings on the 3rd Workshop on Knowledge Engineering and Software Engineering (KESE 2007)*, Osnabrück, September 2007.

[Bach *et al.*, 2010a] Kerstin Bach, Meike Reichle, and Klaus-Dieter Althoff. Case-Based Reasoning in a Travel Medicine Application. In Isabelle Bichindaritz and Lakhmi Jain, editors, *Computational Intelligence in Medicine*, Advanced Information and Knowledge Processing, page to appear. Springer, 2010.

[Bach *et al.*, 2010b] Kerstin Bach, Christian Severin Sauer, and Klaus-Dieter Althoff. Deriving Case Base Vocabulary from Web Community Data. In Cindy Marling, editor, *ICCBR-2010 Workshop Proceedings: Workshop on Reasonng From Experiences On The Web*, page to appear, 2010.

[Bergmann, 2002] Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, volume 2432 of *Lecture Notes in Computer Science*. Springer, 2002.

[Bizer *et al.*, 2008] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (LDOW2008). In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1265–1266, New York, NY, USA, 2008. ACM.

[Hanft *et al.*, 2010] Alexandre Hanft, Régis Newo, Kerstin Bach, Norman Ihle, and Klaus-Dieter Althoff. CookIIS - A successful Recipe Advisor and Menu Advisor. In Stefania Montani and Lakhmi Jain, editors, *Successful Case-based Reasoning applications*. Springer, 2010.

[Plaza, 2008] Enric Plaza. Semantics and Experience in the Future Web. In Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft, editors, *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings*, volume 5239 of *Lecture Notes in Computer Science*, pages 44–58. Springer, 2008.

[Reichle *et al.*, 2009] Meike Reichle, Kerstin Bach, and Klaus-Dieter Althoff. The SEASALT Architecture and its Realization within the docQuery Project. In Bärbel Mertsching, editor, *Proceedings of the 32nd Annual Conference on Artificial Intelligence (KI-2009)*, Lecture Notes in Informatics, pages 556–563, September 2009.

[Richter, 1998] Michael M. Richter. Introduction. In Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, and Stefan Wess, editors, *Case-Based Reasoning Technology – From Foundations to Applications*, LNAI 1400. Springer-Verlag, Berlin, 1998.

[Sauer, 2010] Christian Severin Sauer. Analyse von Webcommunities und Extraktion von Wissen aus Communitydaten für Case-Based Reasoning Systeme. Master's thesis, Institute of Computer Science, University of Hildesheim, 2010.

[Stahl and Roth-Berghofer, 2008] Armin Stahl and Thomas R. Roth-Berghofer. Rapid Prototyping of CBR Applications with the Open Source Tool myCBR. In *ECCBR'08: Proceedings of the 9th European conference on Advances in Case-Based Reasoning*, pages 615–629, Berlin, Heidelberg, 2008. Springer-Verlag.