Expert Systems

Article

DOI: 10.1111/exsy.12034

Extracting knowledge from web communities and linked data for case-based reasoning systems

Christian Severin Sauer and Thomas Roth-Berghofer

University of West London, School of Computing and Technology, St Mary's Road, London, W5 5RF, UK
E-mail: christian.sauer@uwl.ac.uk

Abstract: *Web communities and the Web 2.0 provide a huge amount of experiences and there has been a growing availability of Linked (Open) Data. Making experiences and data available as knowledge to be used in case-based reasoning (CBR) systems is a current research effort. The process of extracting such knowledge from the diverse data types used in web communities, to transform data obtained from Linked Data sources, and then formalising it for CBR, is not an easy task. In this paper, we present a prototype, the Knowledge Extraction Workbench (KEWo), which supports the knowledge engineer in this task. We integrated the KEWo into the open-source case-based reasoning tool myCBR Workbench. We provide details on the abilities of the KEWo to extract vocabularies from Linked Data sources and generate taxonomies from Linked Data as well as from web community data in the form of semi-structured texts.*

Keywords: information extraction, case-based reasoning, experience web, linked data

1. Introduction

The kind of data with the fastest growth in volume on the Web is user-generated content, which is mostly in the form of semi-structured texts. This user-generated content often contains artefacts of user experiences, expressed explicitly or implicitly (Plaza and Baccigalupo, 2009). Additionally, the recent development in the field of Linked (Open) Data (LOD) further added structure and value to the existing vast amount of information that is available (Bizer *et al.*, 2009). Accessing this information is still a task not easily accomplished by a machine, mainly because most of the information is unsystematic and thereby hard to retrieve efficiently and thus not easily available to be reused (Bergmann, 2002). By becoming more user-friendly, more and more users participate in one of the many forms of web communities Web 2.0 offers (Boyd and Ellison, 2007). This development further increases the amount of data, again very often as semi-structured text, such as the 140 character messages in the popular web service Twitter. The availability of such an amount of information suggests also exploiting LOD for the semi-automatic generation of knowledge.

Following the idea of the Experience Web (Smyth *et al.*, 2009), one has also to ask how experience-based technologies such as case-based reasoning (CBR) might be able to benefit from the experience contained in semi-structured texts generated by the users of web communities and social networks. According to Richter (1998), the knowledge of CBR systems comprises four knowledge containers: vocabulary, similarity measures, transformational (or adaptation) knowledge and cases. An approach for extracting a controlled vocabulary and similarity knowledge in the form of taxonomies from semi-structured texts provided by a web community is described by Bach *et al.* (2010). The respective tool is called 'Knowledge Extraction Workbench' (*KEWo*).

In this paper, we show how integrating *KEWo* into the open-source CBR tool and software development kit (SDK) *myCBR 3* (Stahl and Roth-Berghofer, 2008; Roth-Berghofer *et al.*, 2010) supports the knowledge modelling of vocabularies and similarity measures. We further demonstrate how the similarity measure container can be provided with a taxonomy built from LOD as well as from web community data. Our approach aims here at direct automatisation. Thus, we propose a schema that extracts knowledge from LOD/web community sources and feeds it directly to the *myCBR Workbench* and/or any CBR system built with *myCBR 3* SDK.

myCBR focuses on the similarity-based retrieval step of the CBR cycle (Aamodt and Plaza, 1994). A popular class of such retrieval-only systems comprises case-based product recommender systems (Bridge *et al.*, 2006). *myCBR* provides user interfaces for modelling and use of highly sophisticated, knowledge-intensive similarity measures (Stahl, 2003). Such domain-specific similarity measures can improve the retrieval quality substantially. However, they do increase the development effort significantly.

In contrast to earlier versions of *myCBR*, which were plug-ins for the open source ontology editor Protégé[1] (Gennari *et al.*, 2003), *myCBR 3* is a complete reimplementation and consists of a SDK and a new and OSGi-based, eclipse-like graphical user interface, the *myCBR Workbench*. The *myCBR Workbench* still focuses on ease-of-use regarding the creation of the case model, modelling of similarity measures and testing the similarity-based retrieval by offering an easy-to-use graphical user interface. To reduce the effort of the preceding step of defining an appropriate case representation, it includes tools for generating the case representation automatically from existing raw data.

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

The capabilities of *KEWo* to extract elements of a controlled vocabulary and to build similarity measures in the form of taxonomies of terms plus the ability to generate at least limited amounts of adaptation knowledge from the taxonomies are a useful addition to *myCBR*. This belief is further strengthened by the (semi-)automatic extraction of such vocabularies and their respective similarity measures from semi-structured and, to a certain extent, even unstructured texts that *KEWo* supports (Bach *et al*., 2010).

A desirable next step for the extraction of data from the web, to be used in CBR systems, is to enable the access to Linked Data, especially to LOD. LOD is provided without charge and contains comprehensive ontologies based on Semantic Web standards. The complex knowledge repository DBpedia is a prominent example of an LOD repository. It is generated from the online encyclopaedia Wikipedia. The terms are organised in an ontology and are being enriched with further information. Currently, the DBpedia ontology contains 1.83 million instances.[2]

Accessing LOD has the potential to further ease the development of Web CBR systems. Within this paper, we demonstrate the capability of the *KEWo* to extract similarity measures from LOD (Sauer *et al*., 2010). The ability to adapt *KEWo* with a relatively small amount of effort to new data types from which it extracts knowledge for CBR systems further adds to the idea of integrating *KEWo* into *myCBR Workbench* to enable knowledge and system engineers to benefit from the knowledge extraction capabilities of *KEWo*.

The rest of the paper is structured as follows: In Section 2, we give an overview of related work in the field of knowledge extraction from web data for CBR systems. The process model for knowledge extraction used by *KEWo* is described in Section 3, and *KEWo* itself and its functionalities are detailed in Section 4. After introducing *KEWo* and its approaches to knowledge extraction, we take a brief look at the performance of *KEWo* in Section 5. In Section 7, we give a detailed view of the challenges we met during the integration of *KEWo* into *myCBR Workbench*. The last section summarises our approach and gives an outlook on how to further extend the abilities of *KEWo*.

2. Related work

The knowledge of CBR systems comprises the four knowledge containers: vocabulary, similarity measures, transformational (or adaptation) knowledge and the case base (Richter, 1998). To extract data from the Web 2.0, respectively, from a web community, and to use in a CBR system, the extracted data needs to be formalised properly to meet the formal needs of the chosen knowledge container for which it is extracted.

The wide variety of web communities can be classified into certain archetypes and prevalent forms of data types used in these communities (Boyd and Ellison, 2007; Sauer, 2010). One, then, faces a multitude of possible combinations. Combinations consist of the possible forms the source-data types used in a web community, and the formal structure of a target knowledge container for which to extract knowledge from the community data is designed.

However, with respect to the fast growing amount of very diverse data, containing a rich amount of experiences from the users of web communities, the task of extracting knowledge from these data to use in CBR systems would seem to be worth the effort. Given the fact that the underlying methodology of CBR traditionally works upon previously recorded experiences, the development of Web CBR was the next logical step (see, e.g. Recio-García *et al*. (2010)). An issue yet to be solved is the already-mentioned problem of numerous possible combinations of source data and targeted knowledge container(s) for which knowledge is to be extracted.

In contrast to *myCBR*, jCOLIBRI[3] is a framework for developing CBR systems in Java and for modelling knowledge for such systems (Bello-Tomás *et al*., 2004; Recio-García *et al*., 2005). As Recio-García *et al*. point out, there are a variety of opportunities if web-based data sources can be integrated into a development framework for CBR systems such as jCOLIBRI (Recio-García *et al*., 2010). Their approach is similar to the approach described in this paper. However, *myCBR Workbench* follows a tool approach with a rich graphical user interface, providing ease-of-use by itself. Considering that the *KEWo* prototype itself also offers a variety of easy to use GUI features makes them a perfect match.

The *KEWo* focuses on the extraction of knowledge from semi-structured texts plus, from LOD, for the knowledge containers vocabulary and similarity measures. During the development of this approach it was found that, because of the formal needs of the knowledge containers, it was only possible to extract knowledge from web data in a highly customised process. There are plenty of similar approaches for certain combinations of web community data and knowledge containers to extract knowledge from that web community data (e.g. (Ihle *et al*., 2009; Milne *et al*., 2009; Plaza and Baccigalupo, 2009; Smyth *et al*., 2009)). All of these approaches prove the benefits of extracting knowledge from web community data, but they also have in common the need for highly tailored processes to fit the formal needs of the knowledge representation in the knowledge containers.

The integration of *KEWo* into such a tool as *myCBR* provides a starting point for using standard techniques for knowledge extraction from community data and for just trying out initial tests. *KEWo* already enables the extraction from varying forms of source data types for two of the four knowledge containers, thus sparing developers of CBR systems the work of designing customised ways to extract, formalise and integrate knowledge from web sources into their CBR systems.

3. Knowledge extraction process

The knowledge extraction approach utilised by *KEWo* is to extract relevant terms out of a previously specified domain from text data retrieved from web communities. Upon the

[2]http://dbpedia.org/About [Last access: 22 May 2012]

[3]http://gaia.fdi.ucm.es/projects/jcolibri/

extracted terms, a taxonomy is built by assigning the terms in a hierarchy according to an analysis of the term frequency in a given text.

As already mentioned, the *KEWo* provides a limited amount of adaption knowledge by constructing taxonomies of symbols in a given domain. The approach to exploit the taxonomies as a source of adaption knowledge can be described as follows. The nodes in the taxonomy are assigned with similarity values according to their distance in the taxonomy. This approach allows *KEWo* to derive also a limited amount of adaptation knowledge from the structure of the taxonomy by offering the possibility to choose between different siblings, sharing the same level and parent node in the taxonomy (Bach *et al*., 2010).

A possible example for such an adaption in the domain of travel medicine is the following: A case describing a headache contains an attribute 'medicamentation', meaning the medication best used and the best choice offered by the case base is the instance 'Aspirin' for the attribute 'medicamentation'. The user now states to the system that Aspirin is not available in his context. The system then could access the taxonomy of medicaments and adapt the case, namely, the value of the attribute 'medicamentation' with the instance 'Ibuprofen', which is another sibling of the parent node of 'Aspirin', which might be 'Painreliefers' as an abstraction of 'Aspirin' and 'Ibuprofen'. Thus, an adaption is realised using substitution knowledge from the taxonomies.

The *KEWo* knowledge extraction approach (Bach *et al*., 2010) follows a process model based on the knowledge discovery in databases process (Fayyad *et al*., 1996). This process model can be seen as a valid approach in all possible combinations of source data from which to extract knowledge and target knowledge containers for which knowledge is extracted.

The process model described in Figure 1 shows a knowledge extraction for a CBR system with the aim to extract knowledge for two of the CBR systems knowledge containers, namely, the vocabulary and the similarity measure. The extracted knowledge is then used and evaluated in the CBR system itself for which it is extracted.

In addition, extraction of knowledge in the form of symbols for the vocabulary and the construction of taxonomies itself is already implemented as a fully automatic process the last step of the process model, the evaluation still has to be done manually.

Referring to the process model, we will now inspect each step of the process closer and give a brief insight into how the step in question is implemented within *KEWo*.

1. *Domain Detection*. This first step describes the identification of the domain properties and results in the assignment of what kind of information can be extracted and in which knowledge container it should be integrated. For *KEWo*, this can be almost any domain. Targeted knowledge containers are the vocabulary, the similarity measures and to a certain extent the adaptation knowledge.
2. *Web Community Selection*. In this step, a web community is identified from which data should be used for extraction. With regard to *KEWo* currently text-based communities, for example, forums are preferred.
3. *Linked Data Repository*. In this step, a suitable repository or repositories supplying linked data, preferable open data, are identified. The data the repositories provide are checked for its format and suitability for the modelling of the domain at hand with regard to transformation effort from linked data format to the knowledge formalisation of the knowledge container for which the knowledge is extracted.
4. *Content Mining*. This describes the process of acquiring the raw data. This can be accomplished by, for example, web crawlers. A more convenient way is described by Feng *et al*. (2006) with the approach of intelligent web forums.
5. *Data retrieval*. This describes the process of designing a suitable query and acquiring the data from a chosen repository or range of repositories by querying them. This can be accomplished for example by accessing available SPARQL endpoints.
6. *Processing Raw Data*. Noise, stop words and duplicates are removed. These sub steps are already implemented in *KEWo* and are executed automatically if *KEWo* accesses raw text data from a database underlying a web forum
7. *Processed Data*. The automatically refined text data are now ready for analysis by *KEWo*.

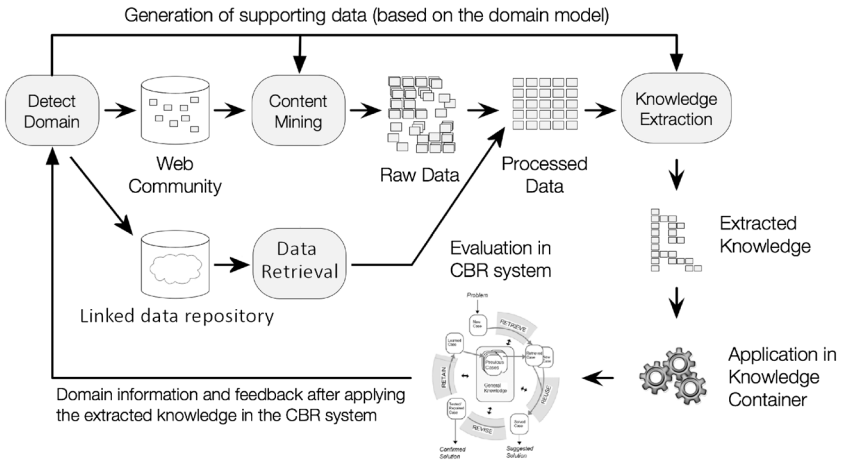

Figure 1: *Knowledge extraction process model for case-based reasoning systems (web communities and linked data).*

8. *Knowledge Extraction*. In this step, *KEWo* extracts relevant terms and builds a taxonomy of these terms as already described. Additional input data for the extraction process, for example, gazetteers, are automatically updated by *KEWo* during the extraction process.
9. *Extracted Knowledge*. Being closely woven in with the preceding step, in this step, the taxonomy generated by *KEWo* is saved back to a *myCBR* and is ready to be used.
10. *Application in Knowledge Container*. The obtained taxonomy can be used in the development and testing of a CBR system using the *myCBR* tool.
11. *Evaluation*. Evaluating the generated taxonomy may deliver hints on how to optimise the auxiliary data and preferences for the A Nearly-New Information Extraction system (ANNIE) application (Cunningham *et al.*, 2002) (see next section) used in the extraction process. The evaluation may result in performance gains and/or gains in quality of the extracted taxonomies. These steps still have to be done manually within *KEWo*.

The *KEWo* offers the user a high degree of interactivity, ranging in modes of operation from fully automatic to manual. The degree of useful automatic analysis strongly depends on the quality of the data to be analysed and the quality of the auxiliary data, for example, the gazetteers and rule sets, used by the ANNIE application within *KEWo*.

The degree of interactivity can be increased from fully automatic by enabling dialogues with the knowledge engineer. Such an increase of interactivity can consist of questioning her if either a found symbol is valid and/or if a newly found symbol, not yet part of the used ANNIE gazetteer, should be integrated in the gazetteer in use. Furthermore, the knowledge engineer can at any time during the extraction process interact with the taxonomy currently being built to make adjustments if so desired.

The *KEWo* is a Java-based middleware for the extraction of knowledge for CBR systems. Currently, *KEWo* relies on *myCBR* for the data type of the taxonomies and the calculation of distance-based similarity measures between the symbols of the taxonomies. The main purpose of *KEWo* is to extract symbols from a given domain and construct taxonomies usable in *myCBR* from the extracted symbols. The process underlying the extraction of symbols is mainly provided by the text engineering tool set GATE (Cunningham *et al.*, 2002), specifically by the ANNIE application, which has been customised for *KEWo* by using specific gazetteers and rule sets that identify terms from a given domain. Thus, *KEWo* relies on a customised ANNIE application to extract symbols from unstructured texts and either build completely new or expand existing taxonomies of symbols to be used in *myCBR*.

4. Knowledge extraction workbench

The *KEWo* in its first version offers the ability to either start the generation of a taxonomy of symbols from scratch or import one from a *myCBR* project to work on. After creating or importing a taxonomy, *KEWo* offers a variety of functions to improve the taxonomy. Besides automatic extraction and addition of further symbols to the taxonomy from analysed text, *KEWo* offers the abilities to recalculate the similarity measures of the symbols with almost any given formula and the possibility to edit the taxonomy symbols manually to refine the taxonomy (Bach *et al.*, 2010).

It provides a minimalistic browser with which the user can navigate a web forum targeted for extraction. The user can choose between two different analysis methods that define the strategy in which the symbols are added to the taxonomy. Furthermore, the user can decide if she either wants to run a fully automatic analysis or an interactive one, in which she can decide if extracted symbols are added to the taxonomy and/or added to support data used by the extraction techniques, for example, gazetteers. As a third option, the user can decide to extract from a given thread posting-by-posting or from the whole thread as one text. See Figures 2 and 3 for the main GUI elements of the *KEWo*.

In addition, the user can define and apply a new formula for recalculating the similarity values of the symbols in the taxonomy as well as load, manually edit and save the taxonomy. The taxonomy can be saved in *myCBR* format. For a complete description of the functionalities of *KEWo*, we refer to Sauer (2010).

The approach of offering two ways for the processing of texts that was mentioned previously has some impact on the resulting taxonomy. Using the approach to analyse each posting of a thread as a singular text *KEWo* tends to generate deeper taxonomies, whereas using the approach to analyse a thread as a whole text generates a shallow taxonomy. The described effects on the depth of the taxonomies generated originate from the numerical approach used by *KEWo* to build the taxonomies. Upon the extracted terms, a taxonomy is built by assigning the terms in a hierarchy according to an analysis of the term frequency in a given text, assuming that two similar terms appear together more often (Church and Hanks, 1990).

With regard to the extraction from LOD, we can query the DBpedia ontology to extend our information retrieval by using the Resource Description Framework Schema (RDFS) (W3C, 2004) to retrieve labels of concepts in different languages. In RDF, meaning is expressed by facts encoded in sets of triples (Bergmann and Schaaf, 2003). We also retrieve the Simple Knowledge Organisation System based (SKOS) information about the categories a concept belongs to. SKOS is a family of formal languages. It is designed for representing a structured controlled vocabulary.[4] SKOS is built upon RDF. Its main objective is to enable easy publication of controlled structured vocabularies for the Semantic Web.

The technique used for these initial retrieval steps are a set of SPARQL queries conducted via the open-source Desktop-SPARQL-Query tool 'Twinkle' 2. The simplest task was the extraction of data for the knowledge container vocabulary. For this task, we simply derived the labels of the concepts from the retrieved data. The building of a taxonomy was then following the same approach as described earlier, just requiring a few steps of pre-processing the LOD retrieval result. Thus, we came up with

[4]http://www.w3.org/TR/2009/REC-skos-reference-20090818/

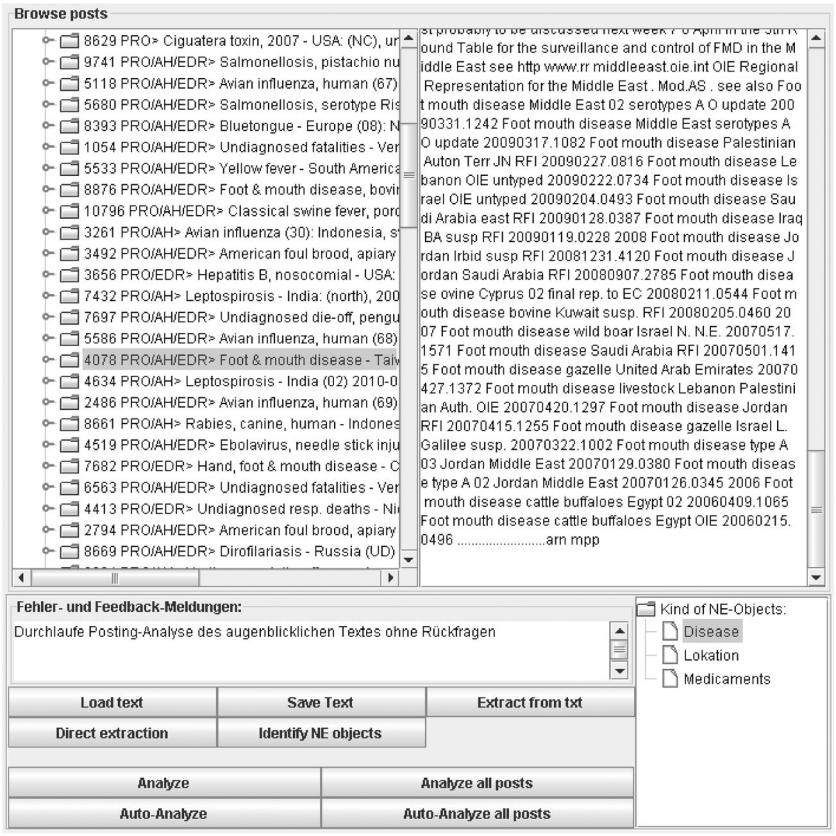

Figure 2: *Screenshot of the knowledge extraction workbench (Sauer and Roth-Berghofer, 2011).*

a process of taxonomy generation as follows: extract the concepts and the categories to which the concepts belong to as described previously. The resulting data are, because of the techniques used for taxonomy generation by the *KEWo*, further formatted in a special way. The special formatting lists the category twice followed by the concept. This results in a chain describing a category-concept-pair, for example, Mammal (category) – Mammal (category) – Dog (concept). This formatting is due to the numerical approach the analysis methods of the *KEWo* employed. The basic requirements, regarding a tool for a first step in the direction of standardised knowledge extraction from the web for specific knowledge containers of a CBR system, are met by *KEWo*. In the next section, we show that the performance requirements are also met.

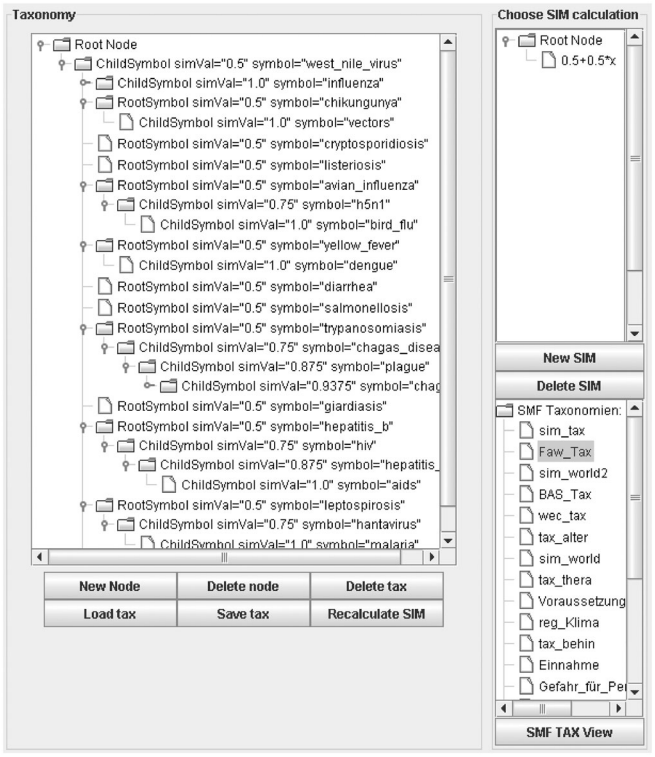

Figure 3: *Screenshot of the forum browser of the knowledge extraction workbench (Sauer and Roth-Berghofer, 2011).*

5. Experiment and results 1: information extraction from text

To show the effectiveness of the knowledge extraction process, experiments were performed on text data provided by a forum of experts in the field of travel medicine (in German). *KEWo* was used to extract taxonomies of terms out of the three domains: diseases, medicaments and geographical locations from the 6500 postings in that forum.

Both of the approaches, to analyse posting-by-posting or whole threads as a single text, were evaluated and delivered the already-mentioned different kinds of taxonomies, regarding the depth of the taxonomies (Section 4). Figure 4 shows the depth profiles of two of the generated taxonomies with the result of the thread analysis in the upper half and the post-by-post analysis in the lower half.

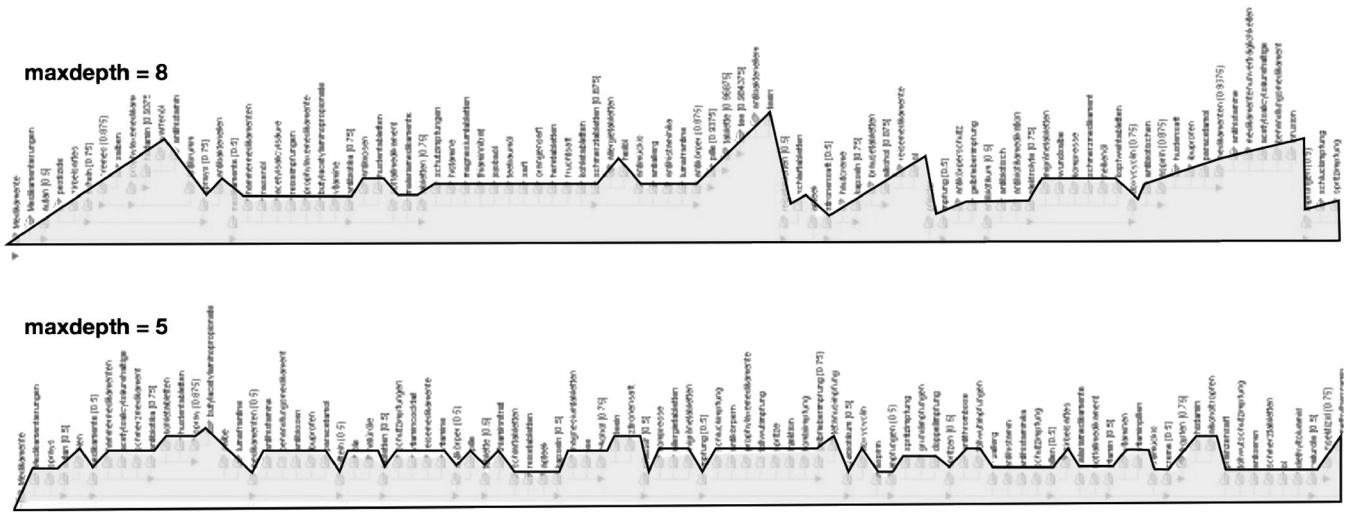

Figure 4: *Taxonomy depth comparison curves (overview) (adapted from (Sauer and Roth-Berghofer, 2011)).*

It is important to note the difference of the term chain length because of the two different approaches causing the differences in the depths of the generated taxonomies.

All generated taxonomies were of acceptable quality with regard to making sense in the hierarchy of extracted terms. Figure 5 shows two snippets of a taxonomy of German terms from the domain of diseases. The taxonomy was built fully automatic by *KEWo*. While processing text data, the F1-score of the term extraction gained by *KEWo* ranged between 68.7 and several numbers in the 80s range, depending on the domain and the degree of auxiliary data provided for the extraction (Sauer, 2010).

For each domain, 'diseases', 'locations' and 'medicaments', a gazetteer and a Jape transducer were designed in the ANNIE application. Each Jape transducer consisting of a set of Jape rules to identify word composites. The new ANNIE application was then used by *KEWo* for term extraction from the postings.

The gazetteers for the three domains contained a randomly chosen set of terms from the given domain. The gazetteer for diseases contained 717 terms, for locations there were 331 terms and 32 terms for medicaments were present at the start of the experiment (Sauer, 2010).

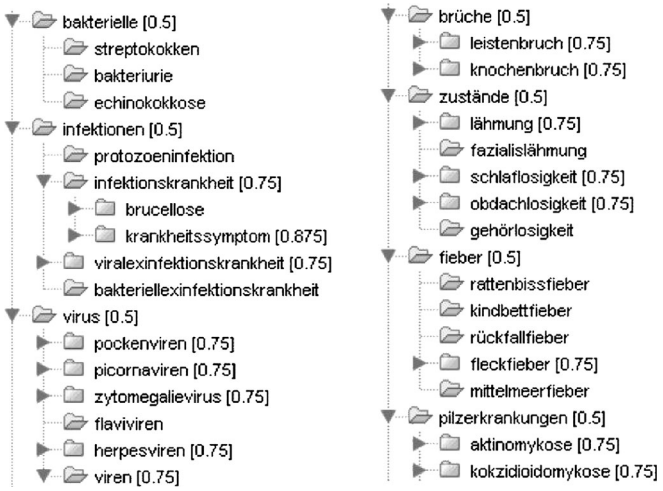

Figure 5: *Snippets of a taxonomy generated by KEWo (Sauer and Roth-Berghofer, 2011).*

To test *KEWo*'s and its underlying ANNIE application's abilities to extract terms from forum postings, a set of experiments was performed. During these experiments, *KEWo* (and its underlying ANNIE application) was able to automatically expand the gazetteers by identifying word composites. The additions were the following: 74 terms for diseases, 47 terms for locations and 123 terms for medicaments. The high number of composites found for the domain of medicaments was partly due to the deliberately low initial population of the gazetteer for this domain to explicitly test *KEWo*'s ability to work with sparse gazetteers and rely on the rule-based Jape transducers the ANNIE application provided (Sauer, 2010).

In another experiment, the first 100 postings of the forum from which the knowledge was extracted from were analysed manually. All occurrences of terms from the domain of medicaments were identified and counted manually, resulting into 38 manually identified terms from the domain medicaments. Of these 38 terms, *KEWo* was able to automatically identify 22 terms correctly, and 4 terms were incorrectly identified as medicaments (Sauer, 2010).

The maximum depth the taxonomies reached was 8. Figure 6 shows the total numbers of symbols *KEWo* identified and integrated into the taxonomies. The numbers are given from left to right for the domains: diseases, locations and medicaments. Please note that the category 'fair' refers to symbols that were correct from their syntactic structure but did not have a correct semantic meaning in the domain for which they were extracted. Most of these symbols were synonyms.

The ability of *KEWo* to also process retrieved data sets from LOD sources had no negative effect on the quality of the generated taxonomies (Sauer *et al.*, 2010). Thus, *KEWo* also proved to enable CBR developers to use LOD as a source of knowledge for their CBR systems.

6. *KEWo* experiment and results 2: retrieval of linked open data

For the generation of the taxonomy, we used retrieved LOD of diseases and the relevant categories they belong to. The *KEWo* was able to process the text containing the

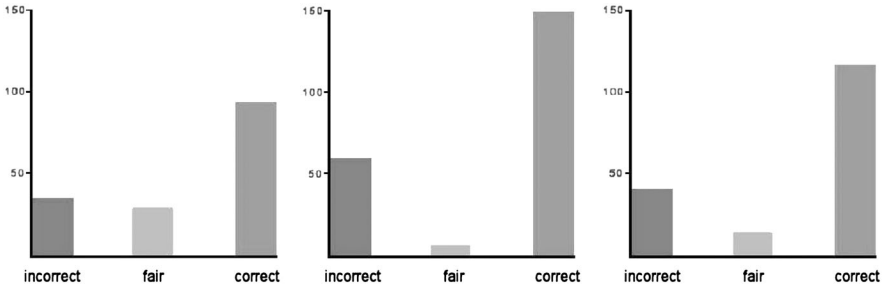

Figure 6: *Number of symbols found and integrated into taxonomies by KEWo for the domains 'diseases', 'locations' and 'medicaments' (Sauer and Roth-Berghofer, 2011).*

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 5 shows a snippet from the generated taxonomy generated purely from LOD.

Our SPARQL query to the DBpedia ontology returned 2004 unique English disease labels. Further queries returned 2000 German disease labels, 2000 English category-concept-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.

For the third knowledge container similarity measure, it was possible to build a taxonomy formalising the similarity of two diseases by their distance within the taxonomy. The generated taxonomy contained 116 disease terms and was built upon 1000 category-concept pairs. The generated taxonomy shows a satisfying quality. Nevertheless, in deeper levels of the taxonomy, the quality of disease item links, for example making 'sense' as parent–child pairs of nodes, deteriorates quickly. Despite the lack of quality regarding the linking of disease terms in the deeper levels of the taxonomy, the generation of a similarity measure in form of a taxonomy can be seen as equally accelerated as the generation of the vocabulary by the use of LOD.

One question to ask is, given the fact that we used 1000 category-concept-pairs, why are there not more diseases in the taxonomy than the 116 present? The comparatively low amount of concepts, here given by diseases, showing up in the generated taxonomy is partly caused by a certain kind of 'misuse' of our own Tool *KEWo*. This 'misuse' occurs as the *KEWo* is optimised for analysing natural language and not such highly structured text as was present during this experiment.

7. Challenges of integrating *KEWo* with *myCBR workbench*

As we have shown, the *KEWo* is a reliable and useful tool for extracting experience from web community knowledge to be used in the development of CBR systems. Now, we want to focus on some challenges we faced when integrating *KEWo* into *myCBR Workbench* and some challenges we foresee for the integration of further features into the *KEWo* being embedded in *myCBR Workbench* now.

Making the data types used by *KEWo* compatible with *myCBR* data types is an already solved problem as *KEWo* was specifically developed for *myCBR 2*. A point more challenging was the extension of *KEWo*'s ability to extract knowledge for the two containers not yet covered by *KEWo*: adaptation knowledge and case base. For the adaptation knowledge, we added the capability to browse in the taxonomy to derive adaptation knowledge from the structure of the taxonomy. For the extraction of cases, we had to implement an extension to *KEWo* aiming at structural cases, because of the ease such cases may be extracted with techniques derived from the well-researched field of template completion.

Currently, we provide a taxonomy of terms annotated with a similarity value for each term derived from its position in the taxonomy. Bringing together the variety of data types given for web data and the strict formalisms of adaptation knowledge and cases, we focussed at first on certain types ranging from at least semi-structured data sources such as annotated documents for the extraction of cases to fully structured data such as RDF-based sources of LOD to extract structural information, for example, hierarchies, to be directly used in generating adaptation knowledge.

To access more data sources, we will integrate more flexible interfaces into *KEWo* allowing it to better parse and thus pre-process the raw data from a wider variety of web sources. Tasks involved in acquiring this goal are the addition of more flexible text and XML parsers, a flexible interface to connect to MySQL-databases and an option to use a crawler on a web source, for example, a forum, from which data are to be extracted.

We already included into *KEWo* the ability to connect to any given online repository of Linked Data. After this prototypical inclusion, *KEWo* is able to query the repository it has connected using SPARQL queries, which are handled by use of the open-source Sesame framework.[5] We are working on exploiting this connectivity to further facilitate the retrieval from highly structured data repositories, which to a high degree will help reducing the effort currently invested in knowledge extraction because the, at best, semi-structured format data are currently mostly available on the net.

[5]http://www.openrdf.org/ [Last access: 8 June 2011]

8. Summary and outlook

In this paper, we emphasised the benefits of integrating a capability for extracting knowledge from web sources for the development of CBR systems into the *myCBR Workbench*.

We examined the performance and limitations of the *KEWo* prototype by pointing out its abilities to extract and correctly formalise knowledge from semi-structured texts and Linked Data for two CBR knowledge containers, vocabulary and similarity measure. The ability to tap into such highly structured sources as LOD was successfully tested in a second *KEWo* prototype. The containers addressed by this prototype were vocabulary and the similarity measure. We have described our experimental setup regarding the methods used to acquire concepts from available LOD repositories. We were able to produce good quantitive and qualitative results for the knowledge containers vocabulary and assign similarity measures in form of a taxonomy to it, both based upon LOD using our customised tool *KEWo*.

During our work with LOD, we discovered that it is occasionally hard to identify relevant LOD repositories. It is further hard to retrieve the specific names of the attributes or of predicates of the items in these repositories. Noticing that there are ongoing efforts to improve the searchability of LOD, we still deem the lack of searchability is hampering the use of LOD.

A future goal, after integrating *KEWo* into *myCBR Workbench*, is given by the shifting of the extraction approach now implemented in *KEWo* in the direction of LOD retrieval, where some first results have been reported elsewhere (Roth-Berghofer *et al*., 2010). This goal will be followed to benefit from the rapidly growing amount of highly structured data available on the web and at the same time reduce the costly process of extracting knowledge from less structured data (Bizer *et al*., 2009).

As another future goal, we look at adding explanation capabilities to the integrated *KEWo* making it easier to use. The *myCBR* tool already possesses some explanation capabilities, namely conceptualisation of symbols from a vocabulary and explaining the similarity calculation (Roth-Berghofer and Bahls, 2008). We aim towards an automatic extraction of both of these sources of conceptualising information from the web and at adding provenance information.

Acknowledgements

This work was supported by the Research Excellence Framework (REF) of the UK (Explanation-aware myCBR development). We would thank our colleague Ray Gumme for his valuable feedback.

References

Plaza, E. and C. Baccigalupo Principle and praxis in the experience web: a case study in social music, in S. J. Delany (Ed.), ICCBR 2009 Workshop Proc., Workshop Reasoning from Experiences on the Web, 55–63.

Bizer, C., T. Heath and T. Berners-Lee (2009) Linked data-the story so far, *International Journal on Semantic Web and Information Systems*, **5**, 1–22.

Bergmann, R. (2002) Experience Management: Foundations, Development Methodology, and Internet-Based Applications, volume **2432** of *LNCS*, Springer.

Boyd, D.M. and N.B. Ellison. (2007) Social network sites: definition, history, and scholarship, *Journal of Computer-Mediated Communication*, **13**, 210–230.

Smyth, B., P.-A. Champin, P. Briggs and M. Coyle. The case-based experience web, in S. J. Delany (Ed.), ICCBR 2009 Workshop Proc., Workshop Reasoning from Experiences on the Web, 74–82.

Richter, M.M. Introduction, (1998) in M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard, S. Wess (eds), Case-Based Reasoning Technology – From Foundations to Applications, LNAI 1400, Berlin: Springer-Verlag.

Bach, K., C.S. Sauer, and K.-D. Althoff Deriving case base vocabulary from web community data, in C. Marling (Ed.), ICCBR-2010 Workshop Proceedings: Workshop on Reasonng From Experiences On The Web, 111–120.

Stahl, A. and T.R. Roth-Berghofer (2008) Rapid prototyping of CBR applications with the open source tool myCBR, in: ECCBR '08: Proc. of the 9th European conference on Advances in Case-Based Reasoning, Springer, Heidelberg, 615–629.

Roth-Berghofer, T., B. Adrian, and A. Dengel (2010) Case acquisition from text: ontology-based information extraction with SCOOBIE for myCBR, in I. Bichindaritz, S. Montani (Eds.), Case-Based Reasoning Research and Development: 18th International Conference on Case-Based Reasoning, ICCBR 2010, number 6176 in LNAI, Springer, Alessandria, Italy.

Aamodt, A. and E. Plaza (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches, *AI Communications*, **7**, 39–59.

Bridge, D., M.H. Göker, L. McGinty, and B. Smyth (2006) Case-based recommender systems, *Knowledge Engineering Review* **20**.

Stahl, A. (2003) Learning of knowledge-intensive similarity measures in case-based reasoning, Ph.D. thesis, University of Kaiserslautern.

Gennari, J.H., M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubézy, H. Eriksson, N.F. Noy and S.W. Tu (2003) The evolution of Protégé an environment for knowledge-based systems development, *International Journal of Human Computer Studies* **58**, 89–123.

Sauer, C.S., K. Bach and K.-D. Althoff (2010) Integration of linked open data in case-based reasoning systems, in M. Atzmüller, D. Benz, A. Hotho, G. Stumme (eds), Proceedings of LWA2010 – Workshop-Woche: Lernen, Wissen & Adaptivitaet, Kassel, Germany, (nd).

Sauer, C.S. (2010) Analyse von Webcommunities und Extraktion von Wissen aus Communitydaten für Case-Based Reasoning Systeme, Master's thesis, Institute of Computer Science, University of Hildesheim

Recio-García, J.A., M. Casado-Hernández and B. Díaz-Agudo (2010) Extending CBR with multiple knowledge sources from web, in I. Bichindaritz, S. Montani (eds), Case-Based Reasoning. Research and Development, volume **6176** of Lecture Notes in Computer Science. Berlin / Heidelberg: Springer, 287–301. DOI:10.1007/978-3-642-14274-1_22.

Bello-Tomás, J., P.A. González-Calero and B. Díaz-Agudo (2004) JColibri: an object-oriented framework for building CBR systems, in: P. A. G. Calero, P. Funk (Eds.), Proceedings of the 7th European Conference on Case-Based Reasoning, Lecture Notes in Artificial Intelligence LNAI, Springer.

Recio-García, J.A., B. Díaz-Agudo, M.A. Gómez-Martín and N. Wiratunga Extending jcolibri for textual cbr, in H. Muñoz-Avila, F. Ricci (eds), Case-Based Reasoning, Research and Development, 6th International Conference, on Case-Based Reasoning, ICCBR 2005, Chicago, IL, USA, August 23-26, 2005, Proceedings, volume 3620 of *Lecture Notes in Computer Science*, Springer, 2005, 421–435.

MILNE, P., N. WIRATUNGA, R. LOTHIAN and D. SONG Reuse of search experience for resource transformation, in S. J. Delany (Ed.), ICCBR 2009 Workshop Proc., Workshop Reasoning from Experiences on the Web, 45–54.
IHLE, N., A. HANFT, K.-D. ALTHOFF Extraction of adaptation knowledge from internet communities, in S. J. Delany (Ed.), ICCBR 2009 Workshop Proc., Workshop Reasoning from Experiences on the Web, 269–278.
FAYYAD, U., G. PIATETSKY-SHAPIRO and P. SMYTH (1996) The KDD process for extracting useful knowledge from volumes of data, *Communications of the ACM* **39**, 27–34.
FENG, D., E. SHAW, J. KIM and E. HOVY An intelligent discussion-bot for answering student queries in threaded discussions, in: IUI '06: Proc. of the 11th Intl Conf. on Intelligent user interfaces, ACM Press, New York, NY, USA, 2006, 171–177.
CUNNINGHAM, H., D. MAYNARD, K. BONTCHEVA and V. TABLAN (2002) Gate: a framework and graphical development environment for robust nlp tools and applications, in Proc. of the 40th Anniv.Meeting of the Assoc. for Comp. Linguistics (ACL'02), (nd).
SAUER, C., T. ROTH-BERGHOFER Web community knowledge extraction for myCBR 3, in: Research and Development in Intelligent Systems XXVIII incorporating Applications and Innovations in Intelligent Systems XIX, Springer Verlag, New York, 2011.
CHURCH, K.W., P. HANKS (1990) Word association norms, mutual information, and lexicography, *Computational Linguistics*, **16**, 22–29.
W3C, Rdf primer, 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ [Last access: 2010-02-26].
BERGMANN, R. and M. SCHAAF (2003) structural case-based reasoning and ontology-based knowledge management: a perfect match?, *Journal of Universal Computer Science* **9**, 608–626. http://www.jucs.org/jucs$_{97}$/structural$_{c}$ase$_{b}$ased$_{r}$easoning [Lastaccess: 2010–02–26].
ROTH-BERGHOFER, T.R. and D. BAHLS, Explanation capabilities of the open source case-based reasoning tool mycbr, in: M. Petridis, N. Wiratunga (eds), Proceedings of the thirteenth UK workshop on Case-Based Reasoning UKCBR 2008, University of Greenwich: London, UK, 2008, 23–34.

The authors

Christian Severin Sauer

Christian Sauer is a Research Assistant at the School of Computing and Technology at the University of West London, UK. Christian holds a Masters Degree in Information Management and Information Technologies gained at the University of Hildesheim, Germany. Christian's research interests are in the fields of explanation generation and more general explanation-capable information systems, case-based reasoning, knowledge-based systems and in context-aware and context-sensitive systems.

Thomas Roth-Berghofer

Thomas Roth-Berghofer is Professor of Artificial Intelligence at the University of West London and head of the Centre for Model-based Software Engineering and Explanation-aware Computing, which is based in the School of Computing and Technology. He has worked in software industry as software developer, technical consultant, and manager quality and support for several years, before he joined the German Research Centre for Artificial Intelligence DFKI GmbH as senior researcher. After a year as visiting professor at the University of Hildesheim he joined the University of West London. Prof Roth-Berghofer's research focuses on aspects of smarter communication with (complex) information systems and how to engineer such capabilities into software systems as the ability to explain reasoning processes, and results can substantially affect the usability and acceptance of a software system. Together with Dr Armin Stahl, DFKI, he initiated the development of the myCBR open-source case-based reasoning workbench and SDK.