Domain Modeling in TCBR Systems: How to Understand a New Application Domain

Kerstin Bach & Alexandre Hanft

University of Hildesheim
Institute of Computer Science
Intelligent Information Systems Laboratory
Marienburger Platz 22, D-31141 Hildesheim, Germany
kerstin.bach|alexandre.hanft@uni-hildesheim.de

Abstract. Domain knowledge is a prerequisite to build a CBR-System. Especially handling unknown application domains requires preprocessing of the raw data to assure that relevant information is accessible. This approach uses the lexical-semantic net GermaNet to recognize terms in unstructured text sections. Furthermore we explain how to deal with complex inflections in the German language and we present the integration heterogeneous sources to enrich our vocabulary for the German language.
Analyzing the source data of large text passages facilitates supporting the knowledge engineer filtering unknown words and describing them in a proper way so they can be used for actual and future application domains. According to assure a certain quality of the domain model we present the Textual Coverage Rate (TCR) which measures the coverage of text sections in cases with modeled terms.

Keywords: textual CBR, domain modeling, Textual Coverage Rate, GermaNet

1 Introduction

Prior setting up a CBR-System based on Case Retrieval Nets (CRN) [6] one has to preprocess the source data to assure it can be accessed by the system. There is a huge amount of information stored in unstructured textual documents which can hardly be processed because it is written in natural language [18] containing unknown words.
In addition comparing the English language with the German language the base form of German words can be affected while building inflections. For that reason we cannot use a stemmer, hence we have to cope with the inflections in a different way.

Most of the domain models of TCBR systems are hand written or adapted from previous applications [10], [7], [2]. We will introduce an approach and its implementation DoMHIR which supports the knowledge engineer to access raw data containing semi-structured or unstructured cases holding a large amount of text.

The approach encloses how to handle large databases of unstructured texts. Furthermore it explains the creating of a vocabulary based on heterogeneous data sources

and applying the vocabulary repository to analyze whether the unknown domain can be represented.

Therefore we describe in section 2 the integration of *GermaNet*[1] to enrich the known base forms and the *Projekt Deutscher Wortschatz*[2] of the University of Leipzig [15] to ensure any kind of German word forms can be recognized.

The approach has been implemented in DoMHIR (*Do*main *M*odeling and *I*ntegration of *H*eterogeneous *R*epositories) which is described in section 3. The application domain used to evaluate this concept are insurance claims consisting of several passages of free text. The database contains more than 9.500 cases with 2.2 million words.

After describing the case format in section 3 we report the analysis of text section to support the knowledge engineer filtering the unknown words and describing them in a proper way so they can be used for the current domains. Because of the fact that we are dealing with large databases we developed a Textual Coverage Rate to determine the right amount of modeled unknown words and increase the quality of the IE coverage in texts. Related Work is concerned in section 5, followed by outlook and conclusion in section 6.

2 Integration of heterogeneous repositories

Heterogeneous repositories can be used building a vocabulary repository to cope with unknown words in new application domains. Dealing with various domains the vocabulary repository has to be comprehensive to cover the required number of Information Entities (IEs) to represent the text section. IEs as smallest entities as used to represent cases in a CRN providing the retrieval. The integration does not only add new words to the repository, it also connects similar words and provides categories in which the word is classified. This section will show how a vocabulary repository can be built or a given repository can be improved.

2.1 Integrating GermaNet

GermaNet [5] is a lexical-semantic net similar to *WordNet*® of the Princeton University[3] developed at the University of Tübingen. We have used *GermaNet* to enhance our vocabulary to be able to cover a new domain.

GermaNet consists of 54 xml-files in different classes with more than 75.000 terms which are summarized in synsets (sets of synonymous words) and linked to each other with basic semantic relations. Each term is described as lexical unit containing information about its syntax and semantic. As explained in [4] *GermaNet* differs from Princeton's *WordNet*® by following linguistic design principals instead of psychological motivations. Furthermore it aims to contain complete taxonomies (by using artificial non-lexicalized standards) and pursues a uniform treatment of meronymy.

[1] http://www.sfs.uni-tuebingen.de/lsd/
[2] http://wortschatz.uni-leipzig.de
[3] http://wordnet.princeton.edu/

To use the terms and their synonyms in a given vocabulary to build up a case base the *GermaNet* entries have to be integrated in the vocabulary. The terms themselves are used to represent IEs and the semantic relations between terms can be used to assign the similarity arcs. Although *GermaNet* only provides the base forms it covers most of general language terms used in German.

2.2 Integrating inflections of the *Projekt deutscher Wortschatz*

The terms described in *GermaNet* contain no inflections which are important to recognize in natural language texts. Especially in the German language the inflections of a term can differ from its base form as is shown in Table 1. Switching from singular to plural the base form has changed from *"Schluss"* to *"Schlüss"*. But not in every word with an 'u' has to be changed into an 'ü' in plural. Furthermore the inflections of German verbs vary frequently as is shown in Table1.

Table 1. Flections of the German word *Schluss* (English: end, finish or conclusion)

	Singular	*Plural*
Nominative	Schluss	Schlüsse
Genitive	Schlusses	Schlüsse
Dative	Schluss(e)	Schlüssen
Accusative	Schluss	Schlüsse

To recognize base terms and inflections the web service provided by the *Projekt Deutscher Wortschatz*[4] can be used, because it is the most comprehensive collection of German words. For each base form the web service returns its inflections which can be stored as terms in the repository and related to its base form.

3 Domain Modeling with DoMIHR

In this section we present DoMIHR (*Do*main *M*odeling and *I*ntegration of *H*eterogeneous *R*epositories) – a tool which supports a knowledge engineer to define a case format for a given database which can be used to create a TCBR system based on CRNs. DoMIHR is implemented in Java and uses PostgreSQL as database.

3.1 Preparation of the repositories

After introducing in section 2 an approach to enhance given repositories we will now show the realization implemented in DoMIHR. The basis for the vocabulary repository we extended was originally built and used in the ExperienceBook II[5] and its predecessors have developed at the Humboldt University of Berlin [10], [11], [2]. First

[4] http://wortschatz.uni-leipzig.de/Webservices/
[5] https://roy.informatik.hu-berlin.de/ExpBookII/

we used *GermaNet* to add general language base forms and synonyms to the vocabulary. In a second step we applied the web service of the *Projekt Deutscher Wortschatz* to adjoin the inflections of each base form. Finally we have created a repository containing the most German nouns which are used in general descriptions. This repository can now be used to determine IEs and terms for the retrieval as well as for the analysis of the IE coverage of text sections as described in section 3.3.

3.2 Creating a new case format in DoMIHR

Considering a new CBR application domain first one has to analyze how the cases are structured and which of the attributes given in the database are necessary to accomplish the retrieval. The case format contains information how to access and process the source data. By means of the existing database structure the information and retrieval attributes are determined. Information attributes are marked to be displayed after the retrieval, but not used to retrieve a case in the opposite to retrieval attributes. This section will illustrate how to generate a new case format matching the new application using DoMIHR.

```
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <!DOCTYPE database SYSTEM "caseformat.dtd">
 3 <database>
 4    <connectiondata>
 5       ...
 6    </connectiondata>
 7    <tables>
 8       <table>
 9          <tblname>claim</tblname>
10          <column>
11             <clmnname>reason</clmnname>
12             <datatype>text</datatype>
13             <isRetrieval>true</isRetrieval>
14             <kindOfRetrieval>Text</kindOfRetrieval>
15          </column>
16          <column>
17             <clmnname>purchase value</clmnname>
18             <datatype>text</datatype>
19             <isRetrieval>true</isRetrieval>
20             <kindOfRetrieval>AV</kindOfRetrieval>
21          </column>
22             ...
23          <column>
24             <clmnname>repairing charges</clmnname>
25             <datatype>text</datatype>
26             <isRetrieval>false</isRetrieval>
27             <kindOfRetrieval></kindOfRetrieval>
28          </column>
29       </table>
30    </tables>
31 </database>
```

Fig. 1. Example of a Case Format produced by the application

The knowledge engineer is supported by DoMIHR guiding him through the following build up process: In the third step the retrieval attributes are described in detail. The user describes their name, data type and which kind of retrieval type has to be applied. To describe the retrieval type it either can be a text section or an attribute-value-pair. If the attribute-value-pair is chosen the knowledge engineer has to specify possible attribute values and how the retrieval should be executed. If the representation is a text section this attribute will be considered to find a match in the vocabulary. Afterwards the case format is stored in an xml-file and provided for further applications (see Fig. 1). This process is described more detailed in [1].

For each attribute (Fig. 1, line 10–15, 16–21, 23–28) its column name, data type, and retrieval type is described. Of course, the retrieval type only has to be characterized if the attribute is considered as retrieval attribute.

3.3 Analyzing text sections with DoMIHR

Before a CRN can be created it is important to ensure that the vocabulary is comprehensive enough to cover the given text sections of the cases of the application domain. Text sections contain texts in natural language and it is challenging to capture the IEs which describe the meaning of the given section. The main content can usually be expressed by using nouns on which we will focus.

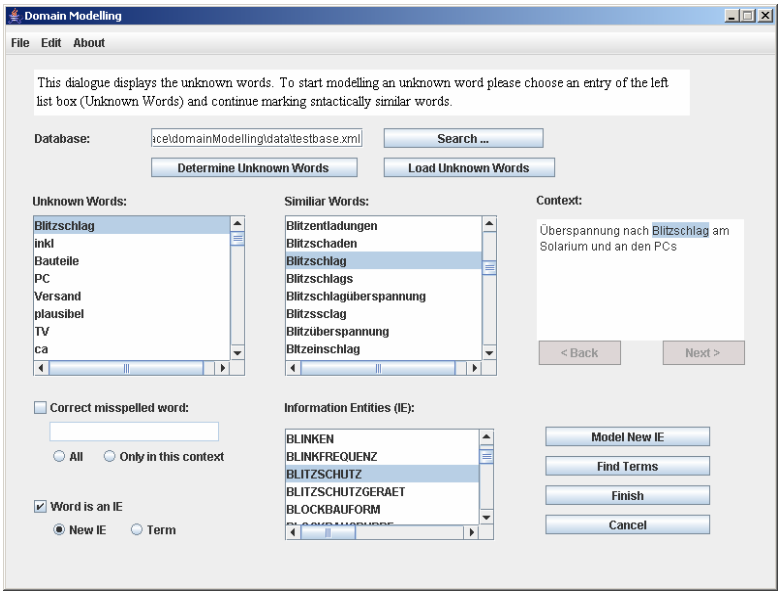

Fig. 2. A screenshot during the build up process modeling “Blitzschlag” (lightning stroke)

We are analyzing the text sections to figure out which words are not described in the vocabulary. In the first step we eliminate all stop words from the given corpus because they have no useful information content. As a second step we remove all words which are contained in the available vocabulary and as a result we get a list of words

the system cannot deal with. Section 2 and 3.1 describe how we enhanced our vocabulary with general terms. Hence, we assume the unknown words are either a mistake in writing or domain specific terms.

DoMIHR now supports modeling the unknown words by showing one list ordered by frequency and another one by alphabet (the middle part of the dialogue). The list orders unknown words by their frequency in the corpus (the left list box) and facilitates that numerous words are modeled in first place. This can be seen in Fig. 2.

The second list box (Similar words) aims to show the knowledge engineer words which are similar spelled to the chosen one. This possibility should motivate the knowledge engineer to model all kinds of terms related to the chosen word at one time and build similarity arcs between them. The text box on the right hand side gives an example in which text section the chosen word occurs.

The bottom part assists to model a new word. It can either be a new IE, term or a misspelled word. The modeling of unknown words enriches the dictionaries used to determine IEs in text sections.

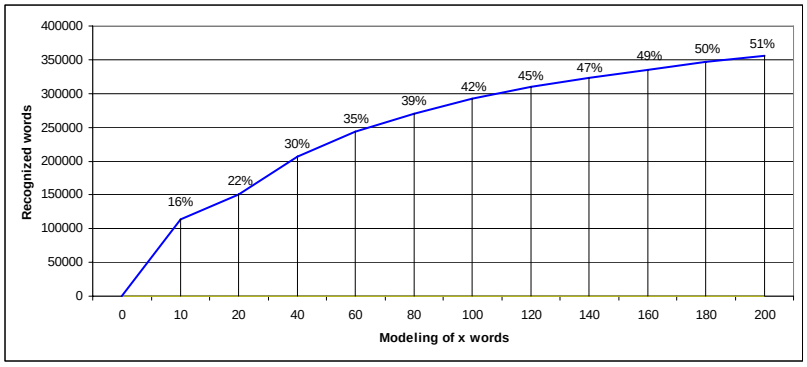

Fig. 3. Effect how the modeling of 200 words describes more than 50% of the unknown words.

Fig. 3 shows how the modeling of 200 words decreases the unknown words down to 50% in comparison when we started.

As mentioned before not every word detected is a new IE. It can also be a term of an already existing IE or a misspelled word. If it is a term it can be added to the IE, but if the word is misspelled it has to be handled in a different way. We decided to handle them in a different way. First of all the knowledge engineer has to figure out which word is meant. The system provides the context in which it was used and the user can decide whether the word should be corrected once or in all occurrences. Instead of adding the misspelled word to the repository like it was done in [10] we augment it with information about the correct term and the corresponding IE. Furthermore the misspelled word can be replaced by the correct one in the source data. By adding these words to our dictionary and relating them to the correct spelled words and IEs we build up a dictionary which can be used to determine the quality of a source corpus. The collection of misspelled terms can also be used to revise misspelled words in new cases and connect them to existing terms to ensure a higher quality of source data. The dictionary of misspelled words is stored separately, but it corresponds with the basic dictionary of terms and IEs.

4 Textual Coverage Rate

After we have used DoMIHR to figure out whether the representation of the text section is guaranteed we found out that DoMIHR does not present those words to remodel which are needed to represent each text section with a certain amount of IEs. Instead of modeling the words with a high frequency in the source data we will introduce an approach which regards each text section and assigns its coverage with IEs. We aim to indicate words which have to be modeled to ensure each text section is represented satisfactorily.

Preparing an unknown corpus for TCBR requires an analysis if the given dictionary holds suitable terms. We will introduce the textual coverage rate (TCR) to describe the potential representation of the source text using the existing dictionary. Therefore we measure the IE coverage of each text section to determine whether it contains a minimum number of terms given in the dictionary or not.

Following [10] a case c with k text sections can be described as c = [S^1, S^2,… , S^k]. Each text section is represented by a set of IEs S^i. In addition, T describes the expected number of IEs in every text section.

(1) and (2) calculate the number of text sections which contain less IEs than given by T. D_{cov} describes the coverage rate of one text section. It is 0 if there are less than T IEs in the tested section S^i. For example a text section is represented by two IEs ($|S^i|$=2) and three IEs are expected (T=3) this section is less covered and D_{cov} for the considered section will be *0*.

$$D_{cov}(S^i, T) = \begin{cases} 0 \quad , & |S^i| < T \\ 1 \quad , & \text{else.} \end{cases} \tag{1}$$

To calculate the *TCR* the number of appropriate covered text sections has to be summed up and the ratio between this sum and the total number of sections gives the *TCR*:

$$TCR(c,T) = \frac{\sum_{i=1}^{k} D_{cov}(S^i, T)}{k}. \tag{2}$$

The *TCR* shown above describes the percentage of text sections represented by at least T IEs. If every text section is adequately covered (for each text section $|S^i| \geq T$ is true) the *TCR* will be 1. Otherwise the knowledge engineer should model more terms to increase the coverage of the given dictionary. To figure out which words should be added to the dictionary the approach described in the previous sections can be used.

Furthermore, if the TCR is 1 the percentage of text sections which contain more that T IEs should be calculated. For that reason the ratio of excess coverage can be examined as shown in (3) and (4). In opposite to (1) and (2) only text sections represented by more than T IEs are factored.

$$D_{excess}(S^i, T) = \begin{cases} 1 & , \quad |S^i| > T \\ 0 & , \quad \text{else.} \end{cases} \tag{3}$$

$$C_{excess}(c, T) = \frac{\sum_{i=1}^{k} D_{excess}(S^i, T)}{k}. \tag{4}$$

A high excess coverage ratio C_{excess} (more than 0.8) points out that more than the expected T IEs represent a text section and the knowledge engineer can consider increasing T. After increasing T the TCR has to be updated and the recalculated C_{excess} helps to decide whether T is chosen correct or still too low. If necessary this step has to be repeated until a C_{excess} of 0.5 or less occurs.

The TCR can be used to explain to the knowledge engineer how many words have to be modeled to achieve a certain quality (given by T) covering the corpus. In addition the C_{excess} can increase the quality of coverage, because it shows how many words have to be modeled to increase T.

5 Related Work

In [6] the domain model was created for a concrete application and the described model is also used to build a similarity model. Each domain model used was build for a specific application domain and purpose.

Another approach dealing with unstructured texts is by Pfuhl [13], but this application covers only one domain. The case model was quiet structured hence many attributes could be handled as AVPs. Pfuhl also built up the repository by hand and used the flections analysis of Lezius [8] to associate words of the same base form, but in the conclusion he pointed out that this algorithm worked out well for his application but will not work for any application domains.

Other approaches which can be applied to deal with flections in the German language are Named Entity Recognition (NER) [9] and stemming [14]. But if we would use a NER a language model for each application domain has to be trained and this is as expensive as building up a repository like we did. Stemming algorithms usually don't work out well analyzing German texts because the declension and conjugation change the root of the word in a complex way the algorithm hardly can deal with.

An approach which has already been used to determine flections in unstructured texts written in Germany is the TreeTagger [16]. It splits up the given text in trigrams and compares each trigram, so changes in the root have not such an influence like the comparison of a string does. But this approach only permits to find syntactically similar words, but no synonyms like we do.

6 Conclusion and Outlook

We have introduced how a repository for a TCBR system can be created using heterogeneous sources. This approach shows how to use the lexical-semantic net *GermaNet* to recognize terms in an unstructured text section. Furthermore we have explained how to deal with the complex inflections in the German language. Also this approach was used to define a case format for a database containing unstructured textual cases and to analyze those cases for unknown words. The presented Textual Coverage Rate (TCR) supports the knowledge engineer achieving and improving a certain quality during the domain modeling.

As a next step *GermaNet* can be used to consider more relations between the given terms to create more detailed similarity measures. Furthermore it can be used to build up *part-of-* and *is-a*-relationships.

Another approach could be the integration of a classification system to reduce the unknown words in application domains like *eCl@ss* [3]. *eCl@ss* is one of the most important horizontal standard categorization for products and services in Europe and is comparable to the North American Industry Classification System (NAICS) [17]. It provides the categorization of products and services based on a hierarchy of classes, dictionary of properties, enumerated value properties and keywords. The categorization can be used to find the kind of product and similar products or relations between them.

References

1. Bach, K.: Domänenmodellierung im Textuellen Fallbasierten Schließen, Master's Thesis, Institute of Computer Science, University of Hildesheim, Hildesheim 2007
2. Hanft, A., and Minor, M.: A Low-Effort, Collaborative Maintenance Model for Textual CBR. In Steffi Brüninghaus (ed): ICCBR 2005 Workshop Proceedings, pp. 138–149, August 2005, DePaul University, Chicago, USA, 2005
3. Hepp, M., Leukel, J., Schmitz, V.: A Quantitative Analysis of eCl@ss, UNSPSC, eOTD, and RNTD: Content, Coverage, and Maintenance. In: Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE 2005), pp. 572–581, Beijing, China, 2005
4. Kunze, C., Lemnitzer, L.: Adapting GermaNet for the Web. In: Proceedings of the First Global Wordnet Conference, Central Institute of Indian Languages, pp. 174-181, Mysore, India, 2002
5. Lemnitzer, L., Kunze, C.: GermaNet – representation, visualization, application. In: Proceedings Conference on Language Resources and Evaluation (LREC) 2002, main conference, Vol V. pp. 1485–1491, 2002
6. Lenz, M.: Case Retrieval Nets as a Model for Building Flexible Information Systems, Dissertation at the Humboldt University of Berlin, Berlin 1999
7. Lenz, M., Hübner, A., Kunze, M:. Textual CBR, In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.), Case-Based Reasoning Technology – From Foundations to Applications, LNAI 1400, Springer Verlag, Berlin, 1998
8. Lezius, W.: Morphy – German Morphology, Part-of-Speech Tagging and Applications. In: Heid, U., Evert, S., Lehmann, E. and Rohrer, C., (eds.): Proceedings of the 9th EURALEX International Congress pp. 619–623 Stuttgart, Germany, 2000

9. Maynard, D., Tablan, V., Ursu, C., Cunningham, H., Wilks, Y.: Named Entity Recognition from Diverse Text Types, In: Proceedings of the Recent Advances in Natural Language Processing 2001 Conference, S. 257–274, 2001. http://gate.ac.uk/sale/ranlp2001/maynard-etal.pdf, last visited on April, 16th 2007
10. Minor, M.: Erfahrungsmanagement mit fallbasierten Assistenzsystemen. Dissertation at the Humboldt University of Berlin, Berlin, May 2006
11. Minor, M.: Experience Management with Case-Based Assistant Systems. In Roth-Berghofer, Th., Göker M. H., and Güvenir H. A., (eds). Advances in Case-Based Reasoning, 8th European Conference, ECCBR 2006, Fethiye, Turkey, Proceedings, LNAI 4106, pages 182–195, Springer Verlag, 2006
12. Minor, M. and Hanft, A.: The Life Cycle of Test Cases in a CBR System. In Blanzieri, E., and Portinale, L., (eds): Advances in Case-Based Reasoning: 5th European Workshop, EWCBR 2000, LNAI 1898, pages 455–466, Berlin, Springer-Verlag, 2000
13. Pfuhl, M.: Case-Based Reasoning auf der Grundlage Relationaler Datenbanken – Eine Anwendung zur strukturierten Suche in Wirtschaftsnachrichten, Dissertation an der Universität Marburg, Deutscher Universitäts-Verlag, Wiesbaden 2003
14. Porter, M. F.: An algorithm for suffix stripping. In: Program, 14(3), S. 130–137, Juli 1980
15. Quasthoff, U.: Projekt der deutsche Wortschatz. In Heyer, G., Wolff, Ch. (eds.). Linguistik und neue Medien, pages 93-99, Wiesbaden: Deutscher. Universitätsverlag, 1998
16. Schmid, H.: Improvements in Part-of-Speech Tagging With an Application To German. In: EACL SIGDAT Workshop, 1995
17. U.S. Census Bureau: North American Industry Classification System 2007, Lanham, MD, Bernan Associates, 2007
18. Wilson, D., Bradshaw, S.: CBR Textuality. In: Proceedings of the Fourth UK Case-Based Reasoning Workshop, 1999