

## Link Proposals with Case-Based Reasoning Techniques\*

Ernst-Georg Haffner, Uwe Roth, Andreas Heuer, Thomas Engel, Christoph Meinel  
Institute of Telematics  
Trier  
Germany  
{Haffner, Roth, Heuer, Engel, Meinel}@ti.fhg.de

**Abstract:** In this paper, we will discuss the problem of proposing links for hypertexts based on Case-Based Reasoning (CBR) techniques. These proposals can be used in addition to traditional textual based methods. At first, we will focus on the basic ideas of CBR. Next, its modeling for Hyperlink-Management Systems will be discussed, and finally the usefulness of CBR in the area of link proposals will be evaluated. The measuring of the quality of link proposals in terms of "recall" and "precision" has to be refined in order to describe the performance of the system adequately.

### Introduction

The importance of high quality hypertexts is increasing as a result of the growth of the World Wide Web (WWW). There are several possibilities to help the user writing hypertexts of high quality. One important aspect of such online authoring tools is the support of finding appropriate links. It would be very convenient yet difficult to provide if the authoring program was able to propose hyperlinks automatically. Unfortunately, it is hardly possible to generate all the hyperlinks of an HTML-document without any human interaction.

Only the so-called *structural* or *navigational links* are easy to generate. For instance, most of the existing web tools provide means for storing files in special directories according to their departmental affiliation (Heuer et al. 1999a). Even though it is very difficult to propose meaningful links for a page on a pure textual base, this type of approach possesses several advantages. Suggestions can be made very fast and - most importantly - with a minimum of user interaction.

On the one side, there are several known possibilities for deriving link proposals on a textual (and statistical) base, but the quality of their proposal is not too persuasive (Cleary et al. 1996). On the other side, if the systems try to improve the accuracy of the proposals by applying additional semantic information, the increased quality of the suggestions must be paid by time-consuming user interaction (Hall et al. 1996).

In this paper, we will introduce a technique similar to Case-Based Reasoning (CBR) (Kolodner et al. 1995) as a possibility to fulfill the aim of providing advanced link proposals of high precision and recall. CBR can be used as a standalone method for suggesting links, but it can also be applied in conjunction with other methods to find very subtle link possibilities. Even though our main focus deals with the use of CBR on a textual analysis of documents, the same ideas can also be applied to approaches based on semantics.

After a more detailed discussion of important areas of current hyperlink research, we will present the main ideas of CBR to solve the problem of link generation. For simplicity, we will speak of "CBR" even though the presented approach is not pure Case-Based Reasoning. We will point out these differences. Our concrete implementation will then be described with special focus on inherent difficulties. Quality - mostly described in terms of "recall" and "precision" - is discussed in section 6 together with some remarks on the usefulness of these measurements and some suggestions for a refinement. Finally, we will present a conclusion and give a short outlook on future work.

### Related Work

---

\* In Proc: World Conference on the WWW and Internet, AACE WebNet'00, San Antonio, Texas, USA, 2000, pp 233-239

Hyperlinks in a (HTML-) text are as important as they are difficult to generate. Research on the area of hyperlinks has a long history already. Kaindl and Kramer present a good and compact summary of the main progresses made so far (Kaindl, Kramer 1999). An important step in hyperlink-proposal research can be found in a contribution of Allan. He proposes three types of links: manual, automatic and pattern-matching ones (Allan 1996). The idea there is to distinguish between links according to the difficulties to calculate or generate them automatically. In this paper, we will present a new approach to retrieve some of the "manual" links and - in addition to the classical methods (e.g. Andersen et al. 1989) - we will try to increase the amount of automatically generated hyperlinks.

The quality of link proposals is traditionally measured in terms of *precision* and *recall* (Cleary et al. 1996). In general, only time-consuming definitions of semantic structures lead to very good link proposals. The tradeoff for less user-interaction using statistical based methods for suggesting links (Gordesch et al. 1993) is mostly a reduction of the precision and the recall in the proposed results.

### Cased-Based Reasoning Systems

The idea of retrieving knowledge in form of Case-Based-Reasoning systems is straightforward and not too complex (Aha 1991). The data is represented in form of cases and each case consists of a *problem* and the affiliated *solutions*. Each problem is described by attributes, which are mostly represented as linear vectors (n-vector P). Often, there are only a few possible solutions so that they can also be represented in form of a (binary) vector (m-vector S). Adequate areas for CBR concepts are diagnosis systems. The attributes of the problems are then called *symptoms* and the solution is the *diagnosis*. A good overview on that topic can be found in (Kolodner et al. 1995). Very valuable work has also been done by Richter et al. (e.g. Richter et al. 1991).

The process of retrieving knowledge based on CBR consists of two phases. In the first phase, the *learning phase*, reasonable cases are "learnt" by storing the problems together with their solutions into the case-base (figure 1).

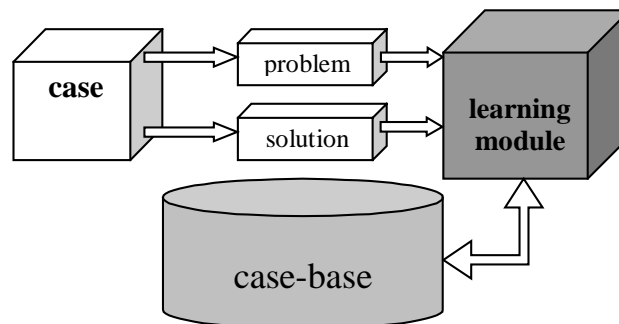


Figure 1: Learning phase of CBR-systems

Then, in the second phase, the *classifying phase*, the CBR-system is confronted with a new problem. It has to retrieve the most adequate case for the presented problem from the case-base and has to transform the according solutions for the new problem (figure 2). If possible, both phases are combined.

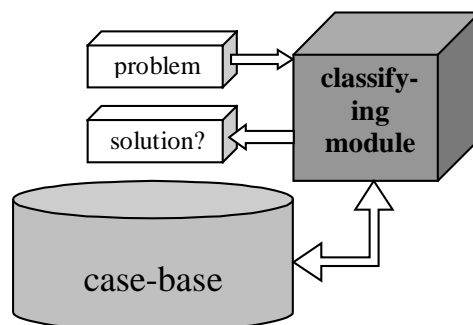
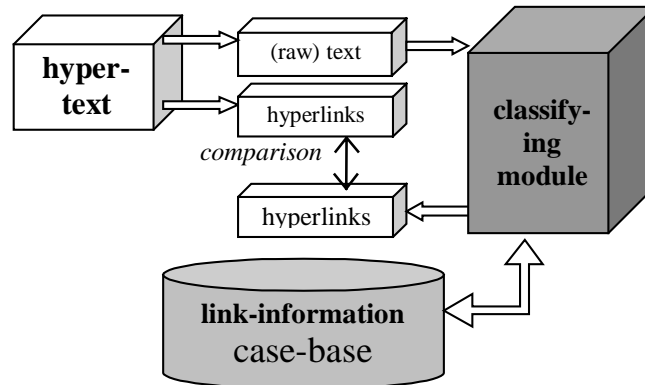


Figure 2: Classifying phase of CBR-systems

Our idea is to model the hyperlink generation problem as a CBR-system and to use the experiences of CBR-research to retrieve high quality links as proposals for the web author. The written texts of the web authors are regarded as the problems and the hyperlinks within are the solutions. A complete hypertext can thus be viewed as a case. In the learning process, (statistical) text attributes are stored together with their attached hyperlinks into the case-base. In the classifying step, raw texts are presented to the system and the CBR process proposes hyperlinks for the text as solutions (figure 3).



**Figure 3:** Learning and classifying applied to the hyperlink context

Furthermore, every new hypertext generates an additional solution – namely the link that refers to itself. Therefore, we designed the CBR-model for the hyperlink environment with an important difference to the classical approach: the cases are not stored explicitly into case-base but more implicitly without a strong relation between problem and solution. Only the importances of certain attributes for the affiliated solutions are represented in a weighted relevance matrix.

### Modeling of an HLM-system Based on CBR

Our idea was the straightforward, relatively simple adaptation of the CBR-concept to the context of hyperlink management systems (HLM) and the evaluation of its potentials. CBR should only be one possibility to propose links and it must collaborate with other methods. As the environment to implement a prototype version we selected an HLM-system as presented in (Heuer et al. 1999b) where problems arising from different supported languages were also modeled.

As already mentioned in the previous section, we choose the keywords of the documents to specify the attributes of hypertexts. Other types of meta-data could also be used here (e.g. author information, creation date, expiration date) if the document was generated with an online authoring tool, e.g. *Daphne* (Heuer et al. 1999a). To find the keywords of a document, we first eliminate all "stopwords" (Chitashvili et al. 1993). All remaining words are weighted according to their position within hyper-tags (e.g. words within <h1>-tags are more important than those within <h2>-tags). At most 100 words of a text with the highest weights are treated as keywords. To investigate the attribute values of the problem vector P, the weights are proportionally transformed into the interval [0..1] (all weights are divided by the maximum value).

### Implementation of the HLM-CBR

We implemented a first version of a CBR-similar algorithm to propose links on the basis of an existing HLM-system (Heuer et al. 1999b) written in Java. Here, the links are represented as objects with source document and target document as variables. Every link has its own description and refers to a default label. For simplicity, the generated link proposals supply only one (default) description.

The core of our CBR-systems consists of a relevance matrix M where the number of rows "n" and the number of columns "m" can be increased dynamically. Every entry (i,j) in M corresponds to the importance of the attribute i for the solution (link-proposal) j. Both sets, the attributes and the solutions, can take up additional

elements any time (e.g. keywords in new HTML-texts and the hyperlinks within). The sum of all relevance values of a solution must always fulfill the following condition:

$$\forall 1 \leq j \leq m : \sum_{i=1}^n M_{ij} = 1 \quad (1)$$

### The classification phase

As we have seen in the last section, a case consists of a pair of vectors, the problem P and the solution S. The aim of the classification is to find the (unknown) solutions of a presented problem P. Thus, the classification phase should supply a resulting m-vector R where the elements of R contain the probability that the according solution is applicable to P. These proposed links are presented to the user in descending order of the corresponding probability. To classify a problem P we must first "normalize" P to  $P^\delta$ :

$$P^\delta = (P_i^\delta) = \begin{cases} 1 : P_i \geq \delta_i \\ 0 : P_i < \delta_i \end{cases} \quad (2)$$

The threshold  $\delta_i$  controls whether an attribute is fulfilled or not. Then  $P^\delta$  simply has to be multiplied with the relevance matrix M to obtain to proposal vector R:

$$R := M \cdot P^\delta \quad (3)$$

### The learning phase

In the learning phase the relevance-indicating values of M have to be adapted so that the problem P of a presented case  $C=(P,S)$  will result in a probability vector R where all elements that correspond to a "1" in S are higher than the corresponding element of threshold  $\delta$ . If necessary, both the number of rows and the number of columns in M can be increased dynamically. All new relevance values are then initialized with  $1/m$  to fulfill condition (1). In the first version of our implementation, an element "0" in S did not lead to an adaptation of entries in M.

To be able to learn the new case C its problem  $P^\delta$  has to be classified. The resulting proposal vector R is then compared to the real solution vector S of C. Four cases have to be distinguished for all elements of both vectors (table 1):

$R_i$	$S_i$	to do
0	0	<i>nothing</i>
0	1	<i>relevance adaptation</i>
1	0	<i>nothing (adaptation in next version)</i>
1	1	<i>nothing</i>

**Table 1:** Relevance adaptation

The process of learning involves a change in the relevance values of M. If a suggested probability is too low so that the corresponding link will not appear on the proposal list of the HLM-system (second case of table 1), the weight of appropriate relevance entries in M must be increased. To fulfill condition (1), the remaining weights have to be decreased by the same amount.

### Evaluation of the Proposed Model

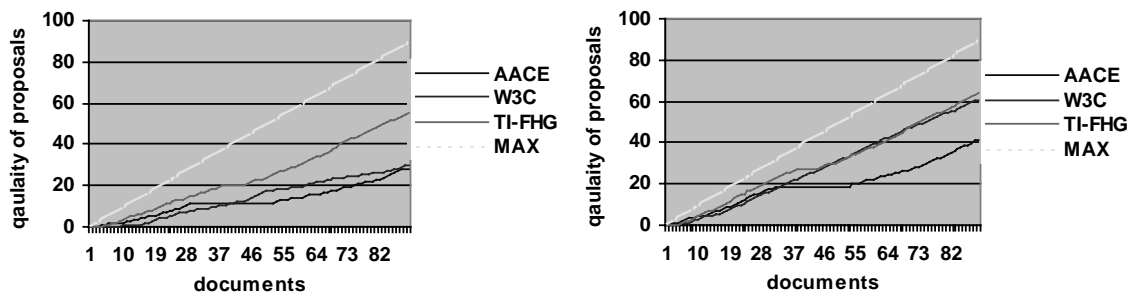
To evaluate the presented model on base of a large amount of data we tested the system "a posteriori" on existing web pages. We extracted the links within the HTML files, classified the raw texts using our model, and finally compared the classification results with the existing hyperlinks. Two classical terms to measure the proposal quality are *recall*, the share of appropriate proposals of all good links and *precision*, and the share of

appropriate proposals of all proposals. Beside this, Cleary and Bareiss mention *ease of use* and *thoroughness* as important factors of link proposals (Cleary et al. 1996).

The system makes many proposals, arranged according to the probability of their usefulness. The user should be able to scroll in the proposal list. In terms of recall and precision this is rather problematic. What are the "appropriate" hits? A web author can, for instance, select several links from the proposal list and can create additional links herself/himself as well. It would be neither correct to count all proposals made nor to ignore the additional ones.

Due to these difficulties, we decided to split the results of our web scans into several parts. There are no unequivocal recall and precision values. Nevertheless - compared to some results so far (e.g. Cleary et al. 1996) - we think that our approach obtains a very good usefulness and implicitly a very good recall and a high precision. As an example, we show the link proposal results for the first ninety pages (beginning at the document root with a breadth-first-search-algorithm) of the *Association for the Advancement of Computing in Education* (AACE) (www.aace.org), the *World Wide Web Consortium* (W3C) (www.w3.org) and the *Institute of Telematics* (TI-FHG) (www.ti.fhg.de).

One of the most important results is shown in the following figures 4a and 4b. Here we can see the increasing sum of all probabilities in the set of proposed links that were really found in the hypertext divided by the number of those links. In figure 4a we considered all links in the scanned document and called the resulting curve the *quantified cumulating recall*, while figure 4b shows the results if we only take care of links that could be proposed, the *qualified cumulating recall*. The gradient of the curves signals the quality of the proposals. The best (maximum) line would be the diagonal. New links in a document that do not exist in the case-base can never be proposed, therefore qualified recalls are "fairer" to the CBR-system.



Figures 4a and 4b: *Quantified cumulating recall* (left) and *Qualified cumulating recall* (right)

High probability values indicate a great correlation between the proposals and the usage of these links. But a system that suggested all links in a case-based would result in even better values. It is necessary to see that mostly very accurate links are proposed ("precision"). Figure 5 presents the (growing) share of links ("hits") among all proposals exceeding the threshold, the *cumulating precision*.

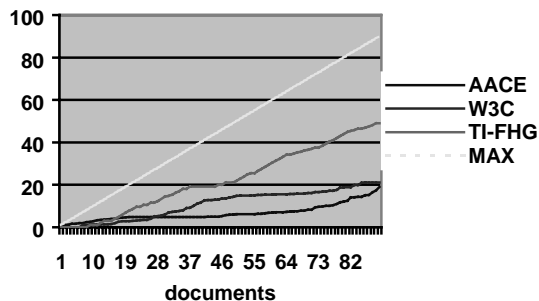


Figure 5: *Cumulating precision*

There are several other interesting relations found in our results so far. A change in the CBR threshold can lead to a very high quality proposal but also to a faster "forgetting" of former cases. If we re-classify cases that have been learnt before, the probability amount does not reach the number of real links, however. A great

number of attributes (e.g. keywords) help to find very subtle proposals, but the handling of the relevance matrix becomes inconvenient. Furthermore, we also compared the highest proposal probabilities of links that were part of a hypertext to those that were not. All in all, we found a high correlation between the complexity and the length of a text and the quality of the suggested links. In some cases, a proposed link that was no part of the hypertext might have been an appropriate supplement.

## Conclusion and outlook

In this paper, we described a system to propose hyperlinks for written raw texts on a statistical basis. We first discussed the underlying concept of Case-Based-Reasoning and outlined the differences to our approach. The core data structure consists of a relevance matrix that provides weighted affiliations between text attributes and possible links. The adaptation of these weights is the essential learning process that can be controlled by a threshold vector and by the mathematical distribution of the changes in relevance. To evaluate the proposed model, we refined the traditional terms of precision and recall in order to adequately measure the quality of the suggested links. We applied the system to several existing web pages “a posteriori” and found rather promising results.

In the near future, we are planning to improve the model by considering both time and document aging on the one side and user link rejection as negative information to learn on the other.

## References

- Heuer, A.; Zhang, Z.; Engel, T.; Meinel, C. (1999a): DAPHNE - Distributed Authoring and Publishing in a Hypertext and Networked Environment. In *Proceedings of the International Conference IuK99 - Dynamic Documents*, 1999. Jena
- Cleary, R.; Bareiss R. (1996): Practical methods for automatically generating typed links. In *Proceedings of the Seventh ACM Conference on Hypertext (Hypertext '96)*, 1996. ACM
- Hall, W.; Davis, H.; Hutchings, G. (1996): *Rethinking Hypermedia: The Microcoms Approach*. Kluwer Academic Publishers, 1996
- Kolodner, J.; Leake, D. (1995): A tutorial introduction to Case-Based Reasoning. In *Case-Based Reasoning*, 1995. AAAI Press, the MIT Press
- Kaindl, H.; Kramer, S. (1999): Semiautomatic generation of glossary links: A Practical Solution. In *Proceedings of the Tenth ACM Conference on Hypertext (Hypertext '99)*, 1999. ACM
- Allan, J. (1996): Automatic hypertext link typing. In *Proceedings of the Seventh ACM Conference on Hypertext (Hypertext '96)*, 1996. ACM
- Andersen, M.H.; Nielsen, J.; Rasmussen, H. (1989): A similarity-based hypertext browser for reading the UNIX network news. *Hypermedia*, 1(3), 1989
- Gordesch, J.; Zapf, A. (1993): Computer-aided formation of concepts. In *Quantitative text analysis, (Quantitative linguistics, Vol. 52)*, 1993. WVT Trier
- Aha, D.W. (1991): Case-Based Learning algorithms. In *Proceedings of the DARPA Workshop on Case Based Reasoning*, 1991. Morgan Kaufmann
- Richter, M.M.; Wess, S. (1991): Similarity, Uncertainty and Case-Based Reasoning in PATDEX. In *Automated Reasoning, Essays in Honor of Woody Bledsoe*, Kluwer Academic Publishers
- Heuer, A.; Haffner, E.-G.; Roth, U.; Zhang, Z.; Engel, T.; Meinel, C. (1999b): Hyperlink management system for multilingual websites. In *Proceedings of the Asia Pacific Web Conference (APWeb '99)*, 1999.

Chitashvili, R.J.; Baayen, R.H. (1993): Word frequency distributions of texts and corpora as large number of rare event distributions. In *Quantitative text analysis, (Quantitative linguistics, Vol. 52)*, 1993. WVT Trier