

Advanced Studies on Link Proposals and Knowledge Retrieval of Hypertexts with CBR*

Ernst-Georg Haffner¹, Andreas Heuer¹, Uwe Roth¹,
Thomas Engel¹, Christoph Meinel¹

¹ Institute of Telematics, Bahnstr. 30-32
D-54292 Trier, Germany
{Haffner,Heuer,Roth,Engel,Meinel}@ti.fhg.de

Abstract. In this paper, several problems in associating hyperlinks to text and the diverse possibilities to overcome these problems are discussed. At the current stage, an important aspect is knowledge retrieval of hypertexts. Our advanced studies on hyperlink management focus mainly on a concept similar to Case-Based Reasoning (CBR) systems as a possibility for the automatic generation of links for hypertexts in addition to traditional textual based methods. A detailed discussion of the basic ideas of CBR and an evaluation of its usefulness follows. Finally, methods to evaluate the quality of the proposals are described.

1 Introduction

The need for high quality hypertexts is increasing as a result of the growing of the World Wide Web (WWW). One of the most difficult tasks when writing hypertexts is the finding of appropriate links. Modern web authoring systems should not only provide possibilities to check link consistencies and help to manage any changes that might occur, but should also propose links in order to improve the usefulness of the documents, which are, in most cases, HTML-files.

It is very difficult to generate hyperlinks based on text semantics without making use of some human interaction. Authoring tools integrate documents of a web site and add structural links which are used to navigate through the various pages of a web site. Often, these web tools provide means for storing files in special folders (departments) according to their content [1].

The reason for these difficulties is the knowledge retrieval of the hypertexts to be able to make meaningful link suggestions. Automatic link proposals possess several advantages. Suggestions can be made very fast and with a minimum of user interaction.

Several possibilities to derive link proposals on a textual base are known, but all of them have their weaknesses, especially concerning the usefulness of the derived link

* In Proc. "EC-Web Conference", Springer LNCS 1875, Greenwich, United Kingdom, 2000, pp. 396-378

suggestions [2], [3]. To improve the accuracy of the proposals, some approaches apply semantic analyses but all of these systems require user feedback of a high quality because it is the job of the web author to create the appropriate semantic model for the pages [4].

In this paper, we focus on a technique derived from *Case-Based Reasoning (CBR)* [5] as a possibility to provide high quality link proposals. The knowledge retrieval used is also based on CBR. Even though our main focus deals with the textual analysis of documents, the same ideas can also be applied to semantic-based approaches.

In a first step, we will discuss the most important areas of current hyperlink research. Next, a short summary of the main ideas of CBR systems and their principles will follow. After this, we will give a detailed description of the methods for retrieving knowledge from hypertexts in order to propose links of high quality. Ideas dealing with the evaluation of these results are mentioned next. Finally, we will present a conclusion and give a short outlook on future work.

2 Progress made in Hyperlink Retrieval

It is very important yet difficult to provide hyperlinks in a (HTML-) document. Hyperlinks dramatically improve content quality by presenting related work, contradictory positions, further information or simply by the continuation of the next page or by giving similar navigational information [6]. The question of how a web author can easily find such information remains, though.

Research on the area of hyperlinks has been carried out since the introduction of the World Wide Web service to the Internet. Kaindl et. al. present a compact history of the progress made so far [7].

Link retrieval research aims at generating hyperlinks if not completely automatically, at least with as little user interaction as possible. Very serious problems arise, though, when trying to retrieve hyperlinks of texts on a statistical base without any semantic knowledge. The results are of low quality [8]. Allan classified link types into three major groups: *manual*, *automatic* and *pattern-matching* [9]. The idea is to retrieve at least the easy-to-find links of the two latter groups and leave most of the former one to the user. This consideration is very useful even though it contains a disadvantage: the classification only works “a posteriori”.

In this paper, we will describe methods to retrieve some of the “manual” links with CBR techniques and focus on the mechanisms to retrieve the corresponding knowledge from the hypertexts. Good examples for classical solutions without using CBR can be found in [10] or [11].

The dilemma of hyperlink retrieval is that a fully automated generation of links on a statistical base [12] leads to relatively bad results in terms of precision and recall, while semantic approaches with very good results require a high degree of user interaction [13]. If hyperlink retrieval is to be used as a tool for supporting web authors in easily adding up links, it would not be appropriate to require the time consuming formulation of a complete model of the semantic dependencies of a text.

Web authors and users (readers) of hypertexts can also be supported without a generation of links. Zellweger et. al. introduce the concept of *fluid links* as a convenient way to deal with temporarily visible information [14].

The technique based on CBR presented in this paper can easily be adapted as part of a web authoring system like *DAPHNE* [1]. It is also appropriate to extend hyperlink management systems such as *Microcosm* [15] with CBR-methods. Another possibility would be the use of CBR in combination with *Distributed Link Services* (DLS) presented by Carr et. al. in [16].

3 Foundations of Case-Based Reasoning Systems

Research in the area of Case-Based Reasoning begun in the early years of the last decade [17]. CBR-systems are well known means of representing knowledge in form of *cases*. Each case can be regarded as a *problem* together with its *solution*. A problem consists of its *description* in form of attributes and one or more *solutions* which refer to it. A typical environment of CBR is the area of diagnostics. Here, the attributes are the *symptoms* and the solution is the *diagnosis* [18].

In general, CBR-systems store their cases in a knowledge database called *case-base*. To solve a new problem, CBR-systems try to find the most similar cases in the case-base. Next, the solutions of these results are transferred to the new problem or are simply regarded as solutions of it. Remarkable efforts have been made to find out how to store only really usable cases (to avoid storage overflow) and how to learn to adapt the rules to compare cases for calculating their similarities [19].

CBR work can be divided into two different phases. The first process, the *learning phase*, builds up the case-base with reasonable cases, e.g. problems together with their solutions. The quality of the resulting case-base is better particularly after the learning phase if the according cases cover the scope of the problem.

The second process, the *classifying phase*, compares a new problem with the existing problems of the cases in the case-base. The solution of the most similar case found is a good proposition for a solution of the new problem. In practice, both phases are combined. The learning of new cases (new “knowledge”) will continue as long as the (real) solutions of formerly posed problems are being recognized.

A main idea of this paper is to model hyperlink generation problems as a case-base and to use the experiences of CBR-systems 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 considered to be the “solutions”. A complete hypertext can be viewed as a *case*. The advantages of CBR-systems to generate hyperlinks are:

- CBR research proves serviceable for extended use (several years) and for use in many areas
- It requires no special user interaction

- The learning process takes place implicitly (i.e. while the user accepts or rejects a link proposal)
- Core functions of CBR are fast and easy to implement
- CBR-systems “learn” to adapt personalized link favorites
- Due to the case model all kinds of (typed) links can be found - not only those that point to documents on the local web side
- The link proposal system can be applied to existent web sites by filling the case-base with hypertexts
- CBR can be used in conjunction with other methods (e.g. the concept model of [4])
- It is not restricted to language characteristics as described in [7]¹
- Link proposals of CBR do not determine non-ambiguous sources of the hyperlinks so that the same keyword can (implicitly) generate more than one link for the hypertext²

On the other hand, there are also some disadvantages of CBR-systems:

- The proposed links do not belong to a small fragment of text but to the whole page so that special link positions must be adapted manually
- CBR generates (many!) link proposals ordered by the probability of their usefulness. Therefore, the classical measurements of recall and precision cannot simply be applied
- The quality of the proposals depends on the structure of the case-base. If it is empty, the system cannot make any proposals. If it overflows, some cases will be “forgotten”

The use of CBR systems can be applied to hyperlink management systems in a straightforward manner. In order to verify the quality of the CBR-system, we scan web pages and take their links as solutions of the problem described by the (raw) text. Before learning those cases we try to classify them first. In the next step, the links proposed by the classifying module are being compared to the really existent hyperlinks of the HTML-pages. Finally, the complete page (text attributes together with the actual hyperlinks) is learnt as a new case for the case-base.

It is clear that at the beginning of such a process the resulting proposals – if any – are not too useful. The quality of the proposals will only improve while increasing the knowledge base and filling it with reasonable cases.

In the area of diagnostics, there are two unusual adaptations of CBR. The first difference arises from a special treatment of hypertexts: a hyperlink that points to the HTML-file forming a case should be added to the number of links (solutions) of the case-base as well, even though there is no single document which contains this hyperlink already. To a certain extent, this means that the problem itself is a part of the solution – of no relevance for the original case but very important for future classification steps of other texts. The other and even more essential difference to classical

¹ Even though we only tested the system for English and German pages

² The exact final location of the hyperlink can be replaced by the user

CBR-systems consists in the storage of the cases themselves. We calculate the most probable link proposals *implicitly* by considering the weights of the according symptoms to the regarded links. Usually, CBR-systems are looking for the most similar cases in the case-base *explicitly* and then transfer the results found to an existing problem.

4 Knowledge Retrieval of Hypertexts

A very crucial question in the context of CBR-systems is the transformation of the problem into certain properties that represent it. Therefore, it is necessary to retrieve the relevant information of the according hypertexts. In this section, we present a technique for knowledge retrieval that is based on statistical and syntactical considerations. Usually, a problem is modeled as an n -vector P where n is the number of attributes used to describe the problem. Every element of P must be normalized into the interval $[0..1]$. The solutions of a case are also represented that way. Here, we speak of an m -vector S , which is mostly a binary vector with elements set to 1 if and only if solution i solves the Problem P , and 0 otherwise. The variable m is the number of all solutions available from the according case-base. Obviously, a problem P can have up to m solutions. In the presented concept, the solutions are hyperlinks within the (problem-) files. To specify the attributes of hypertexts we chose the following settings (if available):

- Every important (weighted) *keyword* of the document is regarded as an attribute
- Every *author* of the document forms an attribute
- The creation date and expiration date of a document are subsumed to one attribute “*validation*”
- The publishing state³ and the version are combined to form the attribute “*availability*”
- The department information is one attribute “*structure*”, but we make the restriction that each document must not belong to more than one department

Thus, we made a statistical approach to apply CBR-ideas. Semantic methods could have been modeled at this point too. An evaluation of our settings will be given in section “5 Evaluation of the CBR-Approach”.

4.1 Keyword Extraction

A very difficult problem is the extraction of keywords from a document on the basis of statistical distribution [20]. We decided to carry out a full text analysis with a special treatment of HTML-tags. All words beside HTML tags, comments and the stop-

³ Allowed states are for instance: *generation in progress, reviewed, exported to the Internet, published ...*

words (e.g. a multilingual list from CD-ISIS [21]) were treated as potential keywords⁴. Beside the classical stopwords we regard in the context of hyperlink management also terms as “homepage” and the company's name as unusable for classification of whole web pages by keywords. A “word” in this context is a sequence of letters without special characters (e.g. hyphens). The following table 1 shows the - arbitrary chosen - weights we attached to every word in a text depending on its relative position between tags. These settings reflect that keywords in titles or headlines are more important than those in the body. In the next version of our CBR-approach, the weights of the keywords should also be part of the learning process.

Position within tag	Weight
<TITLE>	50
<META> (description)	10
<H1>	5
<H2>	4
<H3>	3
<H4>	2
<BODY>	1
<A HREF>	0

Table 1: Distribution of keyword weights

The number of occurrences of a word in a document multiplied with the settings of table 1 results in an absolute weight. Words within the anchor-tag for hyperlink references (HREF) are unconsidered because their information results already in a concrete link.

Only the words that exceed a minimum threshold (depending upon the document length) are treated as keyword attributes. In addition - if there are too many keywords - only the ones with the highest weights are selected⁵. At the end all weights are proportionally transformed into the interval [0..1]. Thus, all weights are divided by the maximum value among them.

Some essential points of the keyword extraction are:

- Keyword extraction does not consider ambiguities in the meaning of the words that are spelled the same
- Abridgments and acronyms can be defined in the text itself and will thus be treated like stopwords
- Even if two texts only have few keywords in common, they can share their solutions in CBR-systems
- The use of full form lexicons for treating different kinds of word-flexion [22] should be applied in the future

⁴ We made our studies for English and German documents. For the latter, we have an additional restriction: only words with beginning capitalized letter (beside the first word of a sentence) are regarded as potential keywords. In German nouns always start with a capitalized letter.

⁵ The number of keywords varied between five and hundred.

4.2 Author Information

If the author of an HTML document is known, this information will form an additional attribute for the according CBR-case. If there is more than one author, the system is able to take care of the varying relevance of the different authors (e.g. the first author is weighted by 1, the second by 1/2, the third by 1/3 and so on; or all authors are weighted by 1 in case of alphabetically sorted authors).

4.3 Document Validation

The idea to consider the “age” of a HTML-file as an attribute to form a CBR-case arises from the perception that the relevance of the content depends on its creation and expiration time. This is also true for the links contained in these documents. To get a linear value between 0.0 and 1.0 for the validation of a file we calculate the “distance in time” between now and the lifetime of the document. There are three possibilities as described in figure 1:

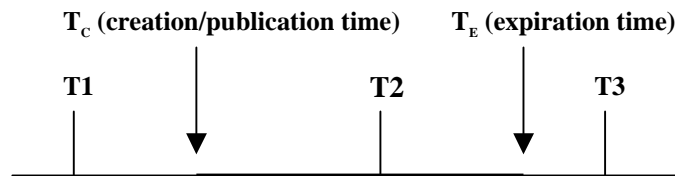


Figure 1: Timeline to calculate validation

If the creation or publication time of a document D is in the past and the expiration time is in the future ($T_2 = \text{“now”}$), the “validation” value v of D will result in:

$$v = 1.0 \quad (1)$$

If the publication time of D is in the future, e.g. D is not yet visible in the Internet/Intranet ($T_1 = \text{“now”}$) the validation v is calculated as:

$$v = \frac{T_E - T_1}{T_C - T_1} \quad (2)$$

If the document is already obsolete the validation attribute of the corresponding documents will obtain the following value ($T_3 = \text{“now”}$):

$$v = \frac{T_3 - T_C}{T_3 - T_E} \quad (3)$$

4.4 Departmental Information

If possible, additional information of the document structure is also used as an attribute for CBR. Here, the idea is that those documents that are positioned “deeper” in the (tree) structure of a web-site obtain a lower value as those on the top level. The usabil-

ity of links with regard to structure depends on how general the contents of the concerned web pages are. It is more probable that links on the top level are not as specific as those in other positions, even though this is not always true. Very often, files all over the web-site refer to the root of the tree (the “home” link).

5 Evaluation of the CBR-Approach

The presented link proposal method implemented as a pure Java application should be a module of a complete hyperlink management system or a standalone program. Even though it is not meant to be used “a posteriori” on finished hypertexts, we think that the comparison of the system proposals with the real links inside existing documents is an appropriate possibility to measure the performance of the system. Therefore, we chose several existing web pages, extracted the links within, classified the texts without considering the link information, and compared, finally, both results. The model we presented is not easy to classify with regard to the terms *precision*, *recall*, *thoroughness* and *ease of use* [4].

- The CBR approach is easy to use because it provides proposals without any prior user interaction (no construction of semantic models etc.)
- All link proposals belong to the whole document. Therefore, the web author has to replace the links if she/he wants to have it at a specific location within the text. This is an inconvenience of the CBR-approach
- CBR can only be as accurate as the according cases in the case-base. It can never propose a link which has not already been learnt
- The system makes many proposals, ordered by the probability of their usefulness. An evaluation in terms of recall and precision this is rather problematic

Therefore, we introduced new terms on base of the probabilities of the link proposals. On the one side, the “*Quantified Cumulating Recall (QCR)*” describes the sum of all probabilities in the set of proposed links that were really found in the hypertext divided by the number of those links. If applied on every link in the hypertext, the QCR becomes an increasing curve. On the other side, the “*Quantified Cumulating Precision (QCP)*” describes the share of hits among those proposals (the “good” links of [4]). The gradient of both curves signals the quality of the proposals. The best curve would be the diagonal.

As a practical evaluation example, we show below the link proposal results for the first ninety pages⁶ of the *Association for Computing Machinery (ACM)* (www.acm.org), the *World Wide Web consortium (W3C)* (www.w3.org) and the *Institute of Telematics (TI-FHG)* (www.ti.fhg.de)⁷ (figures 2 and 3).

⁶ ... with a depth first search tree scan beginning with the document root. No other than HTML-files were considered. For external web scans only the keyword attributes were available.

⁷ We did not begin with the document root here, but with the “no frames” root page.

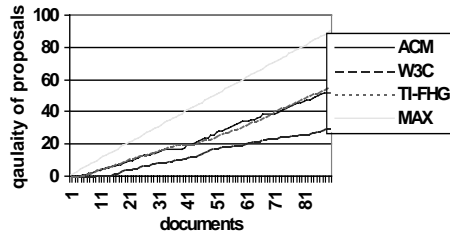


Figure 2: Quantified Cumulating Recall

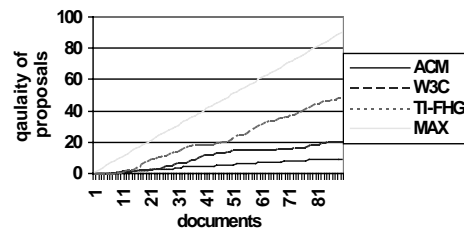


Figure 3: Quantified Cumulating Precision

The QCP of the TI-FHG proposals was higher because we could use here the complete set of attribute values (keywords, author information, validation etc.). The overall recall and precision results are rather good, but a lot of effort has still to be done in order to become perfect.

6 Conclusion and Outlook

We presented a method similar to Case-Based Reasoning to propose links in the context of hyperlink management systems. After a general introduction of CBR-systems we focused on a special model for hyperlink suggestions with some differences to classical approaches. Especially, we decreased the importance of the cases themselves and operated with implicit similarities instead of explicitly looking for the best matching element in the case-base. The difficult task of retrieving knowledge from a hypertext was split into several parts. The most important attributes are the weighted keywords.

In order to adequately measure the quality of the system proposals we refined the terms of recall and precision. In the evaluation part, we presented some promising results of “a posteriori” classifications of web pages.

Even though the proposal mechanism works rather well, several improvements are possible. Our future work aims at increasing the efficiency of the learning algorithm and the finding of the best parameter values. The method must be extended by considering not only proposal acceptance but also the rejection of hyperlink suggestions to improve precision.

References

1. A. Heuer, Z. Zhang, T. Engel and C. Meinel. DAPHNE - Distributed Authoring and Publishing in a Hypertext and Networked Environment. In *Proceedings of the International Conference IuK99 - Dynamic Documents*, 1999. Jena
2. M. Bernstein. An apprentice that discovers hypertext links. In *Proceedings of the First European Conference on Hypertext (ECHT-90)*, 1990

3. D. T. Chang. HieNet: A user-centered approach for automatic link generation. In *Proceedings of the Fifth ACM Conference on Hypertext (Hypertext '93)*, 1993
4. C. Cleary, R. Bareiss. Practical methods for automatically generating typed links. In *Proceedings of the Seventh ACM Conference on Hypertext (Hypertext '96)*, 1996. ACM
5. J. Kolodner, D. Leake. A tutorial introduction to Case-Based Reasoning. In *Case-Based Reasoning*, 1995. AAAI Press, the MIT Press
6. F. J. Ricardo. Stalking the paratext: speculations on hypertext links as second order text. In *Proceedings of the Ninth ACM Conference on Hypertext (Hypertext '98)*, 1998. ACM
7. H. Kaindl, S. Kramer. Semiautomatic generation of glossary links: A Practical Solution. In *Proceedings of the Tenth ACM Conference on Hypertext (Hypertext '99)*, 1999. ACM
8. R. J. Glushko. Design issues for multi-document hypertexts. In *Proceedings of the Second ACM Conference on Hypertext (Hypertext '89)*, 1989. ACM
9. J. Allan. Automatic hypertext link typing. In *Proceedings of the Seventh ACM Conference on Hypertext (Hypertext '96)*, 1996. ACM
10. C. Marshall, F. Shipman. Searching for the missing link: discovering implicit structure in spatial hypertext. In *Proceedings of the Fifth ACM Conference on Hypertext (Hypertext '93)*, 1993. ACM
11. M. H. Andersen, J. Nielsen, H. Rasmussen. A similarity-based hypertext browser for reading the UNIX network news. *Hypermedia*, 1(3), 1989
12. J. Gordesch, A. Zapf. Computer-aided formation of concepts. In *Quantitative text analysis, (Quantitative linguistics, Vol. 52)*, 1993. WVT Trier
13. C. Petrou, D. Martakos, S. Hadjiefthymiades. Adding semantics to hypermedia towards link's enhancement and dynamic linking. In *Hypertext - Information Retrieval - Multimedia '97 (HIM 1997)*, 1997. Universitaetsverlag Konstanz
14. P. Zellweger, B.-W. Chang, J. Mackinlay. Fluid links for informed and incremental link transitions. In *Proceedings of the Ninth ACM Conference on Hypertext (Hypertext '98)*, 1998. ACM
15. W. Hall, H. Davis, G. Hutchings. *Rethinking Hypermedia: The Microcoms Approach*. Kluwer Academic Publishers, 1996
16. L. A. Carr, W. Hall, S. Hitchcock. Link services or link agents? In *Proceedings of the Ninth ACM Conference on Hypertext (Hypertext '98)*, 1998. ACM
17. D. W. Aha. Case-Based Learning algorithms. In *Proceedings of the DARPA Workshop on Case Based Reasoning*, 1991. Morgan Kaufmann
18. K.-D. Althoff, S. Wess. Case-Based Reasoning and Expert System Development. In *Contemporary Knowledge Engineering and Cognition (Schmalhofer, Strube, Wetter)*, 1992. Springer
19. D. Joh. CBR in a changing environment. In *Case-Based Reasoning Research and Development (LNAI 1266)*, 1997. Springer
20. R. J. Chitashvili, R. H. Baayen. 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
21. CD/ISIS Wageningen Agricultural University Library
<http://www.bib.wau.nl/isis/docum.html> (multilin.) 1999
22. R. Sproat. *Morphology and Computation*. The MIT Press, 1992