Information Processing and Management 48 (2012) 183–203

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Updating broken web links: An automatic recommendation system

Juan Martinez-Romo *, Lourdes Araujo

Dpto. Lenguajes y Sistemas Informáticos, NLP & IR Group, UNED, Madrid 28040, Spain

ARTICLE INFO

Article history:
Received 27 July 2009
Received in revised form 15 March 2011
Accepted 24 March 2011
Available online 19 April 2011

Keywords:
Web information retrieval
Link integrity
Recommender system

ABSTRACT

Broken hypertext links are a frequent problem in the Web. Sometimes the page which a link points to has disappeared forever, but in many other cases the page has simply been moved to another location in the same web site or to another one. In some cases the page besides being moved, is updated, becoming a bit different to the original one but rather similar. In all these cases it can be very useful to have a tool that provides us with pages highly related to the broken link, since we could select the most appropriate one. The relationship between the broken link and its possible linkable pages, can be defined as a function of many factors. In this work we have employed several resources both in the context of the link and in the Web to look for pages related to a broken link. From the resources in the context of a link, we have analyzed several sources of information such as the anchor text, the text surrounding the anchor, the URL and the page containing the link. We have also extracted information about a link from the Web infrastructure such as search engines, Internet archives and social tagging systems. We have combined all of these resources to design a system that recommends pages that can be used to recover the broken link. A novel methodology is presented to evaluate the system without resorting to user judgments, thus increasing the objectivity of the results, and helping to adjust the parameters of the algorithm. We have also compiled a web page collection with true broken links, which has been used to test the full system by humans.

Results show that the system is able to recommend the correct page among the first ten results when the page has been moved, and to recommend highly related pages when the original one has disappeared.

1. Introduction

Broken links in the World Wide Web, i.e., links pointing to an unavailable page, are usually the result of pages that have disappeared or have been moved. Broken links affect PageRank (Green, 2004) and discourage page visitors, who might regard the page containing these links as obsolete or unprofessional. In order to check the validity of the links in a page, there exist a number of free systems such as Xenu's Link Sleuth[1] or W3C Link Checker,[2] and commercial software such as CityDesk[3] or Web Link Validator.[4] Once a page owner has detected a broken link – either manually or using one of these checking systems – it is his own responsibility to update or erase the link. However, it is a time-consuming task that has to be performed frequently. Furthermore, when navigating through the web we often reach links that are apparently very interesting for us, but that do

* Corresponding author.
E-mail addresses: juaner@lsi.uned.es (J. Martinez-Romo), lurdes@lsi.uned.es (L. Araujo).

[1] http://home.snafu.de/tilman/xenulink.html.
[2] http://validator.w3.org/checklink/checklink.
[3] http://www.fogcreek.com/citydesk/index.html.
[4] http://www.relsoftware.com/wlv/.

0306-4573/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2011.03.006

not work any more. In those cases we can try a web search using information that the broken link and the page that contains it, suggest to us. But this is a tedious work that we propose to perform automatically, or at least to assist the user with this task.

We can realize that in many cases the page linked by the broken link has not disappeared, but it has been moved inside the web site structure, or to another web site. We have also observed that even in the cases in which the page has in fact disappeared, it is possible to find highly related pages that can be used to repair the broken link.

In most cases the page containing the broken link provides a lot of information regarding the page it points to: the anchor text, the surrounding anchor text, the URL and the text in the page. But we can also use other resources from the Web Infrastructure, such as a cached page stored in a search engine or Internet archive (*Wayback Machine*) (Mearian, 2009), available utilities from search engines and information provided by social tagging websites.

In this work we propose a system that recommends candidate pages to repair a broken link. Our system checks the links of the page given as input. For those which are broken, the system determines if we have available enough information to perform a reliable recommendation. If so, the system provides to the user with a set of candidate pages to replace the broken link. Our method applies information retrieval techniques to extract the most relevant data from several information sources. Some of them, such as the anchor text and the page text have been previously studied in our preliminary works (Martinez-Romo & Araujo, 2008, 2009, 2010). This work extends the previous findings by introducing an exhaustive analysis of different information retrieval techniques in order to use them in two very important phases of the system: extraction of relevant terminology and ranking of results. The candidate pages are obtained by submitting queries to a search engine composed of terms extracted from the different sources. In order to tune the results, the pages recovered in this way are filtered and ranked according to relevance measures obtained by applying information retrieval techniques. The resulting list of pages is presented to the user. Of course, we do not know the actual purpose of an editor when he includes a link in his web page, or the actual interest of a web searcher. Therefore, our system does not automatically associate a page with a broken link, but recommends a ranked list of candidate pages that are highly related to the missing page.

The first step of our work has been the analysis of a large number of web pages and their links in order to determine which ones are the most useful sources of information and which of them are the most appropriate in each case. This study has allowed us to extract criteria to determine, for a particular broken link, whether it makes sense to look for candidate pages to recommend to the user, or whether the available information is not enough to attempt the recovering. Sometimes we can recover a broken link by entering the anchor text as a user query in a search engine. However, there are many cases in which the anchor text does not contain enough information to do that. In these cases, we can compose queries by adding terms extracted from other sources of information (the text of the web page that contains the link, a cached page stored in a search engine – if it exists – the URL, etc.) to the anchor text.

In order to evaluate the different approaches considered, we have developed a method which mainly relies on the random selection of pages and on the use of links that are not true broken, thus allowing us to check whether in the case they were true broken, our techniques would be able to recover the correct page. Later, we have performed both a manual and an automatic evaluation of the resulting system on two different collections of pages containing true broken links.

The remainder of the paper proceeds as follows: Section 2 provides an overview of related work; Section 3 presents a scheme of the proposed model; Section 4 describes the methodology we have followed to evaluate the suitability of the sources of information considered; Section 5 analyzes the utility of the anchor text of the links to recover the page; Section 6 studies the suitability of different sources of information to provide terms for the query expansion process; Section 7 presents several resources from the Web infrastructure that can be used to obtain more information; Section 8 describes the process to rank the candidate documents; Section 9 analyzes some parameter settings of the system; Section 10 presents the scheme resulting from the previous analysis, as well as the results of applying it to two sets of truly broken web links; Finally, Section 11 draws the main conclusions.

2. Related work

Despite the problem of broken links was considered the second most serious problem on the Web (Markwell & Brooks, 2002) several years ago, missing pages are still frequent when users surfing the Internet. Previous works quantified this problem: Kahle (1997) reported the expected life-time of a web page is 44 days. Koehler (2002) performed a longitudinal study of web page availability and found the random test collection of URLs eventually reached a "steady state" after approximately 67% of the URLs were lost over a 4-year period. Markwell and Brooks (2002) monitored the resources of three authentic courses during 14 months, and 16.5% of the links had disappeared or were non-viable.

Most previous attempts to recover broken links are based on information annotated in advance with the link. Davis (2000) studied the causes that provoke the existence of broken links and proposed solutions focused on collecting information on the links in its creation or modification. One of the first works which tried to solve the problem of broken links was (Ingham, Caughey, & Little, 1996), where Ingham et al. presented a model for the provision of referential integrity for Web resources, using for this task a novel object-oriented approach. The Webvise system (Grønbæk, Sloth, & Ørbæk, 1999), integrated with Microsoft software, stores annotations in hypermedia databases external to the web pages. This allows the system to provide a certain degree of capacity to recover integrated broken links. The information is stored when the links are created or modified. Shimada and Futakata (1998) designed the Self-Evolving Database (*SEDB*), which stores only links in a centralized way while documents are left in their native formats at their original locations. When a

document is missing, the *SEDB* reorganizes all links formerly connected to the missing document in order to preserve the topology of links.

Thought with a purpose different to repair broken links, other works have investigated mechanisms to extract information from the links and the context they appear in. Some of these mechanisms have been tested in our system for recovering broken links. McBryan (1994) proposed to use the anchor text as a help to the search of web resources. This work describes the tool *WWWW* intended to locate resources on the Internet. The *WWWW* program surfs the Internet locating web resources and builds a database of these. Each *HTML* file found is indexed with the title string used in there. Each URL referenced in an *HTML* file is also indexed. The system allows searching on document titles, reference hypertext, or within the components of the URL name strings.

Dai and Davison (2009), considered outdated links, alongside broken links, building a classification model based on a number of features of these links. Authors carried out a classification task for links of a public directory with the main target of vetting automatically many links of the web. For the evaluation, the authors used the ODP data set, which was based on the external pages cited by *DMOZ Open Directory Project* and corresponding to historical snapshots provided by the *Wayback Machine* service, offered by the *Internet Archive*. As for their data set for training and testing the classifier, they randomly selected external pages which had complete historical snapshots since the year in which they were first observed in the *ODP* directory up to year 2007.

The INEX Link-the-Wiki task (Huang, Geva, & Trotman, 2009) aims at evaluating the state of the art in automated discovery of document hyperlinks using *Wikipedia* as reference collection. The objective of the task is to create a reusable resource for evaluating and comparing different state of the art systems and approaches to automated link discovery.

Many works also appeared using different techniques in the TREC-10 Web Track (Craswell & Hawking, 2001) that introduced the homepage finding task, where the queries were the name of an entity whose homepage was included in the collection. The challenge in this task was to return all the homepages at the top of the ranking. Among the most relevant works, Westerveld, Kraaij, and Hiemstra (2001) used language models along with four sources of information: page content, number of inlinks, URL depth and the anchor texts of outlinks. Xi, Fox, Tan, and Shu (2002) fulfilled a machine learning study divided into three stages: in a first stage they used the vector space model for getting the similarity between a query and several sources of information. Afterwards they filtered the collection pages by means of a decision tree. Finally, a logistic regression model ranked the remaining pages. Amati, Carpineto, and Romano (2001) used pseudo-relevance feedback and divergence from randomness to extract relevant terms and perform expanded queries.

Although all these works in the TREC environment have in common the page finding task, our work differs in the following respects: (i) Techniques used every year were specific for the intended track. (ii) In the homepage finding task the number of potential candidates is much more reduced and most of papers used the URL depth as a main factor for the page selection. (iii) Mixed tasks used queries that differed from the anchor text and the applied techniques were specific to this scenario. (iv) The size of the collection is much smaller than the whole Internet, and the collections used on the mentioned tracks allow extracting information on the links structure, which is impossible in our study.

Closest to our research are other works (Morishima, Nakamizo, Iida, Sugimoto, & Kitagawa, 2008, 2009a, 2009b; Nakamizo, Iida, Morishima, Sugimoto, & Kitagawa, 2005) which have developed a software tool that finds new URLs of web pages after pages are moved. The tool outputs a list of web pages sorted by their plausibility of being link authorities. This system uses a link authority server which collects links and then it sorts them by plausibility. This plausibility is based on a set of attributes concerning the relations among links and directories.

As the authors state in the paper, the mechanism they propose to exploit locality in the problem is to provide some methods complementary to the content-based approach. For the evaluation in Morishima, Nakamizo, Iida, Sugimoto, and Kitagawa (2009a), the authors collected links contained in Web sites of nine university domains and selected outgoing links from them. Their system found 858 broken links in the collection, 259 of which were identified as having been caused by page movement. These were the only ones used in the evaluation.

Popitsch and Haslhofer (2010), consider a link as broken not only when the target page can not be accesses any more, but also when representations of the target resource were updated in such a way that they underwent a change in meaning that the link-creator had not in mind. The approach proposed in this work for fixing broken links is based on an indexing infrastructure. A monitor periodically accesses considered data sources, creates an item for each resource it encounters and extracts a feature vector from these the representation of these items. The authors have created a dbpedia-eventset that was derived from the person datasets of the DBpedia snapshots 3.2 and 3.3, selecting some particular subsets, appropriate to evaluate their approach.

Harrison and Nelson (2006), Nelson, McCown, Smith, and Klein (2007), and Klein and Nelson (2008) built a framework for locating missing web pages as a digital preservation project. Authors extract several terms called "*lexical signature*" from the pages that they want to preserve, and these terms are used to query a search engine if some of those pages disappear in the future. Terms are extracted using a simple *Tf-Idf* approach. In addition, at least a server have to store these information about the pages joined to the project.

Our work differs from previous proposals since it does not rely on any information about the links annotated in advance, and it can be applied to any web page. Furthermore, we do not try to recover just moved or outdated pages, but any kind of disappeared page. We use a great amount of Web resources to find a missing page by employing new technologies that have arisen recently. Furthermore, we have defined a methodology to evaluate the system without resorting to user judgments, thus increasing the objectivity of the results.

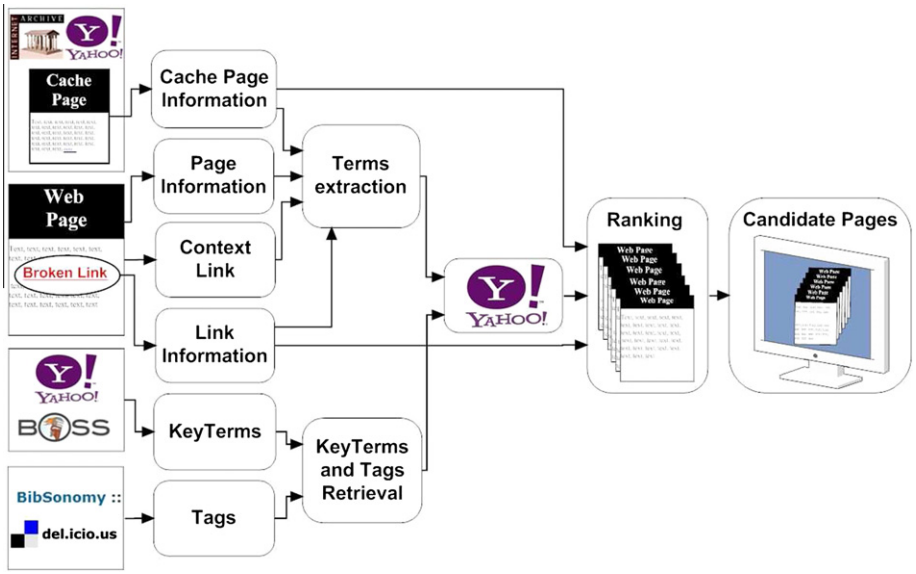

Fig. 1. Scheme of the system for automatic recovering of broken links.

3. Proposed model

Fig. 1 presents a scheme of the proposed model. The broken link and the web page that contains it, provide terms that may be related with the disappeared web page content. The most relevant of those terms are used to build queries that are submitted to a search engine. Our system performs a form of query expansion (Efthimiadis, 1996), a well-known method to improve the performance of information retrieval systems. In this method, the expansion terms have to be very carefully selected to avoid worsening the query performance. Many approaches proposed for query expansion use external collections (Voorhees, 2006, 2005, 2003), such as Web documents, to extract candidate terms for the expansion. There exist other methods to extract the candidate terms from the same collection that the search is performed on. Some of these methods are based on global analysis where the list of candidate terms is generated from the whole collection, but they are computationally very expensive and its effectiveness is not better than one of methods based on local analysis (Jing & Croft, 1994; Qiu & Frei, 1993, 1995; Schütze & Pedersen, 1997). In our case, the original query is composed of the terms extracted from the anchor text, and the sources of expansion terms are the elements of the web page containing the broken link (text, URL, context, etc.), and also, if they exist, a cached page corresponding to the disappeared page that can be stored in the search engine, and other resources from social tagging websites and search engines. There exist many works which have analyzed the importance of the *anchor text* like a source of information. Eiron and McCurley (2003) carried out a study which compared the usefulness of the anchor text and the content page in a web search task. This study shows several aspects of the anchor text, including the frequency of queries and the most frequent terms in the anchor text, content and titles.The anchor text has also been used for the site finding task (Craswell, Hawking, & Robertson, 2001) in TREC, but this source of information has generally been used as a way to represent a page, while in this work, the anchor text is used as a source of information in order to find a missing web page. We have investigated the performance of different approaches to extract the expansion terms from the mentioned sources. Some of these approaches are based on term frequencies, while others are language modeling approaches based on the differences between the probability distribution of terms in a collection and in the considered source of information.

After the term extraction step, a query is submitted to the considered search engine for every one of the expansions. Then, top ranked documents are retrieved in each case and the whole set is ranked. We have also investigated different approaches for this process. One of them is the vector space model, according to which the candidate documents are ranked by similarity with elements from the parent or the cached page. Similarity is measured using different normalized measures (*Cosine*, *Dice* and *Tanimoto*) between the corresponding vectors. Also, language modeling is considered again for this task, which builds a probabilistic language model for each document, and ranks documents according to the probability of the model that generates the query.

4. Evaluation methodology

It is very difficult to evaluate the quality of the candidate pages to replace a link and the usefulness of the different sources of information, if we try to recover broken links directly. Therefore, we have employed random web links which are not

really broken, and we called *pseudobroken* links. Thus, we had available the page at which they point and we were able to evaluate the recommendation of our system.

4.1. Selection of links to recover

To carry out the analysis, we took links from pages randomly selected by means of successive requests to www.random-website.com, a site that provides random web pages. Certain requisites were imposed to our test pages. The language of web sites was restricted to English, considering the following domains: ".com", ".org", ".net", ".gov" and ".edu". Pages with at least 250 words were required for using its text to characterize it. Moreover, the text was required to contain at least ten terms that were not stop words, that is, words that are so common that they are ignored in information retrieval tasks (e.g., articles, pronouns, etc.). We also demanded that the page had at least five potentially *analyzable links*, which means:

- The system analyzes external links, therefore links that pointed to the same site were discarded.
- The anchor text had to neither be empty nor be a number or a URL.
- If the anchor text was only composed of one character and it was a punctuation mark, this link was discarded.

Some preliminary experiments indicated that it is frequent to find pages in which over 95% of the links are alive and others in which most of them are broken. In our data, pages have an average of 17 links, although only 9% of them have more than 100 links and 57% do not exceed 10. When these pages have many links (e.g., 1000 links), they bias the results in some way or another. Because of this, we decided to limit to ten the number of links taken per page. This subset of links was randomly chosen among the analyzable links in the page. Finally, we have a collection of 900 links to study and evaluate our system in the next sections.

4.2. Automatic evaluation

We have used a combination of different tests to consider that a link has been recovered. First of all, it is verified if the URL from the page candidate to replace the link matches the analyzed link (remember that in this first analysis the link is not really broken). Nevertheless, we have found some cases in which the recovered page has the same content as the *pseudobroken* link, but different URL. Therefore, if the URLs do not match, we verify whether the web page content is the same. We have also found several cases in which the page content is not identical, but they were very similar: there are some small changes like advertisements, dates, etc. For this reason, if the contents are not exactly the same, we apply the vector space model (Manning, Raghavan, & Schütze, 2008), i.e we represent each page by a term vector and calculate the cosine distance between them (similarity). Other methods could be used for this task such as *shingling* techniques (Brin, Davis, & Garcıa-Molina, 1995), which take a set of contiguous terms or shingles of documents and compare the number of matching shingles, or similarity methods such as *PageSim* (Lin, Lyu, & King, 2006) or *SimRank* (Jeh & Widom, 2002). We finally decided to use the vector space model since *shingling* techniques focus on copy detection and *PageSim* or *SimRank* comes at a higher computational cost. We performed a study which can be observed in Table 1. This table presents the number of cases in which the broken links have been recovered, querying the search engine with the terms extracted from each anchor text.

As we can see in Table 1, if we use a similarity threshold higher than 0.9, 253 hits are recovered in the first position and 380 hits in the top 10. Lowering the similarity threshold under 0.9 adds very few additional links to the list of recovered ones. Besides, lowering this value the number of wrong results increases, sometimes recovering different pages. This means that using a lower threshold in the evaluation methodology, the system in some cases could incorrectly indicate that the link has been successfully recovered. For these reasons we have set the similarity threshold value to 0.9.

4.3. Manual evaluation

Each candidate web page for each broken link was judged by three different human judges (assessors). These judges were requested for every link to look for the first valid candidate page to replace it. Every assessor was given the broken link and a cached page of the missing page to evaluate if the candidate page was similar enough to replace the missing page.

Table 1
Results of searching the anchor text in Yahoo! in terms of the similarity threshold used. First column indicates the similarity threshold used. *1st pos.* represents the number of *pseudobroken* links recovered in the first position from the results of the search engine, and *1–10 pos.* the number of those recovered among the first 10 positions. *N.R.L.* represents the links that have not been recovered.

Sim. threshold	1st pos.	1–10 pos.	N.R.L.
0.9	253	380	536
0.8	256	384	529
0.7	258	390	521
0.6	262	403	504
0.5	266	425	478

Table 2
Analysis of not recovered (*N.R.L.*) and recovered links (*R.L.*) according to the type of anchor—with (*Named Entities*) and without (*No Named Entities*) named entities—and to the number of anchor terms. 4+ refers to anchors with four or more terms.

Terms	Type of anchor			
	Named entities		No named entities	
	N. R. L.	R. L.	N. R. L.	R. L.
1	102	67	145	7
2	52	75	91	49
3	29	29	27	45
4+	57	61	33	47
Total	240	232	296	148

Reviewers were provided a guidelines consisting of a list of similarity aspects, such as structure, content, title, and data, and a set of examples.

In our experiments, we restricted the broken links considered as "recovered" to those for which all assessors agreed that the system had provided the correct page.

The results of the manual evaluation were statistically analyzed and we conclude that there was a strong agreement between assessor judgments on the comparison of the first valid candidate page detected.

5. Information provided by the anchor text

In many cases the words which compose the anchor text of a hyperlink are the main source of information to identify the pointed page. To verify this theory, it can be observed the previous study about the similarity threshold illustrated in Table 1. This table presents the number of *pseudobroken* links that are recovered among the top ten hits returned by the search engine. We can observe that using a similarity threshold of 0.9, 41% of the links are recovered in the top ten results. In addition, 66% of the recovered links appears in the first position. These results prove that the anchor text is an important source of information to recover a broken link. Accordingly we use the terms extracted from the anchor text to compose a query that can be later extended with some additional terms from other sources.

5.1. Named entities in the anchor text

Sometimes the anchor terms provide little or not descriptive value. Let us imagine a link whose anchor text is "click here". In this case, finding the broken link might be impossible. For this reason it is very important to analyze these terms so as to be able to decide which tasks should be performed depending on their quantity and quality.

In this work we have carried out a recognition of named entities (persons, organizations or places) on the anchor text in order to extract certain terms whose importance is higher than the remaining ones. There exist several software solutions for this task, such as *LingPipe*, *Gate*, and *FreeLing*. There also exist multiple resources, like *gazetteers*. But according to our preliminary experiments, none of these solutions applied to the anchors have provided precise results, perhaps because we are working in a wide domain. In addition, the size of the anchor texts is too small for the kind of analysis usually performed by these systems.

Accordingly, we have decided to use the opposite strategy. Instead of finding named entities, we have chosen to compile a set of dictionaries to discard the common words and numbers, assuming that the rest of words are *pseudo named entities*, which we will from now on call named entities. Although we have found some false negatives, as for example the company *Apple*, we have obtained better results using this technique.

Table 2 shows the number of *pseudobroken* links recovered depending on the presence of named entities in the anchors, and on the number of anchor terms. We can see that when the anchor does not contain any named entity, the number of links that are not recovered is much higher than the number of the recovered ones, whereas both quantities are similar when there exist named entities. This proves that the presence of any named entity in the anchor favors the recovery of the link. The most prominent result is the very small number of cases in which the correct document is recovered when the anchor consists of just a term and it is not a named entity.[5] When the anchor contains named entities, even if there is only one, the number of retrieved cases is significant. Another fact that we can observe is that from two terms, the number of anchor terms does not represent a big change in the results.

[5] These few cases are usually URL domains with a common name, e.g., the anchor "Flock" has allowed recovering www.flock.com, the anchor "moo" the URL www.moo.com/flicker, etc.

6. Additional terminology for the web search

We consider the anchor text as the main source of information to recover a broken link, but we can extract additional terminology in order to complete the information provided by the anchor text. There are several sources of information in the context of a link that can help to recover it. These sources of information have different characteristics such as the amount of text or the relation to the broken link. In some cases we may have sources of information with a large amount of text such as the whole page in which the broken link appears, or just a few words such as a URL. In both cases, our main goal is to synthesize this information and to extract the minimum number of terms that represent a link in the most compact way. Thus, we have applied classical information retrieval techniques to extract the most representative terms. After removing the stop words, we generate a ranked term list. First terms of this list are used to expand the query formed by the anchor text, i.e., the query is expanded with each of those terms, and the top hits are retrieved in each case.

6.1. Different approaches for the extraction of terms

According to the source of information that we analyze, the quality of the extracted terms will be different depending on the used method. Thus, we have used two types of extraction methods: frequency-based and language-model-based approaches.

6.1.1. Frequency-based approaches to select terms

Frequency-based are the most simple approaches to select the most relevant expansion terms. We have considered two different criteria based on frequencies for term selection. The first one is the raw term frequency (TF) in the sources of information. There are some terms with very little or no discriminating power as descriptors of the source, despite they are frequent on it. The reason is that those terms are also frequent in many other documents of the collection considered or in any other web page in our case. To take into account these cases we have also applied the well-known *Tf-Idf* weighting scheme for a term, where *Idf(t)* is the inverse document frequency of that term:

$$Idf(t) = \log \frac{N}{df(t)}$$

being N the size of the collection, and $df(t)$ the number of documents in the collection that contain the term t. We have used an English *Wikipedia* articles dump[6] as reference collection.

6.1.2. Terminology extraction using language modeling

One of the main approaches to query expansion is based on studying the difference between the term distribution in the whole collection and in the subsets of documents that can be relevant for a query. One would expect that terms with little informative content have a similar distribution in any document of the collection. On the contrary, representative terms of a page or document are expected to be more frequent in that page than in other subsets of the considered collection.

One of the most successful methods based on term distribution analysis uses the concept of *Kullback–Liebler Divergence (KLD)* (Cover & Thomas, 1991) to compute the divergence between the probability distributions of terms in the whole collection and the specific considered documents. The most likely terms to expand the query are those with a high probability in the document, which is the source of terms, and low probability in the whole collection. For the term t this divergence is:

$$KLD_{(P_P,P_C)}(t) = P_P(t) \ \log \frac{P_P(t)}{P_C(t)} \quad (1)$$

where $P_P(t)$ is the probability of the term t in the considered page, and $P_C(t)$ is the probability of the term t in the whole collection.

Computing this measure requires a reference collection of documents. The relation between this reference collection and the analyzed document, is an important factor in the results obtained with this approach. Because of general web pages are very different to each other, and can deal with any topic, it is not easy to find an appropriate reference collection. Obviously we can not use the whole web as a corpus. To study the impact of this factor on the results we have used three different collections of web pages indexed with *Lucene* (Gospodnetic & Hatcher, 2004):

- *Enwiki*. This collection contains articles, templates, image descriptions, and primary meta-pages from an English *Wikipedia* dump. The size of this collection is around 3.6 million of documents.
- *Dmoz*. This collection is the result of a crawling process on the set of URLs from the *DMOZ Open Directory Project (ODP)*. The whole set is around 4.5 million of sites. We set the crawling depth to zero, so just a document has been retrieved from each site.
- *Dmoz URLs*. In this work, we have used the terms that compose a URL to extract relevant information. As the previous collections are based in common texts, we have compiled a new collection composed only of URL terms. Thus, we have parsed each URL in the 4.5 million of sites from *ODP* and we have indexed these terms, taking every URL as a document.

[6] http://download.wikimedia.org/enwiki/.

6.2. Sources of information

Both in the context of a link and the Web, there exist sources of information which provide relevant terms to recover a link that no longer exists. These sources of information have different characteristics both in the vocabulary component and length. For this reason, we consider the best approach to extract the most relevant information in a minimal number of terms for each of these sources.

6.2.1. Information provided by the URL

Apart from the anchor text, the URL is the only information directly provided by a link. Terms from a URL are very often highly representative of the content of the pointed page. In fact, most search engines take the URL terms into account in deciding whether a page is relevant to a query.

In addition, URL terms can be a very useful source of information whether a page has been moved within the same site, or the page is in another site but it maintains the same user id, service, application name, resource, etc.

A URL is mainly composed of a protocol, a domain, a path and a file. These elements are composed of terms that can provide rich information about the target page. Moreover in last years, because of the increasing use of search engines, there exist *Search Engine Optimization (SEO)* techniques that try to exploit the importance of URL terms in a request.

In order to select the most relevant terms from a URL, it is important to exclude terms that are very frequent in URLs, e.g., *free*, *online*, *download*, etc. To extract the most relevant terms we have applied a language-model-based approach. First of all, we have built a language model with terms from the *DMOZ URLs* collection. Afterwards, with help of this collection of URLs, we have used different term extraction methods in order to know the most relevant terms in a certain URL.

6.2.2. Information provided by the page that contains the link

The most frequent terms of a web page are a way to characterize the main topic of the cited page. This technique requires the page text to be long enough. A clear example of utility of this information are the links to personal pages. The anchor of a link to a personal page is frequently formed by the name of the person to whom the page corresponds. However, in many cases, the forename and surname do not identify a person in a unique way (Artiles, Gonzalo, & Sekine, 2007), specially if they are very common. If we perform a query to a search engine with only the forename and the surname, the personal page of this person probably will not appear among the first retrieved pages. However, if we expand the query using some terms related to that person, which can be extracted from his web page, then his personal web page will go up to the top positions.

6.2.3. Information provided by the context of a link

Sometimes anchor texts have not enough terms or these terms are too generic. Let us imagine a link whose anchor text is "nlp group". For this reason, text surrounding a link can provide contextual information about the pointed page and the required information to complete a more specific search. Moreover, Benczúr, Bíró, Csalogány, and Uher (2006) measured the relation between a link context and the pointed Web page, getting a good performance when the anchor text was extended with neighboring words. In our experiments, we have used several words around the anchor text to extend it, though we took into account `HTML` block-level elements and punctuation marks as in Pant (2003) and Chauhan and Sharma (2007). Although the format of Web documents changes over time and there are more and more links contained in a sentence or a paragraph (Blogs, *Wikipedia*, etc.), the main problem of this source is the limited number of times that you can find a context for a link. Although this source of information was only found at 30% of the analyzed links, when the context is available, its usefulness is very high.

6.2.4. Information provided by a cached page

One of the most useful sources of information is a cached page stored in a search engine or in an Internet archive (*Wayback Machine*). These websites, with different purposes, store the copy of a large number of Web pages with most of their available resources. The system first tries to retrieve a search engine cached page because it corresponds to the most recent version that can be obtained. However, if this cached page can not be retrieved, the system tries to find the latest cached version stored in the *Wayback Machine*. However, this source of information has two drawbacks: Web is bigger and bigger and servers of these websites are not able to store all the Web pages, although in our experiments we achieved a cached version from 50% to 60% of the analyzed links. On the other hand, the date of the latest version of a stored page may be old, even a year in the case of Internet archives. Although in the latter case, the essence of a site does not usually change in spite of the content has changed slightly. There are exceptions such as sites like *slashdot.com*, which varies almost completely over a few days. In the case of the cached pages, we extract the most relevant terms as in previous Section 6.2.2. These terms are used to expand the query formed by the anchor text.

6.3. Comparing different approaches for the extraction of terms

Fig. 2 shows the results obtained using *Frequency*, *Tf-Idf* and *KLD* for the extraction of the expansion terms from different sources of information. In case of URL as source of information, because of the number of terms that we can extract is very small, we have replaced the *Frequency* approach with a *KLD* approach based on a collection of URLs. According to the

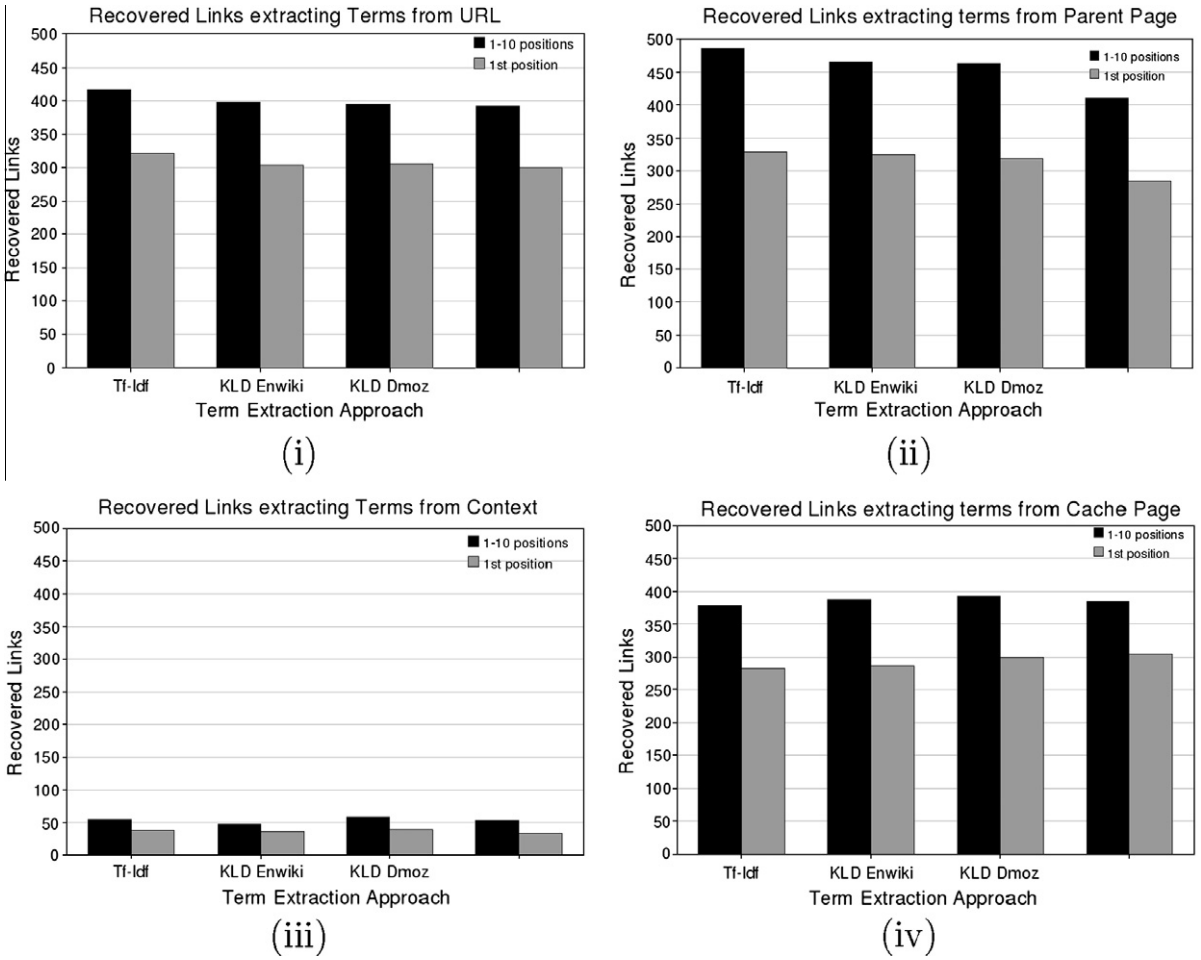

Fig. 2. Results expanding the query with terms from the (i) URL from the anchor of a link (ii) parent page, (iii) context of a link, and (iv) a cached page. The data of the figure indicate the number of cases in which the search engine has provided the required page in the top ten hits or in the first position, when the system expands a query with the extracted terms by applying different approaches.

obtained results depicted in Fig. 2i, the approach that has achieved the best performance is the *KLD* approach which has used the URL collection. It seems clear that a language model formed by the terms that usually appear in a URL is the most appropriate in this case. Both the URL and the page that contains the broken link (parent page) are two sources of information that can always be found, but the parent page, in many cases, has a lot of information not related to the broken link. Thus, the efficiency of the terms extracted from this source is limited. Fig. 2ii shows how the system recovers more links using the *Frequency* approach. As the extracted terms are not precise for this task, a simple method based on *Frequency*, that gets more generic terms, works better than another that gets greater specificity in terms like *KLD*.

Both *context* and *cached* page are two very important sources of information, since *context* usually contains the required terms to complete the information present in the anchor text, and a *cached* page shows a snapshot of what we are really looking for. The main problem in these two cases is that they are not always available but in the case of *context* is even more significant. In both cases, presented in Fig. 2iii and iv, the extraction method that achieves the best performance is *KLD* by using *Wikipedia* as reference collection, because the terms that can be extracted are highly accurate.

From these results we can conclude than the best term extraction approach for the recovery of broken links is *KLD*. We can observe that when we use *KLD*, the results obtained with the *Wikipedia* as reference collection are better (the total number of correct recovered pages). The reason is probably that this collection provides a wider range of topics, although the possible existence of some spam page in DMOZ could also distort the language model, and hence the term extraction.

In spite that *Frequency* is the method that obtains the best results by extracting terms from the parent page, there exists a limited relationship between the terms and the required page in many cases. Thus, this task is a random approximation with the aim of finding the closest terms to the link context or most related to the required page.

In addition, parent page content is sometimes not closely related to the page to recover, and thus refining the methods to select the more representative terms of the parent page does not improve the results.

Table 3
Analysis of the fraction of retrieved documents in the first position (S@1) and top ten positions (S@10), according to the use of query expansion.

Analysis	S@1	S@10
No Expansion (*anchor text*)	**0.42**	0.51
Expansion (*URL terms*)	0.36	0.46
Expansion (*parent page*)	0.36	**0.54**
Expansion (*context*)	0.04	0.06
Expansion (*cached page*)	0.33	0.44

Accordingly, in the remaining experiments we have used *KLD* with the English *Wikipedia*, as reference collection, for extracting terms from the *context* and *cached* page; *KLD* with a URL terms collection for extracting terms from the URL; and the *Frequency* method for extracting terms from the parent page.

6.4. Effect of query expansion in the relationship between success@1 and success@10

In order to study the effect of query expansion methods in the relationship between Success at one (S@1) and Success at ten (S@10), a new comparison is presented. Success at one measures in how many cases the first recommended document is considered an useful replacement of the missing page. In the case of Success at ten, only the first ten recommended documents are considered. In Table 3 we can observe that expansion considerably increases the number of links recovered in the first ten positions (S@10). In spite of this, the number of recovered links in the first position (S@1) is lower when any expansion approach is used. Accordingly, we think that the most suitable mechanism is to recover with and without a expansion approach, and later ranking the whole set of results to present the user the most important ones in top positions.

7. Using the web to track a page trail

We have established the anchor text and the terms extracted from several sources in the context of a link as the main sources of information to recover a broken link. But we can also use other resources from the Web infrastructure, such as available utilities of search engines and information provided by social tagging websites.

Web offers us new resources every day that can be used to obtain more information. Our purpose is to use the resources available in the Web for obtaining more information about those pages that no longer exist, but for which there still exists information in the Web. Search engines (*Google*, *Yahoo!*, *Bing*, etc.) offer applications and services every day more interesting and useful, and they can help us to recover a missing page. Another growing phenomenon are the social tagging systems, where a huge community is willing to generate information in a collaborative way. In this case, the point of view of a group of people on a social website can be a very important source of information to discover something new about a web page.

7.1. Web search utilities of search engines

Search engines work every day for giving us access to information in a simple way. Furthermore, the fact that only a few companies have the ability to index the whole Web (at least the vast majority) means that some of the services they offer have a great usefulness because of their global scope.

Recently, the open search web services platform of Yahoo! (BOSS[7]), offers a new feature through its API development: *Key Terms*. The technology used in *Key Terms* is the same used for Search Assist,[8] which provides search suggestions and enables searchers to explore concepts related to the query. The *Key Terms* feature uses term frequency and positional and contextual heuristics to return ordered lists that describe a web page. Each result returned for a query includes associated meta-data of up to 20 terms that describe that result.

We query to Yahoo! with the URL of a broken link to get an ordered list with these terms, and we use the first N terms to expand the anchor text of the broken link. Table 4 shows an use case after querying to Yahoo! with www.sigir.org, the website of the *Special Interest Group on Information Retrieval*. The search engine provides useful terms, such as *ACM*, *information retrieval*, *web search* and *conference*. In addition, because of the *SIGIR 2009 conference* was held in *Boston*, the search engine provides this location.

7.2. Social tagging systems

Nowadays, the phenomenon *Social Tagging* is revolutionizing the searches in Internet, since behind of websites such as *del.icio.us* or www.bibsonomy.org there exists a community of users that classify pages using labels and also that share their bookmarks with other users. These websites store a huge collection of tagged web pages and they become a very important

[7] http://developer.yahoo.com/search/boss/.
[8] http://tools.search.yahoo.com/newsearch/searchassist.

Table 4
Top ten terms recovered when Yahoo! is queried with www.sigir.org, by using the *Key Terms* feature.

www.sigir.org
SIGIR
ACM SIGIR
Information retrieval
Special Interest Group on Information Retrieval
Conference
Retrieval field
SIGIR Forum
Web search
Text search
Boston

source of information. In addition, there exist algorithms (Bao et al., 2007a) that have shown that this information is useful to improve web searches.

Bao et al. (2007b) explored the use of social annotations in *del.icio.us* to improve web search. They pointed out that annotations are usually good summaries of corresponding web pages and count of annotations indicates the popularity of web pages. Yanbe, Jatowt, Nakamura, and Tanaka (2007) proposed combining the widely used link-based ranking metric with the one derived using social bookmarking data. They also introduced the *SBRank* algorithm which captures the popularity of a page and implemented a web search application by using this social tagging. Noll and Meinel (2008) proposed a personalization technique which separates data collection and user profiling from the information system whose contents and indexed documents are being searched and used social bookmarking and tagging to re-rank web search results.

Our system also considers this source of information and searches in social tagging sites such as *del.icio.us* or www.bibsonomy.org to extract a sorted tag list for every missing page according to the number of users that have used this label to tag that page. Afterwards, first N tags are used to expand the anchor text of the broken link.

8. Ranking the recommended links

After retrieving a set of candidate pages querying to search engines, the system needs to present the results to the user in decreasing order of relevance. To calculate this relevance we have considered different sources of information related to the broken link and different sources of information from the candidate pages. In order to establish the best ranking function for the candidate pages, we performed an analysis to compare different similarity approaches and elements from parent, cache and candidate pages.

8.1. Vector space model and co-occurrence coefficients

Within the comparative study among different ranking approaches, we have used the vector space model Manning et al. (2008) to represent the documents and several co-occurrence Rijsbergen (1977) coefficients to rank them. Methods based on term co-occurrence have been used very frequently to identify semantic relationships among documents. In our experiments we have used the well-known *Tanimoto*, *Dice* and *Cosine* co-occurrence coefficients to measure the similarity between the vectors representing the reference document D_1 and the candidate document D_2:

$$\text{Tanimoto } (\vec{D}_1, \vec{D}_2) = \frac{\vec{D}_1 \vec{D}_2}{|\vec{D}_1|^2 + |\vec{D}_2|^2 - \vec{D}_1 \vec{D}_2} \tag{2}$$

$$\text{Dice } (\vec{D}_1, \vec{D}_2) = \frac{2\vec{D}_1 \vec{D}_2}{|\vec{D}_1|^2 + |\vec{D}_2|^2} \tag{3}$$

$$\text{Cosine } (\vec{D}_1, \vec{D}_2) = \frac{\vec{D}_1 \vec{D}_2}{|\vec{D}_1||\vec{D}_2|} \tag{4}$$

8.2. Language model approach

We have also considered another model to represent the documents and rank them. Thus, we have used a language modeling to represent the documents and a non-symmetric probabilistic measure to rank the set of candidate documents. In this case we look at the difference between two probability distributions, computing the *Kullback–Leibler Divergence (KLD)* between two documents:

$$KLD(D_1 \| D_2) = \sum_{t \in D_1} P_{D_1}(t) log \frac{P_{D_1}(t)}{P_{D_2}(t)} \quad (5)$$

where $P_{D_1}(t)$ is the probability of the term t in the reference document, and $P_{D_2}(t)$ is the probability of the term t in the candidate document.

8.3. Comparing different approaches for the ranking of candidate pages

We have applied the approaches described above to different elements from the parent or the cached pages and from the candidate page. Specifically, we have studied the similarity among the following pairs of elements: (i) parent anchor text and candidate title, (ii) parent anchor text and candidate content, (iii) parent content and candidate content, and (iv) cache content and candidate content.

In addition to these comparisons, we also used the anchor text and the *snippet* of the candidate document, but the results did not improve those showed in Fig. 3. We can observe that in Fig. 3i and ii the results obtained with *KLD* are worse than those obtained with the co-occurrence measures, especially in Fig. 3i. In these figures we are studying the similarity between a very short text, the anchor text, and other short text which is the title of the candidate page (Fig. 3i) or the parent page content (Fig. 3ii). On the contrary, *KLD* performs better than the co-occurrence measures in Fig. 3iii and Fig. 3iv, where we are measuring similarity between the content of two pages, the parent or the cached page, and the candidate page. So we can conclude that *KLD* performs better that the co-occurrence methods only if it is applied to texts long enough, such as the page content. In Fig. 3iv we can observe that, as expected, the obtained results ranked by similarity with the cached

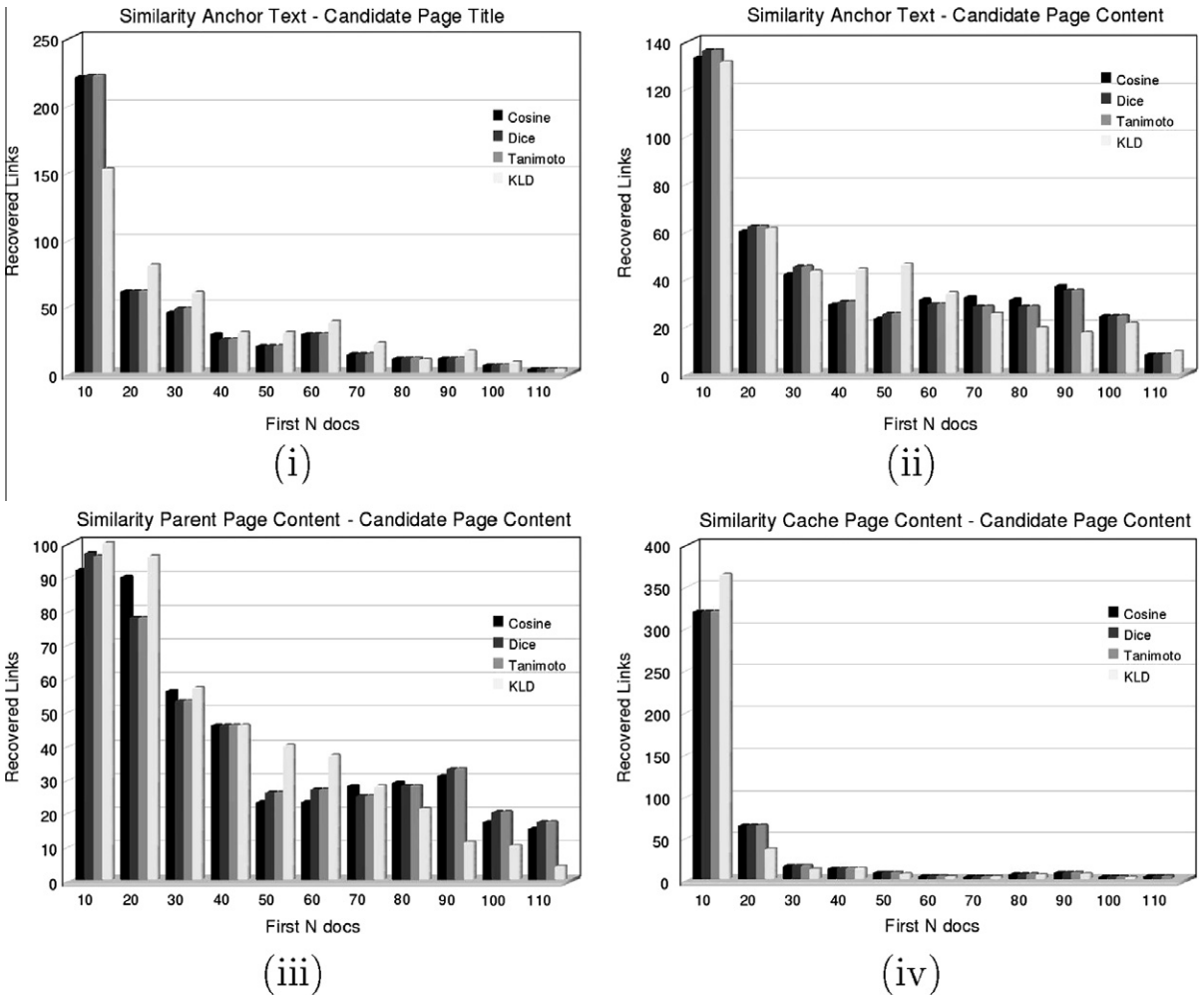

Fig. 3. Results of different approaches (Cosine, Dice, Tanimoto and Kullback–Liebler Divergence) applied to measure the similarity between: (i) the Anchor Text of the broken link and the Title of the candidate page, (ii) the Anchor Text of the broken link and the Content of the candidate page, (iii) the Content of the page where is the broken link and the Content of the candidate page, and (iv) the Content of the Cached page of the broken link and the Content of the candidate page. Results show the position of the best page recovered by the system after the ranking.

page are the best. However, in many cases this page is not available (near a 40–50%). Comparing the remaining cases we can observe that the best results are obtained applying the similarity between the anchor text of the broken link and the title of the candidate page, and using co-occurrence methods. According to these results, if we can retrieve a cached page, we will apply *KLD* to rank the candidates pages with respect to the cached page. Otherwise, we will use the similarity between the anchor text and the candidate page title measured with a co-occurrence method, such a *Dice*, which performs slightly better in some cases.

9. Usage statistics and sources of information effectiveness

Extraction and use of terminology, as well as the recovery of the results from the search engine and subsequent ranking means a significant computational cost for the system. For these reasons, we present a comparative study on different parameters which affect this cost. The system will use 900 links to measure the contribution of each source of information. Furthermore, sources of information have different availability and effectiveness degrees. For this reason, we present a study of these characteristics from every source of information.

An important issue to investigate is the trade-off between the amount of collected information to recover the broken links, and the required time to do it. We can expect that, in general, the more information available, the better the recommendation. However it is important to know the cost incurred for each increment of the collected data.

9.1. Impact on the results according to the number of terms and hits

The amount of information collected mainly depends on two parameters: the number of terms (extracted from several sources of information) used to expand the query (anchor text), and the number of hits taken from the results of the search engine for each query. We performed a set of experiments to evaluate how these parameters affect to the results.

Fig. 4 shows the number of recovered links for different numbers of terms used in the expansion. In this figure are shown the results for the different sources of information employed by the system. In these experiments, ten hits are taken from the results of the search engine. Obviously, the approach without expansion (“Anchor Text”) is not affected by the number of terms used in the expansion. But it is interesting to notice that an expansion with parent terms beats the “Anchor Text” approach when expanding with six or more terms. The main reason is that using more terms increases the probability of expanding with the appropriate term. In addition, it is remarkable the ascending slope when parent terms are used, since these terms are very imprecise. The “All” approach is a combination of all sources of information, but this approach is not the addition of the results from all the sources information since some of these sources recover the same links. Apart from this, we can observe that the number of recovered links increases with the number of expanding terms for all the sources of information. However, the improvement is quite small from 10 to 25, specifically in the “All” approach. Thus, according to this experiment, we can conclude that the best number of terms used to expand the query is 10.

Fig. 5 shows the number of recovered links when different numbers of hits are taken from the results provided by the search engine. This figure presents the results for the different sources of information employed by the system. Ten terms

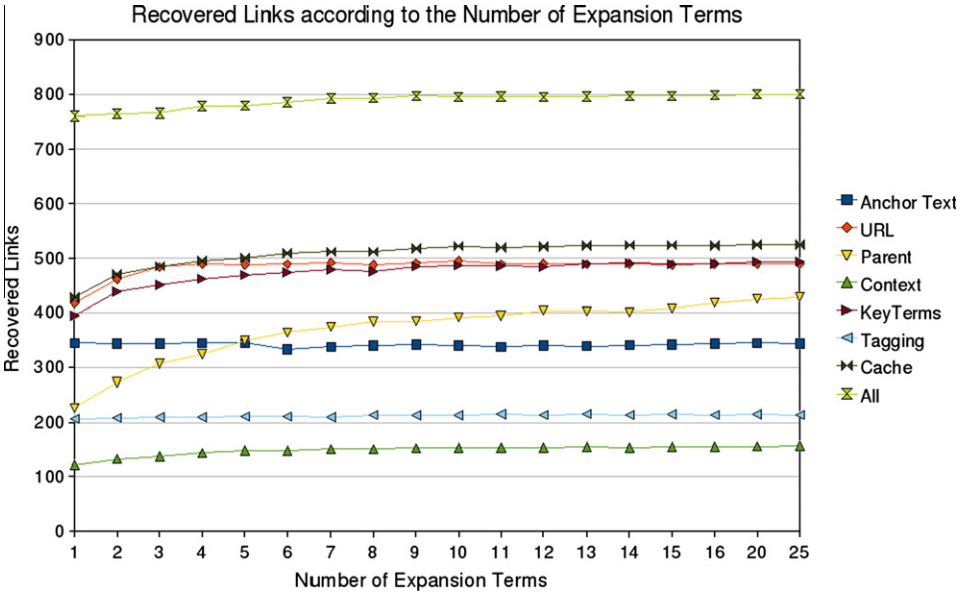

Fig. 4. Recovered Links according to the number of terms used to carry out each query expansion. This figure shows the results obtained by using different sources of information.“All” method stands for a combination of all the sources of information.

are used to expand the query formed by the anchor text in these experiments. It is remarkable the sharp slope when seven URL terms are used, and the low recovery when a single term is used to expand the query. We can observe that the number of recovered links increases with the number of taken hits, though the improvement is much smaller from 10 to 15 hits. In addition, since the "All" approach is a combination of all sources of information and there is not a clear improvement from 10 to 15, we can conclude that the best number of hits taken from the results of the search engine is 10.

9.2. Effectiveness of sources of information

There exists a great diversity in the availability of information sources previously discussed. Sources such as the anchor text, the URL of the missing page or the page containing the broken link are always available. However, a cached page of the missing page or the context of the link that we are trying to recover, are only available in some cases. In addition from the obtained resources from the Web, search engines do not have a list of *key-terms* for each page neither all pages on the Internet are labeled in a social tagging website.

On the other hand and independently of the availability of sources of information, its effectiveness varies greatly from one source to another. That is, there are sources of information that when available, retrieve the missing page over a 90%, while the recovery degree with other sources of information is below 60%.

Fig. 6 shows the availability, the number of recovered (1–10 positions) links and the recovered links in the first position of the returned hits by the search engine, according to the source of information employed.

Fig. 6 depicts the great effectiveness of *Context*, *Cache* and *Tagging* terms. Sometimes the anchor text is too generic and ambiguous, so that some sources can easily find a term that is the key to break this ambiguity as in the case of *Context*, *Cache* and *Tagging* terms. It is remarkable the high level of recovery obtained by *Cache* and *Keyterm* terms. These sources are able to expand the query with very precise terms. In addition, as we mentioned above, there are some sources of information that are available in all the cases. These sources are *Anchor Text*, *Parent* and *URL*. Another important factor to have into account is the precision of each source, in such a way that the more links recovered from the first hit extracted from the search engine, the more precise the source. In addition, the more precise the source of information, the faster the system. Thus, *Context* and *Tagging* are presented in the Fig. 6 as the more precise sources.

9.3. Time impact according to the number of terms and hits

Fig. 7 shows the average execution time to recover all the links using the previously defined sources of information and different number of hits and terms. Results show the recovery according to the number of taken hits, and the number of term used for expansion. In the experiments with different set of hits, the number of terms was fixed to 10, and in the experiments with different sets of terms, the number of hits was fixed to 10. We can observe that, as expected, the time increases with both, the number of hits and the number of terms. Though we expect to reduce this execution time with different improvements in the current implementation, according to the results of this section, in the experiments showed in this work we

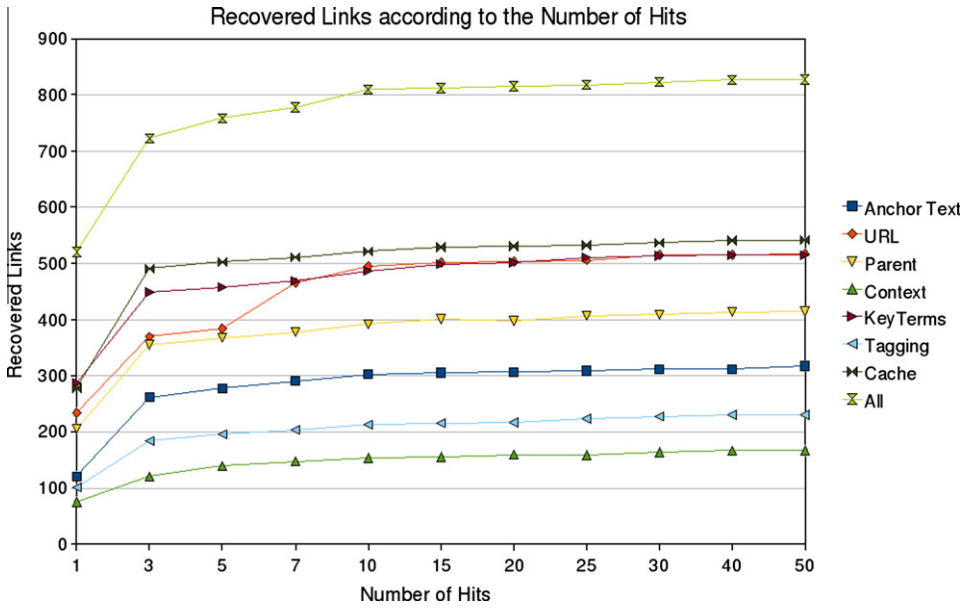

Fig. 5. Recovered Links according to the number of hits from the search engine used to carry out each query expansion. This figure shows the results obtained by using different sources of information."All" method stands for a combination of all the sources of information.

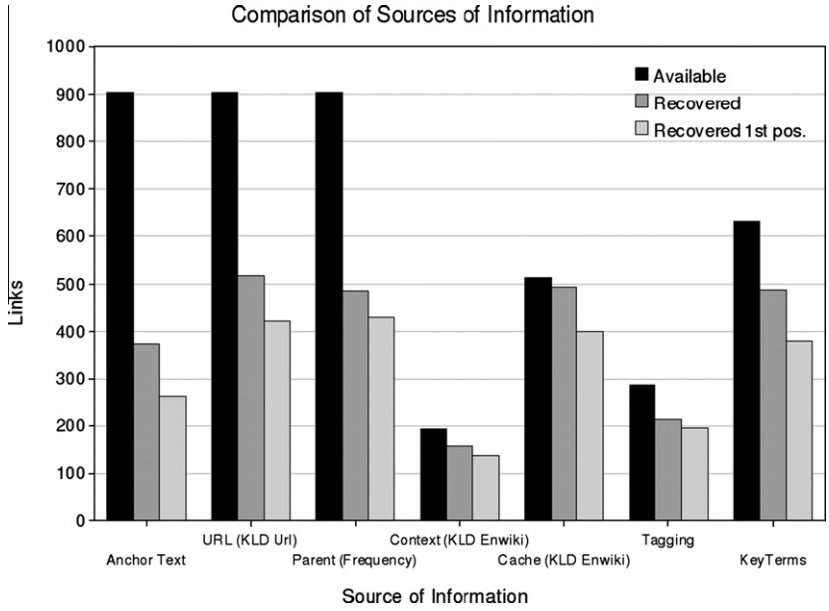

Fig. 6. Availability, Recovered Links and Recovered Links in the first position according to the used source of information.

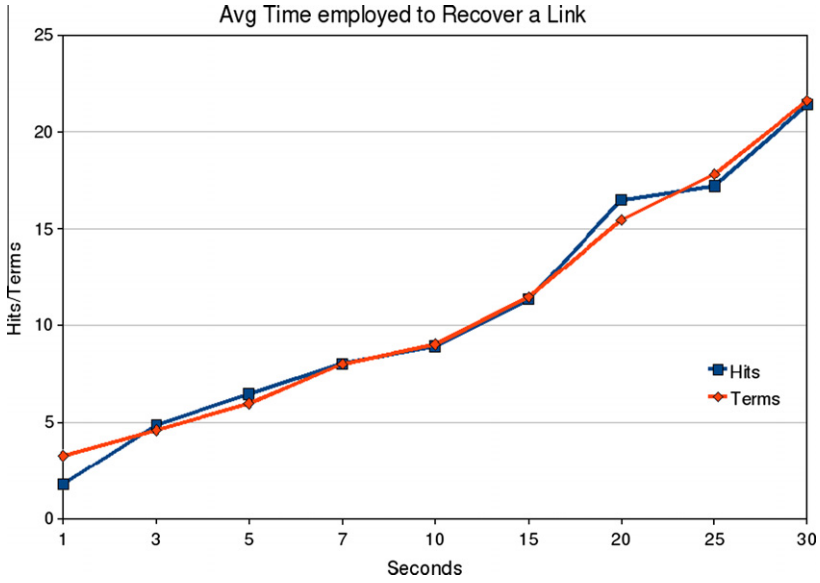

Fig. 7. Average Time required to recover a link according to the number of terms and hits used to carry out each analysis.

have fixed to 10 the number of taken hits of each search, and the number of terms used for the expansion, as a trade-off between the improvement in the performance and the execution time.

This paper presents a system that can be applied in different use cases. The one closer to the experiments corresponds to an online web application where a user can obtain a real-time response to his query. However, there are other use cases where time is not as important, such as a system that works offline. This is the case of a website that runs the recovery system periodically, looking for new broken links and a possible replacement for every one of them, sending that information to the user by email later. Thus, an average time of about ten seconds for every broken link would not be a problem for such a task.

Moreover, returning to the use case of the online web application, we have investigated the highest degree of quality of the results that we can obtain for the problem. However, not all the sources of information are equally useful for the

Table 5
Comparison between different search engines used in the retrieval system for broken links. First row shows the links recovered exclusively using the method without expansion (No Expansion Approach), second row shows the links recovered exclusively through the use of query expansion (Expansion Approach) and third row shows the overall performance of the system.

	Bing	Google	Yahoo!
No Expansion Approach (NE\E)	103	88	114
Expansion Approach (E\NE)	64	40	72
Recovered Links (NE ∪ E)	416	401	484

recovery. Thus, the time could be reduced significantly by excluding some of the sources of information the system uses. For instance, *context*, *tags* and *keyterms* could be excluded without significantly affecting the results.

In general, the system is designed using a parametric approach which allows either improving the response time by reducing the number of sources of information, the number of expansion terms and the number of hits retrieved, or optimizing the precision of the results using a larger number of resources.

9.4. Search engines

In our system, a search engine is used to be queried according to the method described above. Because our system depends on a proprietary technology, we conducted a comparative performance analysis among three of the most used search engines. We have used a collection of 750 broken links to perform these experiments.

Table 5 shows the number of recovered links using each search engine. In addition, it can be seen the links recovered by using the method without expansion (NE), but which are not recovered when expanding the query. It also shows the links recovered exclusively through the use of query expansion (E). For this table, the experiments were performed with a reduced query expansion approach, using only terms from the page where the broken link was found.

According to the data shown in Table 5, Yahoo! gets the best results. Despite this search engine shows a better performance in the recovery of links, the main reasons for using it in our system have been the limitations that the APIs of other search engines present and the possibility of obtaining additional information, such as the "Key Terms" data from *Yahoo!*, with the same query.

10. Algorithm for automatic recovery of links

Results from the analysis described in previous sections suggest several criteria to decide in which cases there is enough information to try the retrieval of the link and which sources of information to use. According to them, we propose the recovery process which appears in Fig. 8. First of all, it is checked whether the anchor number of terms is just one (`length(anchor) = 1`) and whether it does not contain named entities (`NoNE(anchor)`). If both features are found, the retrieval is only attempted if the missing page has an available cached version (`InCache(page)`), and therefore we have reliable information to verify that the proposal presented to the user can be useful. Otherwise, the user is informed that the recommendation is not possible (`No_recovered`). If the page has an available cached version, then the recovery is performed, expanding a query formed by the anchor terms with terms extracted from the cache, context and URL using *KLD*, and from the *Tags* and *Keyterms* otherwise. Then the results are ranked (by similarity between the candidate page and the cached page computed with *KLD*) and only if any of them is sufficiently similar to the cache content (`similarity(docs,cache(page) > 0.9)`), the user is recommended this list of candidate documents. In the remaining cases, that is, when the anchor has more than one term or when it contains some named entity, the recovery is performed expanding the query with the terms from the context and URL (applying *KLD*), and from the *Tags* and *Keyterms* otherwise. If the page has an available cached version, the query is expanded with terms from the cached page using *KLD* or with terms from the page that contains the link (using *Frequency*) otherwise. After that, all documents are grouped and ranked according to the cached page (`rank(docs, cache_content_KLD)`) if it is available, or according to the similarity between the anchor text and the title of the candidate page applying the *Dice* co-occurrence coefficient (`rank(docs, anchor_title_Dice)`) otherwise.

10.1. Results applying the system to broken links

We have applied this algorithm to links that are really broken. We have only used those that had an available stored version of the missing page. The reason is that only in this case, we can evaluate the results in an objective way. In some cases, the system is able to gather at most 700 candidate pages (7 sources of information × 10 terms × 10 hits) for every broken link. Then, according to the algorithm defined above, the system ranks these candidate pages and shows to the user the best 100 candidate pages as a ranked list. In order to evaluate the system, we performed both a manual evaluation by three human judges and an automatic evaluation according to the methodology previously proposed in this work.

To evaluate the recovery system of broken links deeply, we have used two collections with different characteristics. On the one hand, we selected at random a set of active pages to simulate the behavior of the system in a real environment, given

```
if length(anchor) = 1 and NoNE(anchor) then
  if InCache(page) then
    docs = web_search(anchor + cache_KLD)
    docs = docs + web_search(anchor + URL_KLD + context_KLD)
    docs = docs + web_search(anchor + tags + keyterms)
    rank(docs, cache_content_KLD)
    if similarity(docs, cache(page) > 0.9) then
       user_recommendation(docs)
    else
       No_recovered
  else
    No_recovered
else
  if InCache(page) then
    docs = docs + web_search(anchor + cache_KLD)
    docs = docs + web_search(anchor + URL_KLD + context_KLD)
    docs = docs + web_search(anchor + tags + keyterms)
    rank(docs, cache_content_KLD)
  else
    docs = docs + web_search(anchor + parent_FREQ)
    docs = docs + web_search(anchor + URL_KLD + context_KLD)
    docs = docs + web_search(anchor + tags + keyterms)
    rank(docs, anchor_title_Dice)
  user_recommendation(docs)
```

Fig. 8. Links Automatic Recovery Algorithm for broken links.

Table 6
Fraction of recovered broken links (best candidate whose content is very similar to the missing page) according to his cache similarity, with the best candidate somewhere in the top N (S@N).

	Recovered Broken Links	
	Automatic evaluation	Manual evaluation
S@10	0.37	0.38
S@20	0.55	0.57
S@50	0.72	0.75
S@100	0.78	0.81

by a heterogeneous set of pages that are available online. On the other hand, we have tried to test the performance of the system on a set of very old broken links. In this way, we have conducted a set of experiments that reflect in a realistic way the typical conditions to which the system could face.

10.2. Random website collection

Following the methodology presented in Section 4.1, we took every broken link from pages randomly selected by means of successive requests to www.randomwebsite.com, a site that provides random web pages. To evaluate the results in an objective way, we have only used pages that had an available cached version. The cached version is compared to every candidate page to evaluate the quality of the results. However, we do not use this cached page as a source of information.

Results are shown in Table 6 where the measure used was the fraction of recovered links with the best candidate to replace a broken link somewhere in the top 10 candidate pages proposed by the system (“Success at 10” or S@10). In addition, we have considered that a broken link is recovered when a valid candidate page is present among the 100 first documents proposed by the system. Valid pages are those similar enough to the cached one according to the judge opinion or the similarity measure obtained.

According to the manual evaluation, the system recovered 1619 from 1998 links (81% of the total links). On the other hand, the system recovered 78% of the total links using the automatic evaluation. Table 6 shows the ranking obtained for these recovered links. We have verified that in some cases the original page was found (it had been moved to other URL). In some other cases, the system retrieved pages with very similar content. Besides, the system is able to provide useful replacements for web pages among the first 10 positions in 47% of the recovered links, and among the 20 first ones in 71% of the cases.

10.3. UK-domain collection

With the aim of completing a set of experiments that shows the performance of the system, we have also tested the system on a set of very old broken links. To do this, we have used a different collection. A characteristic of this new corpus is that

Table 7
Fraction of recovered broken links (best candidate whose content is very similar to the missing page) according to his cache similarity, with the best candidate somewhere in the top N (S@N).

	Recovered Broken Links	
	Automatic Evaluation	Manual Evaluation
S@10	0.31	0.36
S@20	0.47	0.56
S@50	0.59	0.69
S@100	0.62	0.73

the missing page is available in the collection, although our system has not used this source of information for the recovery of the missing page. The main drawback is that the available version of the missing page is more than three years old. This obsolescence may cause some failures in the recovery because the pages that could replace the broken links have been disappearing gradually.

This new corpus is a publicly available Web Spam collection (Castillo et al., 2006) based on crawls of the *.uk* Web domain done in May 2007. This reference collection was tagged by a group of volunteers labeling domains as "non-spam", "spam" or "borderline". In our experiments, we restricted the dataset using only domains labeled as "non-spam". We imposed this restriction because spammers use techniques to artificially modify the content of web pages and this could introduce a bias in the results.

From the results shown in Table 7, and according to the manual evaluation, we can see that the system recovered 449 from 615 links (73% of the total links). On the other hand, the system recovered 62% of the total links using the automatic evaluation. Table 7 shows the ranking of these recovered links. Furthermore, the system is able to provide useful replacements for web pages among the first 10 positions in 49% of the recovered links, and among the 20 first ones in 76% of the cases.

We have verified that in most cases the system recommended pages with very similar content. In the remaining cases, the system is able to propose pages whose content has changed over time, but have a similar structure and some signs of unique identity, generally Key Terms or sentences.

Comparing these results with those obtained with the previous collection, it can be seen from Table 7 that a lower number of recovered links is achieved. However, this fact can be explained by considering that the pages used in the evaluation methodology are more than three years older than the current candidate pages. Moreover, if we consider the difference between manual and automatic evaluation, there is a greater difference from the previous collection. This fact can be explained taking into account that a judge does not check literally the content but the structure, the most relevant information and some signs of identity. It is very important to note that although the system has only managed to retrieve a cached version in 4% of cases, it has been able to obtain a version stored on the Web Archive in 58% cases.

Furthermore, we have used the UK-Spam collection to identify broken links in the ClueWeb09[9] collection, with the main goal of doing an evaluation of the algorithm that is entirely reproducible using only algorithms known to the scientific community. We have used the default options of the Indri search engine[10] on an index of only the English portion of the Category A ClueWeb09 dataset which consists of roughly the first 500 million English web pages. The system has recovered 54% of the total number of links (332 out of 615 links) with manual evaluation and 51% with automatic evaluation.

10.4. Comparison with other approaches

Although many studies have addressed the issue of broken links, only the work of Morishima et al. (2009a) has shown performance results. The problem that the authors try to solve is different from ours, because they focus on pages that have been moved to other location, while we try to recover all types of links, included disappeared pages. Furthermore, results from both systems have been obtained for different collections. For these reasons, the comparison between both works should be taken with a grain of salt. They represent complementary methods that address problems of different nature. Morishima et al. collected 127,109 links contained in Web sites of nine university domains, and selected outgoing links from them. Their system found 858 broken links in the collection and 259 of them were identify as having been caused by page movement. The data shown in Table 8 corresponds only to the recovery of the latter links.

According to the results shown in Table 8, and considering the manual evaluation, it can be seen how *PageChaser* obtained a 74.9%, while our system recovered a 81% on the random web links collection and 73% on the UK-domain collection. These results indicate that a combination of both methods would give rise to an orthogonal system that could improve the results from both systems separately.

[9] http://boston.lti.cs.cmu.edu/Data/clueweb09/.
[10] http://lemurproject.org/indri.php.

Table 8
Comparison of the recovery system of broken links by Morishima et al. (2009a) (PageChaser) and in this work. Data from *PageChaser* are shown using a manual evaluation. Data in our system are shown using two collections and two different types of evaluation.

Approach/Collection	Total links	Recovered links	Percentage (%)
PageChaser (Manual Evaluation)	259	194	74.9
Random Web Links (Automatic Evaluation)	1998	1559	78
Random Web Links (Manual Evaluation)	1998	1618	81
UK-Domain Collection (Automatic Evaluation)	615	381	62
UK-Domain Collection (Manual Evaluation)	615	449	73

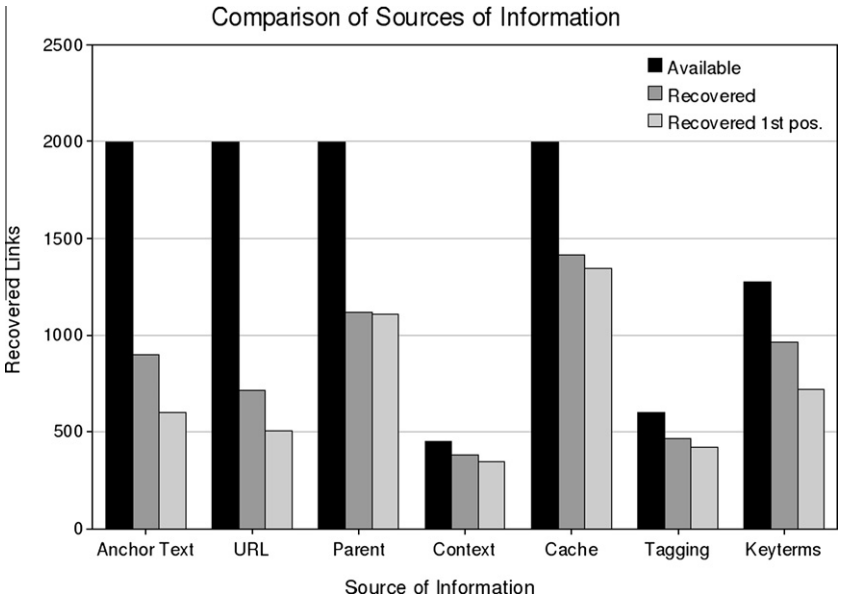

Fig. 9. Availability, Recovered Links and Recovered Links in the first position according to the used source of information.

10.5. Influence of sources of information on the recovery of broken links

Section 9.2 presented a comparative study which showed both availability and amount of links that each source of information had retrieved. Fig. 9 illustrates the same comparative study, but in this case for the broken links considered in Section 10.2. It is important to analyze each source of information separately.

Looking first at the *Cache* source, it can be observed note that its effectiveness (available links/links recovered) is lower though it preserves the precision (recovered links/links recovered in first position).

It is also noticeable that the source *Parent* has become the second best source of information after *Cache* source. One of the main reasons is that the effectiveness of the terms of the URL has been reduced very much, since now the links are really broken. This is the case when you can really assess the effectiveness of the terms of the URL. As for *Keyterms*, it can be seen that their effectiveness has also been increased and they become a really relevant source of information. On the other hand, it is also observed an increase in the effectiveness of the *anchor text* terms which could be somewhat hidden in the experiments with links were not really broken. Finally, the terms extracted from the *context* and *tags* maintain their effectiveness.

11. Conclusions

In this work we have analyzed different sources of information that we can use to carry out an automatic recovery of web links that are not valid anymore. Results indicate that the anchor terms can be very useful, especially if there are more than one and if they contain some named entity. We have studied the effect of using different terminology extraction approaches on sources of information, such as the page that contains the link, the surrounding anchor text, the URL anchor terms, and a cached page stored in some digital library (search engine or web archive). We have also employed resources from the Web infrastructure for obtaining more information about those pages that no longer exist, but for which there still exists information in Internet. Specifically, the system has extracted terms from recently Web technologies such as social tagging systems and available tools from search engines.

This study has shown that the results are better when the query is expanded. Thus, the query expansion reduces the ambiguity that would entail the limited quantity of anchor terms. We have compared different methods for the extraction of terms to expand the anchor text. Experiments have shown that the best results are obtained using a language modeling from sources of information such as the cached page, the context and the URL of a link. On the other hand, a frequency approach should be used to extract terms from the page that contains the link.

We have decided to combine all terminology extraction methods and use a novel ranking approach, in order to present to the user the candidate pages as a ranked list. We have also carried out a comparative study among different ranking approaches by using several sources of information from the source and target pages; We have used several co-occurrence coefficients and a divergence approach based on language models. Thus, the best ranking is obtained applying *KLD* between language models from the cached page and the candidate page, if a cached page is available, and applying a co-occurrence method as *Dice* between the anchor text of the broken link and the title of the candidate page, otherwise.

To evaluate the recommender system, we have developed a novel methodology without resorting to user judgments, thus increasing the objectivity of the results, including two web page collections with true broken links to test the full system. Through our proposed evaluation methodology, we have been able to determine the optimal amount of both terms used for expanding a query and retrieved hits from the search engine. In addition, this methodology has allowed us to perform an empirical evaluation of the availability and effectiveness of sources of information. We also performed a manual evaluation by three human judges to complete the evaluation methodology and to receive feedback about the system performance by assessors.

The result of this analysis has allowed us to design a strategy that has been able to recover a page that could replace the missing one in 81% of the broken links (1618 of 1998) in the case of a random web page collection and in 73% of the broken links (449 of 615) in the case of a UK-domain collection. Moreover, the system is able to provide 47% and 49% from these recovered links in the top ten of the results, and among the top 20 in 71% and 76% using a random web page and a UK-domain collection, respectively.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation within the project Holopedia (TIN2010-21128-C02-01) and the Regional Government of Madrid under the Research Network MA2VICMR (S2009/TIC-1542).

References

Amati, G., Carpineto, C., Romano, G. (2001). Fub at trec-10 web track: A probabilistic framework for topic relevance term weighting. In Proceedings of the 10th text retrieval conference TREC 2001, pp. 182–91.

Artiles, J., Gonzalo, J., Sekine, S. (2007). The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In: Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007). Association for computational linguistics, Prague, Czech Republic, pp. 64–69. <http://www.aclweb.org/anthology/W/W07/W07-2012>.

Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., & Su, Z. (2007a). Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web* (pp. 501–510). New York, NY, USA: ACM.

Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., & Su, Z. (2007b). Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web* (pp. 501–510). New York, NY, USA: ACM.

Benczúr, A. A., Bíró, I., Csalogány, K., & Uher, M. (2006). Detecting nepotistic links by language model disagreement. In *WWW '06: Proceedings of the 15th international conference on World Wide Web* (pp. 939–940). New York, NY, USA: ACM.

Brin, S., Davis, J., & García-Molina, H. (1995). Copy detection mechanisms for digital documents. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on management of data* (pp. 398–409). New York, NY, USA: ACM.

Castillo, C., Donato, D., Becchetti, L., Boldi, P., Leonardi, S., Santini, M., et al (2006). A reference collection for web spam. *SIGIR Forum, 40*(2), 11–24.

Chauhan, N., Sharma, A. K. (2007). Analyzing anchor-links to extract semantic inferences of a web page. International conference on Information technology, pp. 277–282.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York, NY, USA: Wiley-Interscience.

Craswell, N., Hawking, D. (2001). Overview of the TREC-2001 web track. In Proceedings of TREC-2001.

Craswell, N., Hawking, D., & Robertson, S. (2001). Effective site finding using link anchor information. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 250–257). New York, NY, USA: ACM.

Dai, N., & Davison, B. D. (2009). Vetting the links of the web. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management* (pp. 1745–1748). New York, NY, USA: ACM.

Davis, H. (2000). Hypertext link integrity. ACM computing surveys electronic symposium on hypertext and hypermedia 31 (4). <http://eprints.ecs.soton.ac.uk/4473/>.

Efthimiadis, E. N. (1996). Query expansion. *Annual review of information systems and technology, 31*, 121–187.

Eiron, N., & McCurley, K. S. (2003). Analysis of anchor text for web search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 459–460). New York, NY, USA: ACM.

Gospodnetic, O., & Hatcher, E. (2004). Lucene in action. *Manning*.

Green, J.J. (2004). Google pagerank and related technologies. <http://www.lazworld.com/whitepapers/PageRank-Technologies.pdf>.

Grønbæk, K., Sloth, L., & Ørbæk, P. (1999). Webvise: Browser and proxy support for open hypermedia structuring mechanisms on the world wide web. *Computer Networks, 31*(11–16), 1331–1345.

Harrison, T. L., & Nelson, M. L. (2006). Just-in-time recovery of missing web pages. In *HYPERTEXT '06: Proceedings of the seventeenth conference on hypertext and hypermedia* (pp. 145–156). New York, NY, USA: ACM.

Huang, W. C., Geva, S., Trotman, A. (2009). Overview of INEX 2008 link the wiki track. Advances in Focused Retrieval: 7th International workshop of the initiative for the evaluation of XML retrieval (INEX 2008).

Ingham, D., Caughey, S., & Little, M. (1996). Fixing the broken-link problem: The w3objects approach. *Computer Networks and ISDN System, 28*(7–11), 1255–1268.

Jeh, G., & Widom, J. (2002). Simrank: A measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 538–543). New York, NY, USA: ACM.

Jing, Y., Croft, W. B. (1994). An association thesaurus for information retrieval. In Proceedings of RIAO-94, 4th International Conference Recherche d'Information Assistee par Ordinateur (pp. 146–160). New York, US. <citeseer.ist.psu.edu/jing94association.html>.

Kahle, B. (1997). Preserving the internet. *Scientific American, 276*(3), 82–83.

Klein, M., & Nelson, M. L. (2008). Revisiting lexical signatures to (re-)discover web pages. In *ECDL '08: Proceedings of the 12th European conference on research and advanced technology for digital libraries* (pp. 371–382). Berlin, Heidelberg: Springer-Verlag.

Koehler, W. (2002). Web page change and persistence—a four-year longitudinal study. *Journal of the American Society for Information Science and Technology, 53*(2), 162–171.

Lin, Z., Lyu, M. R., & King, I. (2006). Pagesim: A novel link-based measure of web page similarity. In *WWW '06: Proceedings of the 15th international conference on World Wide Web* (pp. 1019–1020). New York, NY, USA: ACM.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Markwell, J., & Brooks, D. W. (2002). Broken links: The ephemeral nature of educational www hyperlinks. *Journal of Science Education and Technology, 11*(2), 105–108.

Martinez-Romo, J., & Araujo, L. (2008). Recommendation system for automatic recovery of broken web links. In *Advances in artificial intelligence – IBERAMIA 2008. Lecture Notes in Computer Science* (Vol. 5290, pp. 302–311). Berlin/Heidelberg: Springer.

Martinez-Romo, J., & Araujo, L. (2009). Retrieving broken web links using an approach based on contextual information. In *HT '09: Proceedings of the 20th ACM conference on hypertext and hypermedia* (pp. 351–352). New York, NY, USA: ACM.

Martinez-Romo, J., & Araujo, L. (2010). Analyzing information retrieval methods to recover broken web links. In *Advances in information retrieval. Lecture notes in computer science* (Vol. 5993, pp. 26–37). Berlin/Heidelberg: Springer.

McBryan, O.A., 1994. GENVL and WWWW: Tools for taming the Web. In Nierstarsz, O. (Ed.), Proceedings of the first international world wide web conference, CERN, Geneva (p. 15). <citeseer.ist.psu.edu/mcbryan94genvl.html>.

Mearian, L. (2009). Internet archive to unveil massive wayback machine data center. *Computerworld* [March (19)].

Morishima, A., Nakamizo, A., Iida, T., Sugimoto, S., & Kitagawa, H. (2008). Pagechaser: A tool for the automatic correction of broken web links. In *ICDE '08: Proceedings of the 2008 IEEE 24th international conference on data engineering* (pp. 1486–1488). Washington, DC, USA: IEEE Computer Society.

Morishima, A., Nakamizo, A., Iida, T., Sugimoto, S., & Kitagawa, H. (2009a). Bringing your dead links back to life: A comprehensive approach and lessons learned. In *HT '09: Proceedings of the 20th ACM conference on hypertext and hypermedia* (pp. 15–24). New York, NY, USA: ACM.

Morishima, A., Nakamizo, A., Iida, T., Sugimoto, S., & Kitagawa, H. (2009b). Why are moved web pages difficult to find? The wish approach. In *WWW '09: Proceedings of the 18th international conference on world wide web* (pp. 1117–1118). New York, NY, USA: ACM.

Nakamizo, A., Iida, T., Morishima, A., Sugimoto, S., & Kitagawa, H. (2005). A tool to compute reliable web links and its applications. In *SWOD '05: Proc. International special workshop on databases for next generation researchers* (pp. 146–149). IEEE Computer Society.

Nelson, M. L., McCown, F., Smith, J. A., & Klein, M. (2007). Using the web infrastructure to preserve web pages. *International Journal on Digital Libraries, 6*(4), 327–349.

Noll, M. G., Meinel, C. (2008). Web search personalization via social bookmarking and tagging. In Proceedings of 6th international semantic web conference (ISWC), pp. 367–380.

Pant, G. (2003). Deriving link-context from html tag tree. In *DMKD '03: Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery* (pp. 49–55). New York, NY, USA: ACM.

Popitsch, N. P., & Haslhofer, B. (2010). Dsnotify: Handling broken links in the web of data. In *WWW '10: Proceedings of the 19th international conference on world wide web* (pp. 761–770). New York, NY, USA: ACM.

Qiu, Y., Frei, H. -P. (1993). Concept based query expansion. In SIGIR, pp. 160–169.

Qiu, Y., Frei, H. -P. (1995). Improving the retrieval effectiveness by a similarity thesaurus. Tech. rep. 225, Zürich, Switzerland. <citeseer.ist.psu.edu/qiu94improving.html>.

Rijsbergen, C. J. V. (1977). A theoretical basis for the use of cooccurrence data in information retrieval. *Journal of Documentation*(33), 106–119.

Schütze, H., & Pedersen, J. O. (1997). A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing & Management, 33*(3), 307–318.

Shimada, T., & Futakata, A. (1998). Automatic link generation and repair mechanism for document management. *HICSS '98: Proceedings of the thirty-first annual Hawaii international conference on system sciences* (Vol. 2, pp. 226). Washington, DC, USA: IEEE Computer Society.

Voorhees, E. M. (2003). Overview of the TREC 2003 robust retrieval track. In: TREC, pp. 69–77.

Voorhees, E. M. (2005). The TREC robust retrieval track. *SIGIR Forum, 39*(1), 11–20.

Voorhees, E. M. (2006). The TREC 2005 robust track. *SIGIR Forum, 40*(1), 41–48.

Westerveld, T., Kraaij, W., Hiemstra, D. (2001). Retrieving web pages using content, links, urls and anchors. In Proceedings of TREC10. Gaithersburg, MD, NIST, pp. 663–672.

Xi, W., Fox, E. A., Tan, R. P., & Shu, J. (2002). Machine learning approach for homepage finding task. In *SPIRE 2002: Proceedings of the 9th international symposium on string processing and information retrieval* (pp. 145–159). London, UK: Springer.

Yanbe, Y., Jatowt, A., Nakamura, S., & Tanaka, K. (2007). Can social bookmarking enhance search in the web? In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on digital libraries* (pp. 107–116). New York, NY, USA: ACM.