Journal of King Saud University – Computer and Information Sciences 31 (2019) 147–160

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Broken link repairing system for constructing contextual information portals

Shariq Bashir

College of Computer and Information Sciences, Information Management Department, Imam Muhammad Ibn Saud University, Riyadh, Saudi Arabia

ARTICLE INFO

Article history:
Received 14 July 2017
Revised 6 December 2017
Accepted 24 December 2017
Available online 28 December 2017

Keywords:
Information retrieval
Machine learning
Broken links
Learning to rank
Contextual information portals for intermittent networks

ABSTRACT

The web is an extremely powerful resource that has the potential to improve education and health. It enables access to new markets. There are, however, fundamental problems with web access in emerging regions. The primary issue is that internet connectivity is not keeping up with web complexity and size. Recently an innovative technology is developed in the form of contextual information portals (CIP) to mitigate the effect of low connectivity. CIP provides offline searchable and browse-able information portal. The information in CIP is composed of vertical slices of the internet about specific topics. CIP is an ideal tool for developing regions which have limited access to internet. It can be used in schools and colleges to enhance lesson plans and educational material. Although, as a standalone portal CIP provides an interactive searching and browsing interface enabling a web-like experience, however, a fundamental problem that users face is broken links. This is because crawling the web for constructing a collection for CIP only makes available a portion of webpages but not all possible documents. This creates several broken links. To address this problem we develop a broken link repairing system *(brLinkRepair)* for repairing broken links. *brLinkRepair* is useful when a user tries to navigate between pages through links and pointed pages of links are missing from the CIP. We provide an information retrieval system for repairing broken links. For each broken link our system recommends related pages that are similar to pointed pages. To further improve the effectiveness of system we combine all information sources using learning to rank approach. Our results indicate learning to rank (by combining information sources) improves effectiveness.

Contents

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

E-mail address: sbbashir@imamu.edu.sa

https://doi.org/10.1016/j.jksuci.2017.12.013
1319-1578/© 2017 The Author. Production and hosting by Elsevier B.V. on behalf of King Saud University.

1. Introduction

Internet connectivity plays an important part in the overall development of any country. It provides a way to acquire knowledge from various study fields and allows people to stay in touch with the current situation in the world without the need to know them personally or travel to long distances. However, connectivity opportunities are largely unequal among different regions (Mishra et al., 2005; Du et al., 2006; Li and Chen, 2013; Zaki et al., 2014; Marentes et al., 2014; Arour et al., 2015; Bouramoul, 2016). High-income countries have continuous rich availability of Internet and have comprehensive contents in the local libraries. Both of these resources are extremely useful for research and education. Whereas local libraries in the developing world are mostly outdated and access to internet is unreliable, expensive and mostly available to only urban areas. People in developing countries have low purchasing power and often have to pay huge amount in order to get a good connection for internet. Slow and intermittent internet connection is a serious issue (Saif et al., 2007; Johnson et al., 2010; Ihm et al., 2010; Pejovic et al., 2012). Satellite connections are slow, mostly provide bandwidth of only a few hundreds of kbps or 1 Mbps. Electricity is often intermittent. Networks are managed either remotely or by poorly trained local staff. In schools and colleges, a single internet connection is shared among large number of students, staff and faculty members resulting in a overall slow page downloading (Pentland et al., 2004). While this is only one issue the quality and the size of webpages on the other hand are significantly advanced over the past few years. This causes significantly increase in page rendering time on slow internet connections.

Information and communications technology (ICT) can offer major opportunities for developing countries. There is evidence that shows that ICT can help developing regions to *"leapfrog"* into the digital economy, catching up by skipping some of the intermediate technology stages and thereby improving the quality of life. For providing connectivity in developing regions, recently an innovate ICT technology is developed called Contextual Information Portal (CIP) for extending the web to developing regions (Chen et al., 2010, 2011). A CIP is a system that provides an offline searchable and browse-able information portal. The topics of CIP are generated from course syllabi that are considered to be insufficiently covered by existing local libraries. Web pages of topics are obtained by crawling the web for pages that are relevant to the topics. The crawled webpages are then indexed, re-ranked locally and shipped to destination on large storage media (e.g. hard disk, DVDs or USB-sticks).

Although as a standalone portal a CIP provides an interactive searching and browsing interface similar to world wide web for the topics covered. However, to make searching of information possible the CIP must turn from mere document repositories into living collection. The development of innovative solutions for searching and exploring (similar to regular web retrieval) it are required. CIP provides web browser that enables search facility to users through queries. In a typical session, user submits a full-text query to CIP and CIP returns result list of results containing a list of top-n pages matching to the query. Each result includes the title of the page. The user then clicks on the results to see and to navigate in the webpages similar to as he/she does in case of world wide web. However, if we compare CIP with regular web then in case of CIP navigating between pages is limited as users frequently experience many broken links, i.e. broken link is a link that has pointed page missing form the collection. This happens because crawling the web for constructing a CIP makes available only a portion of webpages but not all possible pages. In this case, if a user reaches on a link that is apparently very useful and interesting but broken, he/she then moved back to search engine for revising the query to retrieve relevant information. This creates frustration and boredom as during revising the query user often loses the richness of information that was available to him/her on the page containing the broken link.

Main Contribution: The aim of this work is to repair the broken links by retrieving related pages in the collection that are similar to pointed pages of broken links. For each broken link we applied information retrieval technique for retrieving related webpages. For each broken link, our system automatically constructs broken link repairing query for retrieving related page. Constructing a search query is a tedious task that we want to perform automatically as the system needs to know relevant source of information that is useful for retrieving related web page. Possible sources of information are anchor text, surrounding text of anchor text, URL and the full text of page containing missing link. Previous work on repairing broken mainly used the terms of anchor text and URL (Martinez-Romo and Araujo, 2012). Previous work also investigated other sources, however, did not achieved good effectiveness as the technique relied only on *term frequency (tf)* and *document frequent (df)* for extracting terms from the sources. Their technique returns similar terms for queries when a page contains many broken links. This is not suitable for CIP as webpages in a CIP contain many broken links. In this work, we investigate the use of term proximity (position) relationship between the terms of *anchor text*, *URL*, *context around URL* and *full text of webpage* for extracting relevant terms. This not only returns different query terms for different broken links but also increases the effectiveness as the terms that are proximity close to each other reveal more relevance. Furthermore, since the information represented by individual information source is complementary, therefore, we study that whether or not combining all information sources improves effectiveness and for this purpose we use learning to rank approach for combining information sources. Our results indicate learning to rank (by combining information sources) improves effectiveness.

The remainder of this paper is structured as follows. Section 2 reviews related work on enabling web access for emerging regions. This section also reviews related work on repairing broken links. In Section 3 we first show the architecture of CIP and then we describe broken link repairing task. In Section 4 we describe an information retrieval system for repairing broken links. In Section 5 we describe the collections and experimental setup and then in Sections 5 and 6 we show effectiveness of proposed system. In Section 7 we combine information sources for repairing broken link using learning to rank and compare its effectiveness with individual information sources. Finally, Section 8 briefly summarizes the key results of our work.

2. Related work

We divide related work into two parts. In first part, we highlight major work on enabling web access for emerging regions and motivate the need of developing CIP. In second part, we highlight related work on repairing broken links.

2.1. Enabling web access for emerging regions

There has been considerable work in designing effective web search systems for developed regions; however, in the context of developing regions this research area has been largely ignored. In recent years research directions have been investigated. We classify literature review into following four categories.

- **Asynchronous web access for low-bandwidth connectivity:** In recent years offline search engines are developed for addressing the issue of low-bandwidth connectivity. GetWeb,[1] www4-mail[2] and Web2Mail[3] are systems that provide access to internet through emails. Users issue queries to these systems though emails and these systems return relevant list of URLs. Google Email Alert is another example of such category and it provides customized news in response to user queries. TEK (Libby Levison and Amarasinghe, 2002) developed at MIT is another application of such category. It is a client-server based system that provides offline access to the internet. TEK provides a non-interactive search mechanism and a user issues query through simple mail transfer protocol (SMTP) and search results are asynchronously send back to the user through email. DAKNet (Pentland et al., 2004) provides web access through physical transportation links such as buses and vans. DAKNet uses MAPs (Mobile Access Points) mounted on physical transportation links (using buses and vans), which regularly traverses villages to transport required information. These physical transportation links are fitted with omni-directional antennas and kiosks with omni-directional or directional antennas.
- **Web caching:** In the context of web caching there have been several optimizations that have been proposed for increasing the access of web to developing regions (Michel et al., 1998; Rabinovich and Spatschek, 2002; Du et al., 2006; Isaacman and Martonosi, 2008; Chen and Subramanian, 2013). The work by (Du et al., 2006) analyzed web access traces of Cambodia for analyzing the effectiveness of web caching strategies for developing regions. (Isaacman and Martonosi, 2008) in their work showed the benefits of collaborative caching and perfecting pages techniques for developing regions. Their results showed perfecting pages in advance increases the effectiveness for local cache-based search. These perfecting techniques can be used with CIP for enhancing the local search mechanism.
- **Contents adaptation for low bandwidth:** Content adaptation is another area that has been explored for low bandwidth regions. There are number of related works available on filtering and compression. We cannot provide comprehensive literature review of this area as this is out of the scope of this paper. Fox and Brewer (1996) in their work provided techniques for reducing the resolution and color depth of images to suit low-bandwidth users. Fred Douglis et al. (1998) in their work analysed relatedness between webpages of web cache for optimizing bandwidth. Loband[4] is another system developed for low-bandwidth environments that enables users to view filtered text-only versions of webpages.
- **Contextual Information Portal (CIP):** Most of the approaches reviewed above modify the contents of search results for providing the web data to users under limited-internet connectivity. CIP does not modify content of webpages but to leverage proxies to deal with the intermittency of the network in an application specific manner (Chen et al., 2010, 2011). CIP provides an offline searching and browsing facility. The information in CIP is crawled from the web relevant to topics. CIP is designed primarily for those regions and environments where either the connectivity is very poor or not available at all (Saif et al., 2007; Johnson et al., 2010; Ihm et al., 2010; Pejovic et al., 2012). For these regions the standard local search-cache has several problems. For example, the frequent cache misses in these locales result in very slow page downloading and rendering. Moreover, local cache returns only a limited binary answer in the form of *yes* or *no* of whether webpages are available in the cache. This is not suitable because it is possible that a specific webpage does not exist in the cache but the cache could contain many other webpages that are equally similar to missed webpage.

2.2. Repairing broken links

Related work on repairing broken links can be classified into categorizes: (a) repairing broken links by applying information annotation, and (b) repairing broken links using information retrieval. We highlight major works of both categories.

Nakamizo et al. (2005) developed a tool for repairing broken URLs using information annotation technique. Their tool is useful for repairing broken URLs when webpages are moved from their location. Their tool outputs a list of webpages sorted by their plausibility of being link authorities. Their tool first uses a link authority server that collects links and then it sorts the links according to their plausibility. This plausibility is based on a set of attributes concerning the relations among links and directories. Klein and Nelson (2008) utilized document similarity for retrieving related webpages when webpages disappear in the future. To achieve this, their system first extracts a small subset of terms (which they called lexical signature) from the contents of documents for defining the "*aboutness*" of documents. Next, their system utilizes these lexical signatures for retrieving related webpages. Harrison and Nelson (2006) also used lexical signatures of webpages in the context of digital preservation for locating missing webpages. Similar to Klein and Nelson (2008) system, their system first extracts "*lexical signature*" from the pages and then their system uses these lexical signatures as a query to search engine for retrieving related webpages.

Closest to our research work is Martinez-Romo and Araujo (2012) in which information retrieval techniques are applied for repairing broken links. Their work used information retrieval techniques for retrieving related pages by using sources of information that are available in the page containing broken link. The sources that they used are anchor text, URL, context of anchor text and full text of page containing broken link. Their system first used these

[1] GetWeb: Retrieve webpages via e-mail, www.hrea.org/getweb.html.

[2] Web Navigation and Database Search by E-Mail, http://www4mail.org.

[3] Web2Mail, http://www.web2mail.com.

[4] http://www.loband.org.

sources as queries and then their system processed queries through information retrieval methods for retrieving related webpages. Their experiments indicate that the anchor text is quite useful for retrieving related webpages.

Known Items Search: In the context of digital libraries recently some attempts are made for retrieving known items (Azzopardi et al., 2007). The task of retrieving known items is similar to repairing broken links in the sense that the aim of both tasks is to retrieve most relevant items from the collection. Known-items search assumes that a user knows a item (document) in the collection that he/she thinks that it is relevant for his/her need and he/she has already seen this document in the collection. Now there is some need arisen and the user wants to retrieve this item. For retrieving this item he/she tries to recall different terms of the item for constructing a query to search engine that could help for retrieving this item. Azzopardi et al. (2007) in their work developed a user model for identifying relevant terms that a user could recall for retrieving known-items. Their model identifies terms that are either most discriminative or more popular in the desired items. Although both retrieval tasks have close similarity but have difference in this sense that known-item search assumes that user knows some information from the contents of relevant item and he/she uses this information for retrieving that item. Whereas in case of repairing broken link such information is implicitly available in the form of URL and anchor text.

Linking Documents to Encyclopedic Knowledge: The aim of this task is to automatically identify relevant segments from the content of pages that are potential useful candidates for links and then automatically enriching these links with most related webpages (Mihalcea and Csomai, 2007). Most of the studies in this research domain used Wikipedia for analyzing the accuracy of their methods which is rather more structured than regular web collection. Milne and Witten (2008) applied machine learning for identifying significant terms within wikipedia page, and then their technique enriched these terms with links to the appropriate Wikipedia articles. For achieving this, their method first defined features from candidate segments of terms and then their method utilized these features for training a link classifier. The features that they defined are based on: (a) commonness of terms (prior probability) with their surrounding context, (b) term's relatedness with it's surrounding context, and (c) context quality. Although both tasks have close similarity in the sense that the target of both tasks is to retrieve relevant webpages for links, however, in case of repairing broken links our collection is rather more unstructured than Wikipedia and we want to utilize additional sources that are available in the page containing broken such as anchor text, URL, context around URL and content of page.

3. Contextual Information Portal (CIP)

We will explain the broken link repairing system by first explaining the architecture of CIP. Fig. 1 presents the architecture of CIP crawler. Fig. 2 presents the architecture of broken link repairing system. CIP has following two main components:

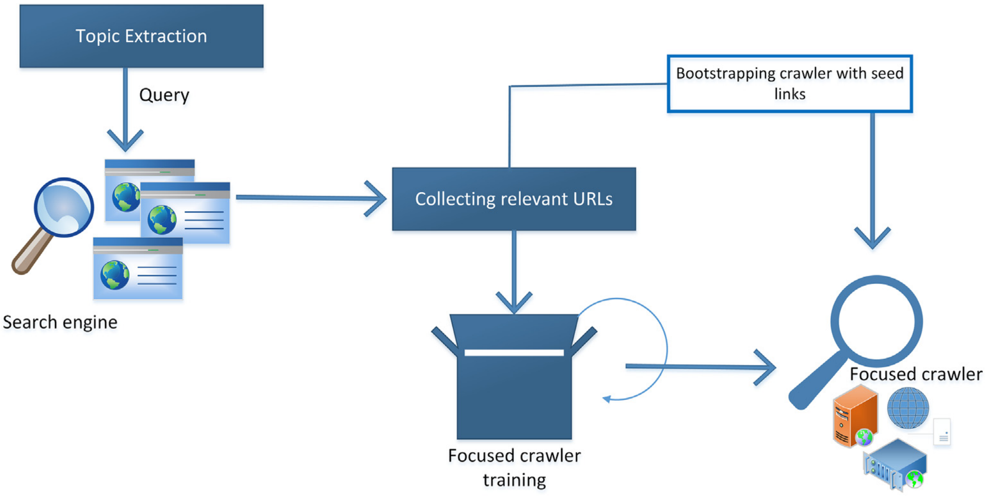

Fig. 1. Architecture of CIP Crawler and different components of CIP Crawler.

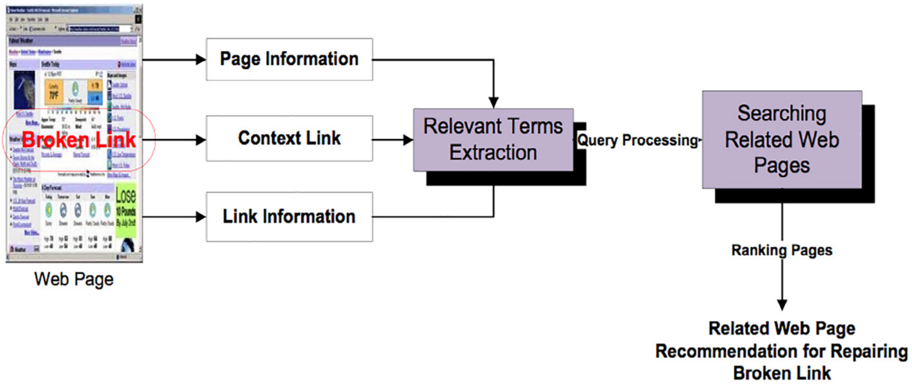
Broken Link
Web Page
Page Information
Context Link
Link Information
Relevant Terms Extraction
Query Processing
Searching Related Web Pages
Ranking Pages
Related Web Page Recommendation for Repairing Broken Link

Fig. 2. Architecture of broken link repairing system.

- (CIP Crawler): This component crawls and constructs collection for CIP.
- (Broken Link Repairing System): CIP collection may have many URLs in the webpages that are broken. This component repairs these broken links using information retrieval technique.

3.1. CIP crawler

The purpose of CIP crawler is to crawl the web and to index only those pages that are relevant to topics. For a single topic, say *"introduction to computers*", the goal of a CIP is to make available a large portion of webpages relevant to the topic. CIP crawler achieves this through focused crawling (Chakrabarti et al., 1999; Arasu et al., 2001; Aggarwal et al., 2001). For each topic, a focused crawler is trained by providing a subset of relevant and irrelevant webpages. For training classifier, we first download a set of top N results from popular search engine (google) by using topics as queries. We mark top N results of search engine as relevant pages. We then again query to search engine with an irrelevant query for instance *"the*" for downloading a set of irrelevant pages. Next, we train a document classifier over these relevant/irrelevant pages for doing focused crawling. After training we start the crawling with the help of trained focused crawler. We bootstrap the crawler by providing relevant webpages of topics as seed links.

3.2. Broken link repairing system

The third important component of CIP architecture is broken links repairing system (*brLinkRepair*). This component is useful when a user tries to navigate in pages through links and pointed pages of links are missing from the CIP (Martinez-Romo and Araujo, 2012). *brLinkRepair* repairs these broken links by retrieving related pages using information retrieval technique. In the architecture, broken link and the webpages that contain broken links provide terms that may be relevant for retrieving related webpages. One important function of *brLinkRepair* is query generation. The task of query generation is to search most relevant terms from information sources that are available in the page containing broken link. Fig. 2 shows the architecture of broken link repairing system. In this method, the relevant terms have to be very carefully selected otherwise irrelevant terms drift the results to noisy webpages. To achieve this, we explore several sources (such as anchor text, context of URL, URL and full text of page containing broken link). We also applied term proximity techniques to determine terms proximity relationship to anchor text and URL. Finally, the generated queries are submitted to the retrieval system, and top ranked webpages are retrieved and page at top position is recommended to the user.

Another important sub component in *brLinkRepair* is link classifier. The purpose of link classifier is to classify repairable and non-repairable links. This is useful for identifying those links for whom the recommendation is possible and for whom the recommendation is not possible (by using the webpages of CIP collection). This is because, in case of highly specialized CIP collection for which the web is crawled with the help of a focused crawler there could be many links for which the recommendation could not be possible using given CIP collection. For instance, those links that are pointing to home pages of websites and their pointed home pages are missing due to their non-relevance with the CIP topics, or links that are pointing to *university/class rooms direction*, or pages that are pointing to *frequent asked questions*, or *authors feedback* etc. In a web collection, there could be many such links. Although *brLinkRepair* can provide recommendation for all kind of links, however, due to the non-relevance of these links with the CIP collection their is a high chance that the retrieved webpages will be irrelevant. This could create frustration; therefore, it is useful to remove non-repairable links from the CIP collection with the help of a link classifier.

4. Automatic query reformulations for repairing broken link

Our task of repairing broken links is similar to as discussed in Martinez-Romo and Araujo (2012) for repairing broken links in the context of world wide web. However, previous work on this task focused only on investigation different sources of information that can be used for generating queries, but ignores how to combine all sources of information to increase the effectiveness of broken link repairing system. Furthermore, query generation techniques used in previous work have a limitation that it retrieves similar pages for multiple broken links when a source page has many broken links. This is because it relies only on term statistics such as *term frequency (tf)* and *document frequency (df)* for identifying relevant terms for queries. The aim of this research is to explore these sources as well as other sources with the help of terms proximity and attempts to combine all information sources using machine learning (learning to rank). To design an effective combination of sources we need to consider several factors: such as *Elements* (where to extract query words), *Weights* (how to calculate relevance of pages for queries), *Proximity* (whether or not to care for closeness of terms).

For Elements, we consider four sources of information, the anchor text of broken links, URL of broken link, context around URL, and full text of page containing broken link. For Weights, we use low level features of terms, such as *term frequency (tf)*, *document frequency (df),* as well as combination of these weights that form high level features such as *tfxidf*, *bm25* (Robertson and Walker, 1994) and *LM2000* (Zhai, 2002). In additional to this we also utilize proximity relationship between terms of anchor text and URL with the terms of context around URL and full text of source page.

4.1. Elements (information sources)

Martinez-Romo and Araujo (2008) in their approach investigate several information sources for identifying relevant terms for generating broken link repairing queries. In our approach we also utilize these sources and generate queries by extracting terms from these sources.

- **Anchor Text:** Anchor text usually provides more reliable information given by a web page designer about the content of pointed page. If we compare it with the URL of pointed page, then URL are ostensibly created by people other than the authors of the target webpages, and thus the anchor text likely includes summary and alternative representation of the content of pointed page. Because these anchor texts are typically short and descriptive similar to queries, commercial search engines widely utilized them as an important part for ranking documents (Eiron and McCurley, 2003; Dang and Croft, 2010).
- **Information provided by the URL (URL):** Apart from the anchor text, terms in the URL are the only information directly provided by a link. Terms of the URL also provide useful information to the pointed page. Similar to anchor text, commercial search engines utilize URLs for determining whether a page is relevant or not to a query (Pant, 2003; Benczúr et al., 2006; Chauhan and Sharma, 2007).
- **Information provided by the source page containing the broken link (SourcePage):** Martinez-Romo and Araujo (2012) found frequent terms of source page useful for repairing broken links.

- **Information provided by the context around URL (URLContext):** Full text of source page may contain many terms that not relevant to the content of pointed page. This drifts the retrieval effectiveness. A more reliable source is to use those terms that are near to the position of the URL. We generate context around URL by taking 20 terms before the URL position and 20 terms after the URL position.

4.2. Selecting terms from elements using term frequency (tf) and document frequency (df)

Anchor text and URL are short text segments. We generate queries from these elements by selecting all those terms that have document frequency less than 40% with respect to overall CIP collection. However, source page and context around URL have usually long text segments. For these elements, we first ordered all terms in the elements using *tf* weight and then select top 15 terms that have high *tf* weights for generating queries and subsequent retrieval.

5. Experiments

5.1. Collection, queries and relevance judgments

We repair broken links using information retrieval. Our system generates query for each broken link and ranks webpages of collection to retrieve related page. In order to analyze the effectiveness of our system we require collection and a set of broken links and their pointed webpages (as a pseudo relevance judgments) for which our system can perform ranking and then we can analyze system effectiveness.

Collection: To create a test collection we crawled the internet using focused crawler and download webpages that are relevant to a topic. To achieve this first we define a manual set of 140 sub topics for *"introduction to computers"* and *"agriculture"* from course syllabi. We want to construct two CIP collections, one for *"introduction to computers"* topics and one for *"agriculture"* topics. Table 1 shows a sample list of few sub topics. Next, for each query (by assuming each sub topic as a query) we train a focused crawler to crawl the world wide web for constructing (topic specific) collection (Chakrabarti et al., 1999). For obtaining relevant and irrelevant samples we query to a popular search engine (google) by assuming the topic terms as query and download top 20 results and marked them as relevant samples. We then again query to search engine using an irrelevant query *"the"* and download 20 queries and marked them as irrelevant samples. Next, we use LibSVM (Chang and Lin, 2011) and trained the focused crawler over these samples. After training, we only use relevant samples of each sub topic as a seed links and crawl the internet for downloading more webpages. We implement the focused crawler using Fish-Search approach (Hersovici et al., 1998) and stop it when it downloads 500 documents for each topic. This result into a total of $140 * 500 = 70,000$ webpages for each CIP collection.

Table 1
Sample list of topics of *"introduction to computers"* and *"agriculture"* collections. CIP crawler uses these topics for focused crawling.

Introduction to computers	Agriculture
Parts of a computer	Principles of crop production
Classification of computers	Soil and water conservation
Stable power supply computers	Kale or cabbage
Keyboard layout	Agricultural marketing
Central Processing Unit CPU	Water supply and irrigation
Computer Processors	List the essential plant nutrients
Computer Processor speed	Describe the various types of stores
Multi tasking operating systems	Sources of water
Computer Files	Soil sampling procedures
Graphical user interface GUI	Farm planning and budgeting

Queries and Relevance Judgments: For broken link queries we need a set of links that have pointed missing from the collections. We search these broken links from CIP collection. We notice that our collections contain three categorizes of links. The first category contains those links which are not broken and have pointed webpages that are relevant to the topics. The second category consists of those links that are broken and contain pointed webpages that are not relevant to the topics. These are the links for which our system cannot perform retrieval and we want to remove these links from webpages. The third category contains those links that are broken, however, these have pointed links that have strong relevance with the topics. For these links, CIP crawler could not download their pointed pages either due to misclassification of classifier or crawling limit (because we download only 500 documents for each topic). Ideally for this retrieval task we can use these links as queries and our system can retrieve relevant webpages from the collection. However, if we use these links as queries then for performing effectiveness analysis (i.e. whether the retrieved webpages are relevant to the broken links) we need a set of human evaluators who can read the webpages of collections and recommend us relevant webpage for each broken link. Although this provides more realistic test-bed for effectiveness analysis, however, this approach requires time and human efforts, as humans have to read a large number of webpages from both collections for each broken link. As an alternative, we can use unbroken links of first category as a query and their pointed webpages as relevance judgments. This approach also makes sense for performing effectiveness analysis as webpage authors already correctly provide the pointed webpages of these links. This provides us a cheep test-bed in order to analyze the effectiveness for broken link repairing system. Next, using links as queries and their pointed webpage as relevant judgments, the task of our retrieval is to rank these pages at top positions. We use this approach and search the collection and randomly select 1000 links as pseudo broken links. Tables 2 and 3 show a list of pseudo broken links of *"introduction to computers"* and *"agriculture"* collections.

5.2. Link classification

Since we construct CIP collection by doing focused crawling, therefore, a CIP collection could contain large number of links that are broken but our (broken link repairing) system could not repair these because the contents of their pointed pages do not make any relevance with the CIP topics. If we keep these links as it is in the collection, then during browsing users can click on these links and our system will recommend irrelevant webpages to the users. We want to remove these links from the webpages prior to loading their contents in the web browser. In order to achieve this, we require a classification system that could help us for classifying repairable and non-repairable broken links. To achieve this, first we randomly collect a subset of links from our collection and manually marked them relevant and irrelevant for recommendation after reading their anchor text, URL and full text of pointed pages. Next, we utilize terms of anchor text, URL, and context around URL and define a set of features on the basis of terms *tf* and *df* statistics to classify these links into repairable and non-repairable categorizes. We use following statistical features for training a link classifier.

- **avg_df_local:** This feature calculates average *df* (document frequency) of terms of the link using all webpages of the collection. This helps in identifying whether terms of a link are common or specific to the topics. Some irrelevant links such as those point-

Table 2
Sample pseudo broken links of "*Introduction to computers*" collection.

Pseudo Broken Link#1 (Introduction to computers)
URL = http://en.m.wikipedia.org//wiki/Clock_rate **Anchor Text** = *clock rate* **Context around URL** = *ed and/wiki/Wikipedia:Verifiability#Burden_of_evidence Wikipedia:Verifiability removed. (September 2009) The megahertz myth, or less commonly the gigahertz myth, refers to the misconception of only using . . ./wiki/Clock_rate Clock rate clock rate (for example measured in/wiki/Hertz#SI_multiples Hertz megahertz or/wiki/Hertz#SI_multiples Hertz gigahertz) to compare the performance of different/wiki/Microproc*
Pseudo Broken Link#2 (Introduction to computers)
URL = http://www.jegsworks.com/Lessons/lesson1-2/lesson1-1.htm **Anchor Text** = *Computer Types* **Context around URL** = *(known) and how you want to use the space (teaching methods, multi-use, etc). Design services are no cost and no obligation. . . .computer-desks-fiseries.asp FI Series Computer Desks flipIT LCD Desks computer-desks-fpseries.asp FP Series Computer Desks Semi-Recessed LCD Computer Desks computer-desks-srseries.asp SR Series Computer Desks Semi-Recessed CRT Computer Desks computer-desks-dtseries.asp*

Table 3
Sample pseudo broken links of "*agriculture*" collection.

Pseudo Broken Link#1 (Agriculture Collection)
URL = http://www.government.nl//issues/agriculture-and-livestock/animals/animal-welfare **Anchor Text** = *Animal welfare* **Context around URL** = *behave towards animals. These rules may be about the care of domestic pets or a harder line on mistreatment of animals, but may equally concern measures to prevent outbreaks of infectious animal diseases./issues/agriculture-and-livestock/animals/animal-welfare Animal welfare/issues/agriculture-and-livestock/animals/prevention-and-control-of-animal-diseases Prevention and control of animal diseases/issues*
Pseudo Broken Link#2 (Agriculture Collection)
URL = http://smallfarm.about.com//od/landpreparation/a/Fall-Soil-Amendments.htm **Anchor Text** = *Fall Soil Amendments* **Context around URL** = *methods from conventional to no-till to reduced tillage will be discussed. Whether you have a homestead, a tiny hobby farm or a small-scale farm, you'll find the right method for tilling your soil here./od/landpreparation/a/Fall-Soil-Amendments.htm Fall Soil Amendments Fall is a great time to spruce up your soil with cover crops, compost, and other amendments./od/landpreparation/a/Why-Test-Your-Soil.htm*

ing to home page with label "*Home*", "*FAQ*", "*feedback*" or "*top of the page*" show large scores for this feature. We calculate *avg_df_local* for all sources (anchor text, URL, context around URL). This result into three sub features.

- **avg_cf_local:** This feature is similar to *avg_df_local*. The only difference is that we use frequency of term within collection instead of document frequency.
- **avg_df_global:** This feature calculates average *document frequency (df)* of all terms. However, for this feature we determine *df* of terms from a global source "*web IT 5-gram Version 1*", a dataset from the Linguistic Data Consortium (LDC)[5] to learn the frequency of occurrences of n-grams on the web. Similar to *avg_df_local*, we calculate *avg_df_global* for all sources.
- **avg_tf:** This feature calculates average of term frequencies of terms from the source page. If the content of a link is relevant to the CIP then likely it would has high *tf* scores for the terms in the source page. We calculate *avg_tf* of the terms for only anchor text and URL. This results into two sub features.

Given above features, we trained a classifier using LibSVM (Chang and Lin, 2011) and found that it achieves around 80% of classification accuracy for classifying reparable and non-repairable links. Next, for retrieval we only use those links that are classified as repairable.

5.3. Effectiveness measures

We use three effectiveness measures of information retrieval in order to test the effectiveness of broken link repairing system.

Recall: Recall is the ratio of the number of retrieved relevant documents relative to the total number of documents in the collection that are desired to retrieve. For this evaluation test we calculate recall of retrieved results at rank position 1 (R@1), at rank position 3 (R@3) and at rank position 10 (R@10).

Mean Reciprocal Rank (MRR): Recall is not sensitive to the ranking position of relevant webpage (i.e., it does not provide evaluation result relative to ranking position of relevant webpage). Mean Reciprocal Rank cares this factor by calculating multiplicative inverse of the rank position of the correct webpage (Voorhees, 2001). A system that retrieves related webpages of broken links at top positions provides high MRR score.

Cosine Similarity: Both evaluation measures explained above only utilize relevance judgments (judged webpages) in order to evaluate the effectiveness of system. These measures cannot calculate effectiveness when relevance judgments are not retrieved at top positions. However, during query processing our system showed several cases when the relevance judgments could not be retrieved at top positions but the contents of webpages at top positions were almost identical to judged webpages. Ideally such kind of retrieved results also show high effectiveness. For this purpose, we apply vector space model to analyze the similarity of top retrieved webpages with relevance judgments. We convert each retrieved webpage into a term vector and calculate its cosine similarity distance with the contents of relevance judgment. We calculate the cosine similarity of retrieved results at position 1 (Sim@1) and at rank position 3 (Sim@3).

5.4. Effectiveness analysis of sources

Tables 4 and 5 show the effectiveness of all information sources that we used for query generation. According to the obtained results the source that has achieved high effectiveness is the content of source page that contains broken link. This indicates that source page contains useful terms and this can increase the effectiveness of retrieved results. Among other sources, URL also showed good effectiveness than anchor text and context around

[5] https://www.ldc.upenn.edu/.

Table 4
Effectiveness of information sources (URL, anchor text, context around URL and source page) on *"introduction to computers"* topics collection.

Query source	Retrieval	Recall			MRR	Cosine similarity	
	Model	R@1	R@3	R@10	MRR	Sim@1	Sim@3
Anchor	*Anchor_tf*	0.09	0.14	0.21	0.12	0.29	0.36
	Anchor_df	0.06	0.09	0.18	0.08	0.31	0.33
	Anchor_tfxidf	0.10	0.17	0.25	0.14	0.29	0.35
	Anchor_bm25	0.16	0.22	0.32	0.18	0.29	0.37
	Anchor_LM2000	0.18	0.26	0.31	0.21	0.29	0.37
URL	*URL_tf*	0.15	0.22	0.36	0.18	0.58	0.61
	URL_df	0.18	0.25	0.39	0.21	0.59	0.66
	URL_tfxidf	0.20	0.24	0.39	0.22	0.62	0.68
	URL_bm25	0.32	0.37	0.51	0.36	0.61	0.73
	URL_LM2000	0.29	0.36	0.50	0.32	0.54	0.66
URLContext	*URLContext_tf*	0.13	0.22	0.33	0.16	0.57	0.61
	URLContext_df	0.14	0.23	0.36	0.18	0.59	0.64
	URLContext_tfxidf	0.13	0.24	0.36	0.17	0.61	0.64
	URLContext_bm25	0.16	0.24	0.35	0.20	0.62	0.70
	URLContext_LM2000	0.18	0.26	0.40	0.22	0.62	0.69
SourcePage	*SourcePage_tf*	0.29	0.36	0.47	0.30	0.62	0.70
	SourcePage_df	0.30	0.36	0.48	0.32	0.61	0.70
	SourcePage_tfxidf	0.29	0.37	0.48	0.31	**★0.63**	0.70
	SourcePage_bm25	0.43	0.48	0.54	0.46	0.62	**★0.78**
	SourcePage_LM2000	**★0.48**	**★0.51**	**★0.57**	**★0.49**	0.61	**★0.78**

Table 5
Effectiveness of information sources (URL, anchor text, context around URL and source page) on *"agriculture"* collection.

Query source	Retrieval	Recall			MRR	Cosine similarity	
	Model	R@1	R@3	R@10	MRR	Sim@1	Sim@3
Anchor	*Anchor_tf*	0.19	0.26	0.40	0.23	0.19	0.26
	Anchor_df	0.16	0.19	0.29	0.17	0.15	0.21
	Anchor_tfxidf	0.29	0.35	0.47	0.29	0.16	0.26
	Anchor_bm25	0.33	0.37	0.53	0.32	0.19	0.30
	Anchor_LM2000	0.36	0.41	0.53	0.37	0.20	0.33
URL	*URL_tf*	0.35	0.40	0.49	0.37	0.37	0.53
	URL_df	0.37	0.41	0.51	0.37	0.35	0.55
	URL_tfxidf	0.44	0.51	0.58	0.45	0.34	0.60
	URL_bm25	0.48	0.57	0.61	0.49	0.35	0.62
	URL_LM2000	0.48	0.59	0.67	0.51	0.34	0.60
URLContext	*URLContext_tf*	0.26	0.32	0.45	0.27	0.36	0.53
	URLContext_df	0.30	0.42	0.54	0.33	0.37	0.59
	URLContext_tfxidf	0.31	0.42	0.53	0.32	0.37	0.59
	URLContext_bm25	0.23	0.33	0.47	0.22	0.35	0.62
	URLContext_LM2000	0.31	0.38	0.50	0.30	0.34	0.61
SourcePage	*SourcePage_tf*	0.51	0.54	0.66	0.51	0.33	0.60
	SourcePage_df	0.55	0.60	0.66	0.57	**★0.39**	0.65
	SourcePage_tfxidf	0.57	0.62	0.68	0.58	0.37	0.65
	SourcePage_bm25	0.61	0.66	0.72	0.62	0.36	**★0.68**
	SourcePage_LM2000	**★0.71**	**★0.74**	**★0.78**	**★0.71**	0.32	0.67

URL. The effectiveness of anchor text is very low than all other sources and this could be because it has very short length.

As source page achieves high effectiveness, however, it contains information from all other sources. Table 6 shows how much individual sources contribute in the effectiveness of source page by excluding the content of individual source from the content of source page. According to the obtained results the source that creates high effect is the content of URL. This indicates that URL contains useful terms and this can increase the effectiveness of retrieved results.

If we compare the effectiveness of different retrieval models, then *LM2000* and *bm25* both have achieved high effectiveness than other retrieval models. We did not notice any major difference between different information sources and retrieval models when they are analyzed with different effectiveness measures (recall, MRR and cosine similarity).

6. Constructing broken link repairing queries using term proximity

In Tables 4 and 5 experiments content of source page shows better effectiveness than all other sources. For running Tables 4 and 5 experiments we select query term from the content of source page on the basis of low *df* and high *tf* weights. Although this approach shows good effectiveness, however, if a page has many broken links then selecting query term using this approach could provide similar query terms for multiple broken links. This recommends similar pages for all broken links. A better approach could be to rely only those terms that are proximity close to the terms of anchor text. Proximity reflects closeness of terms in a text. The underlying intuition is that the more compact or closed the query terms are in the text, the more likely it is that they are topically related, and thus the higher the possibility that the proximity

Table 6
Contribution of individual information sources (URL, anchor text and context around URL) in the effectiveness of source page on *"introduction to computers"* topics collection.

Query source	Retrieval	Recall			MRR	Cosine similarity	
	Model	R@1	R@3	R@10	MRR	Sim@1	Sim@3
SourcePage – URL	*(SourcePage – URL)_tf*	0.18	0.23	0.30	0.19	0.39	0.44
	(SourcePage – URL)_df	0.19	0.23	0.30	0.20	0.38	0.44
	(SourcePage – URL)_tfxidf	0.18	0.23	0.30	0.20	0.40	0.44
	(SourcePage – URL)_bm25	0.27	0.30	0.34	0.29	0.39	0.49
	(SourcePage – URL)_LM2000	0.30	0.32	0.36	0.31	0.38	0.49
SourcePage – URLContext	*(SourcePage – URLContext)_tf*	0.26	0.31	0.41	0.28	0.56	0.63
	(SourcePage – URLContext)_df	0.27	0.33	0.43	0.29	0.55	0.63
	(SourcePage – URLContext)_tfxidf	0.26	0.33	0.43	0.28	0.56	0.63
	(SourcePage – URLContext)_bm25	0.38	0.42	0.49	0.41	0.55	0.70
	(SourcePage – URLContext)_LM2000	0.43	0.46	0.50	0.44	0.55	0.70
SourcePage – Anchor	*(SourcePage – Anchor)_tf*	0.24	0.30	0.39	0.25	0.52	0.59
	(SourcePage – Anchor)_df	0.25	0.30	0.40	0.27	0.51	0.59
	(SourcePage – Anchor)_tfxidf	0.24	0.31	0.40	0.26	0.53	0.59
	(SourcePage – Anchor)_bm25	0.36	0.40	0.45	0.39	0.52	0.66
	(SourcePage – Anchor)_LM2000	0.39	0.42	0.47	0.40	0.51	0.66
SourcePage	*SourcePage_tf*	0.29	0.36	0.47	0.30	0.62	0.70
	SourcePage_df	0.30	0.36	0.48	0.32	0.61	0.70
	SourcePage_tfxidf	0.29	0.37	0.48	0.31	**★0.63**	0.70
	SourcePage_bm25	0.43	0.48	0.54	0.46	0.62	**★0.78**
	SourcePage_LM2000	**★0.48**	**★0.51**	**★0.57**	**★0.49**	0.61	**★0.78**

based selected terms will increase the effectiveness of retrieval results. In information retrieval terms proximity has been widely used in query expansion approaches (Zhao and Yun, 2009; Tao and Zhai, 2007; Cummins and O'Riordan, 2009). It is highly useful for selecting expansion terms from pseudo relevance feedback (PRF) documents. It involves two steps. In first step, terms are weights using proximity measures. In the context of PRF these weights are calculated by utilizing the positions of terms in PRF documents and the positions of terms in the query. Then in second iteration, these proximity measures are used for training a classifier to select good expansion terms from all possible terms. We use similar process for selecting relevant terms from the content of source page. We use following proximity measures in order to calculate the proximity between the term of the page containing broken and the terms of anchor text and URL. Additional descriptions of these features is available in (Zhao and Yun, 2009; Tao and Zhai, 2007; Cummins and O'Riordan, 2009).

We describe features by denoting information sources (URL and anchor text) as s and term of source page (p) as t.

- **(f1): Average Minimum Distance to Information Source:** This feature calculates the score by taking the average of the shortest distance between the occurrences of term t of page containing broken link and terms of s (Cummins and O'Riordan, 2009). This feature rewards all those terms that appear very close to terms of s, e.g. in the same *phrase*, *sentence* or *paragraph*.
  Let $s = \{s_1, s_2, \ldots, s_m\}$ be the set of terms of source (s). $O_{s_i} = \{o_{i_1}, o_{i_2}, \ldots, o_{i_n}\}$ is the set of term occurrence positions of term of source s_i in p. $PD(s_i, t|p)$ is the distance between term s_i and term t of page containing broken link. $f1$ is the distance between the closest occurring positions of term s_i and t. This distance is measured through their occurring positions in the page containing broken link.

$$f_1(t) = \frac{\sum_{s_i \in s} PD(s_i, t|p)}{|s|} \tag{1}$$

$$PD(s_i, t|p) = min_{o_{i_k} \in O_{s_i}, o_{t_k} \in O_t}\{\mathrm{abs}(o_{i_k} - o_{t_k})\} \tag{2}$$

  where $|s|$ is the length of source and O_{s_i} and O_t are the sets of occurrences positions of terms (s_i and t) in p.

- **(f2): Pair-wise Terms Proximity on the basis of Minimum Distance:** $f1$ captures average minimum term distance with individual terms of s. However, a better feature could be to calculate this distance with pairs of source terms (Cummins and O'Riordan, 2009). $f2$ calculates the minimum distance between the term of page containing broken link and pairs of terms in source. This feature is calculated as follows.

$$f_2(t) = min_{\hat{p}(s_i,s_j) \in s, t \neq s_i, t \neq s_j, s_i \neq s_j} \{PD2(s_i, s_j, t|p)\} \tag{3}$$

$$PD2(s_i, s_j, t|p) = min_{o_{i_k} \in O_{s_i}, o_{j_k} \in O_{s_j}, o_{t_k} \in O_t}\{PD(s_i, t|p) + PD(q_j, t|p)\} \tag{4}$$

  $\hat{p}(s_i, s_j)$ enumerates all possible terms pairs in s. $PD2(s_i, s_j, t|p)$ denotes the pair-wise distance between terms pair s_i and s_j in s and term t of page containing broken link. Similar to $f2$, it is the distance between the closest occurring positions of terms s_i, s_j and t.

- **(f3): Pair-wise Terms Proximity on the basis of Average Distance:** This feature calculates the average distance between pairs of source terms s_i, s_j and term t of page containing broken link (Cummins and O'Riordan, 2009). This feature promotes those terms that consistently appear close to term pairs of source in localized areas, e.g. in the same *paragraph*, *sentence* or *phrase*. Given the set $\hat{p}(s_i, s_j)$ of all possible terms pairs of source, the weight of this feature can be calculated as follows.

$$f_3(t) = \frac{\sum_{\hat{p}(s_i,s_j) \in s, t \neq s_i, t \neq s_j, s_i \neq s_j} \{PD2(s_i, s_j, t|p)\}}{|\hat{p}(s_i, s_j)|} \tag{5}$$

- **(f4): Difference of Terms Positions:** This feature calculates the difference between the average positions of terms in the source and term t of page containing broken link. This feature captures where terms of source and t are occurring together.

$$f_4(t) = \frac{\sum_{s_i \in s} abs\{\sum_{o_{i_k} \in O_{s_i}} \frac{o_{i_k}}{|O_{s_i}|} - \sum_{o_{t_k} \in O_t} \frac{o_{t_k}}{|O_t|}\}}{|s|} \tag{6}$$

- **(f5): Co-Occurrence with Information Source:** To learn relevant terms all those terms are assumed relevant that frequently co-occur with terms of s (Cummins and O'Riordan, 2009). f(5) captures this co-occurrence.

$$f_5(t) = log\frac{1}{|s|}\sum_{s_i \in s}\frac{c(s_i, t|p)}{tf(s_i|p)} \quad (7)$$

where $c(s_i, t|p)$ is the frequency of co-occurrences of term of source s_i with term t within text windows of page containing broken link. $tf(s_i|p)$ denotes the term frequency of s_i in p. The window size is empirically set to 20 terms.

- **(f6): Co-occurrence with Terms Pairs:** The previous feature calculates co-occurrence of term t with individual terms of s. This feature captures a stronger co-occurrence relation of term t with all pairs of terms of s (Cummins and O'Riordan, 2009). Given the set $\hat{p}(s_i, s_j)$ of all possible pairs of terms of s, the value of this feature can be calculated as follows.

$$f_6(t) = log\frac{1}{|\hat{p}(s_i, s_j)|}\sum_{\hat{p}(s_i,s_j)\in s, t \neq s_i, t \neq s_j, s_i \neq s_j}\frac{c(s_i, s_j, t|p)}{tf(s_i|p) + tf(s_j|p)} \quad (8)$$

$c(s_i, s_j, t|p)$ denotes the frequency of co-occurrences of term t with term pair s_i and s_j of s within text windows of p. The window size is empirically set to 20 terms.

Table 7
The distribution of good and bad instances of training dataset.

Class	Total instances
All classes	7940
good class	3879
bad class	4061

Table 8
LibSVM classification accuracy on the training dataset using 10-fold cross validation.

Class	True positive rate	False positive rate
All classes	83%	17%
good class	84%	16%
bad class	82%	18%

We train a classifier using term proximity features for selecting good terms from the content of source page. For constructing training queries, a subset of 50 random pseudo broken links (that are not part of 1000 queries) is used for training (term) classifier. URLs of these 50 pseudo broken links are used as a seed queries. Next, we process each seed query and result lists of queries are ranked for retrieving related webpages. We ranked the result lists using *LM2000*. To differentiate bad and good terms of source page we expand the terms of source page individually with seed queries. After expanding each term, we compared the (expanded) seed query effectiveness with (non-expanded) seed query effectiveness. The effectiveness is compared with recall at rank position 1 (*R@1*). In practice, any expanded term may act on the seed query independent to other, resulting in different webpages being ranked for repairing broken links. The good terms are those that improve the effectiveness and bad terms produce opposite effect. Therefore, we can generate two groups of terms; good, and bad. Ideally, we would like to use only the good terms of source page. The characteristics of good, and bad terms are analyzed and defined via a range of term proximity features (as explained above) that serve as a basis for automatically identifying them via a classification approach.

We use LibSVM to train the classification model. The classifier is trained over 10-fold cross validation. Table 7 shows the distribution of good and bad terms on the training dataset. Table 8 shows the classification accuracy of LibSVM on the training dataset. The overall accuracy of positive classified samples is *83%*. The good terms class achieved classification accuracy of *84%*, and the bad terms class achieved classification accuracy of *82%*. After training classifier we use the classifier for generating query terms from the content of source page.

Tables 9 and 10 show the effectiveness when (broken link repairing) queries are generated from the content of source page using term classification approach. We perform experiments with

Table 9
Effectiveness of generating queries from source page using term classification and without term classification on *"introduction to computers"* collection.

	Retrieval	Recall			MRR	Cosine similarity	
	Model	R@1	R@3	R@10	MRR	Sim@1	Sim@3
Without Classifier	*SourcePage_tf*	0.29	0.36	0.47	0.30	0.62	0.70
	SourcePage_df	0.30	0.36	0.48	0.32	0.61	0.70
	SourcePage_tfxidf	0.29	0.37	0.48	0.31	0.63	0.70
	SourcePage_bm25	0.43	0.48	0.54	0.46	0.62	0.78
	SourcePage_LM2000	0.48	0.51	0.57	0.49	0.61	0.78
With Classifier	*SourcePage_tf*	★**0.31**	★**0.38**	★**0.51**	★**0.30**	★**0.64**	★**0.71**
	SourcePage_df	★**0.32**	★**0.37**	★**0.49**	★**0.34**	★**0.63**	★**0.71**
	SourcePage_tfxidf	★**0.30**	★**0.38**	★**0.50**	★**0.32**	★**0.65**	★**0.71**
	SourcePage_bm25	★**0.55**	★**0.59**	★**0.65**	★**0.57**	★**0.64**	★**0.80**
	SourcePage_LM2000	★**0.57**	★**0.60**	★**0.64**	★**0.57**	★**0.65**	★**0.79**

Table 10
Effectiveness of generating queries from source page using term classification and without term classification on *"agriculture"* collection.

	Retrieval	Recall			MRR	Cosine similarity	
	Model	R@1	R@3	R@10	MRR	Sim@1	Sim@3
Without Classifier	*SourcePage_tf*	0.51	0.54	0.66	0.51	0.33	0.60
	SourcePage_df	0.55	0.60	0.66	0.57	0.39	0.65
	SourcePage_tfxidf	0.57	0.62	0.68	0.58	0.37	0.65
	SourcePage_bm25	0.61	0.66	0.72	0.62	0.36	0.68
	SourcePage_LM2000	0.71	0.74	0.78	0.71	0.32	0.67
With Classifier	*SourcePage_tf*	★**0.53**	★**0.56**	★**0.69**	★**0.53**	★**0.38**	★**0.63**
	SourcePage_df	★**0.55**	★**0.63**	★**0.68**	★**0.69**	★**0.42**	★**0.68**
	SourcePage_tfxidf	★**0.59**	★**0.64**	★**0.69**	★**0.62**	★**0.41**	★**0.68**
	SourcePage_bm25	★**0.67**	★**0.72**	★**0.77**	★**0.69**	★**0.38**	★**0.70**
	SourcePage_LM2000	★**0.75**	★**0.80**	★**0.81**	★**0.74**	★**0.37**	★**0.70**

the same collection and queries that are used when the terms of queries from the content of source page are selected without term classification approach. We use URL and anchor text of pseudo broken links for defining feature scores of terms of the source page. Tables 9 and 10 show results of effectiveness when the terms from source page are selected using term classification approach and without term classification approach. According to the obtained results term classification have achieved better effectiveness than generating queries without term classification approach on all effectiveness measures. This is because term classifier filters all those terms of source page that have low relevance with URL and anchor text and this increases the effectiveness of retrieved results.

7. Combining sources using learning to rank

Since the information available in different sources is complementary, it is useful to combine sources (features) to gain improvement in effectiveness. We use learning to rank technique to help *"learn"* the feature combination (Liu, 2009; Wang et al., 2009, 2010). The expectation is that a feature combination that works well on a training set will also generate reasonable effectiveness on unseen queries for repairing broken link. In literature many machine learning techniques are used for learning to rank. We use genetic programming based learning to rank (Yeh et al., 2007; Wang et al., 2009; Cummins and O'Riordan, 2009). Genetic Programming (GP) is a branch of evolutionary computing used to combine features (Poli et al., 2008). It helps to solve problems that require to explore exhaustive search. One important benefit of GP is that user does not need to specify the structure of solution in advance. GP works with the help of two features; (a) generating initial population, and (b) recombination of existing population to evolve better solutions. Initial population is generated randomly and it is represented in the form of trees. Each tree of initial population represents a solution and it is structured with the help of nodes. Nodes can be either operators (features) or operands (terminals). Then recombination occurs (from the initial population) with the help of crossover and mutation to evolve efficient solutions. This is called population of next generation. To evolve better generation fittest individuals from the current population are selected in a large percentage. Fitness of individuals is calculated through a fitness function which measures how well and individual performs in its environment. This process of recombination iterates again and again and stops when either there is no improvement in the fitness function or a predefined number of generations are reached. Important parameters of GP are; (a) the population size, (b) the number of generations, (c) the depth of tree, (d) the function set, and (e) the terminal set.

Our proposed GP based learning to rank framework is as follows.

Every individual *I* of the current population represents a retrieval model which processes queries. It is a functional expression and has two components: *Sv* (features), and *Sop* (operators). *Sv* is a set that it contains all ranking features. We perform a query-based normalization (using min–max normalization) on all features in order to normalize feature values into a range of [0, 1]. *Sop* contains a set of arithmetic operators $(+, /, *)$. Individual *I* is represented in the form of a binary tree structure in which leaf nodes are ranking features and internal nodes are arithmetic operators to combine ranking features. In experiments, we generate the trees up to only 6 (branch) levels and runs the GP for only 150 generations. Furthermore, we generate 50 individuals in each generation.

The evolution process then works as follows. The initial population is generated randomly using ramped half-and-half method (Koza, 1992; Poli et al., 2008) and during generation it is ensured that half of the population must not have all the branches of the maximum tree depth. In order to survive the fittest individuals of current population top 10% fittest individuals of the current generation are moved to the next generation without any modification. The remaining population is then generated through 70% by crossover and 30% by mutation. Parents for crossover are selected using 5-tournament selection approach for removing any kind of bias. Crossover produces two new individuals and it is performed on parents by switching their sub-trees with each other. We randomly select sub-trees for the crossover. To perform mutation we first randomly select an internal node of individual and then we replace the whole sub-tree of node with a randomly generated tree. When the evolution completes its processing a set *O* representing the best individuals are returned as a fittest solutions.

7.1. Training dataset and fitness function

For effective training of learning to rank it is important to select representative samples for training dataset. To avoid over-fitting we select 50 pseudo broken links randomly (that are not part of 1000 queries) from the *"introduction to computers"* collection. We use *R*@1 (recall at rank position 1) as a fitness function.

7.2. Effectiveness

To ensure whether the learning to rank approach presented above is working as expected we analyse through following two factors.

- The maximum-fitness individual from each generation computed on the training data. This allows one to verify that the genetic programming method is functioning, as well as indicating that some improvement over base-line-retrieval models can be made.
- The average fitness for each generation can be examined to further ensure that the system is functioning as desired; a sharp initial rise is expected, after which average fitness should slowly (but not monotonically) rise.

Fig. 3 shows the improvement in effectiveness that is gained by the fittest individuals of each generation. As expected, effectiveness increases in a non-strictly monotonic manner as the generations evolve. Fig. 3 also shows how the average fitness improves within each generation as the generations evolve. Within the first few generations, a large number of individuals yield only

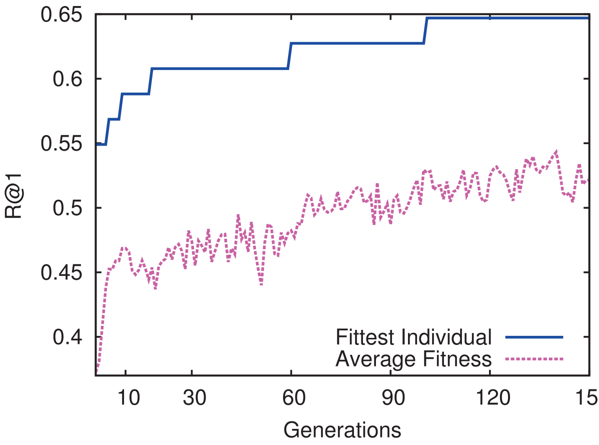

Fig. 3. Improvement gained with R@1 on the basis of average effectiveness of all the individuals and fittest individuals of each generation as the generations evolve.

Table 11
Fittest top three individuals evolved with genetic programming after 150 generations on training dataset.

Rank	Retrieval Model
1	$(((((URL\_tfxidf * (SourcePage\_tf + URL\_tf + Anchor\_tf))$ $*URLContext\_LM2000) + URL\_LM2000) + (Anchor\_df$ $*(Anchor\_lm * \frac{Anchor\_df}{URLContext\_tf}))) + ((SourcePage\_bm25 + URL\_bm25$ $+Anchor\_bm25) + (((URL\_tf * URLContext\_tfxidf) * URL\_tfxidf)$ $*(SourcePage\_LM2000 + URL\_LM2000 + Anchor\_LM2000))))$
2	$(((((URL_tf * (SourcePage\_tfxidf + URL\_tfxidf + Anchor\_tfxidf))$ $*URLContext\_LM2000) + URL\_LM2000)$ $+\left(Anchor\_df * \left(Anchor\_LM2000 * \frac{(SourcePage\_LM2000+URL\_LM2000+Anchor\_LM2000)}{URL\_tfxidf}\right)\right))$ $+((SourcePage\_bm25 + URL\_bm25 + Anchor\_bm25) + ((((SourcePage\_df$ $+URL\_df + Anchor\_df) * (SourcePage\_tfxidf + URL\_tfxidf + Anchor\_tfxidf))$ $*URL\_tfxidf) * (SourcePage\_LM2000 + URL\_LM2000 + Anchor\_LM2000))))$
3	$((((URL\_tfxidf * URLContext_LM2000) + URL\_LM2000)$ $+\left(Anchor\_df * \left(Anchor\_LM2000 * \frac{Anchor\_LM2000}{URLContext\_tfxidf}\right)\right))$ $+((SourcePage\_bm25 + URL\_bm25 + Anchor\_bm25)$ $+(URL\_tfxidf * (SourcePage\_LM2000 + URL\_LM2000 + Anchor\_LM2000))))$
hline	

Table 12
Effectiveness of Learning to Rank on *"introduction to computers"* collection. We compare the effectiveness of *"Learning to Rank"* when terms for query are selected from content of source page using term classification approach.

Query source	**Retrieval**	Recall			MRR	Cosine similarity	
	Model	R@1	R@3	R@10	MRR	Sim@1	Sim@3
SourcePage	*SourcePage_tf*	0.31	0.38	0.51	0.30	0.64	0.71
	SourcePage_df	0.32	0.37	0.49	0.34	0.63	0.71
	SourcePage_tfxidf	0.30	0.38	0.50	0.32	**0.65**	0.71
	SourcePage_bm25	0.55	0.59	**0.65**	**0.57**	0.64	**0.80**
	SourcePage_LM2000	**0.57**	**0.60**	0.64	**0.57**	**0.65**	0.79
	Learning to Rank	**★0.63 (+10.52%)**	**★0.65 (+8.33%)**	**★0.70 (+7.69%)**	**★0.63 (+10.52%)**	**★0.76 (+16.92%)**	**★0.88 (+10.00%)**

Table 13
Effectiveness of Learning to Rank on *"agriculture"* collection. We compare the effectiveness of *"Learning to Rank"* when terms for query are selected from content of source page using term classification approach.

Query source	Retrieval	Recall			MRR	Cosine similarity	
	Model	R@1	R@3	R@10	MRR	Sim@1	Sim@3
SourcePage	*SourcePage_tf*	0.53	0.56	0.69	0.53	0.38	0.63
	SourcePage_df	0.55	0.63	0.68	0.69	**0.42**	0.68
	SourcePage_tfxidf	0.59	0.64	0.69	0.62	0.41	0.68
	SourcePage_bm25	0.67	0.72	0.77	0.69	0.38	**0.70**
	SourcePage_LM2000	**0.75**	**0.80**	**0.81**	**0.74**	0.37	**0.70**
	Learning to Rank	**★0.81 (+8.00%)**	**★0.85 (+6.25%)**	**★0.88 (+8.64%)**	**★0.81 (+9.46%)**	**★0.53 (+26.19%)**	**★0.85 (+21.43%)**

nonsensical weights. These provide poor effectiveness. Then after few generations, these individuals quickly die out, resulting in the dramatic improvements on average fitness for the first few generations. Once the system stabilizes, average fitness rises very slowly over the course of a large number of generations. When all generations complete their execution, the fittest individual having highest $R@1$ score is used for effectiveness analysis. Table 11 shows top three individuals evolved through GP based learning to rank. We use only first individual for analyzing its effectiveness. Tables 12 and 13 show the effectiveness of *Learning to Rank*. It is clear that *Learning to Rank* outperforms all sources on all effectiveness measures. In Tables 12 and 13 results, *Learning to Rank* obtains significant improvement over the most effective source (content of page containing broken link). For instance on *"introduction to computers"* collection, the relative improvements of *Learning to Rank* over content of source page with $R@1, R@3, R@10, MRR, Sim@1$ and $Sim@3$ are +10.52%, +8.33%, +7.69%, 10.52%, 16.92% and 10.00% respectively., and on *"agriculture"* collection, the relative improvements of *Learning to Rank* over content of source page are +8.00%, +6.25%, +8.64%, 9.46%, 26.19% and 21.43% respectively.

8. Conclusion

Broken links is a fundamental problem in contextual information portals. The work presented in the paper provides a method (*brLinkRepair*) to repair broken links using information retrieval. For each broken link, *brLinkRepair* recommends related page that is similar to the pointed page of broken link. One main technique that *brLinkRepair* utilizes is query generation mechanism to search related pages. The task of query generation is to search relevant terms from information sources of the page which contain broken link for constructing queries to retrieve similar web pages. The experiments presented in the paper show that the relevant terms need to be selected very carefully otherwise irrelevant terms drift the retrieved results to noisy (irrelevant) pages. To achieve this, we explore several sources (e.g. anchor text, context around URL, URL and full text of page containing broken link) for searching relevant terms by analzying their proximity relationship to anchor text and URL. To further improve the effectiveness of system we combine all information sources using learning to rank approach. Our results indicate that the combination of sources improves the effectiveness over just relying on a single source.

References

Aggarwal, C.C., Al-Garawi, F., Yu, P.S., 2001. Intelligent crawling on the world wide web with arbitrary predicates. Proceedings of the 10th International Conference on World Wide Web. ACM, New York, NY, USA, pp. 96–105.

Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S., 2001. Searching the web. ACM Trans. Internet Technol. 1 (1), 2–43.

Arour, K., Zammali, S., Bouzeghoub, A., 2015. Test-bed building process for context-aware peer-to-peer information retrieval evaluation. Int. J. Space-Based Situated Comput. 5 (1).

Azzopardi, L., de Rijke, M., Balog, K., 2007. Building simulated queries for known-item topics: an analysis using six european languages. In: SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23–27.

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

Bouramoul, A., 2016. Contextualisation of information retrieval process and document ranking task in web search tools. Int. J. Space-Based Situated Comput. 6 (2).

Chakrabarti, S., van den Berg, M., Dom, B., 1999. Focused crawling: A new approach to topic-specific web resource discovery. In: Proceedings of the Eighth International Conference on World Wide Web, WWW '99. Elsevier North-Holland, Inc., New York, NY, USA, pp. 1623–1640.

Chang, C.-C., Lin, C.-J., 2011. Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3).

Chauhan, N., Sharma, A.K., 2007. Analyzing anchor-links to extract semantic inferences of a web page. In: Proceedings of the 10th International Conference on Information Technology, ICIT '07. IEEE Computer Society, Washington, DC, USA, pp. 277–282.

Chen, J., Kuppusamy, T.K., Subramanian, L., 2010. Contextual information portals. In: AAAI Spring Symposium Artificial Intelligence for Development.

Chen, J., Power, R., Subramanian, L., Ledlie, J., 2011. Design and implementation of contextual information portals. In: WWW (Companion Volume), pp. 453–462.

Chen, J., Subramanian, L., 2013. Interactive web caching for slow or intermittent networks. In: Proceedings of the 4th Annual Symposium on Computing for Development, ACM DEV-4 '13. ACM, New York, NY, USA, pp. 5:1–5:10.

Cummins, R., O'Riordan, C., 2009. Learning in a pairwise term-term proximity framework for information retrieval. In: SIGIR '09: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp. 251–258.

Dang, V., Croft, B.W., 2010. Query reformulation using anchor text. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10. ACM, New York, NY, USA, pp. 41–50.

Du, B., Demmer, M., Brewer, E., 2006. Analysis of www traffic in Cambodia and Ghana. In: Proceedings of the 15th International Conference on World Wide Web, WWW '06, pp. 771–780.

Eiron, N., McCurley, K.S., 2003. Analysis of anchor text for web search. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '03. ACM, New York, NY, USA, pp. 459–460.

Fox, A., Brewer, E.A., 1996. Reducing www latency and bandwidth requirements by real-time distillation. Comput. Networks ISDN Syst. 28 (7–11), 1445–1456.

Fred Douglis, T.B., Tih-Farn Chen, Y.-F.C., Koutsofios, E., 1998. The at&t internet difference engine: tracking and viewing changes on the web. World Wide Web 1 (1), 27–44.

Harrison, T.L., Nelson, M.L., 2006. Just-in-time recovery of missing web pages. In: Proceedings of the 17th Conference on Hypertext and hypermedia, HYPERTEXT '06.

Hersovici, M., Jacovi, M., Maarek, Y.S., Pelleg, D., Shtalhaim, M., Ur, S., 1998. The shark-search algorithm. an application: Tailored web site mapping. Comput. Networks ISDN Syst. 30 (1–7), 317–326.

Ihm, S., Park, K., Pai, V.S., 2010. Towards understanding developing world traffic. In: Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions, NSDR '10, pp. 8:1–8:6.

Isaacman, S., Martonosi, M., 2008. Potential for collaborative caching and prefetching in largely-disconnected villages. In: Proceedings of the 2008 ACM Workshop on Wireless Networks and Systems for Developing Regions, WiNS-DR '08. ACM, New York, NY, USA, pp. 23–30.

Johnson, D.L., Belding, E.M., Almeroth, K., van Stam, G., 2010. Internet usage and performance analysis of a rural wireless network in Macha, Zambia. In: Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions, NSDR '10, pp. 7:1–7:6.

Klein, M., Nelson, M.L., 2008. Revisiting lexical signatures to rediscover web pages. In: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries, ECDL '08.

Koza, J.R., 1992. A genetic approach to the truck backer upper problem and the inter-twined spiral problem. In: Proceedings of IJCNN International Joint Conference on Neural Networks, volume IV. IEEE Press, pp. 310–318.

Li, L.D., Chen, J., 2013. Trotro: web browsing and user interfaces in rural Ghana. In: Proceedings of the Sixth International Conference on Information and Communication Technologies and Development: Full Papers – Volume 1, ICTD '13. ACM, New York, NY, USA, pp. 185–194.

Libby Levison, W.T., Amarasinghe, S., 2002. Providing web search capability for low-connectivity communities. In: Proceedings of the 2002 International Symposium on Technology and Society: Social Implications of Information and Communication Technology, pp. 87–91.

Liu, T.-Y., 2009. Learning to rank for information retrieval. Found. Trends Inf. Retrieval 3 (3), 225–331.

Marentes, L., Wolf, T., Nagurney, A., Donoso, Y., Castro, H., 2014. NETNOMICS: Econ. Res. Elect. Networking 15 (3), 183–213.

Martinez-Romo, J., Araujo, L., 2008. Recommendation system for automatic recovery of broken web links. In: Proceedings of the 11th Ibero-American conference on AI: Advances in Artificial Intelligence, IBERAMIA '08.

Martinez-Romo, J., Araujo, L., 2012. Updating broken web links: an automatic recommendation system. Inf. Process. Manage. 48 (2), 183–203.

Michel, S., Nguyen, K., Rosenstein, A., Zhang, L., Floyd, S., Jacobson, V., 1998. Adaptive web caching: towards a new global caching architecture. Comput. Networks ISDN Syst. 30 (22–23), 2169–2177.

Mihalcea, R., Csomai, A., 2007. Wikify!: Linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07, pp. 233–242.

Milne, D., Witten, I.H., 2008. Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on Information and knowledge management, CIKM '08.

Mishra, S.M., Hwang, J., Filippini, D., Moazzami, R., Subramanian, L., Du, T., 2005. Economic analysis of networking technologies for rural developing regions. In: Proceedings of the First International Conference on Internet and Network Economics, WINE'05. Springer-Verlag, Berlin, Heidelberg, pp. 184–194.

Nakamizo, A., Iida, T., Morishima, A., Sugimoto, S., Kitagawa, H., 2005. A tool to compute reliableweb links and its applications. In: Proceedings of the 21st International Conference on Data Engineering Workshops, ICDEW '05.

Pant, G., 2003. Deriving link-context from html tag tree. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD '03. ACM, New York, NY, USA, pp. 49–55.

Pejovic, V., Johnson, D., Zheleva, M., Belding, E., Parks, L., van Stam, G., 2012. Broadband adoption – the bandwidth divide: Obstacles to efficient broadband adoption in rural sub-saharan africa. Int. J. Commun. 6.

Pentland, A., Fletcher, R., Hasson, A., 2004. Daknet: rethinking connectivity in developing nations. Computer 37 (1), 78–83.

Poli, R., Langdon, W.B., McPhee, N.F., 2008. A Field Guide to Genetic Programming. Lulu Enterprises, UK Ltd.

Rabinovich, M., Spatschek, O., 2002. Web Caching and Replication. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Robertson, S.E., Walker, S., 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, pp. 232–241.

Saif, U., Chudhary, A.L., Butt, S., Butt, N.F., 2007. Poor man's broadband: peer-to-peer dialup networking. Comput. Commun. Rev. 37 (5), 5–16.

Tao, T., Zhai, C., 2007. An exploration of proximity measures in information retrieval. In: SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp. 295–302.

Voorhees, E.M., 2001. The trec question answering track. Natural Language Engineering 7 (4), 361–378.

Wang, S., Ma, J., He, Q., 2010. An immune programming-based ranking function discovery approach for effective information retrieval. Expert Syst. Appl. 37 (8), 5863–5871.

Wang, S., Ma, J., Liu, J., 2009. Learning to rank using evolutionary computation: Immune programming or genetic programming? In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09. ACM, New York, NY, USA, pp. 1879–1882.

Yeh, J.Y., Lin, J.Y., Ke, H.R., Yang, W.P. (2007). Learning to rank for information retrieval using genetic programming. In: SIGIR 2007 workshop: Learning to Rank for Information Retrieval.

Zaki, Y., Chen, J., Pötsch, T., Ahmad, T., Subramanian, L., 2014. Dissecting web latency in Ghana. In: Proceedings of the 2014 Conference on Internet Measurement Conference, IMC '14. ACM, New York, NY, USA, pp. 241–248.

Zhai, C., 2002. Risk minimization and language modeling in text retrieval. ACM SIGIR Forum 36 (2), 100–101.

Zhao, J., Yun, Y., 2009. A proximity language model for information retrieval. In: SIGIR '09: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, pp. 291–298.