Finding Relevant Web Pages Through Equivalent Hyperlinks

Simon Courtenage and Steven Williams,
Cavendish School of Computer Science,
University of Westminster,
115 New Cavendish Street, London, UK.
{courtes, williast}@wmin.ac.uk

Abstract. Finding pages on the web that are relevant to some user-defined criteria is a long-established area of research. Early work on search engines concentrated on the textual content of web pages to find relevant pages, but in recent years, the analysis of information encoded in hyperlinks has been used to vastly improve search engine performance. This paper presents a variation on the use of link analysis for automatically categorizing web pages, by defining a similarity measure and search technique based on hyperlinks. This measure is used to categorize hyperlinks themselves, rather than web pages. This allows us to, given a particular hyperlink, to find others like it in terms of its context and referred-to web pages. We have implemented a prototype version of the technique as a browser application. Initial tests appear to show that the method, as implemented in our browser, returns precise, well-focussed results.

1 Introduction

Given the vast size, structure and state of the web, finding web pages that are in some way relevant is a recurring problem for most people. Unsurprisingly, therefore, search engines such as Google have risen to the level of cultural icons of international prominence through their ability to find web pages that satisfy a user's query.

Many early search techniques and algorithms for finding relevant web pages concentrated on the content of the web pages themselves as a means for assessing relevance, under the assumption that the (usually textual) content of a page gives the best clues to its semantics. In many cases, however, pages do not contain sufficient textual information to allow them to be matched to a textually-formed query. Nor does this approach allow results to be ranked according to their usefulness to the user (other than by how well keywords in a page match keywords in the user query). Hence, over recent years, research has looked at how the structure of the web graph can be used in answering web search queries. For example, looking at keywords in web pages around the anchors that refer to a web page allow you to infer additional information about the page referred to [7] [10] [1] [13]. Link analysis of the global web graph (or focussed sub-graphs) is also used to rank web pages in order of importance (see, for example, [2] [11]).

In this paper, we present a web search technique, based on a form of link analysis, that is designed to find pages relevant to a web page that a user is considering visiting using a hyperlink. Rather than asking the user to make a text-based search query that can be input to a search engine, we use the hyperlink as the basis for the query, with the aim of finding similarly-defined hyperlinks. Our approach may be summed up by saying "find me hyperlinks that could be substituted for the one I am using". For example, if

the user views the web page `http://en.wikipedia.org/wiki/Clint_Eastwood` and is interested in one of the links on that page (for example, to the film *For a Few Dollars More* on page `http://en.wikipedia.org/wiki/For_A_Few_Dollars_More`), then our method attempts to find other hyperlinks to a similar subject from a similar anchor, such that the user could very well visit these other pages instead of the one referred to in the anchor of the currently viewed page. Our work can be viewed as a variation of the localized link analysis used to augment understanding on the semantics of a particular page, as in [7] and [10], but with the difference that, using the context of the anchor of a link and the context of the page referred to, we attach a semantics to the hyperlink rather than to the page.

The benefits of this approach are that it provides links that match the navigational context of the user - given a particular link that the user could follow, it finds other hyperlinks that start from similar anchors and refer to similar pages. Another benefit is that the returned list of found links is very likely to be much smaller in size given the search constraints, and hence more manageable, than if the user had typed in an equivalent search query into a search engine.

The general-purpose search engine approach, on the other hand, does not take into account the overall navigational context within which a user makes a search query. For example, when browsing a web page about Napoleon's defeat at Waterloo, a user may wish to find out more about his exile on Corsica. In a search engine, they may type "Napoleon AND Corsica". In Google, this returns about 17,500 results. Not all of these results are concerned with Napoleon's exile on Corsica following his defeat at Waterloo (in fact, the first result returned by Google is about Napoleon Avenue in Corsica, a district of San Diego, USA[1]. However, we can focus the search on behalf of the user if we take into account their present navigational context. If, on a page about Napoleon at Waterloo, a user decides to click on a link to his exile at Corsica, then we can view the act of clicking on the link as a kind of query within a tightly-defined context, where that context is provided by the information provided by the anchor of the link and by the page referred to by the link.

The rest of the paper is organized as follows: in the following section, we explain the concept of equivalent hyperlinks. In Section 3, we provide a basic algorithm for calculating equivalent hyperlink sets. Finally, we discuss related work and conclude with some discussion of further work.

2 Equivalent Hyperlinks

Most research into web searching using links considers the web to be a graph data structure in which pages form nodes and hyperlinks form edges between the nodes of the graph. In this view of the web, as in Figure 1 below, each hyperlink is separate and distinct from every other hyperlink.

An alternative view of hyperlinks, however, is that of a function. Each hyperlink represents a mapping which is part of a definition of a function H which maps from web pages and their anchors to web pages. The type of such a function H is

$$\mathrm{H} :: (URL, Anchor) \rightarrow URL$$

In other words, given the URL of a web page and the identity of one of the anchors in the page, the function returns the URL of another web page (the URL mentioned in

[1]Search carried out on 24/02/2004.

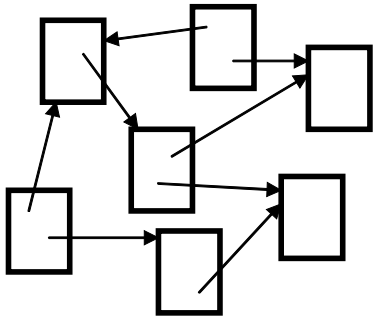

Figure 1: The Graph-Based View of the Web

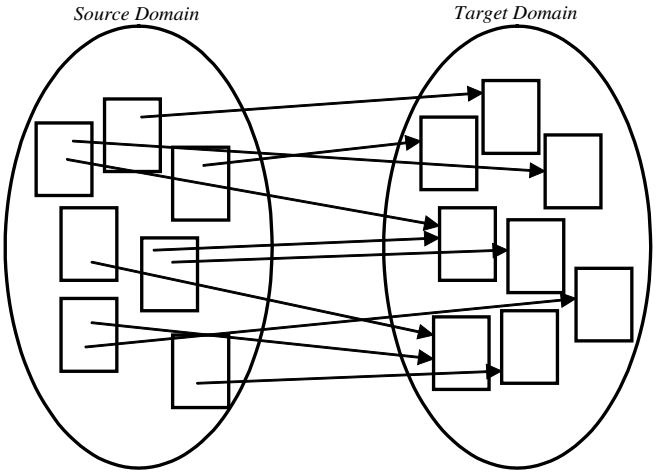

Figure 2: A Functional View of the Web

the anchor's href attribute). Note that this represents a *many-to-one* mapping, since anchors in web pages maps to only one web page, but each web page may be mapped to by many distinct anchors. From this perspective, hyperlinks are not separate and distinct, but represent the same function such that each individual hyperlink forms one of the mappings from one value to another that make up the function definition. Figure 2 describes the functional view of the web, in which web pages and anchors make up values in the source domain, and web pages are values of the target domain. In this model, each hyperlink is shown as a mapping of the hyperlink function from the source domain to the target.

By itself, thinking of hyperlinks as mappings of a single function is not very interesting. Since each mapping in the set of mappings that defines the function is equivalent to the others, it does not yield up any interesting information. It is more informative to think of hyperlinks as making up a set of different functions. One set of hyperlinks mapping from certain anchors in one set of pages to a set of web pages would therefore make up a different function to another set of hyperlinks. One way to divide hyperlinks into distinct sets is on the basis of what they are mapping from to what they are mapping to. In other words, hyperlinks can be distinguished using the text of their anchor (and possibly a wider context) and the content of the page they refer to. For example, consider two web pages A and B, where web page A contains the anchor

```
<a href="Corsica.html">Napoleon on Corsica</a>
```

and where web page B contains the anchor

```
<a href="napoleon_exile.html">Napoleon's exile</a>
```

The first link is to a web page about Napoleon's residence on Corsica, while the second is about Corsica and Napoleon's connection with the island. There is a correlation between the two hyperlinks in terms of the anchor text and the content of the web page referred to for each hyperlink. In our proposed model, each of these hyperlinks is an instance mapping of the same hyperlink function, a function that maps from anchor text references to Napoleon to web pages concerning Corsica.

The benefit of this approach is that given a particular hyperlink, we can consider all the other hyperlinks representing mappings of the same function as leading to pages that are potentially similar to the one the current hyperlink maps to. If the user is sufficiently interested in the current link to follow it and visit the page referred to, then they may also be interested in the other pages referred to by similar hyperlinks. In the example above, therefore, if the user is viewing web page A and is about to click on the anchor linking to the page `Corsica.html`, then they may also be interested in the page `napoleon_exile.html`.

3 Calculating Equivalent Hyperlinks Sets

In this section, we describe an algorithm for calculating sets of equivalent hyperlinks, such that each set of hyperlinks defines a hyperlink function as defined above. The $\mathcal{EH}$ (Equivalent Hyperlink) algorithm is naive in the sense that it requires a large amount of cross-checking of URLs in order to calculate set membership. The algorithm works over hyperlinks such that given a hyperlink, in terms of its anchor text and target page, it produces a list of URLs representing the target pages of hyperlinks similar to the input hyperlink. The means by which similarity of actual page content is judged is independent of the algorithm - hence we assume the use of standard web search techniques to test for similarity of page content where the algorithm requires it.

The algorithm (presented in pseudocode format) is as follows:

```
procedure EH((S,a),T) {
   Similar_to = FindSimilar(T)
   Link_to = { b |∀t ∈ Similar_to∀b ∈ FindBacklinks(t)}
   SimilarAnchor_to = FindSimilarAnchor(a)
   SimilarLinks = {t|∀t ∈ Similar_to;∀u ∈ Link_to ∩ Similar_to;
                   t ∈ Links(Page(u)) }
   return SimilarLinks
}
```

The input to the $\mathcal{EH}$ algorithm is a hyperlink `((S,a),T)`, represented by the source web page `S` and the anchor text `a` in `S`, and the target web page `T`. The result is a set of URLs judged to be similar to the input hyperlink. This set of hyperlinks are taken to define a particular hyperlink function, such that any of the hyperlinks in the set could be substituted for any other hyperlink in the same set. Hence any of the pages pointed to could be visited in place of another.

The algorithm uses functions `FindSimilar` and `FindSimilarAnchor` to perform content-based searching. The `FindSimilar` function, given a particular web page as

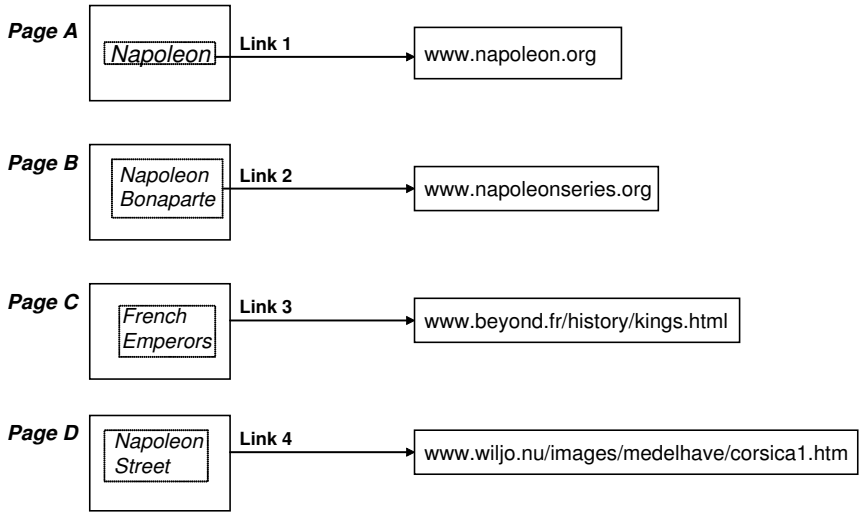

Figure 3: Web pages and links for Napoleon

input, returns a set of URLs for web pages that are judged to be similar in content. The algorithm does not specify how this should be done: in fact, the actual method employed to implement `FindSimilar` is independent of the $\mathcal{EH}$ algorithm. The way in which the `FindSimilarAnchor` function locates pages with anchors similar to the anchor `a` in web page `S` is also left unspecified.

As an example of how the algorithm works, consider the web pages and links in Figure 3. If the user is currently viewing web page A and wants to click on the anchor with text "Napoleon", the input to the $\mathcal{EH}$ algorithm is the value

```
((A,"Napoleon"),www.napoleon.org)
```

The steps taken by the algorithm are as follows:

1. **Calculate the set of web pages similar to www.napoleon.org**
   The function `FindSimilar` is called on the web page www.napoleon.org to find web pages with similar content. This set of web pages is denoted by `Similar_to`. In our example, this set contains URLs

   ```
   www.napoleon.org
   www.napoleonseries.org
   www.beyond.fr/history/kings.html
   ```

2. **Find web pages linking to the set of similar web pages**
   For each page in the set `Similar_to`, find those web pages that link to it. The set of all web pages linking to pages in the set `Similar_to` is denoted by `Link_to`. Our example set in this case contains the web pages **A**, **B** and **C**.

3. **Find all pages with similar anchor text to "Napoleon"**
   Function `FindSimilarAnchor` is called to find all web pages containing an anchor with text similar to "Napoleon". This set of web pages is known as `SimilarAnchor`. In the example, `SimilarAnchor` contains pages **A**, **B**, and **D**.

4. **Cross-reference URLs**
   The URLs of pages that link to pages similar to www.napoleon.org (the set

`Link_to`) and of pages with anchor text similar to "Napoleon" (the set `SimilarAnchor`) are cross-referenced to find pages with anchors whose text is similar to "Napoleon" and whose targets are pages similar to www.napoleon.org (the set `SimilarLinks`). This implements the set intersection in the definition of `SimilarLinks` shown above. If we cross-reference the example `Link_to` set of {**A**,**B**,**C**} and the example `SimilarAnchor` set of {**A**,**B**,**D**}, we obtain the example SimilarLinks set {**A**,**B**}.

4 Related Work

In recent years, there has been a great deal of research into the use of link analysis in searching and processing information on the web. In general, this research divides into two main areas: localized link analysis for the purpose of augmenting traditional text-based page classification using information from the context of anchors to a page in other pages; and global link analysis for the purpose of calculating the relevance or importance of a page.

The first area of research identified concentrates on using the semantic clues provided by the anchor or context of the anchor in the referring page to augment the semantics of the referred-to web page [1] [6] [7] [8] [10] [14].

For example, Fürnkranz [7] [8] investigates the use of anchor text and its context in classifying web pages, based on the observation that web pages often contain insufficient information in order to accurately classify them, while information in or around links to pages is often more helpful. Fürnkranz's method attempts to classify a page solely on the basis of information from links to the page. A learning algorithm is then trained using such example representations of web pages to classify web pages according to their collective context. Attardi et al. [1] also describe a method for the automatic categorization of a web page into a catalogue structure by the context of links to the page. Their tool, Theseus, which analyzes the set of contexts in which a URL was found on other web pages in order to decide which category a URL should be placed in. They report encouraging results compared with Yahoo!'s manual classification.

The aim of link analysis in this case, therefore, is to gather extra knowledge about a web page that is not present in the page itself by looking at the context of anchors of links to the page. Our work is related to this use of link analysis in that we use information from both the anchor (and its context) and the web page linked to, and find web pages similar to the web page linked to. However, we differ in that we do not transfer information from the context of anchors of linking pages to the linked-to page in order to classify the linked-to page. For example, in Fürnkranz's work, two web pages would be considered similar if all anchors in linking pages to the two pages produced similar summary contexts. In our work, two web pages could be considered similar in one navigational context and dissimilar in another, depending on the particular anchor used to create a link to either page. However, if both web pages were linked to by anchors of similar context, then the two pages would also be seen as similar according to Fürnkranz's method of link analysis, and from this perspective, our work is a more refined and more finely detailed version of this kind of localized link analysis.

The second area of research uses knowledge about the context of a web page within the web graph to decide how relevant a page is to a user. Examples of this use of hyperlinks include Brin and Page's work on PageRank [2] [12], and Kleinberg's HITS algorithm [11]. The PageRank algorithm seeks to establish the relevance of a page

by iteratively computing a ranking for all web pages in the web graph based on the relevance of pages that link to them. With HITS, Kleinberg also uses link analysis of a focused web graph based on (but not limited to) pages found by text-based search in order to find authorities (pages that are linked to by a large number of other pages in the graph) and *hubs* (pages that link to a large number of authorities). The HITS algorithm also serves as the basis for CLEVER [5] [4], which modifies the calculation of the web graph by looking at pages two links distant from the set of pages found by text-based search. The work on HITS has also led to work identifying web-based *communities*, collections of web pages that feature a high proportion of authorities on a particular topic linked together by hubs [9].

Our work does not attempt to calculate the importance or relevance of a page from the link structure of the web graph. Hence, Brin and Page's work on PageRank, Kleinberg et al's work on HITS and web communities, nor the work on web graph structure (for example [3]) is not directly relevant. However, the results output by our $\mathcal{EH}$ algorithm could easily be augmented by algorithms such as PageRank or HITS in order to assess whether or not the suggested pages are more relevant to the semantics of the hyperlink than the page actually linked to.

5 Conclusion and Further Work

The aim of our research is to identify similar hyperlinks, such that, given a hyperlink, embedded in a web page, we can provide the user with a list of other hyperlinks that are semantically similar. We have defined similar hyperlinks in terms of the similarity of their anchors (including possibly their contexts) and target pages. This approach yields a list of hyperlinks that can be considered equivalent in the sense that one hyperlink can be substituted for another in a particular context without loss of meaning to the user.

We have implemented the $\mathcal{EH}$ algorithm in a prototype Java-based browser that communiates with the Google search engine [2] via the Google API [3] in order to implement the actual search functions, i.e, the `FindSimilar` and `FindSimilarAnchor` functions. Our experiments with the prototype browser show that in general a small number of links is produced each time, with reasonably high precision. This appears to indicate that the technique will prove useful. However, it depends on the quality of the anchor text (and its surrounding context), as well as the quality of the search techniques to provide good lists of similar pages for both the target page and anchor text. If the anchor text and/or its context is poor, then (as for all link-based analysis methods), the results are poor. We are currently conducting experiments to assess the performance of our method in comparison with other localized link analysis techniques, as well to asses what impact different implementations of `FindSimilar` and `FindSimilarAnchor` have on the outcome of the algorithm.

Also, although our results compare very well with the results of equivalent queries performed directly in Google, the are obviously affected by the actual search results returned via the Google API. Hence, we are currently defining a data model for a search engine based on the $\mathcal{EH}$ algorithm with the aim of developing a web crawler capable of populating the database. In conjunction with a lightweight browser plugin, this will allow similar links to be provided by means of a simple query, rather than the

[2] http://www.google.com

[3] http://www.google.com/apis/

heavyweight process in our prototype browser. This would also enable us to produce a list of similar links before a link was visited, rather than after, as is the case currently.

References

[1] G. Attardi, A. Gullí, and F. Sebastiani. Automatic Web page categorization by link and context analysis. In Chris Hutchison and Gaetano Lanzarone, editors, *Proceedings of THAI-99, 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence*, pages 105–119, Varese, IT, 1999.

[2] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, April 1998.

[3] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, and R. Stata. Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference*, pages 247–256, 2000.

[4] S. Chakrabarti, B. Dom, D. GIbson, J. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Mining the link structure of the World Wide Web. *IEEE Computer*, 1999.

[5] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the Seventh International World-Wide Web Conference*, 1998.

[6] Soumen Chakrabarti, Byron E. Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In Laura M. Haas and Ashutosh Tiwary, editors, *Proceedings of SIGMOD-98, ACM International Conference on Management of Data*, pages 307–318, Seattle, US, 1998. ACM Press, New York, US.

[7] J. Fürnkranz. Exploiting structural information for text classification on the WWW. *Lecture Notes in Computer Science*, 1642:487–497, 1999.

[8] Johannes Fürnkranz. Hyperlink ensembles: A case study in hypertext classification. *Information Fusion*, 3(4):299–312, 2002.

[9] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and spacestructure in hypermedia systems*, pages 225–234. ACM Press, 1998.

[10] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. Using Web structure for classifying and describing Web pages. In *Proceedings of WWW-02, International Conference on the World Wide Web*, pages 562–569, Honolulu, US, 2002. ACM Press, New York, US.

[11] J. M. Kleinberg. "Authoritive Sources in a Hyperlinked Environment". *J. ACM*, 46(5):604–632, September 1999.

[12] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, Stanford University, Stanford, CA, USA, November 1998.

[13] Seán Slattery and Tom Mitchell. Discovering test set regularities in relational domains. In *Proc. 17th International Conf. on Machine Learning*, pages 895–902. Morgan Kaufmann, San Francisco, CA, 2000.

[14] Yiming Yang, Sean Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241, 2002.