Page Retrievability Calculator

Leif Azzopardi, Rosanne English, Colin Wilkie, David Maxwell

School of Computing Science
University of Glasgow, Scotland, UK
{leif.azzopardi, rosanne.english, colin.wilkie}@glasgow.ac.uk
d.maxwell.1@research.glasgow.ac.uk

Abstract. Knowing how easily pages within a website can be retrieved using the site's search functionality provides crucial information to the site designer. If the system is not retrieving particular pages then the system or information may need to be changed to ensure that visitors to the site have the best chance of finding the relevant information. In this demo paper, we present a Page Retrievability Calculator, which estimates the retrievability of a page for a given search engine. To estimate the retrievability, instead of posing all possible queries, we focus on issuing only those likely to retrieve the page and use them to obtain an accurate approximation. We can also rank the queries associated with the page to show the site designer what queries are most likely to retrieve the pages and at what rank. With this application we can now explore how it might be possible to improve the site or content to improve the retrievability.

1 Introduction

Information Architects and Site Designers uses an array of tools to design and evaluate websites. Most of these tools are qualitative in nature, for example card sorting exercises, heuristic evaluations, and usability studies [6]. However there have been a number of attempts to develop more quantitative measures to help in designing and evaluating websites in terms of how findable they make content. For example, in [5], the authors used Information Foraging Theory to build information scent models that predicted how users would interaction with a website, while in [9, 10] the authors developed measures of Navigability using Markov-Models to predict where users on a website would end up. In this paper, rather than considering how people navigate through a website, we consider how people search for information within a website and how the Retrievability of pages can be estimated and used to help Information Architects and Site Designers improve the findability of content.

Retrievability has been defined as the ease with which a document can be found using a retrieval system [2]. Retrievability has been used in a number of application areas within Information Retrieval. For example it has been used to detect bias within retrieval models [2] and search engines [1], tune retrieval models [8], analyse collections [2, 4], and to improve retrieval performance [3, 7]. In this demo paper we set out to develop another use for retrievability where we wish to examine the retrievability of individual pages and determine the queries that retrieve those pages.

2 System and Method Design

To this end we have designed a command line utility that computes the retrievability of a specified URL for different search engines (called the Page Retrievability Calculator, or PRC). To compute the retrievability of a page a set of queries is first extracted from the page, or part of the page. The queries are then issued to the specified search engine. If the page is retrieved by the search engine the query receives a score based on its rank (either cumulative or gravity based, see [2]). This allows us to rank the queries according to how much they contribute to the overall retrievability of a the page. The total retrievability is computed as the sum of all the query scores. As part of the scoring process a number of components can be varied such as the search engine, the part of the page to extract query terms from, and how to select queries.

2.1 System Design

The main components of the system and how they relate to each other is described below, the source code is available on GitHub at `https://github.com/leifos/ifind`:

- **Page Fetcher** This is represented by the Page Capture class. It is responsible for loading a webpage using PhantomJS and selenium. It allows us to capture the HTML of a webpage.
- **Text Extractor** This is represented by the Position Content Extractor. This class is responsible for reading in html and removing or extracting content of divs with given ids. It is also responsible for getting a subset of the content of the html.
- **Query Extractor** This is represented by a superclass for query generation which is responsible for extracting queries from html or text. This involves cleaning the text by a pipeline which removes features like punctuation, special characters, stop words etc. This class also calculates the number of occurrences of each term in the document for use by the query selector. There are currently three subclasses which generate single term queries, biterm queries, and 3-term queries. The biterms are generated by pairing words which are next to each other. The 3-terms are generated by grouping three terms which are next to each other.
- **Query Selector** The query selector is responsible for calculating the probability of each query given the probability of the document and the collection. It then ranks the queries given these probabilities from most to least probable. It is then possible to get the top n queries, so that only the queries most likely to retrieve the page can be issued (instead of all of them).
- **Page Calculator** This is responsible for calculating the score of a page given a list of queries. It generates query objects and issues them to the search engine, noting the result and calculating the cumulative and gravity based scores. It also provides a report which presents a summary of the results such as the number of queries which returned the page, and the scores.
- **Search Engines** A number of wrappers have been implemented so that different search engines can be used. Currently, we have written wrappers for Gov.uk, Bing and SiteBing (which is Bing restricted to a particular site).

3 Demo

A site designer often wants to know how easily a page can be found, how good different search engines are, and what query terms retrieve a page. With this in mind, Table 1 presents some examples of the PRC applied to two webpages - `www.gov.uk/vehicles-you-can-drive` and `www.gov.uk/renew-adult-passport`. For each page we compare how retrievable the pages are using the site search provided by gov.uk, Bing search (via their API on Azure's Datamarket), and Bing Search but restricted to site:gov.uk (referred to as sitebing). We also provide a comparison of different methods for extracting queries, either using all the text on the page or using the text in main content div (called wrapper). Table 1 shows which engine, portion of the page, along with the number of queries issued, number of times the page was retrieved, and the cumulative and gravity based retrievability scores (where the higher is better).

Run	**Engine**	**Portion**	**#Q issued**	**Retrieved**	$r_c(d)$	$r_g(d)$
Examples 1-6 on gov.uk page vehicles-you-can-drive						
1	gov.uk	all	150	25	22.00	10.10
2	bing	all	150	6	3	2.06
3	sitebing:govuk	all	150	33	30.00	14.03
4	gov.uk	wrapper	65	21	19.00	9.10
5	bing	wrapper	65	6	3.00	2.06
6	sitebing:govuk	wrapper	65	31	29.00	13.90
Examples 7-12 on gov.uk page renew-adult-passport						
7	gov.uk	all	250	161	138.00	61.39
8	bing	all	250	6	4.00	0.50
9	sitebing:govuk	all	250	101	85.00	27.35
10	gov.uk	wrapper	250	185	157.00	65.03
11	bing	wrapper	250	8	7.00	0.87
12	sitebing:govuk	wrapper	250	118	90.00	28.20

Table 1. Retrievability scores for different configurations of the PRC.

Comparing Engines: Immediately we can see that searching for the pages on the open web using Bing means that less of the queries retrieve the page resulting in substantially lower retrievability scores. When we compare the two site search variations (run 1 vs. 3), we observed that for `gov.uk/vehicle-you-can-drive`, Sitebing is more successful, retrieving the page more often (33 times vs 25 times) and so makes the page more retrievable. While for `gov.uk/renew-adult-passport` the gov.uk sitesearch is more successful at retrieving the page, retrieving the page more often and at higher ranks (as denoted by retrievability scores). It would be interesting to do a larger comparison across the gov.uk domain to see which site search system performs the best, and to see if this correlates with standard effectiveness measures.

Comparing Page Extraction: Runs 4-6 and 10-12 show the results of the same comparisons. However, here we have used only part of the page to draw queries from. This is because we hypothesized that the terms in the headers, footers, and sidebars, are unlikely to be useful in retrieving the page. So we can reduce the number of queries issued to determine the retrievability of a page. When we compare the results to when the full page was used, we see that we obtain similar retrievability score (though slightly less on most occasions). For runs 7-12, we limited the number of queries issued to 250 (even though there was more possible queries). Here we see that by choosing from the main content div, we can even obtain higher retrievability scores as the queries are more closely related to the document's topic.

Top 5 Queries: Table 2 shows the top five queries which returned the page for examples 1 to 3 for page `gov.uk/vehicles-you-can-drive`. We can see that for Bing, while it only retrieves the page for 3 queries, the queries are pretty sensible, and are likely to be in-line with what a user might type to find information about the topic. For the site search variants we can see similar queries. However, there are also other queries that are unlikely or less likely to be issued (like "tool tells" and "old enough").

Engine	**Top Queries**				
	1	**2**	**3**	**4**	**5**
gov.uk	drive different	driving transport	except otherwise	tool tells	old enough
Bing	licence drive	licence categories	categories driving	N/A	N/A
SiteBing	drive different	licences quick	adding higher	licences elsewhere	tool tells

Table 2. Top 5 Queries for www.gov.uk/vehicles-you-can-drive for runs 1-3.

Summary: The demo shows some of the utility of the application, but also highlights a number of areas of research and development. In future work we will examine a large subset of pages on different domains, add in new query generation methods that are more realistic, examine how to estimate the likelihood of queries to extract the best possible opens, and explore other configurations. In addition we wish to correlate the retrievability of pages with results from usability studies.

Acknowledgements: This work is supported by the EPSRC Project, *Models and Measures of Findability* (EP/K000330/1).

References

1. Azzopardi, L., Owens, C.: Search engine predilection towards news media providers. In: Proc. of the 32nd ACM SIGIR. pp. 774–775 (2009)
2. Azzopardi, L., Vinay, V.: Retrievability: An evaluation measure for higher order information access tasks. In: Proc. of the 17th ACM CIKM. pp. 561–570 (2008)
3. Bashir, S., Rauber, A.: Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In: Proc. of the 18th ACM CIKM. pp. 1863–1866 (2009)
4. Bashir, S., Rauber, A.: Improving retrievability of patents in prior-art search. In: Proc. of the 32nd ECIR. pp. 457–470 (2010)
5. Chi, E.H., Pirolli, P., Chen, K., Pitkow, J.: Using information scent to model user information needs and actions and the web. In: Proc. of the SIGCHI Conference. pp. 490–497. CHI '01, ACM (2001)
6. Morville, P.: Ambient Findability. O'Reilly Media (2005)
7. Pickens, J., Cooper, M., Golovchinsky, G.: Reverted indexing for feedback and expansion. In: Proc. of the 19th ACM CIKM. pp. 1049–1058 (2010)
8. Wilkie, C., Azzopardi, L.: Relating retrievability, performance and length. In: Proc. of the 36th ACM SIGIR conference. pp. 937–940. SIGIR '13 (2013)
9. Zhang, Y., Zhu, H., Greenwood, S.: Web site complexity metrics for measuring navigability. In: Proc. of the 4th QSIC. pp. 172–179 (2004)
10. Zhou, Y., Leung, H., Winoto, P.: Mnav: A markov model-based web site navigability measure. IEEE Transactions on Software Engineering 33, 869–890 (2007)