

Using String-matching to Analyze Hypertext Navigation

Roy A. Ruddle
University of Leeds, Leeds, UK
+44 (0)113 343 5430
royr@comp.leeds.ac.uk

ABSTRACT

A method of using string-matching to analyze hypertext navigation was developed, and evaluated using two weeks of website logfile data. The method is divided into phases that use: (i) exact string-matching to calculate subsequences of links that were repeated in different navigation sessions (common trails through the website), and then (ii) inexact matching to find other similar sessions (a community of users with a similar interest). The evaluation showed how subsequences could be used to understand the information pathways users chose to follow within a website, and that exact and inexact matching provided complementary ways of identifying information that may have been of interest to a whole community of users, but which was only found by a minority. This illustrates how string-matching could be used to improve the structure of hypertext collections.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and retrieval – clustering, search process. H.5.4 [Information Interfaces and Presentation (e.g., HCI)]: Hypertext/Hypermedia – navigation.

General Terms

Algorithms, Measurement, Human Factors.

Keywords

Navigation, String-matching, Analysis.

1. INTRODUCTION

One continuing challenge in the field of hypertext is to understand how users navigate. Progress in this area allows us to improve the way information is structured [4], enhance browser functionality [6], and develop new techniques for identifying groups (virtual communities) of users who have similar interests, so they can benefit from the navigational efforts of their predecessors.

This article investigates how string-matching can be used to analyze navigation. It is divided into two main sections, which discuss how string algorithms may perform inexact pattern matching of long navigational sequences, and evaluate string-matching using two weeks of data from a case study website.

© ACM, (2006). This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version will be published in the Proceedings of the 17th ACM Conference on Hypertext and Hypermedia (HT'06).

2. String-matching

If hypertext is modeled as a graph, with each link given a unique code and each navigational session performed by a user represented by the sequence of links visited, then string-matching algorithms may be used to find navigational sequences that are either identical, or similar, to each other.

One approach calculates the *longest repeated subsequences* (LRSs) to find identical subsequences of two or more links. For example, comparing Sequences 1 and 2 (see Figure 1a) there are LRSs of length three (ABCDE) and two (DE). In practice it is rare to find long LRSs in hypertext navigation, and in one study the overwhelming majority had a length ≤ 3 [11]. However, many of the sequences were similar (most links were the same, but at least one in each sequence was unique), which led to the suggestion that inexact string matching would be more appropriate.

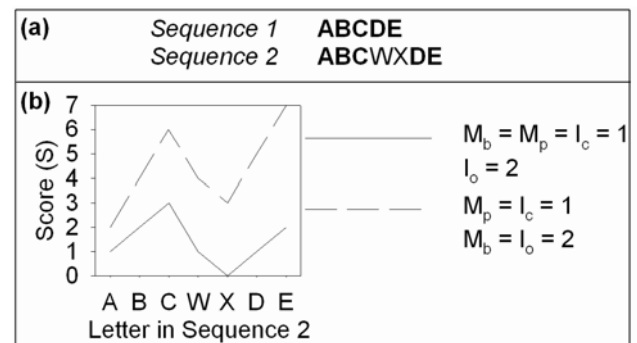


Figure 1. (a) Hypothetical navigation sequences (each letter corresponds to a different hyperlink), and (b) Score generated by inexact matching of Sequence 2 with Sequence 1.

One method of performing inexact string matching involves calculating the *Levenshtein* (edit) distance [5], which is the number of insertions, deletions and substitutions needed to change one string into another. Once an allowable distance d has been defined, LRSs can be calculated. For example, if $d = 2$ then the LRS of Sequences 1 and 2 is ABCDE (W and X are deleted from Sequence 2), but if $d = 1$ then the LRSs are the same as when exact matching was performed (see above). In the real world this approach has been used to analyze people's movements around science exhibitions [7]. In hypertext, the Levenshtein distance has been used with limited success to reconstruct users' navigation paths from logfile data [8] and suggested for use within recommender systems [1].

A more powerful inexact string matching technique is widely used to analyze bioinformatics data (for a detailed description, see [3]), and effectively increases the allowable Levenshtein distance with the length of an LRS. This is achieved by using four parameters: the *match bonus* (M_b) rewards pairs of matching elements, the *mismatch penalty* (M_p) penalizes substitutions, the *insertion origin penalty* (I_o) penalizes the first element in each

insertion, and the *insertion continuation penalty* (I_c) penalizes each subsequent element in that insertion. The score for similar parts of two strings is calculated as $S = aM_b + bM_p + cI_o + dI_c$, where a , b , c and d are the number of matches, mismatches, insertion origins and continuations, respectively.

This technique identifies subsequences from the profile of S (see Figure 1b). For example, if $M_b = M_p = I_c = 1$ and $I_o = 2$, then two subsequences are generated (ABC and DE), but if $M_b = 2$ then the whole of Sequence 2 forms a single subsequence because E's peak is higher than C's. As the ratio of $M_b:M_p$ or $I_o:I_c$ increases, fewer identical elements are needed for two strings to "match".

In bioinformatics $I_o:I_c$ is often in the range 5:1 to 10:1. Ratios of 100:10:10:1 ($M_b:M_p:I_o:I_c$) have been used to match users' navigation paths in virtual reality [9], and the technique has also been investigated for aiding navigation through shared file directories, but few details were provided [2].

3. EVALUATION

3.1 Method

To evaluate string-matching, we analyzed two weeks of logfile data (26 September - 9 October 2005) from the website of the School of Computing at Leeds. This evaluation focuses on how string-matching may be used. A comprehensive comparison with other approaches for analyzing hypertext navigation is outside the scope of this paper.

Prior to analysis, the data were cleaned so only successful requests (status code = 200) for the main types of content (.html, .htm and .pdf file extensions) were included. Next, the navigation path (sequence of links) followed in each user session was reconstructed, with sessions defined using the IP-Timeout method (each IP address was a different session, and new sessions were triggered if more than 25 minutes had elapsed since the last request from a given IP address [8]). Finally, sessions from the IP addresses of our School's computers (almost certainly human users) and known search engine robots (see www.briandunning.com) were flagged (see Table 1).

Table 1. Summary of the logfile data.

Session type	No. sessions	No. successful requests	No. different webpages
School's computers	4,567	27,686	1,777
Known robots	4,329	20,319	6,628
All sessions	75,658	248,734	8,756

Analysis was carried out using two general approaches, both of which utilized the Smith-Waterman algorithm [10] for string matching. This algorithm is computationally efficient and can be used to find either exact matches (i.e., LRSs) or inexact matches (using the parameters M_b , M_p , I_c , and I_o).

The first approach performed inexact matching at a session level, but some aspects of the results proved to be rather sensitive to the particular parameter values that were chosen.

Results from the second approach are described in the following subsection. This approach was divided into two phases:

- i) Calculate all subsequences that contained at least three links and occurred in at least two sessions.
- ii) Perform inexact matching at a subsequence level to find all sessions that contained a sequence similar to each given subsequence.

3.2 Results

3.2.1 Parameter Choice

The first stage of the evaluation investigated the effect that changes in the ratio $M_b:M_p:I_o:I_c$ had on the number of sessions that "matched" each subsequence (one set of parameters required the match to be exact, whereas the others allowed different amounts of mismatch). This was carried out using just the sessions from our School's computers (human users) so, for any given subsequence, the matching sessions represented a group of people with a similar interest.

There were 922 subsequences (578 of length 3, the minimum allowed, and the longest was of length 41). The ratios $M_b:M_p$ and $I_o:I_c$ were varied separately (these dictated the amount and type of mismatch that was allowed; M_p always equaled I_o), with exact matching being included for comparison.

The results are summarized in Figure 2. For all the parameter sets, approximately half the subsequences only matched the two sessions that had generated that subsequence in the first place. Increasing the ratio $M_b:M_p$ modestly increased the number of inexact matches that were found, and this reach an asymptote when the ratio was approximately 10:1. Variations in the ratio $I_o:I_c$ had little effect on the number of inexact matches. This indicates that variations between sessions took the form of several short side trails rather than one long side trail, and that the overall approach being used was robust.

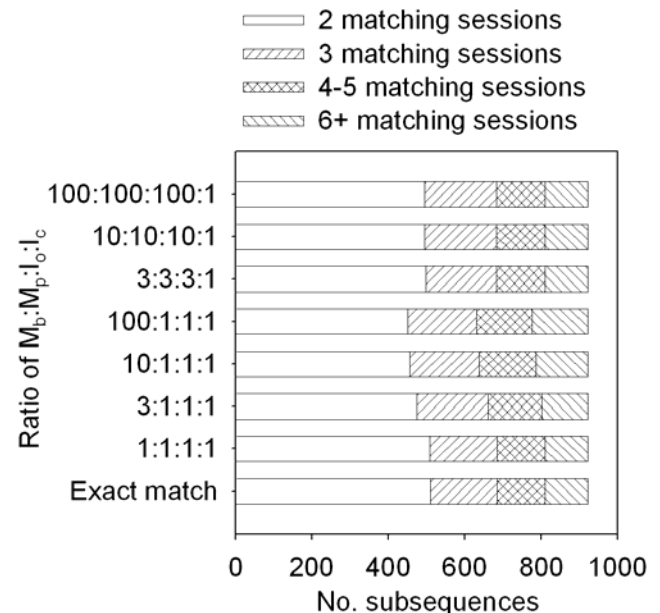


Figure 2. Effect of the ratios $M_b:M_p$ and $I_o:I_c$ on the number of subsequences matching a given number of School sessions (depending on the parameters, a match could be either exact or inexact).

3.2.2 Human vs. Robot Navigation

The second stage of the evaluation compared the navigation patterns of human users (sessions from our School's computers) with those of known robots, using parameters of $M_b = 10$, and $M_p = I_o = I_c = 1$.

The number of human and robot sessions was similar (see Table 1), but there was far more commonality in the navigation paths taken in the former. The human sessions generated many more repeated subsequences than the robot sessions (922 vs. 71), and a larger proportion of those human subsequences matched the paths taken in several sessions (say, six or more; see Figure 3). The longest subsequences had 41 links (humans) vs. 42 links (robots).

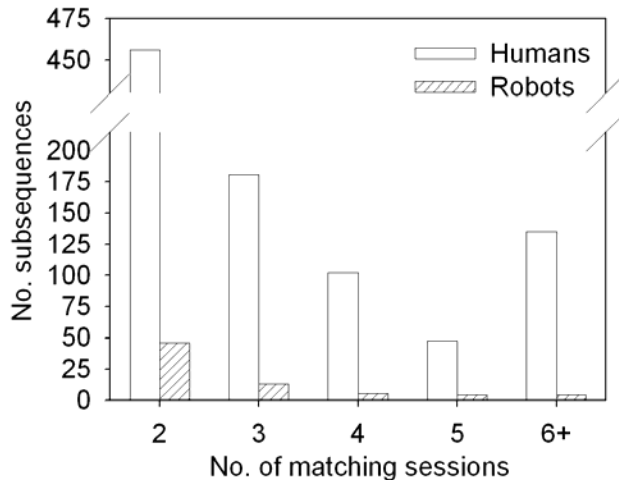


Figure 3. Number of subsequences matching a given number of human and robot sessions ($M_b = 10$, $M_p = I_o = I_c = 1$).

Each time a session matched a subsequence, the match's score was calculated as $S = aM_b + bM_p + cI_o + dI_c$ (for an explanation of the parameters, see §2). The subsequences were then ranked according to an aggregate score, $\sum S^2$ (squaring each individual score helped to emphasize longer subsequences, which would inevitably match with fewer sessions than short subsequences).

For the robot sessions, 69 of the 71 subsequences only had exact matches, which indicates the same basic algorithm was being used to make navigational decisions. Comparison of the subsequences with the relevant website pages showed that robots usually performed breadth-first searches, for example, requesting all 42 research reports from 2001 and 2002 in the order they were listed (see <http://www.comp.leeds.ac.uk/research/pubs/reports.shtml>).

For the human sessions, in 160 subsequences users had navigated part of the Frequently Asked Questions (FAQ) section of the website, often pressing the "next" or "previous" button to navigate between pages, following a route through related topics that had been provided by the FAQ's designer. This included the longest subsequence (41 links), which also had the highest aggregate score. In 529 other subsequences users had navigated through online teaching material, involving a total of 25 of the School's taught modules. Data relating to one of these, *GI31: Advanced computer graphics* (taught by the author), were analyzed in detail.

Thirteen subsequences traversed at least one link within GI31's online material. Figure 4 illustrates that the primary information

pathway lay from the module's home page to OpenGL exercises on modeling ([gi31-ex-modelling.html](#)), which students were tackling during the period the logfile data were captured. In the following weeks students tackled six other sets of exercises (the next was viewing; [gi31-ex-viewing.html](#)), so if we had captured and analyzed the data over an extended period of time the primary information pathway would have been expected to change.

The module's resources page ([gi31-resources.html](#)) is a collection of useful internal and external links, but it is surprising that none of the subsequences included the page dedicated to the basics of OpenGL ([gi31-ogl.html](#)), which was linked to from both the resources page and the home page. Instead, one subsequence included information about the basics of OpenGL ([gi21/resources.html](#)) developed for a different module (*GI21: Introduction to computer graphics*), which all students taking GI31 would have studied in one of the two previous years.

Assuming that most access to GI31's material was by students studying the module then the activity we recorded can be considered to belong to users who were all part of the same community. The subsequences highlighted a connection that some users made between GI31's practical work and the resources provided for GI21, even though no actual link was provided within the website. The subsequences also highlighted a potential problem that students were unaware of (or unable to find) GI31's information about the basics of OpenGL.

3.2.3 Analysis of all Sessions

The final stage of the evaluation involved analyzing the logfile data for all the sessions, using the same string matching parameters as in the previous stage. Within the 6781 subsequences that were generated, the most popular topics were *Hidden Markov Models*, *Perl*, *Latex* and *Undergraduate Admissions*.

The 230 subsequences that included at least one link within the admissions topic (<http://www.comp.leeds.ac.uk/ugadmit/>) were selected for further investigation, to see whether the inexact string-matching process identified other "interesting" topics (links that occurred within a subsequence for at least one session, but were not part of the subsequence itself).

Three types of topic stood out. One was details of particular modules, which users could access directly from the page describing the option streams ([/ugadmit/streams.html](#)) or each course's content (e.g., [/ugadmit/CS/course_content.html](#)). The other topics were people working in the School and our research groups, neither of which were linked to from the admissions pages. Given that most of the option streams are closely allied to the research groups' interests, and this is partly what sells "Computing at Leeds", explicit links or on-the-fly recommendations should perhaps be added.

4. CONCLUSIONS

This paper investigated how inexact string-matching techniques, widely used to analyze bioinformatics data, may be applied to analyze hypertext navigation. A two phase method of analysis was developed, which: (i) calculated all the subsequences that were repeated between sessions, and then (ii) used inexact string-matching with four parameters (M_b , M_p , I_o and I_c) to find sets of similar sessions.

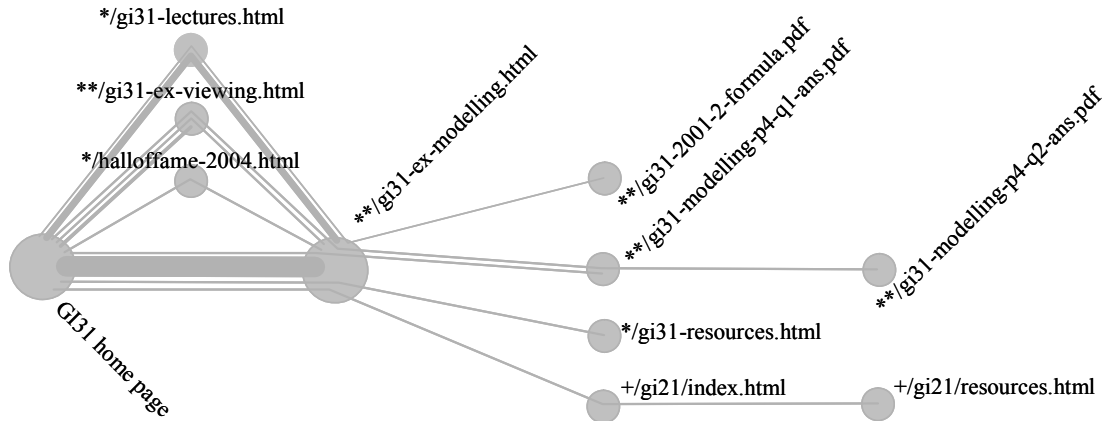


Figure 4. Overview of the parts of subsequences that traversed a link(s) within the GI31 module's online material. Line thickness represents number of sessions that matched each subsequence. Module homepage is <http://www.comp.leeds.ac.uk/royr/gi31/>. * <http://www.comp.leeds.ac.uk/royr/gi31> ** <http://www.comp.leeds.ac.uk/royr/gi31/exercises> + <http://www.comp.leeds.ac.uk>.

The method was evaluated using two weeks of logfile data from our School's website. The method was shown to be robust with respect to the exact values of the four parameters, with the magnitude of M_b in relation to M_p and I_o being most important.

Sets of subsequences that accessed the same topic were used to identify a community of users with the same interest, and one of these (the GI31 module) was used to show how sets of subsequences can be used to understand the pathways people use to access information.

Both phases of the method identified information (the GI21 resources; the School's research groups) that may have been of interest to a whole community of users, but which was only found by a minority. This shows how string-matching can identify ways of improving the structure of hypertext collections.

Clearly, a substantial amount of further research is still needed, for example, to determine how to automate the identification of topics of interest, deal with the dynamic nature of most hypertext collections, and use semantic relationships to perform the analysis at a variety of levels of detail. However, the basic design of a full system is clear, and would involve components that: (i) periodically mined logfiles to create a database of subsequences and associated topics, (ii) matched users navigation paths to this database to generate recommendations in real time, and (iii) allowed systems designers to visualize the overall process.

5. ACKNOWLEDGMENTS

I thank Jonathan Ainsworth for useful background information about the logfile data, and Vania Dimitrova for helpful comments about a draft of this paper.

6. REFERENCES

- [1] Barra, M., Malandrino, D., and Scarano, V. "Common" web paths in a group adaptive system. In *Proceedings of the ACM Conference on Hypertext and Hypermedia (HT'03)* (Nottingham, UK, August 26-30, 2003). ACM Press, New York, 218-219.
- [2] Gams, E., and Reich, S. "Common" web paths in a group adaptive system. In *Proceedings of the ACM Conference on Hypertext and Hypermedia (HT'04)* (Santa Cruz, CA, August 9-13, 2004). ACM Press, New York, 89-90.
- [3] Krane, D. E., and Raymer, M. L. *Fundamental concepts of bioinformatics*. Pearson Education, New York, 2003.
- [4] Large, S., and Arnold, K. Evaluating how users interact with NHS Direct Online. In *Proceedings of the workshop on Personalisation for e-Health* (Edinburgh, UK, July 29, 2005). <http://www.csc.liv.ac.uk/~floriana/UM05-eHealth/Large.pdf> [last accessed 4 May 2006].
- [5] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10, 8 (1966), 707-710.
- [6] Milic-Frayling, N., Jones, R., Rodden, K., Smyth, G., Blackwell, A., and Sommerer, R. (2004). SmartBack: Supporting users in back navigation. In *Proceedings of the 13th International World Wide Web Conference (WWW'04)* (New York, USA, May 17-22, 2004). ACM Press, New York, NY, 2004, 63-71.
- [7] Peponis, J., Conroy Dalton, R., Wineman, J., and Dalton, N. Measuring the effects of layout upon visitors' spatial behaviors in open plan exhibition settings. *Environment and Planning B: Planning and Design*, 31 (2004), 453-473.
- [8] Pirolli, P. L. T., and Pitkow, J. E. Distribution of surfers' paths through the World Wide Web: Empirical characterizations. *World Wide Web*, 2 (1999), 29-45.
- [9] Ruddle, R. A. (2005). The effect of trails on first-time and subsequent navigation in a virtual environment. In *Proceedings of the IEEE Virtual Reality Conference (VR'05)* (Bonn, Germany, March 12-16, 2005). IEEE Press, Los Alamitos, CA, 115-122.
- [10] Smith, F. F., and Waterman, M. S. Identification of common molecular sequences. *Journal of Molecular Biology*, 147 (1981), 195-197.
- [11] Tauscher, L., and Greenberg, S. (1997). How people revisit web pages: Empirical findings and implications for the design of history systems. *International Journal of Human Computer Studies*, 47 (1997), 97-137.