Exploiting Web Browsing Histories to Identify User Needs

Fabio Gasparetti
Roma Tre University
Via della Vasca Navale, 79
00146 Rome, Italy
Tel:39-06-5517-3223
E-mail: gaspare@dia.uniroma3.it

Alessandro Micarelli
Roma Tre University
Via della Vasca Navale, 79
00146 Rome, Italy
Tel:39-06-5517-3205
E-mail: micarel@dia.uniroma3.it

ABSTRACT
Browsing activities are an important source of information to build profiles of the user interests and personalize the human-computer interaction during information seeking tasks. Visited pages are easily collectible, e.g., from browsers' histories and toolbars, or desktop search tools, and they often contain documents related to the current user needs. Nevertheless, menus, advertisements or pages that cover multiple topics affect negatively the advantages of an implicit feedback technique that exploits these data to build and keep updated user profiles. This work describes a technique to collect text relevant to the current needs from sequences of pages visited by the user. The evaluation shows how it outperforms other techniques that consider the whole page contents. We also introduce an improvement based on machine learning techniques that is currently under evaluation.

ACM Classification: H5.4 [Hypertext/Hypermedia]: User issues

General terms: Algorithms, Human Factors

Keywords: implicit feedback, user model, user profile, information needs

1. INTRODUCTION

Internet users have the opportunity to access digital libraries or other large collections of hypertext information such as the Web. Traditional Information Retrieval (IR) systems allow users to quickly sift through the documents by means of given keywords. Nevertheless, users need time to realize that their knowledge is not sufficient for their tasks, to formalize clear queries to submit to search tools and to access particular documents. The time spent for these activities is sometimes longer than the time to find out and comprehend the concepts related to their current needs. Moreover, many users are not able to accurately formulate their needs into valid search expressions. Sometimes information needs are vague or users are not familiar with the retrieval process or the system's interface. Relevance feedback (RF) techniques have been introduced to allow users to explicitly suggest what information is relevant and help the systems to build a better representation of their needs. Nevertheless, RF requires additional time during the seeking process. The burden on the users is still high and the benefits are not always clear compared with other approaches [11].

IUI'07, January 28–31, 2007, Honolulu, Hawaii, USA.

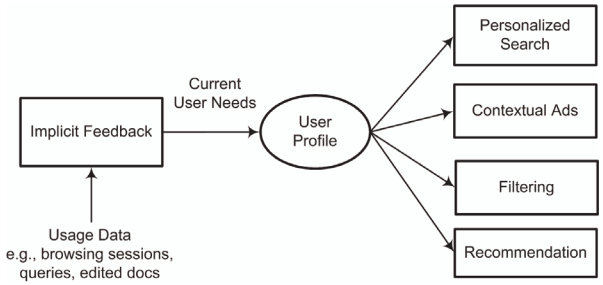

Figure 1: Implicit Feedback techniques extract information related to the current user needs

Implicit Feedback techniques [7] passively monitor the user behavior gathering usage data to build a profile of the user needs (see Fig.1). Users do not have to explicitly indicate which documents are relevant. Typical sources of usage data are: viewed or edited documents, query histories, emails, purchased items, annotations, etc.. Browsing/query histories in particular have been considered in some personalized search systems, e.g., [10,11]. Search engines' toolbars and desktop search tools can easily access that information, which has proven to be very useful to disambiguate query terms and personalize the search results. In spite of this, implicit feedback suffers from important drawbacks. Many Web pages contain various information that is not related to the topic they are concerned, e.g., navigation menu, advertisements, contact information, copyrights. Furthermore, some Web pages cover multiple topics that are not necessarily related one another. Analyzing the semantic content structure of Web pages to identify the real content related to the user needs can undoubtedly improve the performance of personalized systems.

In this paper, we introduce an algorithm that, given single a browsing session extracts the textual content from Web pages and processes it in order to find information related to the user's current needs. This information can be exploited by user modeling systems in order to improve the internal representation of the users' interests, knowledge

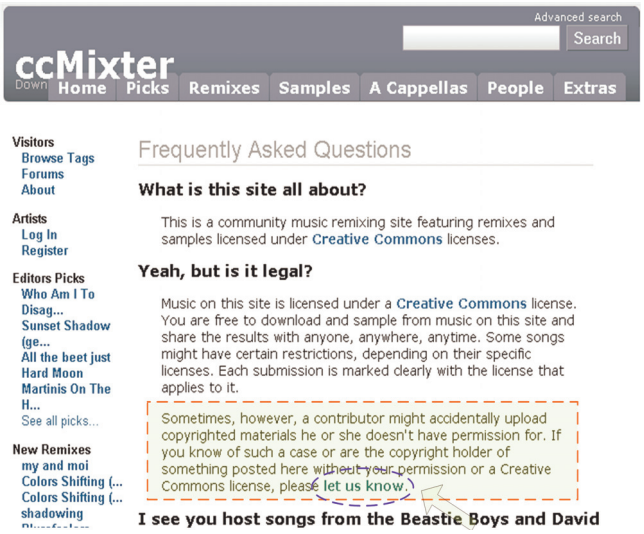

Figure 2: A common Web page, where the links to related content and the HTML contexts of the links are highlighted. *Page from ccmixter.org*

and goals. An evaluation of the algorithm is to be found in Sect.5. Section 4 introduces an improvement of the proposed algorithm considering machine learning techniques.

2. RELATED WORKS

As far as our knowledge, the problem of filtering Web page content to draw relevant information for user modeling is still to be addressed. Some Information Extraction prototypes analyze the structure of sets of pages downloaded from a given Web site in order to find repeated patterns and retrieve data to store in data bases [8]. VIPS performs visual analysis of Web page structures in order to improve query expansion [2]. Other techniques focus to page segmentation to filter out advertisements or adapt content to mobile devices, e.g., [1].

3. EXPLOITING BROWSING HISTORIES

The approach to analyze browsing sessions is based on the notion of Information scent developed in the context of Information foraging theory [9], which has been already used in different domains [4]. It analyzes text snippets associated to links, i.e., proximal cues, which users use to decide whether to access the distal content, that is, the page at the other end of the link. Formally, the Information scent is the imperfect, subjective perception of the value or cost of the information sources obtained from proximal cues. We assume that if users select a particular link, they are expressing a particular interest that corresponds with their perception of the information source pointed by the link. Because this perception depends on the link's anchor, this text can be considered related to the current user needs, which are governing the browsing activity. Collecting this information during a browsing session might be valuable for adapting the user profile in a personalized search or recommending system. Nevertheless, anchor text is usually vague and imprecise to recognize user needs, especially if it consists only of a few words or even worse these words are for surfing support, e.g., "full story", "page 2", "link" (see also the IBM Web site analysis of the most frequent terms in anchor text, and titles [5]). Large IR search systems, such as Google, are able to collect several anchors from incoming links sifting through a corpus of billions of pages. Browsing sessions most of the times do not provide this breadth.

For these reasons the proposed algorithm does not ignore anchor text but this information is extended with further text that can be retrieved from the current browsing sessions, such as the current page and the page pointed by the link that the user has selected. The outcome of this expanding process can be exploited to build user profiles. The basic steps are summarized as follow:

$A_T \leftarrow$ anchor text of the selected link
$T_T \leftarrow$ title of the page pointed by the link
$C_T \leftarrow$ context of the selected link {where *context* is the text in the link's html element}
$TXT \leftarrow A_T +_s T_T +_s C_T$ {where $+_s$ is the string concat operator}
$OUT \leftarrow TXT$
$L \leftarrow$ leaves of the current page's DOM tree
while L is not empty **do**
  $l \leftarrow$ a node in L
  $L \leftarrow L - \{l\}$
  $sim \leftarrow s(TXT; text(l))$ {where *text* returns the text contained in a DOM node, and the function $s(S1,S2)$ draws the textual similarity between the two strings $S1$ and $S2$}
  if $sim > sim\_threshold$ **then**
    $OUT \leftarrow OUT +_s text(l)$
  end if
  if $l \neq ROOT$ **then**
    $L \leftarrow L + \{parent(l)\}$
  end if
end while

At the beginning, we collect the available text most likely related to the main topic of the page pointed by the selected link, i.e., the anchor text of the link, the title of the pointed page and the context of the selected link in the current page. For this latter information, we analyze the DOM view of the HTML page, i.e., the tree structure of HTML elements. Web pages are usually structured in logic parts, such as menu, advertising, copyrights, etc., defined by a physical organization of the HTML tags, most of the times based on <TABLE> and <P> tags. This organization is useful to recognize text related to the link, that is, its context. For instance, Fig.2 shows how the content of the paragraph that contains a link is very helpful to summarize the content of the pointed page (i.e., copyright violation). If we have a look at the physical organization of the page, we discover how these paragraphs are enclosed in <P>...</P> tags. Generalizing the procedure and considering further tags, such as HR, UL, OL, LI, we are able to split pages in units whose boundaries are arranged by HTML tags, and the text of the deepest unit that contains the link is set as its context.

Once the context of the link has been drawn, we perform a search through the content of the current page to find correlated information. The search covers the set L of all the

HTML units built in the previous step. Each time a unit with similar text is found, it is included in a buffer. At the end of the procedure, the buffer corresponds with the information related to the user needs. To compare textual content we employed the similarity measure defined in the TextTiling [6], a document segmentation algorithm for subdividing texts into contiguous, non-overlapping units that represent passages or subtopics. This function compares pairs of text blocks by a cosine rule:

$$s(S1,S2) = \frac{\sum_{t \in S1} w_{t,S1} w_{t,S2}}{\sqrt{\sum_{t \in S1} w_{t,S1}^2} \sqrt{\sum_{t \in S2} w_{t,S2}^2}}$$

where the weighting scheme is based on the term frequency measure. An evaluation of the proposed technique is presented in Sect.5.

4. ENHANCE THE PROPOSED ALGORITHM

We are currently working on an improvement of the proposed algorithm that takes into consideration learning techniques to exploit previous outcomes in order to improve the accuracy of the output. Many Web sites are based on layout templates that organize the Web page content. In many cases, such as News Web sites, most of the pages are based on a few templates. As stated by Cai *et al.* [2], a link from page p_1 to page p_2 frequently suggests that there might be some relationship between some certain part of p_1 and some part of p_2. If we are able to identify and measure this relationship, further content related to the choice of a particular link can be collected analyzing the pointed pages.

In order to group pages with the same or similar DOM representation, i.e., sharing the same template, we define a similarity measure based on the number of differences between the two DOM trees drawn by the Chawathe's algorithm [3]. The measure corresponds to the number of edit operations, i.e., insertions, deletions and updates, to transform one tree and make it identical to a second. When the number of operations does not exceed a given threshold, the two pages share the same layout template. By means of a cluster analysis, we are able to sort different browsed pages by their templates. For each cluster we build a centroid tree merging all the DOM trees. If a browsed page is similar to a given centroid, it is included in the cluster and the related centroid is updated. Otherwise, a new cluster is created. To improve the similarity measure of two pages, we take also into consideration the frequency of html tags. The more frequent they are, the more important they weight when the similarity measure is drawn.

We extended the previous algorithm considering similarities between two blocks (i.e., a part of the Web page delimited by HTML tags used to make the layout) contained in two different pages. When the user visits one page, we collect the textual description as described in the previous section. Afterwards we compare this description with the content of the page pointed by the selected link (by means of the same similarity measure and algorithm previously defined). If we find a block *j* with similar content, we record the tuple $<c(p_i), c(p_j), id_i, id_j>$, where p_i and p_j are two contiguous pages in the browsing session. Given a page *p*, the function *c(p)* returns its cluster. id_i is the id of the block that contains the selected link, and id_j is the block of the pointed page *j* that contains text related to the textual description of the user needs. Considering page clusters allows us to generalize the relationship of similarity between two blocks to any pair of pages that share the template of p_i and p_j respectively.

The occurrence *n* of one tuple is a measure of the relatedness of two blocks, but it is to be seen also as a measure of reliability of the textual comparison between two blocks. If we have found a content similarity between two blocks id_1 and id_2 several times, that is, we have a high number of occurrences of $<c(p_i), c(p_j), id_i, id_j>$, we can assign a higher level of importance to the textual comparison and therefore to the text collected from the pointed page. For instance, given two pages p_i and p_j, two different blocks id_1 and id_2, and the related texts *S1* and *S2*, the new similarity measure between them is:

$$\hat{s}(S1,S2) = \frac{n}{N} s(S1,S2)$$

where *n* is the occurrences of $<c(p_i), c(p_j), id_i, id_j>$ normalized by *N*, that is, the times we compared the two blocks id_i and id_j considering also the cases we have not found any textual similarity. The obtained result correspond to the weight we assign to the text S2 that is included as output of the algorithm. While the approach described in Sect.3 is suitable for any sequence of pages, in order to estimate *n* for the proposed learning technique, the user has to browse several sequences of pages $p_i \rightarrow p_j$ characterized by sharing the same pairs of templates $<c(p_i), c(p_j)>$.

5. EVALUATION

The evaluation of the algorithm described in Sect.3 is based on sequences of browser's histories collected from different users. We removed not interesting Web sites, e.g., Web mail services or Intranet hosts, and sequences related to casual behaviors. Finally, we assigned a brief textual description of the information needs for each subsequence. Afterwards, a person outside of our research group chose a subset of the extracted sequences. After a preliminary evaluation, we set the similarity threshold to 0.65, the only parameter of the algorithm. Given a browsing subsequence of pages, our goal is to analyze the affinity between the output of the proposed algorithm and the textual description related to the user needs. The comparison is based on the cosine rule between the output of the algorithm and the user needs. As benchmark, the whole content of the pages in the subsequence has been considered and compared with the user needs. This approach is usually employed by traditional user models to extract usage data from browsing activities.

Figure 3 shows the results of the evaluation for 14 subsessions composed of two pages and 33 subsessions of three pages. The proposed algorithm outperforms the traditional approach in most of the cases. In other words, the proposed algorithm is able to filter out information that is not related

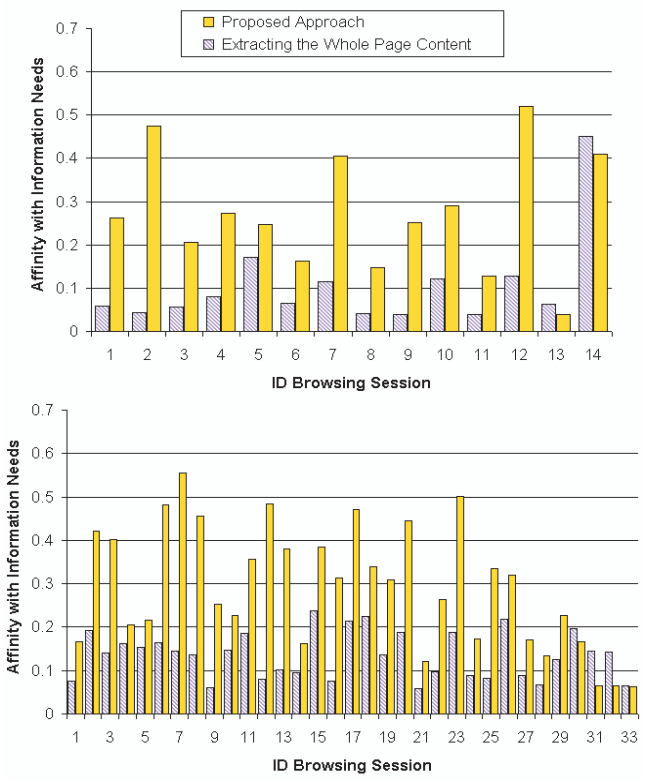

Figure 3: For each browsing sequence of 2 pages (the upper diagram) and 3 pages, the related affinity of the output with the user needs.

to the needs that guide the user during the browsing activity. The average affinity for all the subsessions is summarized in Fig.4.

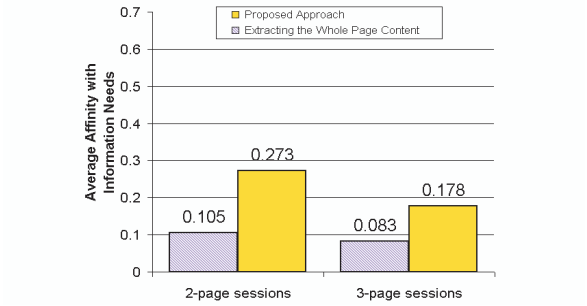

Figure 4: Average affinity with the user needs

6. CONCLUSIONS

The proposed implicit feedback technique extracts information related to the current user needs. The better performance compared to traditional approaches which consider the whole content of pages suggests how user models can take advantage of this technique improving their performance. We provide an evaluation of the proposed algorithm without the learning technique described in Sect.4.

We are currently analyzing longer browsing sessions, characterized by having many pages from the same Web sites. In this way, it is possible to collect enough information to represent clusters of Web pages with similar templates and provide measures of relatedness between html blocks. Moreover, we will include in the evaluation different approaches to extract information from Web pages, e.g., advertisement removal techniques.

REFERENCES

1. Buyukkokten O. and Garcia-Molina H. and Paepcke A. Seeing the whole in parts: text summarization for web browsing on handheld devices. *In Proc. of the 10th World Wide Web conference* (May 1-5 Hong Kong), ACM Press, New York, NY, USA, 2001, pp. 652-662

2. Cai D. and Yu S., Wen J.-R. and Ma W.-Y. VIPS: a vision-based page segmentation algorithm. *Microsoft Technical Report, MSR-TR-2003-79,* 2003.

3. Chawathe, S.S. Comparing Hierarchical Data in External Memory. In *Proc. of the 25th International Conference on Very Large Data Bases* (September 7-10, Edinburh, Scotland, UK), Morgan Kaufmann Publishers Inc., SF, 1999, pp. 90–101.

4. Chi, E. H. Transient user profiling. In *Proc. of the Workshop on User Profiling (CHI2004)* (April 24-29, Vienna, Austria), 2004, pp. 521-523

5. Eiron, N. and McCurly K. Analysis of anchor text for web search. In *Proc. of the 26th ACM SIGIR conference* (July 28-August 1, Toronto, Canada), ACM Press, New York, NY, USA, 2003, pp. 459-460.

6. Hearst, M. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 21, 1 (1997), pp. 33-64

7. Kelly D. and Teevan J. Implicit Feedback for Inferring User Preference: A Bibliography. *ACM SIGIR Forum.*, 32, 2 (Fall 2003), pp. 18-28

8. Laender A.H.F. and Ribeiro-Neto B.A. and da Silva A.S. and Teixeira J.S. A brief survey of web data extraction tools. *SIGMOD Rec.*, 31, 2 (2002), pp. 84-93

9. Pirolli, P. and Card, S. Information Foraging. *Psychology Review*, 106, 4 (1999), pp. 643-675

10. Speretta, M. and Gauch, S. Personalized search based on user search histories. In *Proc. of Web Intelligence (WI2005)* (September 19-22, Compiegne, France), IEEE Press, Los Alamitos, CA, 2005, pp. 622-628

11. Teevan, J. and Dumais, S.T. and Horvitz E. Personalizing search via automated analysis of interests and activities. In *Proc. of the 28th ACM SIGIR conference* (August 15-19, Salvador, Brazil), ACM Press, NY, 2005, pp. 449-456.