Data Mining and Knowledge Discovery manuscript No.
(will be inserted by the editor)

Modeling user interests from web browsing activities

Fabio Gasparetti

the date of receipt and acceptance should be inserted later

Abstract Browsing sessions are rich in elements useful to build profiles of user interests, but at the same time HTML pages include noisy data such as advertisements, navigation menus and privacy notes. Moreover, some pages cover several different topics making it difficult to identify the most relevant to the user. For these reasons, they are often ignored by personalized search and recommender systems. We propose a novel approach for recognizing valuable text descriptions of current user information needs —namely *cues* —based on the data mined from browsing interactions over the web. The approach combines page clustering techniques based on DOM-based representations for acquiring evidence about relevant correlations between text contents. This evidence is exploited for better filtering out irrelevant information and facilitating the construction of interest profiles. A comparative framework proves the accuracy of the extracted cues in the personalize search task, where results are re-ranked according to the last browsed resources.

Keywords Information needs · User modeling · Clustering · Web browsing

1 Introduction

Over the past two decades the time spent using web browsers on a wide variety of tasks such as research activities, shopping or planning holidays is increased. Among the Internet activities, surfing the web is the most relevant accounting for 63% of the overall time [7]. Search engines are crucial interfaces the users turn to in order to submit queries representing their information needs and access

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

to relevant information, however only 12.5% of the visited pages are part of browsing activities which include at least one visit to these tools [92]. The remaining pages lie somewhere on the browsing path flowing away from the listing of results, therefore, their content is basically ignored by the search engines.

More services aim at personalizing the provided content in any format that is relevant to the individual user, current context and material being currently read [64]. Recommender systems for e-commerce, contextual advertising and personalized search are all popular examples of user-adapted interaction. Less approximate representations of user needs generally lead to more precise search results, recommendations and, more in general, less time to complete tasks.

Clickthrough data, namely, query-logs of search engines in connection with the log of links the users clicked on in the presented ranking, have been proven to be a valuable source for adapting retrieval strategies [34]. But general browsing behaviors far outweigh search engine interactions alone as a broader source for effectively predicting users' future interests [9,89]. White and Huang [93] analyze millions of trails originating from search engines' result pages. They prove how topics in the visited pages provide significantly more coverage, diversity and novelty versus the pages lying at the beginning or end of the trail. The users also spend significant more time looking through inner-pages that means they may be deriving utility from those trails.

More than half of the user tasks on the web are related to fact finding, information gathering and, more in general, browsing behaviors including "see what's new" goals [41]. Although browsing sessions contain important hints about the interests driving these tasks [79], empirical evaluations show that a large fraction of elements on web pages (i.e., from 40% to 50%) can be considered irrelevant w.r.t. the present interests [27]. For example, navigation bars, privacy notes, contact information and ads blocks, which are not related to the main topics of the page, represent noise content. While they do not pose any problem for human users to find significant elements, they are difficult to handle when the pages are automatically processed by personalized systems. The overwhelming amount of low quality information makes it difficult to obtain relevant cues useful to adapt the human-computer interaction.

When the users are involved in fact finding and information gathering tasks, they usually submit many short queries, visiting several domains in complex sense-making tasks [92,35]. They typically

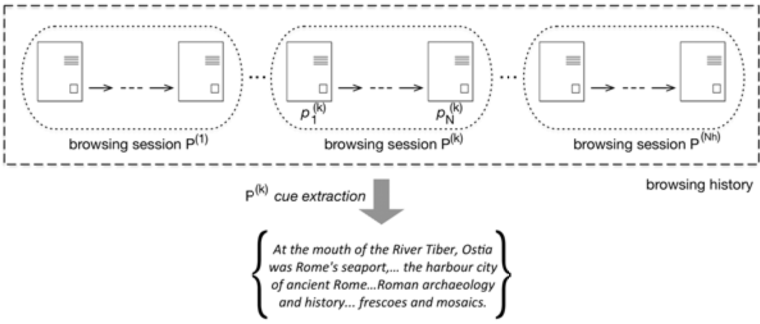

Fig. 1 An example of the output obtained extracting cues from a given browsing session.

show various needs at different times based on current circumstances [66, 49]. Pages often contain mixtures of topics that are not necessarily interrelated one another, and relevant information is often described over a series of connected blocks which make the cue extraction entangled [93]. Wobbling nature of human behaviors may consider pieces of information, such as trivial and entertainment pages, which are of little future utility once they have digested it, and potentially reduce the overall adaptation accuracy [97].

Personalized approaches based on statistical profiles of long-term interests usually produce satisfactory recommendations [25], yet the user sometimes spends time seeking recommendations on new or ephemeral topics, e.g., new books, breaking news or places to go on vacation. In these scenarios, novel suggestions based on selected content extracted from last visited pages are certainly more accurate and preferred by the users [89, 8], but the well-known *filter bubble* phenomenon may prevent it from happening [59]. During the interaction with information sources, user interests continually shift and new content may lead to unanticipated needs [6, 56, 23]. This new content does not match the profile of long-term interests therefore it will be ignored by the system. Finally, long-term profiles sometime require numerous examples of relevant information before it can generate valid representations of user intents [95].

Only a very few attempts have been reported aiming at recognizing relevant cues w.r.t. current user interests and intentions by leveraging browsing sessions. It is our opinion that strategies based on implicit feedback, which analyzes navigation behaviors and operate without human effort, have the chance to better represent both short-term and long-term profiles of users. If specific information linked with the current situation and goals is provided, logical reasoning can also take place, inferring intentions and plans through observed actions or effects on the environment [2], and predicting future interests [90].

In order to achieve the purpose of recognizing cues related to the current interests we propose an innovative approach for selectively collecting text information from visited pages based on implicit signals that naturally exist throughout the browsing sessions. Figure 1 shows an example where, given a session part of the navigation history, the approach aims at extracting terms related to the facts the user wanted to acquire in that moment.

The research questions we intend to address are summarized as follows:

– How to make capital of the browsing activity for identifying relevant cues associated to the current user interests without any human effort?
– Acknowledging that browsing sessions contain noisy content, is the identification of the interests by means of the current state-of-the-art approaches negatively affected by that?
– What is the effectiveness and the efficiency of the proposed approach?
– How does our extraction approach perform compared to the state-of-the-art techniques?

In order to answer these questions, the paper provides the following contributions:

I We propose a novel approach that combines clustering techniques based on DOM-based representations of web pages for identifying relevant correlations between text contents on visited pages.

II We show how the acquired evidence obtained by analyzing the browsing sessions can be used for filtering out irrelevant information and facilitating the construction of *profiles* of current user interests.

III An extended comparative evaluation estimates the effectiveness of state-of-the-art techniques and their efficiency in the well-known personalized search task.

The paper is laid out as follows. Section 2 introduces some relevant issues about profiling user interests in the web domain. Section 3 gives a detailed review of the approaches proposed in the literature. After the problem formulation (Sect. 4), the proposed extraction approach is introduced in Sect. 5. In particular, the representation of browsing histories (Sect. 5.2), the clustering of pages with similar structural templates (Sect. 5.3), the extraction of relevant correlations between text contents (Sect. 5.4.2), and how to exploit that evidence for identifying current information needs in Section. 5.5. The computational complexity of approach and its comparison with others in the literature is discussed in Sect. 6. Experimental comparative results are presented in Section 7, by

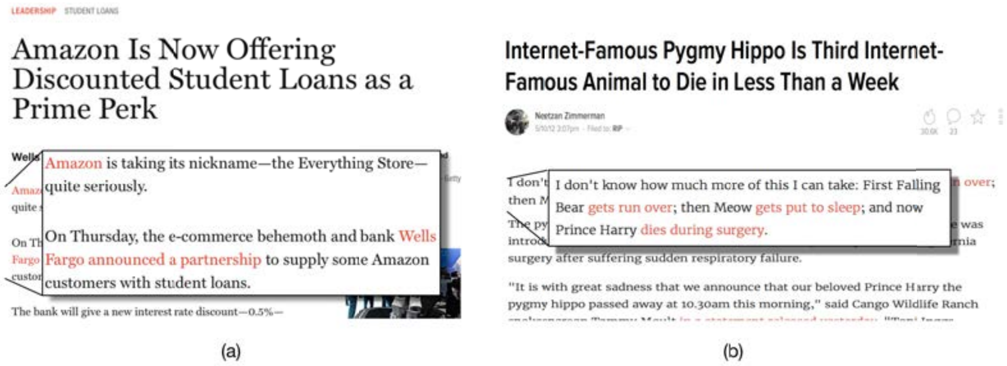

Fig. 2 Links and surrounding text in two web pages.

assessing the performances both on large-scale synthetic corpus of news pages (Sect. 9.1) and on a real-world dataset of browsing histories (Sect. 9.2). Section 10 summarizes the conclusions and points out future work.

2 Identifying user interests from browsing sessions

Empirical observations prove how the anchor (i.e., the visible part of the link text) and its surrounding context are found to be useful for guessing the target page's topic [3,28,81]. Figure 2(a) shows a web page where both anchors (e.g., "Wells Fargo announced a partnership") and near text ("...supply some Amazon customers with student loans.") identify the topic of the pointed resource.

According with the Information scent concept developed in the context of Information foraging theory [61], users decide whether or not access the distal content, that is, the target page, by analyzing this information. If the user decides to follow a link, she is expressing a particular interest that corresponds to her perception of the information resource pointed by that link. In other words, links convey recommendations and users make judgments about which links to follow according with the potential value of the distal objects w.r.t. their needs [98]. Because this perception depends on the text related to the link, it can be considered strongly correlated to the current user needs governing the browsing activity.

On the web, however, hyperlinks bind documents of varying quality and purposes. In particular, anchors and surrounding text can sometimes introduce noise and degrade potential representations of user current interests. In particular, the anchor text is often vague and imprecise especially if consisting only of a few words or, even worse, these words are chosen from a restricted vocabulary of common terms, e.g., "full story", "page 2", "link". Figure. 2(b) illustrates a typical example of anchor text that does not clearly represent the content of the pointed pages. Large retrieval systems on the web, such as Google, are able to collect anchors from incoming links by sifting through a corpus of tens of billions of pages and, thus, statistically filtering out less useful information. Browsing sessions of one user do not provide this breadth. Moreover, if the link is used only for navigational aid, it conveys no recommendation to the user as it frequently happens in site maps and tables of contents.

One more interesting phenomena worth of consideration is page revisiting. People tend to revisit pages, frequently access only a few pages, browse in very small clusters of related pages and generate only short sequences of repeated URL paths [51]. For example, one study found that revisits make up 58% of all browsed pages [78]. More recent studies suggest that revisitation is more common, with 81% of web pages having been previously visited [14,58]. Examples of frequently visited web pages belong to blogs, social networks, online news and e-commerce services. If we limit ourselves considering the hostnames of the visited pages, the percentage of revisits is even higher. This implies that if we group pages according to the templates that generate them, we obtain a small number of clusters of similar pages in comparison with the total number of browsed ones.

The goal of the proposed approach is performing deep analyses on pages visited by the user. Instead of extracting the whole content of pages, our goal is to selectively and unobtrusively extract text that is more likely related to the user's current needs. The semi-structured nature of web pages drives the extraction, which is based on clustering techniques trained during the usual browsing activities.

3 Related work

Early attempts show that clickthrough data have the chance to recognize the current search context improving the retrieval task [74,17]. Other techniques take advantage of large-scale aggregated click-through datasets from search engines [91,80,39], providing a direct measure of relevance based on the overall popularity of entities. Since most of computation is operated offline by analyzing aggregated logs of submitted queries, the ranking does not depend on the particular user intention that motivated the interaction. Recent attempts provide more fine-grained modeling of the interests of each user trying to group different search activities motivated by similar intents [39,33]. Whereas this form of dynamic IR has the ability to incorporate implicit feedback for better representing the user, aggregated click-through data remain an exclusively advantage of large search engines and, therefore, out of reach of individuals [38].

Speretta and Gauch use queries and snippets — few lines of text that appear under every search result — from the listings of results of search engines for inferring user interests and providing personalized search [73]. Natural language queries are inherently short and ambiguous, and the approach identifies the most relevant terms that are used for the query expansion. Snippets are regarded by several authors as query-focused summaries of documents and are therefore used to extract terms relevant to the context of the query. While several other approaches follow this kind of intent identification, e.g., [72,75,100,48,87,68], they all ignore the content of the visited pages beyond the results pages.

User profiles built on visited pages are somehow richer and may contain useful discriminating terms that are not present in the top results from a search engine [4]. These profiles are also more effective in the personalization task in comparison with traditional relevance feedback techniques [94, 76]. Nevertheless, a very few attempts exploit full browsing histories for the identification of the user interests.

(BP) Boilerpipe is a well-known approach used to extract relevant content from pages by filtering out components that are common to many pages and, therefore, considered less relevant or noisy [43]. It describes web pages' text blocks with "shallow text features" and builds a decision tree used to classify these blocks as relevant content or not. Since there is not any explicit representation of the user interests, the approach does not adapt the extraction according to the user but takes advantage of the structural elements of the pages.

(MR) Matthijs and Radlinski [50] build profiles of user interests for re-ranking the top results returned by a search engine to bring up documents that are more relevant to the user.

The authors experimented several combinations of input data sources and scoring functions. Best performances are obtained by extracting metadata keywords, titles and noun phrases from the visited pages' content, and weighting that information with a tf·idf scheme [4].

(PX) The approach is based on the notion of Information scent developed in the context of Information foraging theory [63]. In short, text snippets associated with links, such as visited links' anchors and the text surrounding them, are used by users to decide whether to access the distal content. The approach exploits that information for building profiles of interests related to the current browsing activity [24]. Whereas, DOM-based representations of web pages are considered, past browsing histories and potential relevant evidence extracted from them are ignored in the extraction task.

(SHY) In the *pure browsing history* approach proposed by Sugiyama *et al* [76], the entire browsing history is analyzed for extracting specific content from the the visited documents. The preferences of each user are partitioned in persistent (or long term) and ephemeral (or short term). The latter are gathered during the current session and, therefore, may well represent the actual interests. The approach does not take advantage of the structure of web documents and, except for the collected text content, no further statistical evidence is analyzed.

(TDH) Teevan *et al.* [79] propose a model of interests built by combining a variety of sources, such as received emails messages, browsed pages and search engines' snippets. This model is exploited for improving the ranking of search engines. They prove how the content extracted from the visited pages has some sort of affinity to the user interests, but snippets of results pages usually contain more discriminating keywords. This emphasizes further that more advanced techniques for filtering out irrelevant information from web pages are required. They perform web search personalization by modifying the well-known BM25 probabilistic weighting scheme [40] and indexing the visited pages in a local search engine. The approach shares the same limitations of SHY.

The proposed approach distances itself from the above-mentioned techniques based on a combination of full text unigrams and noun phrase extraction, removing infrequent words or the ones that are not into predefined dictionaries. The filtering is based on the spatial organization and potential correlations of the visited content.

4 Problem formulation

As for the problem formulation, the input consists of the k-th *browsing session* (or trail) $P^{(k)} = (p_1^{(k)}, p_2^{(k)}, \ldots, p_N^{(k)})$, of N pages visited over the time interval $[\lambda, \lambda + \Delta T]$. If the sequence of tokens $\Omega^{(k)} = (t_1, t_2, \ldots, t_M)$ is extracted from the text content in $P^{(k)}$, we want to obtain an interest *model*

Table 1 Symbol legend.

Sym.	**Description**
$c(T_p)$	function that returns the most similar cluster for p
$d(T_p, T_c)$	tree edit distance between T_p and T_c
$depth(v, T_p)$	depth of v in T_p
$freq(T_c, v)$	occurrences of v in the cluster T_c
$g(v, T_p, T_c)$	tree-edit cost for updating the node v in the page p inside T_c
KB_c	KB of the identified clusters
$KB^{(+)}$	KB of the semantic correlation between pairs of pages
$KB^{(\Gamma)}$	KB of the occurrences of pairs of clusters corresponding to two successively visited pages
$N^{(+)}_{p_i \rightarrow p_{i+1}}$	number of times a specific correlation between p_i and p_{i+1} occurs in $KB^{(+)}$
$N^{(\Gamma)}_{p_i \rightarrow p_{i+1}}$	number of times $c(p_i)$ and $c(p_{i+1})$ sequentially occur in $KB^{(\Gamma)}$
$p_i^{(k)}$	i-th visited page in the browsing session k
$P^{(k)}$	k-th browsing session
$s_{p_i \rightarrow p_{i+1}}$	semantic region in p_i containing a link to p_{i+1}
S_{p_i}	subset of semantic regions Φ_{p_i} in p_i
T_c	tree-based representation of the c cluster containing pages with similar template
T_p	tree-based representation of the p page
T'_p	set of nodes in T_p
$\|T_p\|$	number of nodes in T_p
$\|T_p\|_d$	maximum depth in T_p from the root
v	node in T_p
V_t	vocabulary of terms
$w_{p_i \rightarrow p_{i+1}}$	boosting factor that weights the content extracted from $p_i \rightarrow p_{i+1}$
w_h	tree-edit cost associated with high relevant HTML tags
w_l	tree-edit cost associated with low relevant HTML tags
wt	generic term on a web page
Γ	set of potential tree-based representations
Γ'	set of potential nodes in the tree-based representations in Γ
$\Theta^{(k)}$	subset of $\Omega^{(k)}$ related to the current user interests
Φ_{p_i}	set of semantic regions in p_i
$\Omega^{(k)}$	sequence of tokens extracted from the text content in $P^{(k)}$

$\Theta^{(k)}$ as follows:

$$\Theta^{(k)} \subseteq \Omega^{(k)} \tag{1}$$

Since each page $p_i^{(k)}$ can deal with multiple topics and contain content related to navigation support, advertisement or further less relevant elements, $\Theta^{(k)}$ corresponds to the smallest subset that better describes the interest driving the browsing activity over $P^{(k)}$. Since the interest model is built by limiting the extraction to the k-th session, we only perform short-term analysis of user interests.

The example in Figure 3 shows a common page with several text regions. By extracting the whole content, entities such as Nokia, Apple, NATO and YouTube would have been included in the output model. But our goal is to limit the extraction to the most relevant regions, highlighted in pink.

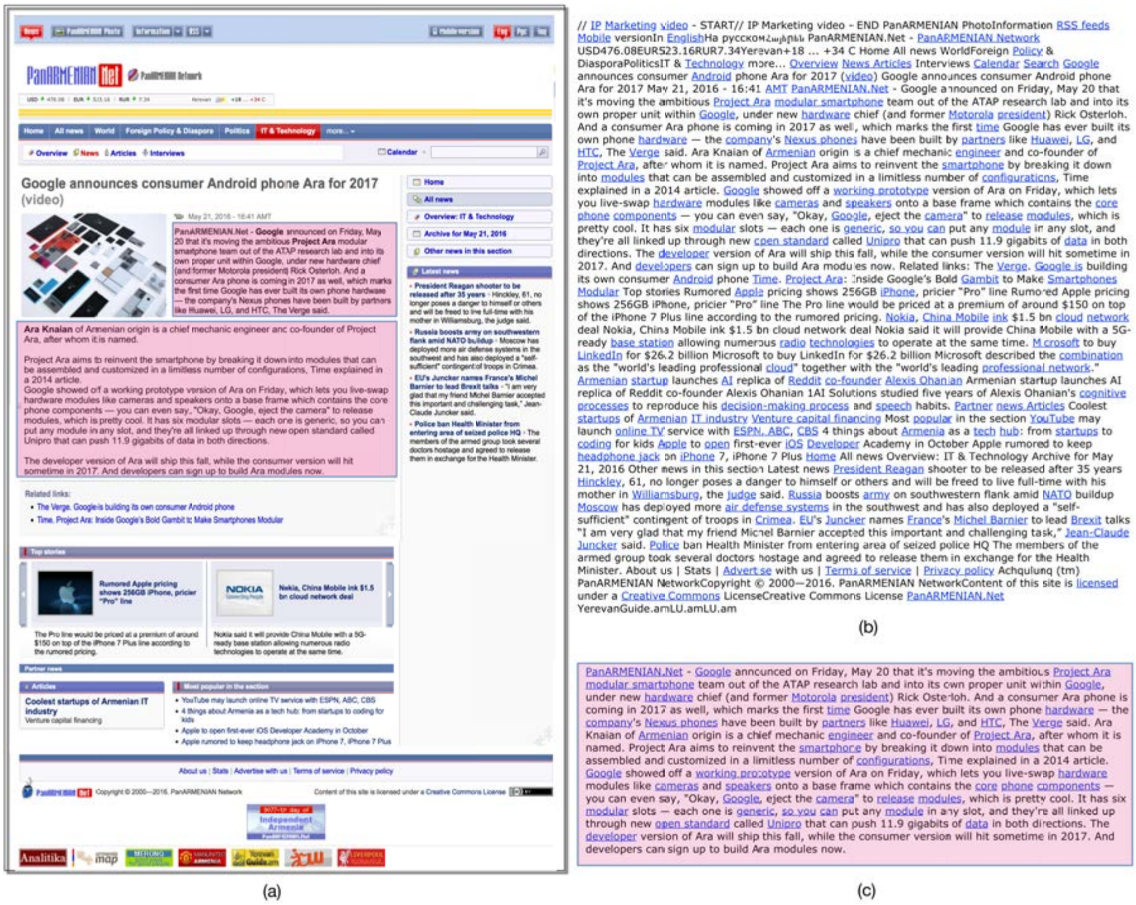

Fig. 3 Two text outputs extracted from the whole page (b), and from the content related to the current user intents (c). Highlighted are the concepts obtained from a common Named-entity recognition tool. Content courtesy of `Panarmenian.net`

When the intent behind a browsing session is *informational*, that is, the acquisition of particular information [12], $\Theta^{(k)}$ overlaps the text representation of the searchers' needs at the time λ [6,56]. Informational searchers typically try to maximize the amount of relevant information they are viewing while minimizing the paths to irrelevant ones [62], that is pages whose text content is not related to $\Theta^{(k)}$.

A more common representation of profiles of user interests consists of an estimated relevance distribution over a set of keywords [26]. Real-value weights have the chance to associate a single degree of relevance with each keyword in the set. Without loss of generality, we can define a vector $\vec{\Theta}^{(k)} \in \mathbb{R}^{|V_t|}$ as follows:

$$\vec{\Theta}^{(k)} = < wt_1, wt_2, \cdots, wt_{|V_t|} > \tag{2}$$

where each dimension corresponds to a distinct term in the vocabulary V_t and wt_i is the weight for the term associated with the i-dimension in V_t.

	rome	ostia	lido	metro	train	restaurants	ruins	ancient	town	harvor	...	museums
$\overrightarrow{\Theta}^{(k)}$	8	7	5	4	3	3	3	3	3	3	...	2

Fig. 4 An example of vector representation of the user interest.

Figure 4 shows an example of the model $\overrightarrow{\Theta}^{(k)}$ obtained from the browsing session depicted in Fig. 5(a). The weights are computed by counting the occurrences of the keywords in the relevant regions and filtering out the most common ones, e.g., 'the', 'to' and 'about'.

5 The proposed extraction approach

The proposed approach can be broken down into two stages, as shown in Fig. 6. They can be summarized as follows:

Stage ***I*** (Sect. 5.2): In the initial stage, single pages and pairs of visited pages in browsing sessions are considered. The goal of this stage is twofold:

- ***I.i*** (Sect. 5.3) representing groups of pages with a similar template by means of a common tree structure;
- ***I.ii*** (Sect. 5.4) finding correlations between the contents of text regions on two consecutive pages.

Stage ***II*** (Sect. 5.5) This stage considers the whole session currently browsed. It weights the retrieved text in each pair of visited pages in the current session in accordance with the times the semantic relationship between the two regions has occurred in the past.

In particular, in the I stage, each visited page is represented by a traditional DOM-based tree, which consists of the hierarchy of HTML elements. The obtained tree is also subjected to a agglomerative clustering to group pages with similar templates. The obtained clusters are stored in a local knowledge base KB_c.

Each time the user follows a link between two pages $p_i \to p_{i+1}$, two sets of *semantic regions* Φ'_{p_i} and $\Phi''_{p_{i+1}}$ are identified. A semantic region is defined as a region of coherent content w.r.t. a certain topic. Examples of those regions are shown in the dashed blocks in Figure 5a. The HTML structural elements defining the content layouts, such as <DIV> and <TABLE>, support the identification

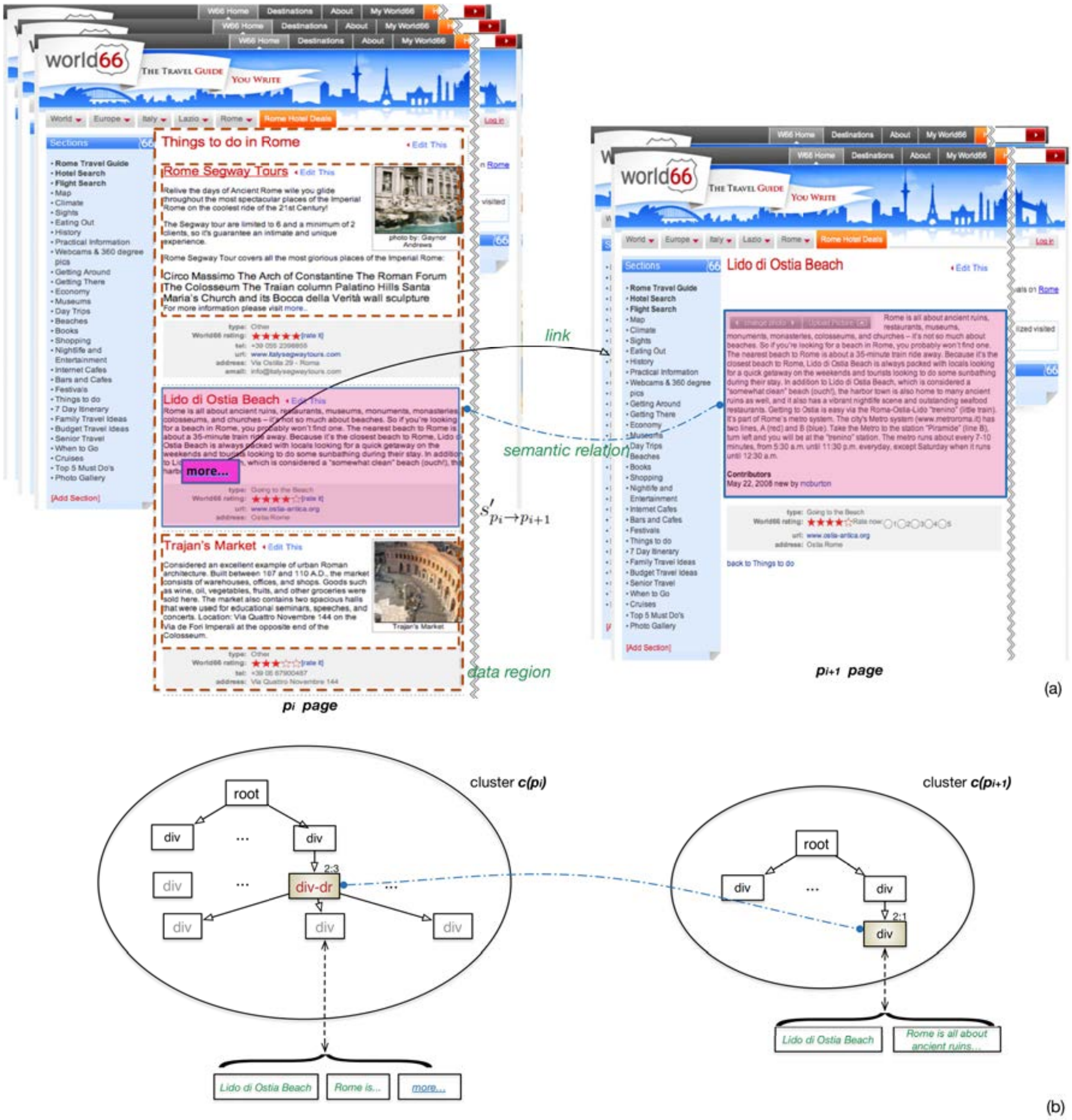

Fig. 5 Two visited pages and the two textual regions that correspond to the current user needs (a). The internal DOM-based representation of the visited pages and the correlation between two blocks (b). Content courtesy of `World66.com`

task (see Sect. 1). The semantic regions $s'_{p_i \to p_{i+1}} \in \Phi'_{p_i}$ in the page p_i, shown as a solid block, has the characteristic of containing the link $p_i \to p_{i+1}$ (e.g., the HTML *href* attribute of the <A> element). Often this kind of regions include additional text surrounding the link. In the example, the link with anchor *"more.. "* is associated with the surrounded text *"Lido di Ostia Beach - Rome is all about..."* enclosed in the inner solid square.

The content of $s'_{p_i \to p_{i+1}}$ is then compared with each semantic region $s''_{p_i \to p_{i+1}} \in \Phi''_{p_{i+1}}$. When the two regions are found semantically related, the correlation between $s'_{p_i \to p_{i+1}}$ and $s''_{p_i \to p_{i+1}}$, represented by a dot-dash line, is stored in the knowledge base $KB^{(+)}$. The knowledge base $KB^{(\Gamma)}$ keeps track of

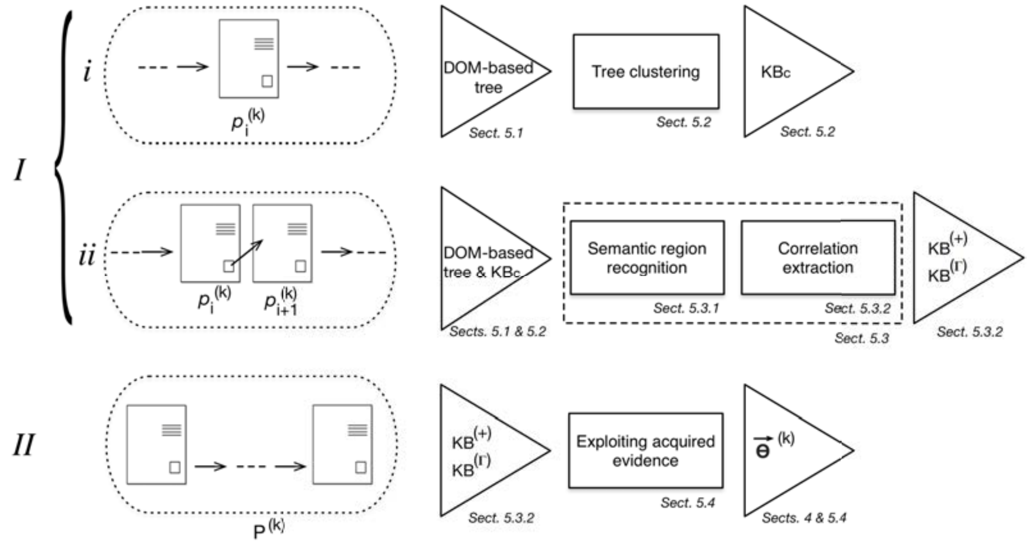

Fig. 6 The proposed approach consists of multiple stages.

the occurrences of pairs of clusters, corresponding to two successively visited pages. The occurrences are computed without regard to their semantic correlation. It is relevant for normalizing the statistics extracted by $KB^{(+)}$.

By clustering the pages with similar template, it is possible to generalize the relationship between two regions making it independent of the current browsing session, and, consequently, the particular text content. In other words, the correlations are generalized to any future page that shares the same template with the current ones. Figure 5b shows two clusters, each associated with two or more pages that share the same HTML template. The semantic relationship connects two structural elements from the two clusters stored in KB_c. For this reason, the input of $I.ii$ stage includes this knowledge base obtained in the previous stage.

The last stage considers the whole browsed session. It weights the retrieved text in each

For each pair of visited pages, the semantic regions $s''_{p_i \to p_{i+1}}$ are extracted and their content weighted in accordance with the times the relationship between the regions has occurred in the past. The above-mentioned knowledge-bases (namely, $KB^{(+)}$ and $KB^{(\Gamma)}$) store the results of the analysis of the previous browsing sessions of the user. If two regions have been found statistically related, the extracted text has more chance to be correlated with the intent that drove the user to read the anchor, click on the link and visit the pointed page.

If statistically significant evidence indicates that two blocks have had strong semantic correlation, higher relevance is assigned to the retrieved text extracted from the pointed page. By repeating this

Fig. 7 Two browsing sessions where semantic correlations between similar blocks are repeated. Content courtesy of `GreenBiz.com`

extraction activity on the whole trail $P^{(k)}$, the text can be combined in an interest model $\overrightarrow{\Theta}^{(k)}$, which is assumed to contain most of the significant themes for the given browsing activity.

Figure 7 shows two trails where semantic relationships between blocks occurs more than one time. Because some of those relationships refer to the same pairs of clusters, the content of the related regions is increasingly weighted, and so, is highlighted by darker colors.

5.1 A comparison with a traditional content-based approach

Figure 7 shows two sessions where text content relevant to the interests of the user that drove the browsing activity is highlighted. As already stated in the previous section, a traditional content-based approach that extracts the whole text from the visited pages would return several misleading elements on the web pages. However, it is interesting to note how the navigation path connects pages that often are similar one another. In particular, the followed links bind two elements on different pages whose content is related to the same concepts. Indeed, the author decides to include a hyperlink on a page to make a reference to a different document the reader can directly follow. But the HTML link does not state the specific target document fragment it refers to, so a filtering approach is required to take into consideration only the fragments whose content is correlated with the followed link. By limiting the analysis on each single page, current approaches, such as BP [43] and MR [50], do not take advantage of the explicit references of hyperlinks.

The benefits of the proposed approach are manifold. For instance, in the two browsing sessions in Fig. 7, the semantic correlation between blocks is recognized only in the pairs of pages $p_3^{(k)} \to p_4^{(k)}$ and $p_2^{(k+1)} \to p_3^{(k+1)}$ (blue dot-dash line). In these cases, the content extraction may be performed by limiting the analysis to the identified blocks, ignoring the remaining page. This results in a more accurate weighting of the text content extracted from the visited pages. Blocks related to other information, ads and navigation elements are not considered (see Fig. 3).

Due to short text snippets or vocabulary problems (i.e., different words used to express similar meanings), the other pairs of consecutive pages do not show any correlation. Anyway, it is still possible to identify relevant content. In the i-session in Fig. 7 the overlapping content between the pages $p_3^{(k)} \to p_4^{(k)}$ results in a few keywords therefore a semantic relationship is hard to be established. In spite of that, the two regions have already be found similar on $p_2^{(k)} \to p_3^{(k)}$. This evidence allows the text in $p_4^{(k)}$'s region to see its content noticeably weighted. Same criteria is met in $k+1$-session between $p_2^{(k+1)} \to p_3^{(k+1)}$ pages.

This statistical approach comes in handy to address the circumstances where two regions are accidentally found similar one another because they contain short and misleading common contents. Most of the times that content is very far from the user intentions. Since the approach utilizes previously acquired evidence in subsequent ranking of relevant information, if the two regions have been seldom found similar, the weight of the retrieved content is relatively low. It makes the extraction less affected by false-positive matching.

Since our ultimate goal is to extract and weight relevant content from the browsing sessions, the evaluation methodology discussed in Sect. 7 is focused on assessing this aspect in the typical personalized search task.

In the following sections we give account of the techniques required for the execution of the approach under discussion, namely, the representation of web pages, clustering of templates and semantic similarity measures.

5.2 Representation of web pages

The generalization ability of a classification algorithm depends on the appropriateness of the representation of the instances for the given task. In this step, a tree-based representation is assigned to each visited page. These representations correspond to the input of the I stage.

Web pages can be treated as semi-structured documents. Generally, web authors organize the page content to make it easy for reading. Thus, semantically coherent content is usually grouped together and the entire page is divided into regions with the help of visual separators such as lines, blank areas, images, different font size and colors defined by specific HTML elements.

Web pages can be naturally represented as labeled ordered rooted trees, where labels represent the tags proper of the HTML mark-up language syntax, e.g., *Document*, *DocumentType*, *Element*, *Text*, *Comment*. Tree hierarchies represent the nesting levels of the elements constituting the page. That representation is named *DOM*-based (Document Object Model) [32]. Among the available node types, those most relevant to our purpose are *Element* and *Text* nodes, corresponding to HTML tags and textual content, respectively. An example of a tree representation is to be found in Figure 8b.

A pre-processing step involves a syntax checker [15] that cleans up most of the malformed code generated by faulty templates. Since the content is organized by a limited number of tag, a simplified DOM-based tree (Figure 8c illustrates an example) is obtained by filtering out unnecessary tags and considering the following most relevant ones:

$$Tags \leftarrow \left\{ \begin{array}{c} \text{<DIV>}, \text{<SPAN>}, \text{<TABLE>} \\ \text{<TD>}, \text{<TH>}, \text{<TR>}, \text{<OL>}, \text{<OPTION>}, \text{<UL>} \end{array} \right\} \tag{3}$$

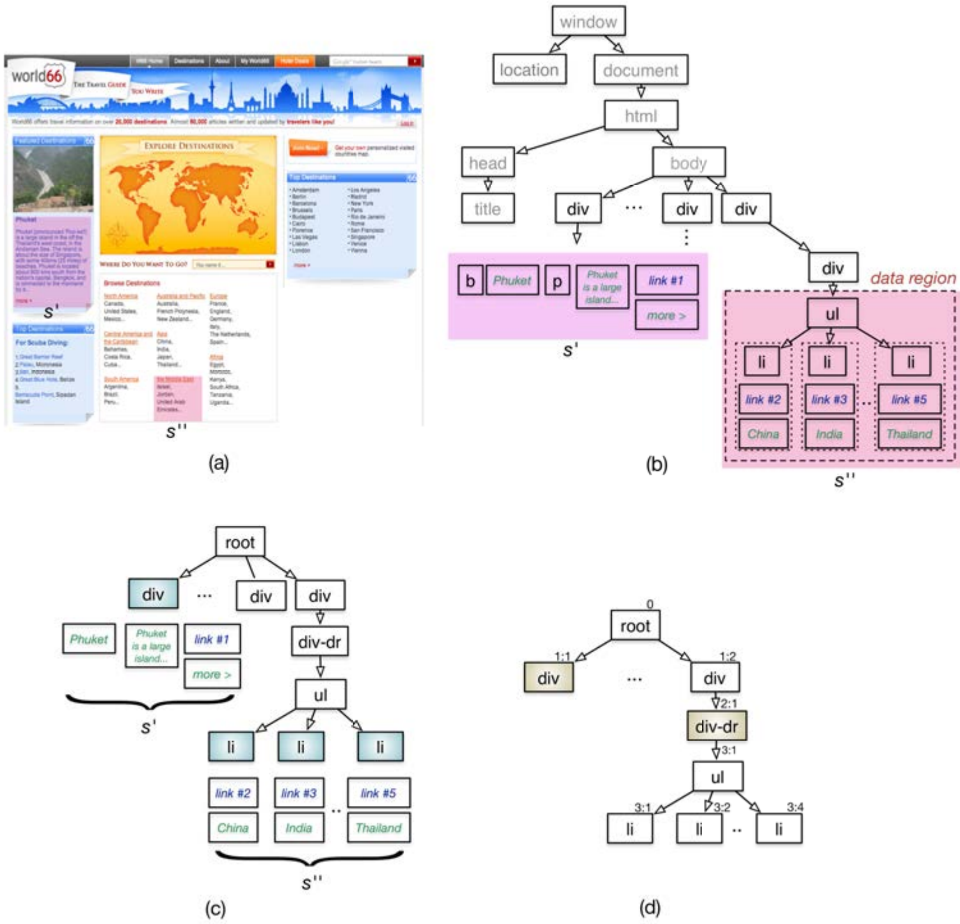

Fig. 8 A traditional DOM-based representation of a web page (b), a simplified version where data regions are identified (b), and a representation of a cluster of similar pages (c).

The representation of a web page p is therefore defined as an ordered and labeled tree T_p, where each node is being assigned a symbol from a fixed finite alphabet $Tags$. The following function:

$$tag : T_p' \rightarrow Tags \tag{4}$$

merely returns the tag of a node. The notation T_p' represents the set of nodes of T_p.

The *size* of T_p, denoted by $|T_p|$, corresponds to the number of its nodes. The depth of a node $v \in T_p$, output of the function $depth(v, T_p)$, is the number of edges on the path from v to the root of T_p. By extension, $|T_p|_d$ denotes the maximum depth in T_p.

5.2.1 Data region mining

Web pages may contain several repeated regions with different contents. Since a correlation binds only one of these regions with the subsequent page, the chance to see the same specific block in the future rarely happens. That is why one more step aims at generalizing groups of blocks forming a single *data region*. These regions are part of the tree-based representation and replace a sub-tree with a special node. Potential correlations starting from one of the sub-tree's nodes take on the reference to that special node.

A sequence of data records containing descriptions of a set of similar objects are typically rendered in a contiguous region of a page and formatted using similar tags. Typical examples of data regions are ordered lists, menus, results of search engines and lateral navigation bars. The p_i page in Fig. 5a shows three records in a data region named *Things to do in Rome*. Because our intent is to define semantic relationships between regions regardless of the specific HTML element containing the followed link, the relationship is generalized to the whole data region dr containing that link. Figure. 5b shows how the semantic relationship binds the element <DIV>, root of the data region *Things to do in Rome*, with the region on p_{i+1}.

Formally, a data region can be defined as a subset $V \subset T'_p$ of two or more nodes satisfying the following properties:

(1) $\forall v \in V$, $depth(v, T_p) = c$, where c is a constant.

(2) $\forall v \in V$, the parent of v is v', where $v' \in T'_p$.

(3) All the nodes in V are adjacent.

(4) The normalized edit distance between two adjacent $v', v'' \in V$ is less than a fixed threshold.

Figure 8c clearly shows one data region whose root is the <UL> tag that has repeated sub-trees beneath.

The root of a data region correspond to a node in T_p. In the example, it is denoted by the <DIV-DR> tag. A definition of the edit distance for two DOM-subtrees is introduced in the following section.

The identification of data regions on web pages relies on the popular iterative approach proposed by Liu *et al.* [47]. Basically, the algorithm follows a depth-first traversal of the DOM-tree from the root downward. At each internal node it compares various combinations of the children sub-trees by means of the same previously-mentioned tree edit distance. When two or more sub-trees satisfy the data region properties, they will be considered as potential candidates along with their parent node.

The process ends up with the candidate data regions at the highest depth, that is, the ones that include the other candidates.

5.3 Clustering

An increasing number of documents on the web are automatically generated according to predefined templates. Various studies report that templates represent between 40% and 50% of the content on the Web [85], with a trend that seems to be increasing this set at a rate of 6% per year [27].

Templates provide the authors with an easy to manage uniform look and feel, and can be seen as frameworks which are filled with different contents to compile the final pages. As side effect, the source code of template-generated documents is always very similar, resulting in slight alterations of the DOM-tree structure among pages. Empirically, dynamically generated pages from a particular site tend to fall into a few clusters representing each a template structure, a phenomenon massively exploited by information extraction algorithms for mining large amount of structured data [21]. Statistics about the ratio between the number of clusters created during browsed sessions collected from a groups of users are to be found in Sect. 9.3.

The goal of this step ($I.i$ stage) is to populate a knowledge base KB_c with groups of pages sharing similar templates. The input of this step is the tree-based representation obtained by the elaboration described in Sect. 5.2.

The cluster of the p page is an approximate representation of the HTML template that the web server uses to generate p. Therefore, the cluster itself is a tree structure T_c, where each node is a symbol extracted from the same finite alphabet used for the page representation. An example of two clusters grouping similar pages is shown in Fig. 5b.

KB_c is the set of clusters in the local knowledge base which is incrementally updated each time a visited page has a template that is not similar to the stored ones. If Γ is the set of potential ordered trees, KB_c corresponds to the a subset of Γ, therefore, $KB_c \in \mathcal{P}(\Gamma)$. The principal task of clustering is to define a function $c : \Gamma \mapsto \Gamma$ that, given a tree T_p, returns the most similar cluster T_c to p. The task is based on the similarity measure $d(T_p, T_c)$, which is expressed by the *tree edit distance*. It represents the minimum-cost sequence of node edit operations that transform T_p into T_c.

Calculating the tree edit costs for DOM-based trees have some advantages in comparison with a general purpose tree edit distance because the root node is known, the sibling nodes are ordered and similar sub-trees from different pages are hardly ever changing their distance to the root node [31]. The Restricted Top-Down Mapping (RTDM) algorithm has proven to perform well in calculating the distance in the web scenario [67]. Similar techniques reach precision and F1 levels of more than 90% [86,1,85].

In short, the algorithm first determines the identical sub-trees occurring at the same level of the input trees. Once the vertices in those sub-trees are grouped in equivalent classes, the minimal restricted top-down mapping between the trees is obtained. While it shows a worst-case complexity of $O(|T_p||T_c|)$, in practice it performs much better due to the above-mentioned characteristics of the DOM-based representations.

To put it more formally, for a given page p we define the clustering function $c(\cdot)$ as follows:

$$c(T_p) = \begin{cases} T_{c'} \leftarrow \text{argmin}_{T_{c''} \in KB_c}\, d(T_p, T_{c''}), & \text{if } d(T_p, T_{c''}) < k_d \\ T_{c'} \leftarrow T_p, KB_c \leftarrow KB_c \cup \{T_p\}, & \text{otherwise} \end{cases} \tag{5}$$

where the page p is assigned to the cluster T_c that has the minimum distance to p. If the distance is above a given threshold k_d, the function returns a new cluster corresponding to the tree representation of the page T_p. KB_c is incrementally updated each time a new cluster is built.

The following three edit operations at the level of single nodes in a tree T are considered:

- **Deletion** Delete a non-root node v in T with parent v', making the children of v become the children of v'. The children are inserted in the place of v as a subsequence in the left-to-right order of the children of v'.
- **Insertion** The complement of delete. Insert a node v as a child of v' in T making v the parent of a consecutive subsequence of the children of v'.
- **Relabel** Change the HTML element assigned to a node v in T.

In order to obtain the tree edit distance between the page p and centroid T_c, the sequence of operations for transforming T_p into T_c, i.e., the *mapping*, is obtained. If the function $c(T_p)$ returns a previous stored cluster, these operations are used to update T_c. In particular, the new nodes that the current page p introduces but which have never seen in the pages already belonging to the cluster are merged with T_c.

We find that most of recent web pages use style sheets, where <DIV> and <SPAN> define the structural organization, whereas older pages use HTML table tags, e.g., <TABLE> and <TD>. Nevertheless, additional tags are sometimes employed for further refining layouts. To improve the accuracy of the tree comparison, nodes are arranged in the following two categories:

$$\begin{aligned} Tags_{Hi} &\leftarrow \{\text{<DIV>}, \text{<SPAN>}, \text{<TABLE>}\} \\ Tags_{Lo} &\leftarrow \{\text{<TD>}, \text{<TH>}, \text{<TR>}, \text{<OL>}, \text{<OPTION>}, \text{<UL>}\} \end{aligned} \tag{6}$$

Given a node v in a cluster T_c, we define the following function:

$$freq : \mathcal{P}(\mathit{\Gamma}) \times T'_p \to \mathbb{N} \tag{7}$$

that, given a knowledge base KB_c, returns the number of pages associated to T_c containing the node $v \in T'_p$. Basically, it assigns greater significance to nodes that best represent the template. Hereafter, $|T_c|_p$ denotes the total number of pages included in T_c.

The cost model of the vertex insertion, removal and replacement is defined by the function g as follows:

$$g(v, T_p, T_c) = \begin{cases} w(v)\ \frac{freq(T_c, v)}{|T_c|_p} & \text{for delete op} \\ w(v)\left[1 - \frac{depth(v, T_p)}{|T_p|_d}\right] & \text{for insert op} \\ w(v) & \text{for relabel op} \end{cases} \tag{8}$$

where $w(\cdot)$ is a surjective function mapping the v's categories to $\mathbb{R}$:

$$w(v) = \begin{cases} w_h & \text{if } tag(v) \in Tags_{Hi} \\ w_l & \text{if } tag(v) \in Tags_{Lo} \end{cases} \tag{9}$$

where w_h and w_l are two constants. Basically, the cost function $g(\cdot)$ returns high values for delete operations if the cluster T_c has a node missing in the current page and its frequency is high (i.e., it has seen in most of the pages in the cluster). By contrast, if the page T_p contains a node never seen before, the insertion cost gets high values if the node is at the top of the tree. The rationale is to give more importance to elements frequently occurring in a cluster and top elements that determine the essential structure of web pages.

As a result of the clustering step, each cluster tends to grow and include slight alterations of the templates that website managers may consider over time. Node frequencies in the cluster allow us to increase the influence of the subtrees that better represent the associated template.

5.4 Extracting relevant correlations between semantic regions

Once we defined a representation suitable for clustering pages according to their content structure, the semantic correlations between two consecutive pages are considered. This stage ($I.ii$) is decomposed in two steps: the identification of the semantic regions and the extraction of potential correlations between these regions.

5.4.1 Semantic region recognition

The first step takes as input the tree-based representation (Sect. 5.2) of each visited page $p \in P^{(k)}$ and identifies the semantic regions Φ'_p on p. Web authors organize semantically coherent content in such a way that it is surrounded by structural elements, such as margins, paddings and borders [21]. In terms of HTML elements, these layouts are mostly defined by the <DIV>, <SPAN>, <TABLE> tags and the others included in the $Tags$ set.

The authors of [24] propose to solve this problem by first starting of the leaves of the DOM-based representation of a page, collects each node v whose tag is in $Tags$, which contains significant amount of text. The pages is therefore split in units whose boundaries are arranged by HTML tags and the text is retrieved by the deepest units. Because each region can be identified by its root node in the DOM-based representation, we have $\Phi'_p \subset T'_p$. A high-level description can be summarized as follows:

input : A labeled tree T_p

output: The set of semantic regions Φ'_p

$\Phi'_p \leftarrow \emptyset$;

$V' \leftarrow \emptyset$;

$V \leftarrow$ leafs of T_p;

while V *is not empty* **do**

  foreach *element* v *of* V **do**

    $V' \leftarrow V' + \{v\}$; $V \leftarrow V - \{v\}$;

    if $tag(v) \in Tags \wedge$ `text(`v`)` *length is above* k_t *words* **then**

      $\Phi'_p \leftarrow \Phi'_p + \{v\}$;

      $V' \leftarrow V' +$ `children(`v`)` ;

    end

    if `parent(`v`)` $\neq$ root **then**

      $V \leftarrow V + (\{$`parent(`v`)`$\} \cap V')$;

    end

  end

end

Algorithm 1: Retrieval of semantic regions.

where the functions `parent`(v) and `children`(v) return the parent and the children nodes of v, respectively; `text`(v) collects the text in the form of sequence of words contained in v and its descendants, and, finally, k_t is a constant.

Figure. 8b shows two semantic regions, s' and s'', identified by the highlighted boxes whose roots are two <DIV> nodes. The leaf nodes of the simplified tree in Fig. 8c corresponds to the text content of the two regions. Because every page tree T_p is associated to a cluster, for the sake of clarity, a unique serial identifier is assigned to each node in $c(T_p)$, as shown Figure. 8d.

5.4.2 Correlation extraction

Once each browsed page p is split to a set of non-overlapping text fragments, we begin analyzing pairs of contiguous pages $p_i \rightarrow p_{i+1}$. The goal of this step is building up relevant statistics between pairs of regions whose content is frequently similar one another. Those statistics are stored in two knowledge bases, namely, $KB^{(+)}$ and $KB^{(\Gamma)}$. The former actually stores the correlations, the latter how many times pairs of clusters sequentially appear in the past sessions and is used for normalization.

Given the semantic region $s'_{p_i \to p_{i+1}}$ that includes that followed link, we identify the set:

$$S_{p_{i+1}} = \{s'' | s'' \in \phi_{p_{i+1}} \wedge sim(\texttt{text}(s'_{p_i \to p_{i+1}}), \texttt{text}(s'')) > k_s\} \tag{10}$$

that consists of pointed page's semantic regions which have a content correlated with $s'_{p_i \to p_{i+1}}$ (see diagram in Fig. 9). The function $sim(\cdot, \cdot) \to [0, 1]$ performs a similarity measure between two textual contents while k_s is a constant threshold. Section 8 discusses comparative accuracies of different measures in the task under discussion.

The identified semantic correlations of each pair of visited pages are incrementally stored in a local knowledge base $KB^{(+)}$ composed of a multiset of tuples member of the following data domain:

$$KB_c \times \Gamma' \times KB_c \times \Gamma' \tag{11}$$

In particular, the multiset of tuples is obtained as follows:

$$\{< c(p_i), s'_{p_i \to pi+1}, c(p_{i+1}), s'' > | s'' \in S_{p_{i+1}}\} \tag{12}$$

Intuitively, the set of tuples summarizes the semantic connections found between pages by analyzing the browsing activity. For instance, given the pair of pages in Figure 5, the following tuple will be stored in the KB: $< c(p_i)$,*2:3*, $c(p_{i+1})$,*2:1*$>$.

A further multiset of tuples of interest, denoted by $KB^{(\Gamma)}$ with domain $\Gamma \times \Gamma$, is obtained as follows:

$$\{< c(p_i), c(p_{i+1}) >\} \tag{13}$$

It merely keeps track of the times a pair of regions, part of two successively visited pages, respectively, occurred in the past. The occurrences are counted without regard to their content correlation.

By examining the sessions in Figure 7, assuming that pages $p_1^{(k)}$ and $p_1^{(k+1)}$ are clustered in c_1, $p_2^{(k)}$, $p_3^{(k)}$, $p_4^{(k)}$ and $p_2^{(k+1)}$ in c_2; and $p_3^{(k+1)}$ in c_3; $KB^{(\Gamma)}$ would store the following tuples:

$$\{< c_1, c_2 >, < c_1, c_2 >, < c_2, c_2 >, < c_2, c_2 >, < c_2, c_3 >\} \tag{14}$$

The proposed formalism extends the tree representation of clusters with a set of ordered pairs of vertices, that is, directed edges that connect two nodes from the same or different clusters in KB_c. Let us recall the example of that kind on two clusters connected by a dash-dot line in Figure 5b.

The assumption at the root of link analysis is that hyperlinks establish relationships between two pages. In our approach, a link from p_i to p_{i+1} indicates a relationship between $s'_{p_i \to p_{i+1}}$ on p_i, which includes the link, and one or more regions s'' on p_{i+1}. According to the acquired evidence, we are able to distinguish between informative and organizational links. The former kind of links provides better evidence related to the current user interests because they are used to build semantic connections between different contents. Organizational links usually connect unrelated blocks, therefore the knowledge base $KB^{(+)}$ has less chance to include tuples related to those ones.

5.5 Exploiting the acquired evidence

The last stage (II) retrieves text information related to the current user interests. More formally, given a browsing session $P^{(k+1)}$ and the experience $KB^{(+)}$ and $KB^{(\Gamma)}$ acquired during the previous browsing activities $(P^{(1)}, P^{(2)}, \cdots, P^{(k)})$, our goal is to output the $\Theta^{(k+1)}$ model of the interests related to $P^{(k+1)}$.

Each pair of consecutive pages $p_i \to p_{i+1}$ in $P^{(k+1)}$ of the input session is subjected to clustering and extraction of relevant correlations, as described in Sect. 5.3 and 5.4, respectively. Therefore, the semantic region $s'_{p_i \to p_{i+1}}$ in p_i and the associated set $S_{p_{i+1}}$ of regions in p_{i+1} correlated with $s'_{p_i \to p_{i+1}}$ are obtained. Figure 9 depicts these elements.

In principle, once the semantic region $s'_{p_i \to p_{i+1}}$ that includes the followed link, and the pointed regions $S_{p_{i+1}}$ that show some sort of correlation with the former, the text retrieved by these regions (pink blocks in Fig. 9) can be considered part of the model of interest. By iterating this task over the session $P^{(k+1)}$, the entire text can be assigned to $\Theta^{(k+1)}$. However, our goal is to exploit any evidence that two regions have been previously found similar in order to better determining the relevant keywords in the model.

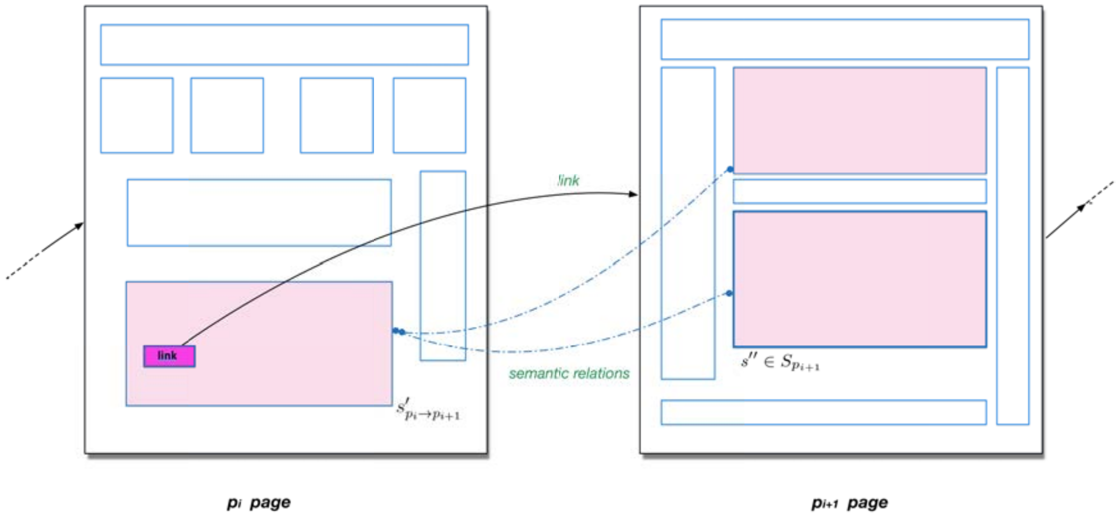

Fig. 9 Two pages in the browsing session $P^{(k+1)}$.

Applying the relational algebra's projection operator to the knowledge bases $KB^{(+)}$ and $KB^{(\Gamma)}$, the subset of correlations related to pairs of structurally similar regions can be obtained as follows:

$$N^{(+)}_{p_i \to p_{i+1}} = \bigcup_{s'' \in S_{p_{i+1}}} \pi_{c(p_i), s'_{p_i \to p_{i+1}}, c(p_{i+1}), s''}(KB^{(+)}) \tag{15a}$$

$$N^{(\Gamma)}_{p_i \to p_{i+1}} = \bigcup_{s'' \in S_{p_{i+1}}} \pi_{s'_{p_i \to p_{i+1}}, s''}(KB^{(\Gamma)}) \tag{15b}$$

The projection's attributes correspond to the clusters assigned to p_i and p_{i+1} and the semantic regions under consideration. The Eq. 15a collects the stored correlations that bind two clusters matching $c(p_i)$ and $c(p_{i+1})$, respectively. Moreover, the correlations must also bind the same regions in $p_i \to p_{i+1}$ that are currently identified as semantically correlated. Similarly, Eq. 15b returns pairs of regions part of two successively visited pages, with no regard to their semantic correlation.

Since we are interested in the number of occurrences of the so obtained sets, $n^{(+)}_{p_i \to p_{i+1}}$ and $n^{(\Gamma)}_{p_i \to p_{i+1}}$ denote the cardinalities of the two multisets obtained by the Eq. 15a and 15b, respectively.

Finally, the *boosting factor* w can be introduced as follows:

$$w_{p_i \to p_{i+1}} = \begin{cases} \frac{n^{(+)}_{p_i \to p_{i+1}}}{n^{(\Gamma)}_{p_i \to p_{i+1}}}, & \text{if } n^{(\Gamma)}_{p_i \to p_{i+1}} > 0 \\ 1, & \text{otherwise} \end{cases} \tag{16}$$

That factor is computed at each pair of visited pages $p_i \to p_{i+1}$. It gets low values if the two regions were rarely being judged similar on previous browsing sessions. Instead, it shows increased values if the regions have always been found strongly semantically correlated. The $n^{(\Gamma)}$ value is basically used for normalization. If the templates of the current pages occurred many times in the previous sessions, the boosting factor gets high values only if many semantic correlations were also found.

In the case the KBs do not provide any evidence from the past sessions, $n^{(\Gamma)}$ is 0 and the boosting factor gets value 1. In other words, the model is built by considering only the current semantic correlations extracted from each pair of visited pages.

Similarly to $\texttt{text}(\cdot)$, the function $\overrightarrow{\texttt{text}}(v)$ returns a vector representation of the text enclosed in v, where the weights are assigned by means of a tf·idf weighting scheme. The idf values are computed by taking into consideration the text content of the whole collection of sessions up to the currently visited page.

The interest model $\overrightarrow{\Theta}^{(k+1)}$ is incrementally updated at each pair of sequential pages belonging to the same session. In particular, the contribution for the pair $p_i \rightarrow p_{i+1}$ is given as follows:

$$\overrightarrow{\Theta}^{(k+1)}_{p_i \rightarrow p_{i+1}} = \overrightarrow{\texttt{text}}(s'_{p_i \rightarrow p_{i+1}}) + w_{p_i \rightarrow p_{i+1}} \sum_{s'' \in S_{p_{i+1}}} \overrightarrow{\texttt{text}}(s'') \tag{17}$$

where the former contribution is derived by the content of the semantic region that contains the followed link, and the latter is built by collecting the content of correlated regions identified by the Eq. 10. This form of term weighting is inspired by the well-known relevance feedback approach proposed in the Rocchio algorithm [70].

By iterating over the browsing session $P^{(k+1)}$, the expected interest model is obtained with the following:

$$\overrightarrow{\Theta}^{(k+1)} = \sum_{i=1}^{|P^{(k+1)}|-1} \overrightarrow{\Theta}^{(k+1)}_{p_i \rightarrow p_{i+1}} \tag{18}$$

For instance, by considering the session $k+1$ in Figure 7, the pair of clusters $< c(p_1^{(k+1)}), c(p_2^{(k+1)}) >$ corresponds to $< c(p_1^{(k)}), c(p_2^{(k)}) >$, both stored in $KB^{(\Gamma)}$. Moreover, between $p_1^{(k)} \rightarrow p_2^{(k)}$ a semantic correlation has been recognized. By analyzing the pair $p_1^{(k+1)} \rightarrow p_2^{(k+1)}$, the following statistics are therefore obtained:

$$n^{(+)}_{p_1^{(k+1)} \rightarrow p_2^{(k+1)}} = 2 \tag{19a}$$

$$n^{(\Gamma)}_{p_1^{(k+1)} \rightarrow p_2^{(k+1)}} = 2 \tag{19b}$$

According to Equation 16, the boosting factor $w_{p_1^{(k+1)} \rightarrow p_2^{(k+1)}}$ is 1. If the semantic correlation on $p_1^{(k)} \rightarrow p_2^{(k)}$ was missing, less evidence would suggest that the text extracted from the identified region in $p_2^{(k+1)}$ was relevant. Indeed, the $KB^{(+)}$ would miss the tuple associated with that missing correlation, obtaining the following variation:

$$n^{(+)}_{p_1^{(k+1)} \rightarrow p_2^{(k+1)}} = 1 \tag{20}$$

and a boosting factor of 0.5. In other words, the text extracted from the $p_2^{(k+1)}$'s region is half-weighted in the construction of the interest model.

The just described approach (from now on named **EXP**), which builds interest models by analyzing input browsing sessions, suffers of one drawback. In circumstances in which new templates are found (e.g., websites with templates never seen in the past), $KB^{(\Gamma)}$ does not provide any evidence from the past sessions. As already mentioned, EXP is still able to find semantic correlations by exploiting the function *sim* introduced in the Sect. 5.4.2, but since the boosting factor w would get value 1, the accuracy of the extraction is limited. By analyzing the outcomes of the comparative evaluation (Sect. 7), content-based approaches that take into account specific elements of each single browsed page, e.g., titles and anchors, generate adequate approximations of the interest models. For this reason, an hybrid approach named **HEM** is introduced. It simply combines EXP and MR, that is, the approaches that obtained the best performances during the experiments. The EXP approach is considered under normal circumstances. In case $KB^{(\Gamma)}$ does not provide any evidence from the past sessions, which is represented by the condition $n^{(\Gamma)}_{p_i \rightarrow p_{i+1}} = 0$ in Eq. 16, the MR approach is taking over in the construction of the current interest model. The assumption is that, whenever pages with templates never seen before are visited, a content-based approach based on noun phrases, titles and metadata keywords provides better outcomes.

6 Analysis of the computational complexity

The computational complexity of the approach we just described is linearly dependent with the number of clusters stored in KB_c. Specifically, the overall complexity of the tree-edit distance and clustering approach is $O(N|T_p||T_c||KB_c|)$, where N is the length of the input browsing session. The semantic region recognition is obtained during the parsing of the web page required for the tree-based representation.

In a real scenario, most of the visited pages are grouped in few clusters and KB_c assumes bounded cardinality, hypothesis empirically supported by the evaluation of a dataset of browsing sessions discussed in Sect. 9.2. Nevertheless, as the browsing sessions tend to mount up spanning several months, the chance to see pages with new templates increases and, therefore, the number of new clusters.

Since the approach strongly relies on the tree-based representations of pages and clusters, the computational complexity differs from other modeling approaches as shown in Table 2. The complexity of MR is influenced by the noun phrase extraction. Most of the natural language parsers for noun phrase extraction exploit probabilistic context-free grammars and are particularly slow in case of long input [42]. It gets very difficult to obtain the output from pages with long text, making the approach not feasible for daily use. Specific adaptations have been implemented to include it in the evaluation, see Sect. 7.

Because TDH is based on a local search engine that is not affected by any form of *forgetting*, it sees its capacity growing more and more. So that the complexity is a function of the number of pages visited so far, and of the set of keywords extracted, i.e., the search engine's dictionary.

By contrast, the SHY approach builds the profile by considering a limited number of recent browsing sessions, and takes advantage of the quick Rocchio algorithm for the construction of the interest model. For this reason, it shows the lowest complexity among the considered techniques, which show any form of adaptation to the visited pages.

As for wall-clock running times, the build-up of local indexes related to the visited content or the identification of semantic relationships between regions, makes the computational requirement of TDH, SHY and EXP between 2 and 4 times higher than others.

In order to process the 15,5 thousands sessions of the corpus-based evaluation (Sect. 7.1), TDH, SHY and EXP required 161, 73.4 and 166.8 hours, respectively, whereas BP, MR and PX needed 5.1, 40.8 and 33.3 hours. It must be also said that, the backend implemented in the EXP prototype is based on a standard SQL database, which is less adequate for storing and retrieving binary tree-based structures.

Table 2 Complexity of the most relevant approaches in the literature.

Approach	Ref.	Complexity
MR	[50]	$O(NL_t^3)$, where L_t is the average number of words on a web page.
SHY	[76]	$O(NL_t)$
TDH	[79]	$O(NL_p\|V_t\|)$, where L_p is the total number of visited pages.
EXP	-	$O(N\|T_p\|\|T_c\|\|KB_c\|)$

A number of workarounds have been developed to keep the EXP complexity bound so that computational resources of common personal computers are enough for the algorithm execution. Hereafter, we briefly introduce solutions to scale our approach.

6.0.1 Hostname-based matching priority

Since performing an exhaustive search over a large set of clusters is infeasible, the key insight is to prune the search space.

In particular, the tree edit distance will be evaluated first on the clusters that include pages from the same hostname of the current one. On the circumstance when no cluster matches with a distance below the k_d threshold constant, the calculation will be extended to the rest of clusters in KB_c. The idea is favoring the templates generated by the same website because, more likely than not, those templates are distinctive of the page layouts of the site itself. Nevertheless, seldom templates are not associated to a particular domain but are shared among several websites. Popular cases are themes of popular public forums or content management system (e.g., Wordpress, Drupal). For this reason, the rest of the KB_c will not be ignored if the clusters containing pages from the exact same hostname are not deemed similar enough.

6.0.2 Simple-tree matching

Even though RTDM is reported to usually behave better in practice, it still does have a worst-case quadratic time complexity. If the trees are particularly complex, the calculus of the distance measure is compute-intensive. In this scenario, we introduce a lightweight distance to identify tree-pair candidates with high similarity.

A simplified tree is built and kept updated for each cluster in KB_c by considering only tags in the *HiRel* category. An example is depicted in Fig. 8. In other words, each cluster is mapped to a smaller tree, empirically 35.3% percent of the original on average according to the browsing histories considered for evaluation in Sect. 9.2.

During the clustering, the tree edit distance is first calculated between the simplified versions of each potential cluster and the simplified tree built from T_p, respectively. If that measure is below the threshold k_d, the distance is than evaluated on the standard representation. Consequently, the number of nodes analyzed for each cluster that does not represent the current page's template is substantially reduced.

It is easy to prove that if the distance measure on simplified pairs of trees is above k_d, the same measure evaluated on the corresponding standard trees is still above the threshold, therefore, the clustering accuracy is not affected.

6.0.3 Pre-Pruning

One more optimization is performed during the tree edit distance calculus. The recursive formula used by the RTDM algorithm for the $d(T_p, T_c)$ calculus has the characteristic of updating the temporary distance with positive increments, that is, the cost of the operation on the node currently under consideration. In other words, the calculus of the distance will never decrease its value.

Therefore, once a cluster with distance d' is found, that value is assigned to the maximum threshold the future clusters must satisfy. If it happens that the partial distance obtained by the RTDM algorithm on the current cluster gets a value higher than d', the calculus can be early-stopped. If the final distance gets values less than d', the latter is updated accordingly. This optimization reduces the time spent on templates that clearly show different structures with the current page.

6.0.4 Forgetting

Finally, in order to combat the proliferation of clusters after many browsing sessions, we monitor long periods of inactivity (i.e., 60 days). The clusters that have not been subjected of any alteration in terms of new pages that have been put in, are removed from KB_c. This step helps us to limit the number of comparisons and keep the storage requirements bounded.

7 Evaluation methodology

In order to assess the effectiveness of the proposed approach, the accuracy of the content represented by the interest model, which defines a level of preference over a set of keywords after a browsing activity, has to be evaluated. Due to the subjectivity of human perception, assessing the effective relevance of each keyword is challenging and requires time-consuming procedures. For this reason, evaluation methodologies often exploit these models for collecting additional resources w.r.t. the current interests and, accordingly, assessing their relevance [83].

In particular, given a browsing session $P^{(k)}$, the top-ranked keywords extracted from the interest model $\overrightarrow{\Theta}^{(k)}$ compose a web search query. In this scenario, users are asked to provide relevance assessments over the content of the recommended resources retrieved by a search engine. This strategy allows us to take up the traditional IR evaluation metrics for performance comparison [52], e.g., how many of the retrieved results are judged useful by the user. Similar evaluation approaches have been undertaken by a number of authors, see for example [77, 19, 93, 50, 84].

Two different experiments are discussed. A *corpus-based* experimental methodology is first described in Sect. 7.1. It consists of a large-scale off-line evaluation of different interest modeling strategies. A comparative evaluation framework over a dataset of news pages is defined for simulating short browsing activities. Section 7.2 describes a *field-based* experiment for accuracy assessment in a real-world environment. In this on-line study we analyze the feedback of the users exposed to recommendations generated by considering their histories.

Field-based evaluations are complementary to the batch processing approach. They are fundamental from the qualitative point of view since the effectiveness is evaluated by humans in real-world environments [84]. Nevertheless, users are required to judge large sets of documents so, due to the cognitive burden and long time to complete the tasks, they are limited in its realization [50]. On the other hand, corpus-based experiments provide comparable results within the same retrieval scenarios considering larger sets of input data.

So far as we are aware, this is the first comparative framework that aims at estimating the effectiveness of different strategies for representing interest models by analyzing browsing activities.

The comparative evaluation includes the algorithms reported in Table 3, with specific adaptations for the kind of considered experiments. Section 3 reports a brief description of each approach.

Table 3 Approaches considered in the evaluation.

Approach	Ref.	Notes
W	-	Simple strategy that collects the text content from all the visited sessions' pages and extracts the most frequent terms, ignoring common stopwords.
BP	[43]	Extraction limited to the pages in the current session.
MR	[50]	The inverse document frequency is estimated by extracting statistics from the Google N-Gram corpus [29]. The initial queries correspond to the titles and the anchors of the current browsing session. Because the noun phrase extraction of the MR approach is a compute-intensive task, the extraction has been limited to the first sentences of the text extracted from each page.
PX	[24]	-
SHY	[76]	10 sessions per day, with a history of browsing activities spanning 10 days (i.e., a total amount of 100 browsing sessions profiled). According to the definition of the approach, the Rocchio algorithm [4] expands the initial query considering both the short and the long-term collected preferences by considering the previous browsing activities.
TDH	[79]	-
EXP	Sect.5	The proposed approach.
HEM	Sect.5.5	A hybrid approach that combines EXP and MR.

With the exception of the two baselines W and BP, the considered approaches make explicit representation of short-term information needs. The SHY, TDH, EXP and HEM approaches, in different ways, build these representations by considering also the content of past browsing activities.

Significance tests between every pair of approaches have all been empirically validated in both the experimental setups by the paired t-test ($P < 0.05$). The preliminary assumption, or null hypothesis H_0, is that two extraction approaches being tested are equivalent in terms of performance.

Experimental outcomes are reported in Sections 9.1 and 9.2, respectively, following the procedure for tuning the parameters of the mining approaches under examination (Sect. 8).

7.1 Corpus-based evaluation setup

In the batch processing paradigm [13], a set of queries is run against a static collection of documents. The task of a retrieval system is to identify those documents relevant to the query. Basically, the user-system interactions are simulated through a well-defined retrieval scenario. This method is worthwhile since it maintains complete control over situational variables and measurements, testing the effectiveness of the algorithms underling the considered approaches in a variety of topics. However, obtaining a large test collection of browsing sessions motivated by clear information needs is a complex task that requires a long time to be accomplished and raises privacy concerns. As far as we

Table 4 Statistics of the news collection aggregated by the four macro-categories: business, entertainment, science & tech and health.

	News corpus G			*2-page session corpus*		
	news pages	stories	sources	sessions	stories	sources
Business	152,746	2,019	5,637	6,091	603	179
Entertainment	152,746	2,076	5,620	9,425	342	306
Science & Tech	108,465	1,789	5,399	-	-	-
Health	45,615	1,347	4,492	-	-	-
All Categories	422,937	7,231	9,311	15,516	945	408

are aware, public domain datasets are not available and, for this reason, our first attempt is to build this collection.

Online newspapers create a rich information landscape of gigantic proportions. The intents behind browsing sessions that include news pages are foundamentally *informational*, that is, the acquisition of some information assumed to be present on one or more pages [12]. Unlike blogs, online forums or discussion boards, each newspaper usually deals with several macro-categories (e.g., Sports, Technology) and hundreds of different topics each day. The availability of continuously updated news content provides great value, but it represents yet another case of information overload problem [10, 55], which often does not help the audience obtaining meaningful and consistent insights.

News aggregators' purpose is to periodically check for new contents from several sources creating unique points of access. Some of these aggregators provide an organized view of the content, clustering all the news about the same *story* or topic s. So we might have the following stories: “Trump and Clinton debate”, “Samsung's Note 7 recall” and “Nestle Recalls Drumstick Ice Cream Cones After Listeria Test”; each one collecting *news pages* from various *sources* (e.g., CNN, Washington Post and Reuters) about the topic at issue.

A corpus G of 422,937 English-language news pages have been collected during a time period of just over 5 months by monitoring four macro-categories of a popular online aggregator [30], namely: Business, Science & Technology, Entertainment and Health. Table 4 shows statistics about each category. The total number of monitored publishers, or sources, is 9,311. On average, each story clusters 58.4 news pages discussing similar topics ($\sigma = 55.084$).

A local text search engine [22] based on the vector-space model indexes the G corpus that becomes the document collection used for testing. In order to build a set of browsing sessions from the news

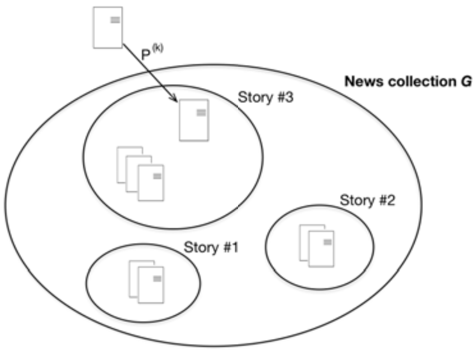

Fig. 10 Partition of the news page collection used for the evaluation.

collection, we begin looking for backlinks, that is, web pages containing a link to one of the pages $p \in G$. The backlink retrieval is conducted by querying a search engine with specific query operators.

Each time a link is found, a 2-page browsing session is identified. Nearly all of those sessions are composed of patterns such as *homepage* $\rightarrow$ *news page*, or *blog page* $\rightarrow$ *news page*, that is, paths frequently followed by users in their everyday browsing activity. A total of 15,516 sessions Π_G have been identified covering two categories, namely, Business and Entertainment. The average number of sessions per story is 16.4 ($\sigma = 60.346$). The entire dataset is made publicly available for download[1] for encouraging the objective comparison with future studies.

Our task is to suggest news in G belonging to the story s of the pages visited by the users (see Fig. 10). Formally, after having visited a browsing session $P^{(k)} \in \Pi_G$, the interest model $\overrightarrow{\Theta}^{(k)}$ is built according to the visited sessions up to k. The vector $\overrightarrow{\Theta}^{(k)}$ is then converted into a query that is submitted to the local search engine and the stories associated with the top retrieved news are evaluated by means of traditional IR measures on sets.

The Π_G collection is chronologically partitioned in 10 equal-sized sub-samples so that the evaluation process is repeated 10 times with each of the 10 sub-samples used once as validation (k-fold cross validation with k=10). The results are averaged to reduce the variability of the outcomes due to the particular ordering of the input sessions. A chronological split is realistic since user profiling usually requires training on currently available material, and then applying the filtering to material that is received later.

The proposed test is tailored to investigate retrieval performances allowing additional insight into the strengths and weaknesses of different extraction mechanisms. The synthetic dataset models each

[1] UCI Machine Learning Repository `https://archive.ics.uci.edu/ml/datasets/News+Aggregator` (Last visited on 15 April 2016)

news article as having a fixed number of properties, namely, the HTML content, the set of browsing sessions that include the news page and the story associated with the news. The so-built dataset does not suffer by data sparsity because, given a topic, all the items in the dataset have been classified as relevant or irrelevant. Moreover, it falls in the test-retest reliability class, where future approaches can be easily taken into consideration for measuring potential performance improvements.

7.2 Field-based evaluation setup

As pointed out by Matthijs and Radlinski [50], it is crucial that interest models are evaluated by users performing regular day-to-day searches driven by information needs so that the hypothesis of the personalization yielding an actual improvement in the search experience is properly evaluated. Furthermore, it allows us to strengthen the corpus-based outcomes on a different dataset.

Subjects involved in this study are graduate students enrolled in Computer Science courses at the Faculty of Engineering with a mean age of 26 years. These 50 students, assumed to have experience with a broad range of software, are required to complete a pre-experiment questionnaire to establish their level of experience in conducting on-line searches. Of course, all the subjects reported that they have a great deal of knowledge with search engines and web browsers; the median search frequency among the user population was 4.16 on-line search per day. Every user underwent a training session of half an hour to ensure they were familiar with the task before beginning the experiments proper.

For a 4-week period each user was asked to analyze her browsing histories. A Java tool installed in the user's personal computer had the function of retrieving the history, irrespectively from the default browser (e.g., Explorer, Firefox, Safari or Chrome). Each browse trail consists of a temporally ordered sequence of URLs per web browser instance. The tool performs a session boundary recognition based on the presence of links that connect two consecutive pages. A traditional session inactivity timeout of 30 minutes is also used to demarcate two different sessions [36, 92]. On average, 67.62 sessions per day have been identified.

The potential intents behind single browsing sessions are complex, covering *informational*, *navigational* and *transactional* goals [12]. For this reason, we asked each user to identify the browsing sessions whose aim is acquiring relevant information, ignoring other kinds of motivations. An upper bound limit of 10 sessions was given to each user. Subjects were asked to think about their online

Table 5 Categories of information seeking activities of the retrieved browsing sessions.

Seeking activity	# sessions (%)
News and Weather	13.5
Technology	15.6
Shopping	11.8
Education	21.2
Travel	18.4
Recreation (e.g., Videos, Games)	14.9
Others	4.6

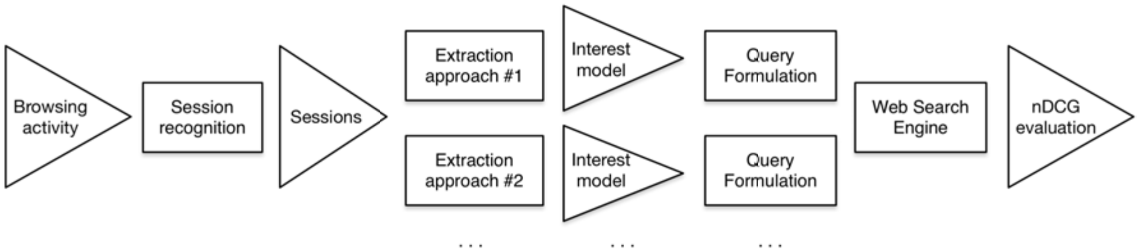

Fig. 11 Field-based evaluation steps.

information-seeking activities in terms of tasks by creating text labels for each session. A summary of the seeking categories grouping those labels are shown in Table 5.

The obtained sessions are subjected to the extraction of user interests. The remaining sessions, that is, the ones initially discarded by the user, is however input to each approach, similarly to the evaluation proposed by White *et al.* [96]. They correspond to the training set that the extraction algorithms can exploit in order to learn related or different interests, and analyze patterns on the visited hypertext information. The micro-average session length is 2.42 pages (σ=1.73).

The output of each considered extraction approach corresponds to the model of user interests related to the current browsing session. The 10 top ranked keywords in the model are submitted to the Microsoft Bing search engine [53] through its API, and the first 10 results are retrieved. The test document collection is therefore the whole Bing's index.

The participants are asked to determine whether they personally find each result relevant or not based on the intents that drove the particular browsing activity. The user relevance is expressed in a three point Likert-type scale: (1) high-relevant to the browsing session; (2) partially relevant and (3) not relevant. So as not to bias the participants, the three sets of results are presented mixed and in random order. An outline of the evaluation steps is depicted in Fig. 11.

Table 6 Quantitative parameters and references.

$w_h = 0.75$	: Eq. 9 (definition of tree edit distance)
$w_l = 0.20$	: Eq. 9 (definition of tree edit distance)
$k_d = 0.28$	: Sect. 6.0.2 (page cluster similarity)
$k_s = 0.80$	: Eq. 10 (text similarity)
$k_t = 10$	: Algo. 1 (semantic region identification)

In order to keep reasonable the number of sessions the user is asked to express judgments, the extraction approaches have been chosen from the ones that obtained better performances in the corpus-based evaluation, namely, BP, MR, EXP and HEM. Accordingly, each of the 50 users submitted 400 judgments.

This experiment falls in the class of evaluations defined for the JITIR (Just-in-Time-Information-Retrieval), where software agents proactively present potentially valuable information based on a person's local context [69, 95]. Because the user determines if an item meets her taste requirements, the relevance is more inherently subjective in this evaluation compared to the corpus-based setting.

8 Parameter tuning

We report the threshold settings and the values of the parameters used in the evaluation for the proposed approach.

As for the clustering, a test set composed of 1,000 web pages randomly chosen from about 100 websites, mostly popular blogs and online newspaper, have been assembled. The websites' hostnames do not overlap with the ones in the corpus-based dataset. Web pages are manually clustered according to common templates.

The thresholds are obtained if the approach produces the most similar cluster for each given page, minimizing the global number of errors (misses and false alarms) in the decisions made. They are automatically tuned by varying their values until the global performance of the classifier obtains good results on the validation set. The iterative gradient-descent is used for this task.

Similarly, a small subset of 500 pages obtained with the same procedure discussed in Sect. 7.1 has been manually examined for tuning the remaining parameters. The values found to be best w.r.t. precision measurements are reported in Table. 6.

As far as the text similarity measure is concerned (Eq. 10), the described approach does not require a particular algorithm to be implemented. Nevertheless, evaluating similarities among text contents is fundamental for recognizing dependencies between regions belonging to different pages. For this reason, we perform a comparison on various similarity measures in the domain under discussion for determining the most accurate. The considered measures are the following:

(CM) Corley and Mihalcea [16] model the similarity of texts as a function of the semantic similarity of the component words. A combination of six different word-based metrics is considered by the authors for determining the similarity between pairs of keywords.

(CS) A traditional cosine similarity, that is, a normalized inner product of two vectors, with a tf·idf weighting scheme [4]. In short, the semantic similarity of two texts is determined by the lexical overlap, i.e., how many words they have in common.

(GR) Mihalcea *et al.* [54] propose a greedy method based on word-to-word similarity measures. For each word in the text $t1$, the maximum similarity score to any word in text $t2$ is determined. Different word-word similarity measures can be considered for this task. In our experiments, we take into consideration: Latent Semantic Analysis (**LSA**) [44,60], Latent Dirichlet Allocation (**LDA**) [11] and the statistical similarity proposed by Lesk (**L**) [5] extended to use WordNet, an online publicly available hand-crafted lexical database [20]. Both LDA and LSA models are developed from the lemmatized Touchstone Applied Science Associates (TASA) corpus [44].

(LSA) The approach proposed by Lintean and Rus [46] for estimating the semantic similarity between two short texts by using the LSA word-word similarity.

(OP) Similarly to (**CM**), Rus and Lintean [71] cast the similarity to a measure between words. Instead of a greedy paradigm, the authors propose to find the best matching using the sailor assignment problem, also known as job assignment, a well-known combinatorial optimization problem. Again, three different word-word similarities are considered, based on **LDA**, **LSA** and Lesnik's similarity **L**, respectively.

The open source SIMILAR toolkit [45] has been employed for the implementation of some of the above-mentioned measures.

To test the effectiveness of the text semantic similarity metrics, the test set used for tuning the cluster algorithm has been extended considering pages that can be reached by a link in that set. Pairs of related regions between two connected pages have been manually identified. A total amount of 2,180 pairs have been used as unsupervised setting. Experimental results in terms of residual sum of

Table 7 Wall-clock running times in seconds to complete the task and Residual sum of squares (RSS) for various text similarity measures.

Similarity Measure	Time (secs)	RSS
CM	7.643	**0.477**
CS	**0.26**	0.489
GR-LDA	7.537	0.559
GR-LSA	7.616	0.556
GR-L	23.035	0.553
LSA	7.582	0.546
OP-LDA	7.588	0.610
OP-LSA	7.584	0.607
OP-L	16.065	0.606

squares (RSS) are reported in Table 7. The RSS is calculated by averaging the discrepancies between the estimated similarity and the expected correlation between text regions, that is, 1 for correlated regions, 0 otherwise.

The CM semantic similarity measure obtains the best results. Intuitively, semantic analysis of text contents has the chance to identify correlations in circumstances where the lexical overlap between texts is missing. This deeper analysis comes at the expense of computational complexity, which is significantly higher, as expressed by the required time to complete the task (7.6 sec). Interestingly, a traditional cosine similarity obtains good outcomes, in spite of its relative simplicity of implementation. At first sight, the good performance of this non-semantic measure looks counterintuitive but it must be said that many of the collected text region pairs are included in pages sharing the same hostnames. It is likely that these pages have been authored by the same person and, therefore, the vocabulary of terms appearing in correlated regions corresponds. In this case, a traditional keywords-based measure looks mostly adequate to draw accurate similarities. Moreover, keyword-based approaches have the advantage to be language-independent, bearing the whole described approach adaptable to a larger amount of web content. For these reasons, the CS similarity has been chosen for the experimental evaluation.

9 Experimental results and discussion

After giving an account of the two experimental methodologies, the outcomes are reported and discussed in the following sections.

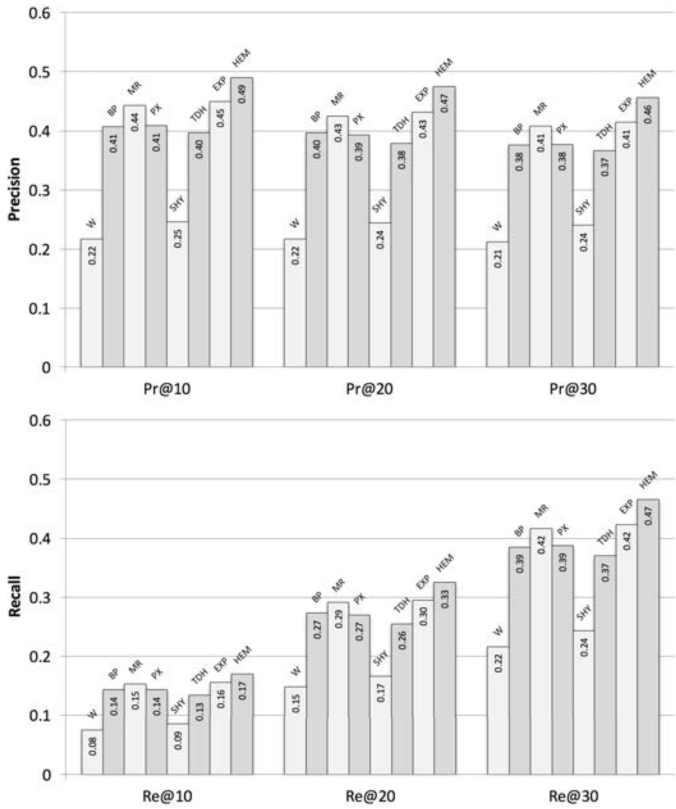

Fig. 12 Overall precision and recall considering the top-ranked pages, with N ∈ {10,20,30}.

9.1 Corpus-based evaluation results

Because the task is finding all the relevant items with binary granularity of true preferences (i.e., the news page belongs to the given story or not), traditional set-based measures such as precision Pr and recall Re measures [4] over the list of documents returned by the local search engine are evaluated as follows:

$$Pr = \frac{tp}{tp + fp} \qquad \text{and} \qquad Re = \frac{tp}{tp + fn} \tag{21a}$$

where tp is the number of retrieved news pages belonging to the same story of the current browsed session (true positives), fp is the number of retrieved web pages that do not belong to the current story (false positives), and fn is the number of web pages related to the story that are not retrieved (false negatives). These measures are computed at different cut-off values, namely, $\{10, 20, 30\}$. Since precision and recall are defined only for a single classification task (i.e., input session), the results of multiple sessions need to be macro averaged to get to a single performance value [99].

The average precision and recall measures considering the top-ranked pages (i.e., 10, 20 and 30 results) retrieved by the local search engine are reported in Fig. 12. The hybrid approach HEM performs the best among the eight considered approaches in terms of both precision and recall. It shows 11% more accuracy in comparison with the single approaches MR and EXP.

BP, PX and TDH comparatively show around 22% less accuracy. The baseline W and the SHY approach behave even less accurately. In the first case, the whole content of the current browsed session contains noise that does not allow identifying relevant terms. In spite of its limited computational requirements, the SHY approach builds up interest profiles by considering the whole content of the last browsed sessions. Therefore, exhibiting similar inaccuracies.

It must be said that Google News aggregator tends to group news pages very selectively, creating several clusters for related or developing stories. For example, each of the following related news belong to distinct stories:

SoftBank acquisition of US telco threatened by $15B offer from French rival

Hostname: `techinasia.com` URL: `http://goo.gl/UxxTxE`

SoftBank Vows 'Price War' if T-Mobile Deal Approved

Hostname: `moneynews.com` URL: `http://goo.gl/y22hs4`

SoftBank CEO hopeful of T-Mobile merger, AT&T chief says it's impossible

Hostname: `techtimes.com` URL: `http://goo.gl/edH28o`

In terms of absolute performances, even if the extraction algorithms are able to identify relevant cues related to the current interests for querying the local repository, good chances are that relevant pages associated to different stories will be retrieved, with adversely effects on the overall estimated precision.

Since some of the considered approaches make a sort of inference based on the acquired evidence from previously visited sessions, both in terms of text content and page structure, it is worth analyzing the performances of the approaches as more data is made available. As the amount of visited pages increase, the quality of the predictions should increase as well.

The diagrams in Fig. 13 show the Mean Average Precision (MAP) for a certain number of browsed sessions. While the approaches that explicitly take into account past interests and, more in general, the visited content are SHY, TDH EXP and HEM; only EXP and HEM alone are able to exploit that

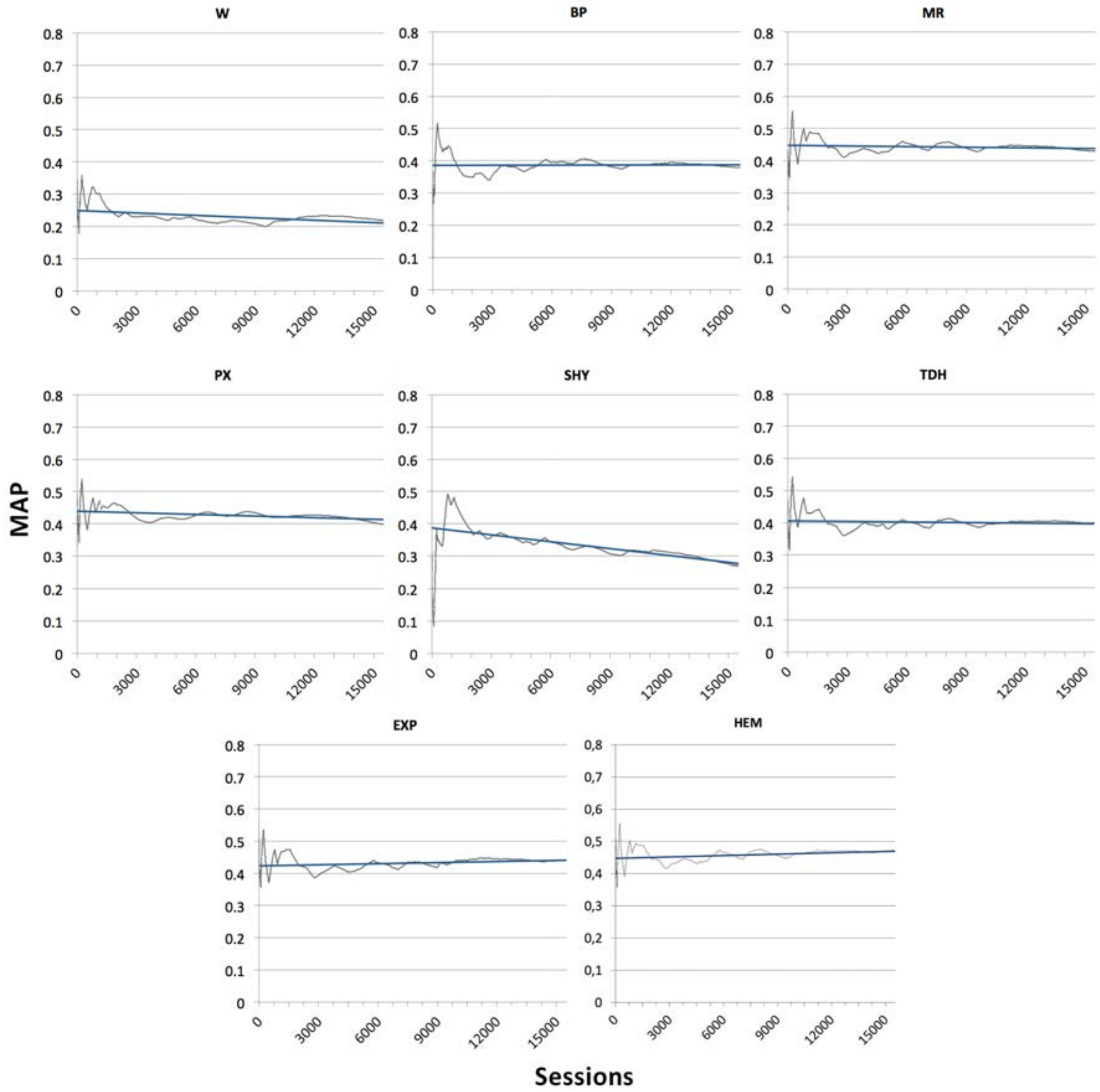

Fig. 13 Mean Average Precision measurements over the analyzed sessions.

evidence improving the performances over time. In other words, the unsupervised learning paradigm is able to infer significant features in the visited pages to improve the accuracy of the profiling.

All the remaining approaches exhibit low and stationary average precision values, or they are subjected to reduction, such as in the SHY case.

9.2 Field-based evaluation results

Since searchers typically exhibit limited interaction with search results, it is important to ensure that most of the documents they interact with are relevant. At any point in the ranking we want the current item to be more relevant than all items lower in the ranking. So the widely used measure of

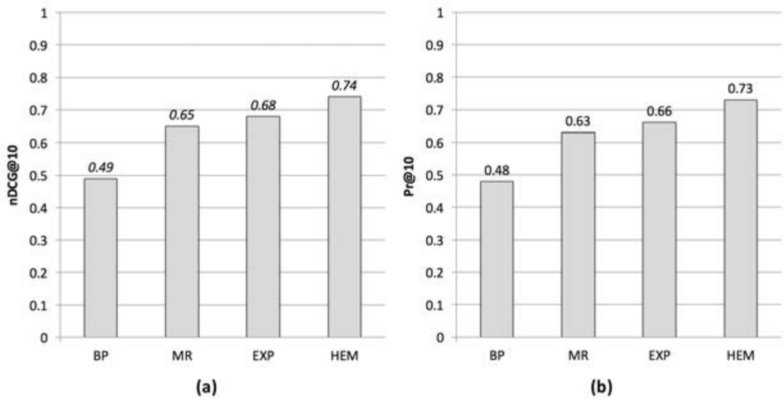

Fig. 14 nDCG (a) and precision (b) outcomes.

Normalized Discounted Cumulative Gain (nDCG) [37] has been considered. It well suits situations of non-binary relevance expressed by users and it involves a discount function over the rank while many other measures uniformly weight all positions.

It is formalized as follows:

$$nDCG@N_{cut} = \frac{1}{IDCG_{N_{cut}}} \sum_{i=1}^{N_{cut}} \frac{2^{rel(d_i)} - 1}{log(i+1)} \quad (22)$$

where N_{cut} is the cut-off value, $IDCG_{N_{cut}}$ is the ideal DCG value used for normalization, i is the ranking position of the document being evaluated, d_i is the document at position i and $rel(d_i)$ is the degree of relevancy of d_i. $nDCG$ values close to 1 prove that the system is able to pull the most relevant documents on top.

The nDCG values reported at the retrieval depth $N_{cut} = 10$ are shown in Figure 14. Clearly, similar gaps between the three considered approaches obtained by the corpus-based evaluation methodology persist also on data collected from real-life scenarios, supporting the validity of the previous tests.

To allow for a direct comparison with the corpus-based evaluation setup, the precision $P@10$ has also been computed. In particular, a true positive corresponds to a positive feedback by the user, represented by *high-relevant* or *partially relevant*. Since the overall number of interesting results for each user in the web search engine's index is not realistically computable, the recall values are omitted.

Table 8 Summary of the outcomes.

	Corpus-based						*Field-based*	
	Pr@10	Pr@20	Pr@30	Re@10	Re@20	Re@30	nDCG@10	Pr@10
W	0.22	0.22	0.21	0.08	0.15	0.22	-	-
BP	0.41	0.40	0.38	0.14	0.27	0.39	0.49	0.48
EXP	0.45	0.43	0.41	0.16	0.30	0.42	0.68	0.66
HEM	0.49	0.47	0.46	0.17	0.33	0.47	0.74	0.73
MR	0.44	0.43	0.41	0.15	0.29	0.42	0.65	0.63
PX	0.41	0.39	0.38	0.14	0.27	0.39	-	-
SHY	0.25	0.24	0.24	0.09	0.17	0.24	-	-
TDH	0.40	0.38	0.37	0.13	0.26	0.37	-	-

The relative difference between the precision values between the considered approaches is comparable with the values obtained in the corpus-based evaluation (see Fig. 12). Since the size of the collection of documents on which the personalized retrieval is performed has orders of magnitude more than the dataset of the corpus-based evaluation (see Tab. 9), it seems counterintuitive. But, as has been said, even if several news pages in the corpus-based dataset are similar, they belong to different stories. The overall performances in terms of precision and recall are negatively affected. This phenomenon does not occur in the field-based experiment.

9.3 Discussion

By way of a summary, Table 8 reports the outcomes of both corpus-based (Sect. 9.1) and field-based (Sect. 9.2) experiments. Since the best performances are obtained by the MR and EXP approaches, it is possible to say that:

- Titles, metadata keywords, titles and noun phrases extracted from the first paragraphs of the pages are a good approximation of the models.
- Statistical correlations between text regions of visited pages can be exploited to identify the most relevant elements of the future browsing sessions.

By combining the content-based extraction of the MR approach with the analysis of the statistical correlations extracted by tree-based representations of the visited pages implemented in EXP, significant improvements of the accuracy (approximately 11%) are obtained. In particular, HEM combines the two approaches in such a way that:

- Whenever relevant statistical correlations about the affinity of pairs of text regions are missing, the MR content-based approach kicks in. Since the EXP approach can be cast to a traditional

Table 9 Statistics about the two considered datasets.

	Corpus-based	Field-based
Number of sessions	15,516	1,893
Avg session length	2	2.42
Size of test collection	422,937	$> 13 \cdot 10^9$

unsupervised learning task, the prediction is considered only if its estimation is based on a significant number of samples. Content-based approaches that operate on the current browsed pages do not depend on the amount of collected samples.

Both SHY and TDH do not provide comparable results. In particular, the outcomes of TDH seem counterintuitive because it selectively chooses elements extracted from visited pages. Titles and anchors from the current session probe its local index looking for keywords occurring in spatial vicinity and, therefore, less relevant content should be ignored. One hypothesis is that, pages stored without any filtering technique introduce noise that negatively affects the co-occurrence based selection of additional keywords, thereby gaining outcomes no better than BP, which does not take into consideration past sessions.

In different ways, in order to represent the current interest model, both SHY and TDH go beyond the current sessions and extend the extraction of relevant information to past browsing activities. As already demonstrated [8], incremental profiles based on user activities spanning long periods (long-term profiles) are not very good at determining short-term interests. Besides, incremental profiles normally require numerous examples of relevant information before it can generate valid representations of information needs [95], an event that does not often turn out that way for ephemeral preferences.

Thus, we can claim that:

– Short-term interests can be better represented by algorithms that overcome less relevant information content from currently browsed resources instead of considering concepts extracted from several browsed sessions.

A further comment is about an empirical investigation of the instances where the EXP approach fails to identify relevant cues. As a matter of fact, the HTML parser[2] used to build and represent DOM models of browsed pages often fail to correctly handle JavaScript, malformed code or recent

[2] `http://htmlparser.sourceforge.net` (Last visited on 15 August 2016)

HTML versions. It negatively impacts the prediction accuracy in two ways: misleading correlations between regions are stored in the knowledge base, or relevant ones are being ignored. Both imply that text content of relevant blocks fails to be retrieved. Of courses, modern browsers implement robust layout engines that parse HTML into a DOM, such as WebKit [88] and Gecko [65]. Approaches based on DOM-based representations may take advantage of these tools.

With regard to a comparison of the two kinds of evaluation, Table 9 summarizes the principal statistics. The interest model obtained in the field-based experiments is exploited for querying a web search engine, for this reason the test collection has orders of magnitude more than the dataset of the corpus-based evaluation. The estimate is provided by `WorldWideWebSize.com` website [18], by applying the approach detailed in [82].

Besides the test collection, the average length of the considered sessions does not substantially differ between the corpus-based dataset (i.e., fixed at 2-page per session) and field-based (2.42 clicks on average). The principal difference between the datasets is related to the considered topics. In one case, it was limited to online newspapers and recent hyperlink paths the take the visitors to published news pages. In the field-based experiments, the users were asked to select their browsing sessions motivated by informational intents. And it has already been mentioned that their seeking activities span different intents in addition to content on news pages (Table 5), so that the outcomes of the field-based evaluation look more comprehensive.

One more way to analyze the difference between the two datasets is through the EXP's knowledge bases built during the two experiments. Figure 15(a) shows how many pages are stored in the clusters at the end of the exploration. Around 50% of the clusters contain more than two pages in both of the datasets. On average, the corpus-based dataset bears an average of 6.59 pages per cluster, and 6.10 in the other case. Even if the the corpus-based dataset is much larger w.r.t. the field-based one, the former has been collected by considering a wider number of different sources (or hostnames), and the differences between the two is almost irrelevant.

Looking at how the execution is affected, Figure 15(b) proves the expected increment of the ratio between created clusters and browsed sessions as new sources are being analyzed. At the end of the exploration, the ratio is 0.28 and 0.16 for the field-based and corpus-based, respectively. The difference is justified also by the different number of sessions in the two evaluations, that is, 1,893 and

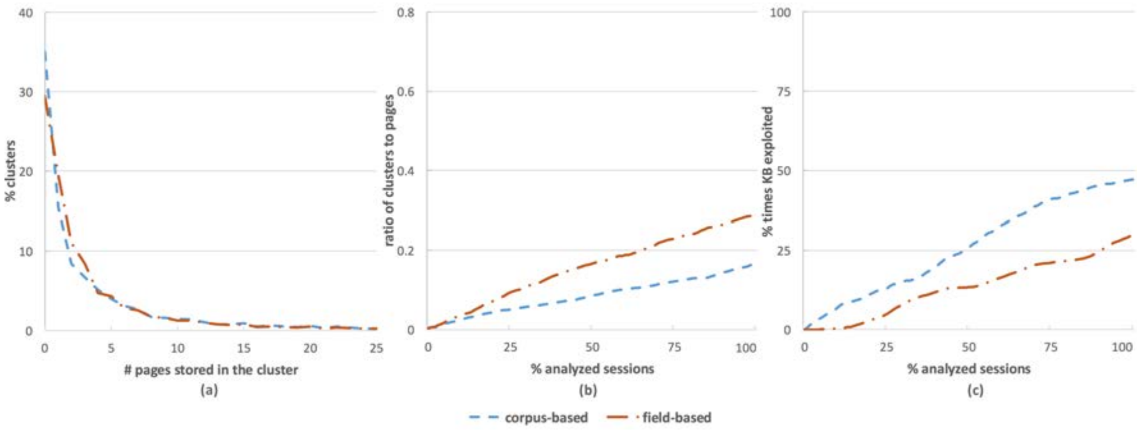

Fig. 15 Statistics related to the two datasets considered in the evaluation.

15,516, respectively. Since the corpus-based is focused on pages from a specific domain (i.e., online newspapers), slightly more chances exist to see similar templates between visited pages and websites.

One more interesting result is about the number of times the EXP approach has taken advantage of the statistics in the KBs during the construction of the interest model. 47.19% of the visited pages in the corpus-based dataset exploited the KBs to obtain a relevant boosting factor (see Eq. 16), and weight the text retrieved from significant regions accordingly. The percentage reaches 30.05% in the case of the field-based experiments. The different size of the datasets is still the main reason of this variance. The ever-increasing ratio between the use of KBs and the visited sessions makes sense inasmuch as the unsupervised learning benefits from the statistical evidence collected during the browsed sessions.

Whenever the KBs do not provide any relevant statistics, the EXP approach falls back to the extraction based solely on the semantic similarity between pages' regions discussed in Sect. 5.4. In this event, the boosting factor gets value 1 and, therefore, the content avoids to get weighted by the missing evidence from past activity.

10 Conclusions and future work

The extraction of current interests from browsing histories is a complex task that calls for elaborated analyses. The approaches such as the one being discussed here, lie on the evidence acquired by analyzing visited pages and the organization of hypertext contents for identifying relevant correlations among text regions.

An extended comparative evaluation proves the effectiveness both on a corpus composed of informational content and in a field-based evaluation involving humans in every-day tasks.

Significant observations can be summarized as follows:

- Browsing sessions have the chance to contain relevant information that can be exploited for better representing current user interests.
- Noise in the form of advertisements, navigation bars, links to other content, etc.; and pages dealing with multiple topics overshadow the benefits of extraction approaches based on the whole page content.
- Whereas past browsing activities might contain relevant information w.r.t. the current interests, more advanced techniques are required to automatically isolate it for any further analysis.
- DOM and template analysis on visited pages enables the identification of relationships between text regions that can be exploited for filtering out less relevant content. As this knowledge builds up, its statistical analysis improves the accuracy of the extraction of current interests.

These observations open up an interesting research pathway to future strategies able to combine multiple evidence. Whereas most of the extraction approaches are based on information retrieval techniques based on natural language processing on text content, the proposed strategy exploits structural knowledge acquired in the course of browsing. In the absence of this kind of knowledge, the extraction may instead rely on text features, such as metadata keywords, titles, link anchors and noun phrases extracted from the very start of the last visited pages, which proved to be good approximation of current interests.

In the near future, we hope to extend this approach to embody content and signals extracted from social networks, where new forms of interactions and correlations between content play an important role in the identification of user needs.

As for old browsing sessions, co-occurrence or semantic similarity-based inferences w.r.t. current activities are often inadequate for highlighting the most related visited content and representing current interests. By deploying the proposed approach over past sessions for obtaining and combining additional information to enrich the present model, chances are to improve the extraction accuracy over already established approaches (e.g., [76, 79]). However, it is necessary to define more completely the complex process of information consumption, that is, gathering, organizing and analyzing informational units in a particular context or use environment in order to build selective personalized systems able to deliver the information needed at the time the user's need was to be met. This issue cannot be addressed without a proper combination of long-term and short-term modeling of interests and explicit representations of higher layers dealing with the information-seeking strategies and plans users undertake when a particular task ought to be accomplished. Estimating short-term interests is a required step toward the development of this comprehensive modeling approach.

As a brief comment on privacy issues, users may be uncomfortable with having personal information broadcast across the Internet to search engines, other services or uncertain destinations [57]. The analysis of the visited pages required for building the knowledge base of the proposed approach can be operated on the client side, guaranteeing that user information will not be submitted to a remote server. Interests models can be communicated to the server by the explicit consent of users who are keen to have the human-computer interaction personalized.

References

1. Julián Alarte, David Insa, Josep Silva, and Salvador Tamarit. Temex: The web template extractor. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 155–158, New York, NY, USA, 2015. ACM.
2. Han The Anh and Luís Moniz Pereira. State-of-the-art of intention recognition and its use in decision making. *AI Commun.*, 26(2):237–246, 2013.
3. Giuseppe Attardi, Antonio Gullí, and Fabrizio 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.
4. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England, 2011.
5. Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 136–145, London, UK, UK, 2002. Springer-Verlag.
6. Marcia J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–431, 1989.
7. Thomas Beauvisage. Computer usage in daily life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 575–584, New York, NY, USA, 2009. ACM.
8. Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the impact of short- and long-term behavior on search personalization. In *Proceedings of the*

35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, pages 185–194, New York, NY, USA, 2012. ACM.

9. Mikhail Bilenko and Ryen W. White. Mining the search trails of surfing crowds: Identifying relevant websites from user activity. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 51–60, New York, NY, USA, 2008. ACM.
10. Daniel Billsus and MichaelJ. Pazzani. Adaptive news access. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 550–570. Springer Berlin Heidelberg, 2007.
11. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.
12. Andrei Broder. A taxonomy of web search. *SIGIR FORUM*, 36(2):3–10, 2002.
13. Cyril Cleverdon. The cranfield tests on index language devices. In Karen Sparck Jones and Peter Willett, editors, *Readings in Information Retrieval*, pages 47–59. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
14. Andy Cockburn and Bruce McKenzie. What do web users do? an empirical analysis of web use. *Int. J. Hum.-Comput. Stud.*, 54(6):903–922, June 2001.
15. World Wide Web Consortium. Tidy. Last visited on 15 August 2016.
16. Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
17. Mariam Daoud, Lynda Tamine-Lechani, Mohand Boughanem, and Bilal Chebaro. A session based personalized search using an ontological user profile. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, SAC '09, pages 1732–1736, New York, NY, USA, 2009. ACM.
18. Maurice de Kunder. Worldwidewebsize - the size of the world wide web (the internet). Last visited on 15 August 2016.
19. Chen Ding and Jagdish C. Patra. User modeling for personalized web search with self-organizing map. *Journal of the American Society for Information Science and Technology*, 58(4):494–507, 2007.
20. Christiane Fellbaum. *WordNet: An Electronic Lexical Database.* Bradford Books, 1998.
21. Emilio Ferrara, Pasquale De Meo, Giacomo Fiumara, and Robert Baumgartner. Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, 70:301 – 323, 2014.
22. The Apache Software Foundation. Apache lucene. Last visited on 15 August 2016.
23. Sarah Gallacher, Eliza Papadopoulou, Nick K. Taylor, and M. Howard Williams. Learning user preferences for adaptive pervasive environments: An incremental and temporal approach. *ACM Trans. Auton. Adapt. Syst.*, 8(1):5:1–5:26, April 2013.
24. Fabio Gasparetti and Alessandro Micarelli. Exploiting web browsing histories to identify user needs. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, pages 325–328, New York, NY, USA, 2007. ACM Press.

25. M. Rami Ghorab, Dong Zhou, Alexander O'connor, and Vincent Wade. Personalised information retrieval: Survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443, September 2013.
26. M.Rami Ghorab, Dong Zhou, Alexander OConnor, and Vincent Wade. Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443, 2013.
27. David Gibson, Kunal Punera, and Andrew Tomkins. The volume and evolution of web page templates. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, WWW '05, pages 830–839, New York, NY, USA, 2005. ACM.
28. 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 the 11th International Conference on World Wide Web*, WWW '02, pages 562–569, New York, NY, USA, 2002. ACM.
29. Google. Google books ngram viewer. Last visited on 15 August 2016.
30. Google. Google news. Last visited on 15 August 2016.
31. Thomas Gottron. Clustering template based web documents. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and RyenW. White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 40–51. Springer Berlin Heidelberg, 2008.
32. W3C DOM Working Group. Document object model (dom). Last visited on 15 August 2016.
33. Ramanathan Guha, Vineet Gupta, Vivek Raghunathan, and Ramakrishnan Srikant. User modeling for a personal assistant. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 275–284, New York, NY, USA, 2015. ACM.
34. Katja Hofmann, Shimon Whiteson, Anne Schuth, and Maarten de Rijke. Learning to rank for information retrieval from user interactions. *SIGWEB Newsl.*, 5(Spring):5–7, April 2014.
35. Wen Hua, Yangqiu Song, Haixun Wang, and Xiaofang Zhou. Identifying users' topical tasks in web search. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 93–102, New York, NY, USA, 2013. ACM.
36. Bernard J. Jansen, Amanda Spink, Chris Blakely, and Sherry Koshman. Defining a session on web search engines: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 58(6):862–871, April 2007.
37. Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.
38. Daxin Jiang, Jian Pei, and Hang Li. Mining search and browse logs for web search: A survey. *ACM Trans. Intell. Syst. Technol.*, 4(4):57:1–57:37, October 2013.
39. Xiaoran Jin, Marc Sloan, and Jun Wang. Interactive exploratory search for multi page search results. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 655–666, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.
40. K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments. *Inf. Process. Manage.*, 36(6):779–808, November 2000.
41. Melanie Kellar, Carolyn Watters, and Michael Shepherd. A Goal-based Classification of Web Information Tasks. *Proceedings of the American Society for Information Science and Technology*, 43(1):1–22, 2006.

42. Philipp Koehn. *Statistical Machine Translation.* Cambridge University Press, New York, NY, USA, 1st edition, 2010.

43. Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 441–450, New York, NY, USA, 2010. ACM.

44. Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284, 1998.

45. Language and Information Processing Research Group @ University of Memphis. Semilar: A semantic similarity toolkit. Last visited on 15 August 2016.

46. Mihai C. Lintean, Cristian Moldovan, Vasile Rus, and Danielle S. McNamara. The role of local and global weighting in assessing the semantic similarity of texts using latent semantic analysis. In Hans W. Guesgen and R. Charles Murray, editors, *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference, May 19-21, 2010, Daytona Beach, Florida.* AAAI Press, 2010.

47. Bing Liu, Robert Grossman, and Yanhong Zhai. Mining data records in web pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 601–606, New York, NY, USA, 2003. ACM.

48. Yiqun Liu, Junwei Miao, Min Zhang, Shaoping Ma, and Liyun Ru. How do users describe their information need: Query recommendation based on snippet click model. *Expert Syst. Appl.*, 38(11):13847–13856, 2011.

49. Takuya Maekawa, Yutaka Yanagisawa, Yasushi Sakurai, Yasue Kishino, Koji Kamei, and Takeshi Okadome. Context-aware web search in ubiquitous sensor environments. *ACM Trans. Internet Technol.*, 11(3):12:1–12:23, February 2012.

50. Nicolaas Matthijs and Filip Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 25–34, New York, NY, USA, 2011. ACM.

51. B. McKenzie and A. Cockburn. An empirical analysis of web page revisitation. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 5 - Volume 5*, HICSS '01, page 5019, Washington, DC, USA, 2001. IEEE Computer Society.

52. Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. Personalized search on the world wide web. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 195–230. Springer Berlin, Heidelberg, Berlin, Heidelberg, and New York, 2007.

53. Microsoft. Bing. Last visited on 15 August 2016.

54. Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press, 2006.

55. Bree Nordenson. Overload! *Columbia Journalism Review*, 47(4):30–42, 2008.

56. Vicki L. O'Day and Robin Jeffries. Orienteering in an information landscape: How information seekers get from here to there. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 438–445, New York, NY, USA, 1993. ACM.
57. Saurabh Panjwani, Nisheeth Shrivastava, Saurabh Shukla, and Sharad Jaiswal. Understanding the privacy-personalization dilemma for web search: A user perspective. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3427–3430, New York, NY, USA, 2013. ACM.
58. George Papadakis, Ricardo Kawase, Eelco Herder, and Wolfgang Nejdl. Methods for web revisitation prediction: survey and experimentation. *User Modeling and User-Adapted Interaction*, pages 1–39, 2015.
59. Eli Pariser. *The Filter Bubble: What the Internet Is Hiding from You.* Penguin Group , The, 2011.
60. Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 91–100, New York, NY, USA, 2008. ACM.
61. P. Pirolli and Stuart K. Card. Information foraging. *Psychological Review*, 106(4):643–675, 1999.
62. Peter Pirolli and Stuart Card. Information foraging in information access environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 51–58, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
63. Peter L. T. Pirolli. *Information Foraging Theory: Adaptive Interaction with Information.* Oxford University Press, Inc., New York, NY, USA, 1 edition, 2007.
64. James Pitkow, Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. Personalized search. *Commun. ACM*, 45(9):50–55, September 2002.
65. Mozilla Project. Gecko. Last visited on 15 August 2016.
66. Mandar Rahurkar and Silviu Cucerzan. Predicting when browsing context is relevant to search. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 841–842, New York, NY, USA, 2008. ACM.
67. D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 502–511, New York, NY, USA, 2004. ACM.
68. Xiang Ren, Yujing Wang, Xiao Yu, Jun Yan, Zheng Chen, and Jiawei Han. Heterogeneous graph-based intent learning with queries, web pages and wikipedia concepts. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 23–32, New York, NY, USA, 2014. ACM.
69. B. J. Rhodes and P. Maes. Just-in-time information retrieval agents. *IBM Syst. J.*, 39(3-4):685–704, July 2000.
70. Josh J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice-Hall Inc., Englewood Cliffs, NJ, USA, 1971.
71. Vasile Rus and Arthur C. Graesser. Deeper natural language processing for evaluating student answers in intelligent tutoring systems. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence*

and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA, pages 1495–1500. AAAI Press, 2006.

72. Barry Smyth and Evelyn Balfe. Anonymous personalization in collaborative web search. *Inf. Retr.*, 9(2):165–190, March 2006.
73. Mirco Speretta. Personalized search based on user search histories. In *In Proc. of International Conference of Knowledge Management(CIKM), Washington D.C., 2004*, pages 622–628, 2005.
74. Smitha Sriram, Xuehua Shen, and Chengxiang Zhai. A session-based search engine. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 492–493, New York, NY, USA, 2004. ACM.
75. Sofia Stamou and Alexandros Ntoulas. Search personalization through query and page topical analysis. *User Modeling and User-Adapted Interaction*, 19(1-2):5–33, February 2009.
76. Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of WWW'04*, pages 685–684, New York, USA, May 17–22 2004.
77. Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 675–684, New York, NY, USA, 2004. ACM.
78. Linda Tauscher and Saul Greenberg. How people revisit web pages: Empirical findings and implications for the design of history systems. *Int. J. Hum.-Comput. Stud.*, 47(1):97–137, July 1997.
79. Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM Press.
80. Yury Ustinovskiy and Pavel Serdyukov. Personalization of web-search using short-term browsing context. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 1979–1988, New York, NY, USA, 2013. ACM.
81. Hervé Utard and Johannes Fürnkranz. Link-local features for hypertext classification. In Markus Ackermann, Bettina Berendt, Marko Grobelnik, Andreas Hotho, Dunja Mladeni, Giovanni Semeraro, Myra Spiliopoulou, Gerd Stumme, Vojtch Svtek, and Maarten van Someren, editors, *Semantics, Web and Mining*, volume 4289 of *Lecture Notes in Computer Science*, pages 51–64. Springer Berlin Heidelberg, 2006.
82. Antal van den Bosch, Toine Bogers, and Maurice de Kunder. Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics*, 107(2):839–856, 2016.
83. Eduardo Vicente-Lpez, LuisM. de Campos, JuanM. Fernndez-Luna, JuanF. Huete, Antonio Tagua-Jimnez, and Carmen Tur-Vigil. An automatic methodology to evaluate personalized information retrieval systems. *User Modeling and User-Adapted Interaction*, pages 1–37, 2014.
84. Eduardo Vicente-Lpez, LuisM. de Campos, JuanM. Fernndez-Luna, JuanF. Huete, Antonio Tagua-Jimnez, and Carmen Tur-Vigil. An automatic methodology to evaluate personalized information retrieval systems. *User Modeling and User-Adapted Interaction*, 25(1):1–37, 2015.

85. Karane Vieira, André Luiz da Costa Carvalho, Klessius Berlt, Edleno S. de Moura, Altigran S. da Silva, and Juliana Freire. On finding templates on web collections. *World Wide Web*, 12(2):171–211, 2009.

86. Karane Vieira, Altigran S. da Silva, Nick Pinto, Edleno S. de Moura, João M. B. Cavalcanti, and Juliana Freire. A fast and robust method for web page template detection and removal. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 258–267, New York, NY, USA, 2006. ACM.

87. Hongning Wang, ChengXiang Zhai, Feng Liang, Anlei Dong, and Yi Chang. User modeling in search logs via a nonparametric bayesian approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 203–212, New York, NY, USA, 2014. ACM.

88. Webkit. Webkit - open source web browser engine. Last visited on 15 August 2016.

89. Ryen W. White, Peter Bailey, and Liwei Chen. Predicting user interests from contextual information. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 363–370, New York, NY, USA, 2009. ACM.

90. Ryen W. White, Paul N. Bennett, and Susan T. Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1009–1018, New York, NY, USA, 2010. ACM.

91. Ryen W. White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1411–1420, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

92. Ryen W. White and Steven M. Drucker. Investigating behavioral variability in web search. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 21–30, New York, NY, USA, 2007. ACM.

93. Ryen W. White and Jeff Huang. Assessing the scenic route: Measuring the value of search trails in web logs. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 587–594, New York, NY, USA, 2010. ACM.

94. Ryen W. White, Joemon M. Jose, and Ian Ruthven. An approach for implicitly detecting information needs. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 504–507, New York, NY, USA, 2003. ACM.

95. Ryen W. White and Diane Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 297–306, New York, NY, USA, 2006. ACM.

96. Ryen W. White, Ian Ruthven, Joemon M. Jose, and C. J. Van Rijsbergen. Evaluating implicit feedback models using searcher simulations. *ACM Trans. Inf. Syst.*, 23(3):325–361, July 2005.

97. Steve Whittaker. Personal information management: From information consumption to curation. *ARIST*, 45(1):1–62, 2011.

98. Mingfang Wu, David Hawking, Andrew Turpin, and Falk Scholer. Using anchor text for homepage and topic distillation search tasks. *Journal of the American Society for Information Science and Technology*, 63(6):1235–1255, 2012.

99. Yiming Yang. An evaluation of statistical approaches to text categorization. *Inf. Retr.*, 1(1-2):69–90, May 1999.

100. Zhijun Yin, Milad Shokouhi, and Nick Craswell. Query expansion using external evidence. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 362–374, Berlin, Heidelberg, 2009. Springer-Verlag.