User Profiling for Interest-focused Browsing History

Miha Grčar, Dunja Mladenič, Marko Grobelnik

Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{Miha.Grcar, Dunja.Mladenic, Marko.Grobelnik}@ijs.si
http://kt.ijs.si

Abstract. User profiling is an important part of the Semantic Web as it integrates the user into the concept of Web data with machine-readable semantics. In this paper, user profiling is presented as a way of providing the user with his/her interest-focused browsing history. We present a system that is incorporated into the Internet Explorer and maintains a dynamic user profile in a form of automatically constructed topic ontology. A subset of previously visited Web pages is associated with each topic in the ontology. By selecting a topic, the user can view the set of associated pages and choose to navigate to the page of his/her interest. Each topic can be seen as an interest of the user (hence the term *interest-focused* browsing history). The ontology is constructed by transforming the textual contents of the pages into sparse word-vectors and applying bisecting k-means clustering (i.e. a form of hierarchical clustering) on the set of sparse vectors. The most recently visited pages are used to identify the user's current interest and map it to the ontology. The user can clearly see which topics, and their corresponding pages, are related (or are not related, for that matter) to his/her current interest. We see this as a useful way of organizing the user's browsing history. To illustrate the functioning of the system, we demonstrate its behavior in one particular real-life scenario.

1 Introduction

In this paper, user profiling is presented as a way of providing the user with his/her interest-focused browsing history. We present a system that is incorporated into the Internet Explorer and maintains a dynamic user profile in a form of automatically constructed topic ontology.

Let us begin by briefly summarizing some of the related work in the field of user profile construction. The most related work is that of (KIM AND CHAN, 2003). They propose a tree-like hierarchy of interests, the root being the user's general interest (i.e. long-term interest) and leaves representing domains the user is – was ever – interested in (i.e. short-term interests). User interest hierarchies are built using a form of hierarchical clustering on a set of Web pages visited by a user.

Another less related way of constructing a user profile is to analyze the user's browsing history and apply modified collaborative filtering techniques (SUGIYAMA ET AL., 2004). Here, the user profile is also a combination of both (i) user's persistent preferences (long-term preferences) and (ii) user's ephemeral preferences (short-term preferences – "today's" preferences) and is represented as a vector of term weights. Modified collaborative filtering is then applied to a user-term matrix (in contrast to being applied to a user-item matrix as is the case with the original collaborative filtering approach – hence the word "modified") to predict the missing term weights

in each user profile. Clustering is used (in one of their approaches) to determine user communities. Cluster centroids are compared to the active user's term vector to find the user's neighborhood (a threshold is used to discard less relevant communities). The latter approach, according to (SUGIYAMA ET AL., 2004), achieves the best results.

In Foxtrot recommender system (MIDDLETON ET AL., 2003), an ontology (taxonomy) based on CORA digital library is used – new documents are classified into the taxonomy by using a variant of the nearest neighbor algorithm. A user profile holds a set of topics and their corresponding interest values. Each topic adds 50% of its interest value to its super-class. They also used "static knowledge" ontologies to alleviate the cold-start problem. Visualization of profiles is used to encourage immediate users' feedbacks. For evaluation, collaborative filtering is performed on a user-topic matrix (they term this technique "collaborative and content-based recommendations").

The rest of the paper is arranged as follows. In Section 2, user profiling is viewed from the perspective of the Semantic Web. Architecture of our system is presented in Section 3. In Section 4, we demonstrate the functioning of the system in a real-life scenario. The paper concludes with the discussion and some ideas for future work in Section 5.

2 User Profiling from the Perspective of the Semantic Web

When thinking of the Semantic Web we can say that the Semantic Web is a Web focused on the exchange of information between computers that does not explicitly involve human users. Although computers could be quite busy communicating to each other, there still needs to be some space left for human users in the whole process – there is where user profiling comes into the play.

Technically speaking, the Semantic Web is mainly about the data that are self-explanatory, or in other words, about the data which are annotated in some standard fashion that enables efficient computer-to-computer communication. The main purpose of the Semantic Web is to enable better services for the end-users. Since in general the data can be understood in more than one way – especially when talking about the more abstract categories which cannot be annotated explicitly – one of the possible sources of annotations (i.e. meta-data) may also be the information about the user. This information can be represented in several ways. Typically, if we talk about more abstract and aggregated information, we talk about *user profiles* or *user models*. Their main characteristic is the ability to generalize the collected data about the user's behavior (such as click-stream data of the user's browsing behavior). Such *user-models* are then used to annotate the data in such a way that Web services are able to deliver personalized information, aiming at increasing the user's efficiency when he/she is communicating with the computer.

We can conclude this short description of user profiling from the perspective of the Semantic Web by saying that user profiling is an important source of meta-data on the user's understanding of the data semantics. In particular, this compensates for the differences in users' understanding of the data by using an alternative annotation, which is more of the *soft* nature (the *softness* comes from the fact that the data are

annotated implicitly and dynamically by taking a user profile into the account). The main goal of user modeling is increasing the efficiency of user activities by delivering more personalized information.

3 Architecture of the System

The system provides a dynamic user profile in a form of topic ontology. After a page is viewed, the textual content is extracted and stored as a text file as described in Section 3.1. Pages are represented as word-vectors (also termed *bags of words*) as explained in Section 3.2. To construct the topic ontology, we perform a variant of hierarchical clustering (see Section 3.3). By using the cosine similarity measure, we are able to map the user's current interest to the topic ontology (more details in Section 3.4). The latter identifies the ontology nodes that are in the context of the user's current interest. The whole process is illustrated in the system architecture figure (Figure 1) which also includes the references to Sections 3.1 through 3.4. These sections contain a detailed description of individual phases of the process.

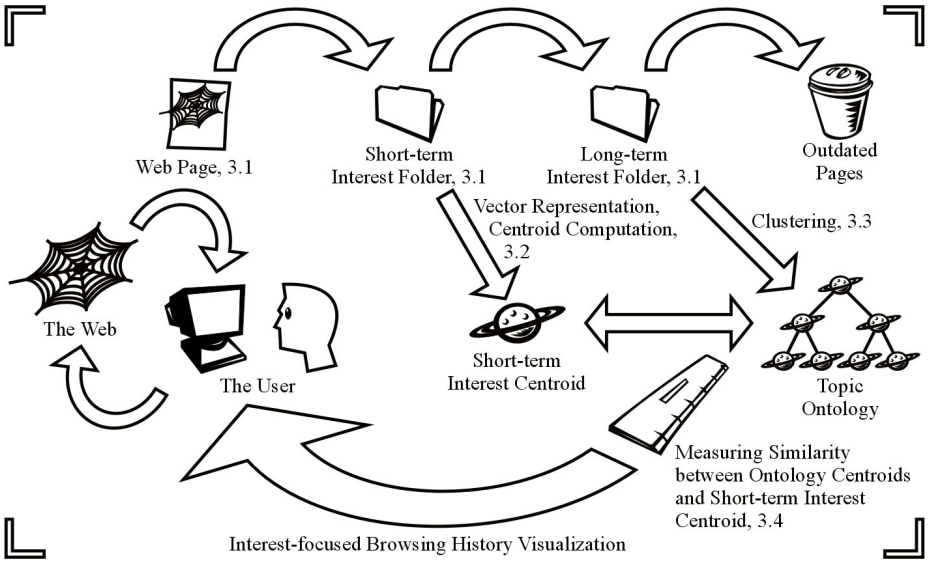

Figure 1. The Interest-focused Browsing History Architecture. The process is described throughout Section 3. The corresponding subsections are noted in the figure (3.1, 3.2, 3.3, 3.4).

3.1 Handling a Page-view

After a page is viewed, the textual content is extracted and stored as a text file. The text extraction is done in two relatively simple steps:

(i) text segments between and including "<script>" and "</script>" or "<style>" and "</style>" are discarded,
(ii) substrings starting with "<" and ending with ">" are removed.

A collection of such text files (from now on simply termed *pages*) is maintained in two folders. The first folder holds *m* most recently viewed pages (*the short-term interest folder*). In our experiments, *m* is set to 5. The second folder contains the last *n* viewed pages, where $n > m$ (*the long-term interest folder*). In our experiments, *n* is set to 300. When a page is first visited, it is placed into both folders. Eventually it gets pushed out by other pages that are viewed afterwards. A page stays in the long-term interest folder much longer than in the short-term interest folder (hence the terms *long-* and *short-term*), the reason for this being a much higher number of new pages that need to be viewed for the page to be pushed out of the long-term interest folder.

Pages are named after their 128-bit MD5 hash codes. In this way we are able to, at least to some extent, detect a page that was already visited and handle this scenario. Currently we simply update the timestamp of the file (i.e. the page) to mark it as *recently interesting*. This action is carried out in both folders.

3.2 Word-vector Representation of a Page

To build a user profile, we first take the pages from the short-term interest folder and compute their TFIDF vector representations of the textual content, ignoring the order of words (thus such vectors are also termed *bags of words*, see Figure 2 for illustration), as introduced in (SALTON AND BUCKLEY, 1987). Each vector component is calculated as the product of Term Frequency (*TF*) – the number of times a word *W* occurs in the page – and Inverse Document Frequency (*IDF*), as explained by the following equation:

$$d^{(i)} = TF(W_i, d)IDF(W_i), \ \ where \ \ IDF(W_i) = \log \frac{D}{DF(W_i)}$$

where *D* is the number of pages and document frequency $DF(W)$ is the number of documents in which word *W* occurred at least once.

Prior to transforming pages into vectors, stop-words are removed and stemming is applied. After vectors are obtained, the centroid of short-term interest pages is computed by averaging corresponding TFIDF vectors component-by-component. This process combines the short-term interest pages, regardless of their count, into one single construct – the short-term interest centroid.

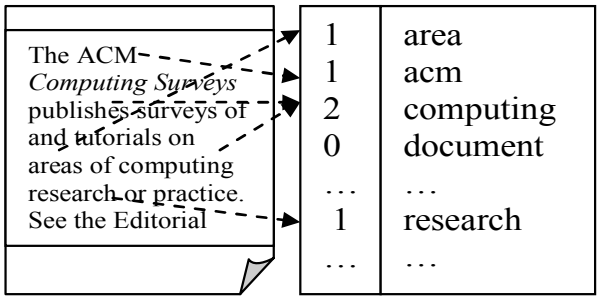

Figure 2. An illustration of a word-vector (term-frequency vector, in this particular case) representation of a document.

3.3 Constructing the Topic Ontology

The long-term interest pages are treated slightly differently from the short-term interest pages. We first perform the bisecting k-means clustering (i.e. a variant of hierarchical clustering) over the long-term interest TFIDF vectors. This clustering method is computationally efficient and was already successfully applied on text documents (STEINBACH ET AL., 2000). At start, all the pages form the root cluster which is first divided into two child clusters (hence the term *bisecting* clustering). The same procedure is repeated for each of the two newly obtained clusters and recursively further down the hierarchy. We perform the splitting until the size of the clusters (i.e. the number of pages the cluster contains) is smaller than the predefined minimum size (usually set to 10% of the initial collection size). During the clustering process, the similarity between two vectors is computed as the cosine of the angle between the two vectors.

The result of the clustering is a binary tree (in this text termed *topic ontology*), with a set of pages at each node. Later on, for each node a centroid is computed in the same way as for the short-term interest pages.

The root of the topic ontology holds the user's general interest while the leaves represent his/her specific interests. By our understanding the term *general interest* is not synonymous with *long-term interest* and in that same perspective the term *specific interest* is not a synonym for *short-term interest*. While the terms *long-term* and *short-term* (i.e. *recent* or *current*) *interest* emphasize the chronological order of page-views, this is not the case with the terms *general* (i.e. *global*) *interest* and *specific interest*. *General interest* stands for all the topics the user is – or ever was – interested in, while the term *specific interest* usually describes one more-or-less isolated topic that is – or ever was – of interest to the user.

3.4 Current Browsing Interest of the User

By using the cosine similarity measure, we are able to compare the centroid at each node to the short-term interest centroid. In other words, we are able to map the user's current interest to the topic ontology. The mapping reveals the extent to which a node (i.e. a set of pages) is related to the user's short-term interest. By highlighting nodes with the intensity proportional to the similarity score, we can clearly expose the topic ontology segments that are (or are not, for that matter) of current interest to the user.

Due to the highlighting the user can clearly see which parts of the topic ontology are relevant to his/her current interest. He/she can also access previously visited pages by selecting a node in the hierarchy which is visualized in the application window. This can be explained as the user's interest-focused Web browsing history, the interest being defined by the selected node.

4 Implementation of the System

The user profile is visualized on the Internet Explorer toolbar that we developed for this purpose. The user can select a node (i.e. his/her more or less general interest) to see its specific keywords and the associated Web pages.

4.1 Toolbar as the User Interface

Generally, an Internet Explorer (IE) toolbar is an extension of the IE's GUI, as well as an application that extends the IE with additional functions. Since it is highly integrated into the IE, a toolbar can also:
(i) receive notifications and information about the user's actions in the IE; most notably the user's requests to "navigate to" (the user's requests can be filtered or preprocessed in some other way),
(ii) access the contents of the currently loaded Web page,
(iii) apply any kind of changes to the content of the currently loaded page (e.g. highlight links to recommended pages, highlight some parts of the text, etc.),
(iv) easily access the Web as well as the local computer.

We have developed an IE toolbar to construct and visualize the user's interest-focused browsing history. The toolbar is placed into the left side of the IE's application window. It is divided into two panels, one showing the user's topic ontology and the other showing the most characteristic keywords and the set of pages corresponding to the selected node (see Figures 4 and 5). The user can select any page from the list and navigate to that page.

The user's current interests are highlighted (see screenshots in Picture 4) in the ontology visualization panel. The color intensity of the highlighting corresponds to the relevance of the node to the user's current interest. The user can thus clearly see which pages that he/she already visited are in the context of his/her current interest.

4.2 Example of the System Usage

We will demonstrate how the system works in a real-life scenario. Let us say that the user is interested in three distinct topics. He/she searches the Web for "whale tooth", "triumph tr4" and "semantic web", in this same order. After viewing several pages (55 altogether in our case), his/her topic ontology looks as shown in Figure 3.

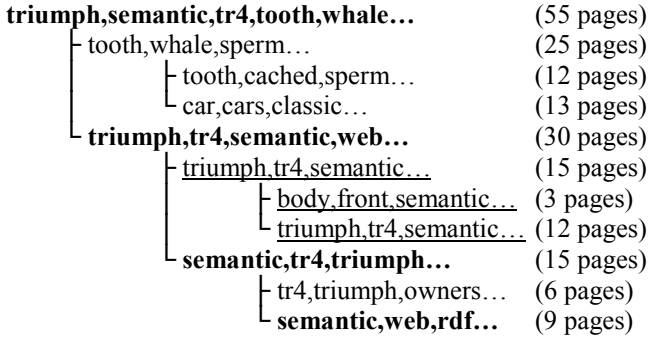

Figure 3. The topic ontology as automatically constructed after viewing 55 Web pages about "whale tooth", "triumph tr4" and "semantic web", in this same order.

Each node is named after the three keywords from the centroid vector that have the highest averaged TFIDF weights. The root node represents the user's general interest – they appear to be about *tooth*, *whale*, *triumph*, *tr4*, *semantic* and *web*, which is exactly what the user searched for. Note that the user's search-engine queries are not included in the profiling process and that these keywords were actually reconstructed from the textual contents of the pages that the user visited.

The root node is first partitioned into two clusters – one containing the pages about *whale tooth* and the other containing the pages about *triumph tr4* and *semantic web*. We can see that the partitioning is not perfect. The cluster talking about *classic cars*, for example, is contained within the *whale tooth* cluster. It would make more sense if it was included into the *triumph tr4* cluster. Furthermore, we see that the second cluster (*triumph tr4 & semantic web*) is not clearly partitioned into the *triumph tr4* cluster and the *semantic web* cluster in the next step. However, since we are using fully automated methods, we can say that the results are reasonably good.

Since *semantic web* was the user's latest interest, the nodes containing mainly pages related to this topic are highlighted (in Figure 3 bolded or underlined). We can see that highlighting works quite well in this particular example. Bolded clusters are highly relevant, underlined clusters are less relevant, and other clusters are irrelevant to the user's current interest. Two screenshots of the system's GUI are given in Figure 4 and Figure 5.

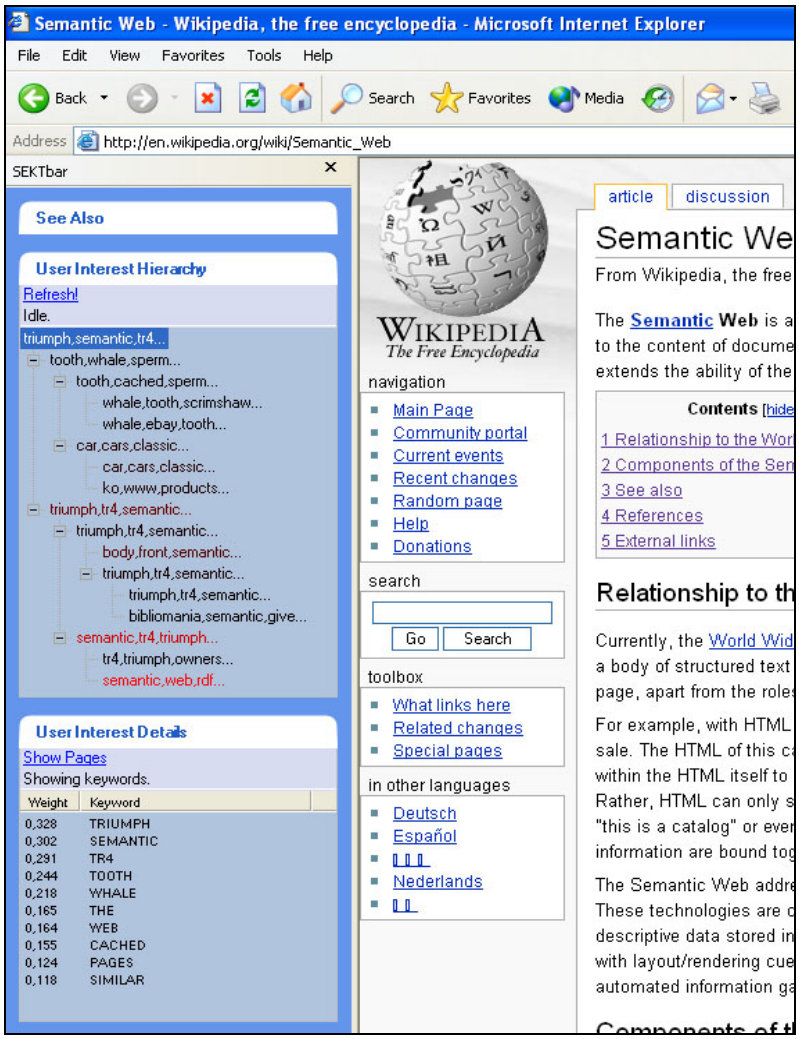

Figure 4. Screenshot of the system's GUI, captured after the user visited all the pages used for the demonstration in Section 4. Screenshot shows the topic ontology of the user's interests and the most characteristic keywords from the root cluster. The user's most recent interest is highlighted with red color (the brighter the more relevant).

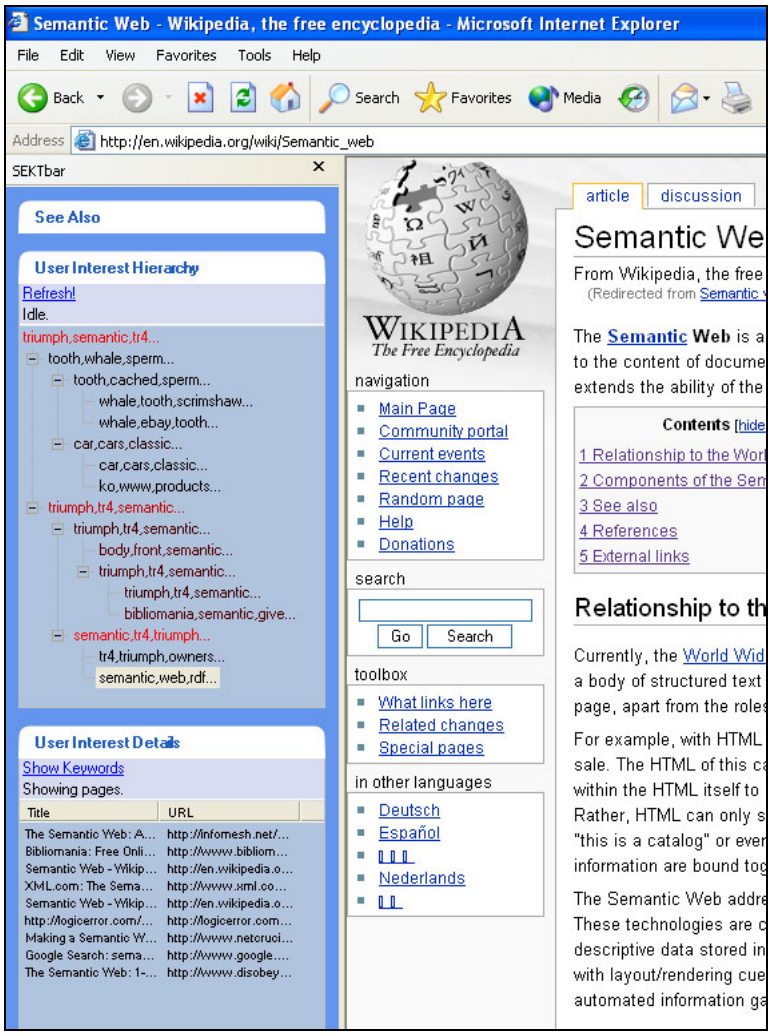

Figure 5. Screenshot of the system's GUI, captured after the user visited all the pages used for the demonstration in Section 4. Screenshot shows the topic ontology of the user's interests and the list of Web pages that corresponds to the *semantic-web-rdf* cluster. The user's most recent interest is highlighted with red color (the brighter the more relevant).

5 Discussion

Many research issues and technical details still need to be investigated. We noticed that when extracting the textual content of a Web page, a lot of interest-irrelevant text segments are also processed (e.g. standard navigation menus and adds). A simple heuristic that could be used to alleviate the problem is discarding text segments (i.e.

chunks of text between two HTML tags) that are shorter than some predefined length. This solution has not been applied yet but we are planning to try it out in near future.

Since our software resides on the client side and we are able to track the Web browser's events, we could also efficiently measure the time the user spends on a page and use this information to additionally weight pages that were viewed by the user. In this same context, the pages that were visited more than once should be weighted by the sum of their page-view durations.

Currently we treat all Web pages equally. In the future, we should identify pages that are not suitable for the user profiling process. Such pages may be Web mail pages, search engine results and portal entry pages. To weaken the negative impact of such pages on the user profile construction, we could extend our stop-word collection with most frequent common Internet words. Another approach would be to allow the user to specify URLs (in the form of regular expressions, for instance) that should be excluded from the profiling process.

There is some work on document profiling that extends the vector representation by using word sequences (also termed *n-grams*) instead of single words (MLADENIĆ AND GROBELNIK, 2003). This work suggests that using single words and also word pairs for features in the vector representation of short documents improves the accuracy with which these documents are classified. We should incorporate these findings into our TFIDF vector representation of Web pages.

In our current implementation we are using the *nearest neighbor* approach to map the current interest to the topic ontology. Other more sophisticated machine learning techniques might provide better results in this process (e.g. classification with Naïve Bayes or SVM).

In this implementation, each time a page is viewed, the entire profile is rebuilt from scratch. We need to consider ways to update the topic ontology rather than rebuild it.

The clustering method used was not evaluated. We need to define evaluation methods for the profile generation process and, on the other hand, for the page classification process. This is not a trivial task and needs to be investigated in great detail. Once we are able to evaluate the algorithms, we will also be able to apply other approaches and see how they measure up to the one described in this paper.

The system was not tested in a real-life scenario. We should carry out an experiment involving test-users to see how useful the system really is.

Acknowledgement

This work was supported by the 6FP IP SEKT (2004–2006) (IST-1-506826-IP) and the Slovenian Ministry of Education, Science and Sport. The authors would also like to thank Tanja Brajnik for help in improving the overall clarity of this text.

References

1. KIM, H. R., CHAN, P. K. (2003): Learning Implicit User Interest Hierarchy for Context in Personalization.
2. PAZZANI, M., BILLSUS, D. (1997): Learning And Revising User Profiles: The Identification of Interesting Web Sites.
3. CHAN, P. K. (1999): A Non-Invasive Learning Approach to Building Web User Profiles.
4. BILLSUS, D., PAZZANI, M. J. (1999): A Hybrid User Model for News Story Classification.
5. SUGIYAMA, K., HATANO, K., YOSHIKAWA, M. (2004): Adaptive Web Search Based on User Profile Construction without Any Effort from Users.
6. MIDDLETON, S. E., SHADBOLT, N. R., DE ROURE, D. C. (2003): Capturing Interest through Inference and Visualization: Ontological User Profiling in Recommender Systems.
7. ADOMAVICIUS, G., TUZHILIN, A. (1999): User Profiling in Personalization Applications through Rule Discovery and Validation.
8. WASFI, A. M. (1999): Collecting User Access Patterns for Building User Profiles and Collaborative Filtering.
9. SALTON, G., BUCKLEY, C. (1987): Term Weighting Approaches in Automatic Text Retrieval. *Technical Report, COR-87-881, Department of Computer Science, Cornell University*.
10. STEINBACH, M., KARYPIS, G., KUMAR, V. (2000): A comparison of Document Clustering Techniques. In: *Proceedings of KDD-2000 Workshop on Text Mining, 109–110*.
11. MLADENIĆ, D., GROBELNIK, M. (2003): Feature Selection on Hierarchy of Web Documents. In: *Journal of Decission Support Sysems, 35, 45–87*.