Self-tuning Personalized Information Retrieval in an Ontology-Based Framework

Pablo Castells[1], Miriam Fernández[1], David Vallet[1],
Phivos Mylonas[2], and Yannis Avrithis[2]

[1] Universidad Autónoma de Madrid, Escuela Politécnica Superior,
Campus de Cantoblanco, 28049 Madrid, Spain
{david.vallet, miriam.fernandez, pablo.castells}@uam.es
[2] National Technical University of Athens, School of Electrical and Computer Engineering,
GR-15773 Zographou, Athens, Greece
{fmylonas, iavr}@image.ntua.gr

Abstract. Reliability is a well-known concern in the field of personalization technologies. We propose the extension of an ontology-based retrieval system with semantic-based personalization techniques, upon which automatic mechanisms are devised that dynamically gauge the degree of personalization, so as to benefit from adaptivity but yet reduce the risk of obtrusiveness and loss of user control. On the basis of a common domain ontology KB, the personalization framework represents, captures and exploits user preferences to bias search results towards personal user interests. Upon this, the intensity of personalization is automatically increased or decreased according to an assessment of the imprecision contained in user requests and system responses before personalization is applied.

1 Introduction

Broadly speaking, information retrieval deals with modeling information needs, content semantics, and the relation between them [9]. Personalized retrieval widens the notion of information need to comprise implicit user needs, not directly conveyed by the user in terms of explicit information requests [7]. Again, this involves modeling and capturing such user interests, and relating them to content semantics in order to predict the relevance of content objects, considering not only a specific user request but the overall needs of the user.

When it comes to the representation of semantics (to describe content, user interests, or user requests), ontologies provide a highly expressive ground for describing units of meaning and a rich variety of interrelations among them. Ontologies achieve a reduction of ambiguity, and bring powerful inferencing schemes for reasoning and querying. Not surprisingly, there is a growing body of literature in the last few years that studies the use of ontologies to improve the effectiveness of information retrieval [5,8,10,11] and personalized search [4]. In this paper, we present a comprehensive personalized retrieval framework where the advantages of ontologies are exploited in different parts of the retrieval cycle: query-based relevance measures, semantic user preference representation, automatic preference update, and personalized result ranking. The framework is set up in such a way that the models benefit from each other

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 977 – 986, 2005.
© Springer-Verlag Berlin Heidelberg 2005

and from the common, ontology-based grounding. In particular, the formal semantics are exploited to improve the reliability of personalization.

Personalization can indeed enhance the subjective performance of retrieval, as perceived by users, and is therefore a desirable feature in many situations, but it can easily be perceived as erratic and obtrusive if not handled adequately. Two key aspects to avoid such pitfalls are a) to appropriately manage the inevitable risk of error derived from the uncertainty inherent to the automatic user preference acquisition by the system, and b) to correctly identify the situations where it is, or it is not appropriate to personalize, and to what extent. With this aim, our proposed framework incorporates a module to control the degree of personalization that is applied in the search result ranking, automatically adjusting it depending on the uncertainty contained in the search before personalization. The precision of ontology-driven semantics enables sharper observations within the system, upon which such uncertainty is assessed.

The rest of the paper is organized as follows. The semantic search framework is described in the next section. Section 3 explains the personalization model built on top of this framework. Section 4 is devoted to the techniques for the dynamic adjustment of the personalization effect. Section 5 describes our experimental setup for this system, after which some final remarks are given.

2 Ontology-Based Content Retrieval

Our ontology-based retrieval framework [11] assumes the availability of a corpus Δ of text or multimedia documents, annotated by domain concepts (instances or classes) from an ontology-based KB O. The KB is implemented using any ontology representation language for which appropriate processing tools (query and inference engines, programming APIs) are available. In our semantic search model, Δ rather than O is the final search space. Since the metadata attached to a document provide only, in general, a subset of the full document semantics, we advocate for a model of *imprecise semantic search*, where documents may satisfy a query to different degrees on a continuous scale (rather than a boolean relevance value), according to which they can be ranked.

Our retrieval model works in two phases (see Figure 1). In the first one, a formal ontology-based query (e.g. in RDQL) is issued by some form of query interface (e.g. NLP-based) which formalizes a user information need. The query is processed against the KB using any desired inferencing or query execution tools, outputting a set of ontology concept tuples that satisfy the query. From this point, the second retrieval phase is based on an adaptation of the classic vector-space Information Retrieval model, where the axes of the vector space are the concepts of O, instead of text keywords. Like in the classic model, in ours the query and each document are represented by vectors q and d, so that the degree of satisfaction of a query by a document can be computed by the cosine measure:

$$\mathrm{sim}(d,q) = \frac{d \bullet q}{|d| \cdot |q|}$$

The problem remains to build the d and q vectors, which is summarized next.

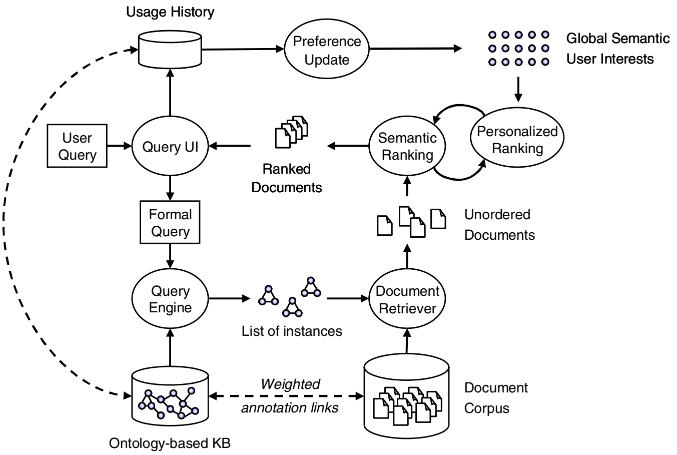

Fig. 1. The personalized ontology-based retrieval framework

Document vectors. Each content item in the search space Δ is represented by a vector d of concept weights, where for each domain concept $x \in \mathrm{O}$ annotating d, d_x represents the importance of the concept x in the document (if x does not annotate d, then $d_x = 0$). The weight of annotations can be assigned by hand or automatically. If the document contains text, d_x can be computed automatically by a TF-IDF algorithm [9] as:

$$d_x = \frac{freq_{x,d}}{\max_y freq_{y,d}} \cdot \log \frac{|D|}{n_x}$$

where $freq_{x,d}$ is the number of occurrences of x in d, $\max_y freq_{y,d}$ is the frequency of the most repeated instance in d, and n_x is the number of documents annotated by x. This requires that an appropriate mapping of concepts to text keywords be available, whereby the number of occurrences of a concept in a document can be defined as the count of concept keywords in the text. What an appropriate mapping is in this context, and how it can be automated is a subject of active research [8].

For audiovisual documents, a variety of strategies can be used to weight the relevance of concepts in the content, based on automatic content analysis techniques, such as the size, movement, or relative position (e.g. foreground vs. background) of automatically recognized objects [2], measures of recognition certainty, text-based processing of speech transcripts, etc.

Query vector. The proposed construction of the query vector defines $q_x = 1$ if x appears in some tuple of the query result set, and 0 otherwise. The weights q_x can be further refined with a TF-IDF scheme, as suggested by [9]:

$$q_x = 0.5 + 0.5 \cdot \frac{freq_{x,q}}{\max_y freq_{y,q}} \cdot \log \frac{|D|}{n_x}$$

where we define $freq_{x,d}$ as the number of tuples of the result set where x occurs.

Our experiments confirm that this model outperforms keyword-based or image-based schemes, but not surprisingly degrades as the knowledge needed to answer queries is missing from the KB. Since it is not realistic in general to expect a complete coverage of the semantic space involved in large real-world document collections by means of domain KBs, our model is complemented with classic techniques to perform acceptably (i.e. not worse than standard retrieval techniques) when the knowledge is missing. Further details can be found in [11].

The proposed retrieval model takes advantage of the additional semantics (class hierarchies, precise and formalized relations) expressed by the ontology, that cannot be expressed using keywords. Moreover, it supports a notion of conceptual search based on fuzzy annotation of unstructured contents (text, media) by concepts, not supported by the traditional multifaceted search by document fields (e.g. title, author, date, etc.). Additionally, it provides elaborated tools for measuring the vagueness or uncertainty in the expression of an information need, as will be shown in Section 4.2, which will provide a way to assess the adequacy of personalizing the search and to what extent.

3 Ontology-Based Personalization

Personalization is a means to improve the performance retrieval (e.g. measured in terms of precision and relevance) as subjectively perceived by users [7]. The key aspects involved include the representation of user interests (beyond a specific one-shot query), the dynamic acquisition of such interests by the system, and the exploitation of user preferences. Our personalization framework is built as an extension of the ontology-based retrieval model described in the previous section. It shares the concept-based representation proposed for retrieval, and the expressiveness of ontologies to define user interests on the basis of the same concept space that is used to describe contents.

In our personalization framework, the semantic preferences of a user are represented as a vector $u\in[0,1]^{|O|}$ of concept weights, where for each domain concept $x\in O$, $u_x\in[0,1]$ represents the intensity of the user interest for x. With respect to other approaches, where user interests are described in terms of preferred documents, words, or categories, here an explicit conceptual representation brings all the advantages of ontology-based semantics, such as reduction of ambiguity, formal relations and class hierarchies. Our representation can also be interpreted as fuzzy sets defined on the sets of concepts, where the degree of membership of a concept to a preference corresponds to the degree of preference of the user for the concept. This interpretation is used in the definition of automatic preference extraction techniques based on the observation of user actions as is shown in the next section.

3.1 Automatic Preference Update

The approach followed for extracting user preferences for personalization is based on a formal methodology that is founded on fuzzy relational algebra and the existence of semantic relations amongst concepts. The extraction of preferences for semantic concepts is achieved by applying clustering algorithms on usage information data. The considered usage data consists of documents selected by the user for viewing them, or explicitly marked as relevant in relevance feedback sessions. Our approach for extracting preferences from the history of user interaction consists of the clustering of documents based on the semantic annotation that matches concepts to documents, by which common topics implicit in clusters of concepts are detected.

The concept-vector representation of documents described in Section 2 can be reformulated to an equivalent interpretation of a document *d* as a normal fuzzy set on the set of concepts. Based on this set, and the knowledge contained in the form of available relations between the concepts, we aim to detect the degree to which a given document *d* is indeed related to a topic *t*. We will refer to this degree as $R(d,t)$. In other words, we attempt to calculate a relation $R : \Delta \times \mathrm{T} \rightarrow [0,1]$, where Δ is the set of available documents and T is the set of topics. Note that T is not known by the system beforehand, but emerges as a result of the algorithm. In designing an algorithm that is able to calculate this relation in a meaningful manner, three main issues need to be tackled. First of all, it is necessary for the algorithm to be able to determine which of the *topics* are indeed related to a given document, since a concept may be related to multiple, unrelated topics. In order for this task to be performed in a meaningful manner, the common meaning of the remaining concepts that annotate the given document needs to be considered as well. On the other hand, when a *document* is related to more than one, unrelated topics, we should not expect all the concepts that index it to be related to each one of the topics in question. Therefore, a clustering of concepts, based on their common meaning, needs to be applied. In this process, concepts that are misleading (e.g. concepts that resulted from incorrect annotation of the document) will probably not be found similar with other concepts that index the given document and therefore, the cardinality of the clusters may be used to tackle this issue. The main steps of the proposed algorithm are summarized in the following: (i) create a single relation that is suitable for use by thematic categorization, (ii) determine the count of distinct topics that a document is related to, by performing a partitioning of concepts, using their common meaning as clustering criterion, (iii) fuzzify the partitioning, in order to allow for overlapping of clusters and fuzzy membership degrees, (iv) take each cluster as a thematic topic and (v) aggregate the topics for distinct clusters in order to acquire an overall result for the document.

The topics that interest the user, and should be classified as positive interests are the ones that characterize the detected clusters. Degrees of preference can be determined based on the cardinality of the clusters, i.e., clusters of low cardinality should be ignored as misleading and the weights of topics in the context of the clusters, i.e., high weights indicate intense interest. The notion of high cardinality is modeled with the use of a *large fuzzy number* $L(\cdot)$, where $L(t)$ is the truth value of the preposition "the cardinality of cluster *t* is high". Therefore, each of the detected clusters *t* is mapped to positive interests by $u_x = \sum_{t \in T} \mu(x,t) \cdot L(t) \cdot K(t)$ for each $x \in \mathrm{O}$, where $\mu(x,t)$ denotes the degree of membership of the concept *x* to the cluster *t*, and $K(t) = \bigcap_{d \in t} R(d,t)$.

3.2 Personalization Effect

Once a semantic profile of user preferences is obtained, either automatically as described in the previous section, and/or refined manually, our notion of preference-based content retrieval is based on the definition of a matching algorithm that provides a personal relevance measure prm(*d*,*u*) of a document *d* for a user *u*. This measure is set according to the semantic preferences of the user, and the semantic annotations of the document, weighted as explained in Section 2. The procedure for matching *d* and *u* is based on a cosine function for vector similarity computation:

$$\text{prm}\left(d,u\right)=\frac{d\bullet u}{\left|d\right|\cdot\left|u\right|}$$

In order to bias the result of a search (the ranking) to the preferences of the user, the measure above has to be combined with the query-based score without personalization sim(*d*,*q*) defined in Section 2, to produce a combined ranking. The combination of several sources of ranking has been the object of active research in the field of IR [3]. We have adopted the so-called combSUM model, by which the two rankings are merged by a linear combination of the relevance scores:

$$\text{score}\ (d,\ q,\ u) = \lambda \cdot \text{prm}\ (d,\ u) + (1 - \lambda)\ \text{sim}\ (d,\ q)$$

where $\lambda \in [0,1]$. The choice of the λ coefficient in the linear combination above is critical and provides a way to gauge the degree of personalization, from $\lambda = 0$ producing no personalization at all, to $\lambda = 1$, where the query (local user interests) is ignored and results are ranked only on the basis of global user interests.

Given the inherent ambiguity of user actions upon which user preferences are automatically inferred, the automatic preference extraction techniques have an unavoidable risk of guessing wrong preferences, the negative effects of which increase with λ. Even when the extraction is most successful, there is considerable risk of contradicting explicit user requests if λ is too high, and λ should be therefore set with great care. It is commonly agreed that the user should have the means to turn personalization off (λ=0), or even tune λ as a free parameter (see e.g. Google Personalized[1]). Other than this, a fixed moderate value for λ can be typically set by experimental tuning, but we argue that the same λ is not necessarily appropriate for all situations. Further, we claim that it is possible to find hints in the context of a search according to which the level of personalization can be automatically self-adjusted, as is explained in the next section.

4 Gauging the Impact of Personalization

The degree to which the query dominates the retrieval process should vary in a manner that optimizes the retrieval result, i.e. in a manner that minimizes its uncertainty. As a general principle, if there is a high certainty that the results without personalization are relevant for an information need, personalization should be kept to a minimum. Put otherwise, the intensity of personalization should increase monotonically

[1] http://labs.google.com/personalized

with the degree of uncertainty in the search. Assessing (or even defining) such uncertainty with the information available in the system, before the results are presented to the user, is a fairly difficult problem in general. However, we propose an approximation to such assessment, by taking the *vagueness* in user requests and system responses as an approximation of the uncertainty in the search.

4.1 Assessing the Vagueness of the Search

In our proposal, the vagueness of a search (or equivalently, its *specificity* counterpart) is defined in terms of the specificity of the formal query, the query result set (concepts), and the final result set (documents), based on the retrieval model described in Section 2. Each of these three aspects is examined separately, and the corresponding results are combined into a single measure. More formally, given an ontology query q, containing a set of variables V_q, let $T_q \subset O^{|V_q|}$ be the result set of the execution of q (first retrieval phase as described in Section 2), and let $R_q = \{ d \in \Delta \mid \text{sim}(d, q) > 0 \}$ be the final result set in terms of documents (second retrieval phase). The specificity of a search is defined as a function spec $(q) = f(\text{spec}_1(q), \text{spec}_2(T_q), \text{spec}_3(R_q))$, where $f : [0,1]^3 \rightarrow [0,1]$ should be monotonically increasing with respect to its three variables. Our current empirical choice is the geometric mean $f(x, y, z) = (x \cdot y \cdot z)^{\frac{1}{3}}$.

The three partial specificity measures are defined as follows. First, we define $\text{spec}_1(q) = 1 - \frac{|V_q|}{3|C_q|}$, where $|V_q|$ is the number of variables in the query, and $|C_q|$ is the number of conditions. Thus, a query with many conditions and few variables is taken as more specific. Second, we take $\text{spec}_2(T_q) = 1 - \frac{\log(1+m)}{\log(1+|O|)}$, where $m = \left|\{x \in O \mid \exists t \in T_q, \exists v \in V_q, t_v = x\}\right|$, i.e. m is the number of distinct ontology elements occurring in the query result set. According to this measure, a query that is satisfied by many ontology instances (in relation to the size of the KB) is considered more unspecific. Finally, $\text{spec}_3(R_q) = 1 - \frac{\log(1+|R_q|)}{\log(1+|D|)}$, by which the fewer results a search returns, the more specific it is considered.

4.2 Adjusting Personalization by Impact

Once a notion of the vagueness of a query is established, a method for setting the degree of personalization in relation to that measure has to be defined. A very simple approach would be to set $\lambda = 1 - \text{spec}(q)$, which would implicitly take λ as a synonym for the "degree of personalization". But a better definition of "personalization impact" can be given in terms of the effective change of position of documents in the ranking. Dwork et al [3] propose the Spearman footrule distance to measure the difference between two search result lists as the average displacement of each document across the rankings. We argue that a more significant measure of impact is whether or not a result will be seen at all by the user. Since in general the user will not browse the entire list of results, but stop at some top k in the ranking, there are a number of documents that the user would not see (the ones ranked after the k-th result) in the ranking without personalization, but would see as a result of a personalized reordering, and vice versa. If we count the rate of documents in the whole collection that cross the line for each possible value of k, and multiply it by the probability $P(k)$ that the user stops at each k, we get a loss function ranging in [0,1] that provides a measure of the effective impact (thus, the risk) of personalization in the retrieval process:[2]

$$\gamma(q,u) = \frac{1}{|D|}\sum_{k=1}^{|D|} P(k) \sum_{d\in D} \chi_k(d,q,u),$$

where $\chi_k(d,q,u) = \begin{cases} 1 \text{ if } \mathrm{sim}(d,q) \le k \text{ and } \mathrm{score}(d,q,u) > k \\ 1 \text{ if } \mathrm{sim}(d,q) > k \text{ and } \mathrm{score}(d,q,u) \le k \\ 0 \text{ otherwise} \end{cases}$

Now, rather than setting λ (i.e. the amount of personalization input) as a function of the vagueness of the search, we fix a desired output value for the effective impact of personalization in terms of that vagueness, and then set the value of λ that would yield this impact. Put formally, we equate $\gamma\ (q,u) = g$ (spec (q)), and find λ from this equation, which is achieved as follows. To make $\gamma\ (q,u)$ linearly increasing with the uncertainty of the search, we define g (spec (q)) = (1 – spec (q)) · $\gamma\ (q,u)|_{\lambda=1}$, where $\gamma\ (q,u)|_{\lambda=1}$ is the maximum value of $\gamma\ (q,u)$ for a given query, reached when $\lambda = 1$. $\gamma\ (q,u)$ is in fact a (monotonically increasing) function of λ, but it is not simple to invert this function analytically. However, it can be inverted empirically at runtime with high precision by computing $\gamma_i\ (q,u)$ for a discrete set of values $\lambda_i = i/n \in [0,1]$ with $1 \le i \le n$ (e.g. $n = 20$ which is not expensive[3]), and then defining $\lambda = \lambda_i$ for $\gamma_i\ (q,u) \le \gamma_n\ (q,u) \cdot$ (1 – spec (q)) $< \gamma_{i+1}\ (q,u)$.[4] For the computation of $\gamma_i\ (q,u)$, we have taken an approximation to the distribution function for $P(k)$ by interpolation of data taken from a statistical study [6]. The final effect of this approach is that it is $\gamma\ (q,u)$, rather than λ, that is proportional to the vagueness of the search.

5 Early Experiments

We are testing our techniques on a corpus of documents from the CNN web site,[5] comprising 145,316 documents (445 MB). The domain ontology KB was taken from the KIM Platform [8], developed by Ontotext Lab,[6] with minor adjustments, plus the specific extensions of our framework (see [11] for a description of these). The domain KB includes 278 classes, 131 properties, 34,689 instances, and 462,848 sentences,

[2] We assume sequential browsing, i.e. the user does not see the i-th result before the $(i–1)$-th.

[3] Computing $\gamma_i\ (q,u)$ is $O(n \cdot m^2)$, where m is the number of elements in $\{d\in\Delta \mid \mathrm{sim}\ (d, q) > 0$ or $\mathrm{prm}\ (d,u) > 0\}$.

[4] If $\gamma_i\ (q,u) = \gamma_{i+1}\ (q,u)$ for some i we would remove λ_i from the set without loss of precision.

[5] http://dmoz.org/News/Online_Archives/CNN.com

[6] http://www.ontotext.com/kim

taking a total of 70,5 MB in RDF text format (though in practice it is stored within a MySQL back-end using Jena 2.2). Based on the concept-keyword mapping available in the KIM KB, we automatically generated over $3 \cdot 10^6$ annotations (i.e. over 25 per document on average).

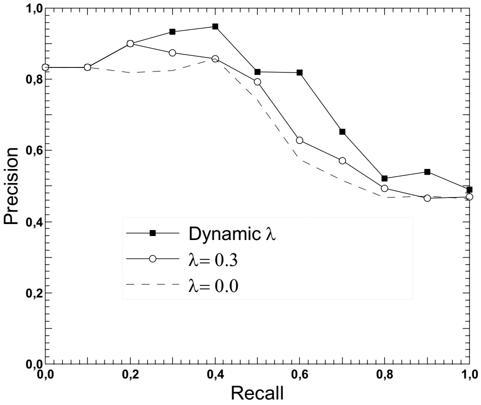

Fig. 2. The graphic illustrates the performance of our technique for the query "child organizations of public companies", using the standard precision vs. recall curve [9], according to different options to gauge personalization: i) personalization impact proportional to vagueness of the search, ii) personalization with fixed λ = 0.3, and iii) no personalization.

Figure 2 illustrates the performance of our techniques on the query "child organizations of public companies", compared to the results obtained without self-adjustment, and without personalization. In this case, the dynamic adjustment raises λ to 0.6 because the query is rather vague, according to the principles explained in Section 4.1, perceptively improving performance. The evaluation is done on the basis of a manual rating of document relevance on a scale from 0 to 5. Though our initial experiments are showing promising results, a more extensive testing is needed (and actually under way at the time of this writing), in order to complete and extend these first observations.

The experiments were run on a standard PC. Although systematic efficiency tests have not been conducted yet, the typical observed time to process a query takes below one minute. The main bottleneck is in the traversal of annotations (for the calculation of sim and prm), which are currently stored as an extension to the ontology. This cost grows linearly with the size of the result sets ($|T_q|$ and $|R_q|$). We expect to reduce this overhead by storing the annotations in a separate DB.

6 Conclusions

Reliability is a well-known concern in the field of personalization technologies. Since automatic preference modeling involves guessing implicit user's interests, it is impossible to approximate a total accuracy in meeting actual user needs. However it is possible to predict when the effect of potential failures can be serious, or close to harmless. Our proposal aims at an automatic prediction of this effect, in order to raise or lower the level of personalization accordingly. The techniques are built upon a comprehensive framework that reaps the benefits from the expressive power and precision of ontologies in the different phases of the retrieval and personalization process.

The directions for the continuation of our work are manifold. To mention but a few, we are studying the integration of further ontology-based specificity measures (see e.g. [1,10]) in our uncertainty assessment techniques. Also, we plan to research a finer, qualitative, context-sensitive activation of user preferences, by which the level of personalization would not be uniform, but would be selectively distributed with a higher weight on the most context-relevant preferences.

Acknowledgements

This research was supported by the European Commission under contract FP6-001765 aceMedia. The expressed content is the view of the authors but not necessarily the view of the aceMedia project as a whole.

References

1. Anyanwu, K., et al: SemRank: Ranking Complex Relationship Search Results on the Semantic Web. 14th Intl. World Wide Web Conference (WWW 2005). Chiba, Japan (2005)
2. Bloehdorn, S., Petridis, K., Saathoff, C., et al: Semantic Annotation of Images and Videos for Multimedia. 2nd European Semantic Web Conference (ESWC 2005). Lecture Notes in Computer Science, Vol. 3532. Springer Verlag, Berlin Heidelberg New York (2005)
3. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank Aggregation Methods for the Web. In Proc. of the 10th Intl. World Wide Web Conference (WWW10), Hong Kong (2001)
4. Gauch, S., Chaffee, J., and Pretschner, A.: Ontology-based personalized search and browsing. Web Intelligence and Agent Systems 1, Issue 3-4, IOS Press (2003) 219-234
5. Guha, R. V., McCool, R., Miller, E.: Semantic search. Proc. of the 12th Intl. World Wide Web Conference (WWW 2003), Budapest, Hungary (2003) 700-709
6. Jansen, B. J., Spink, A.: An Analysis of Web Documents Retrieved and Viewed. Proc. of the 4th International Conference on Internet Computing. Las Vegas, Nevada (2003) 65-69
7. Micarelli, A., Sciarrone, F.: Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. User Modelling and User-Adapted Interaction 14, Issue 2-3, Springer Science (2004) 159-200
8. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Journal of Web Sematics 2, Issue 1, Elsevier (2004) 47-49
9. Salton, G. and McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
10. Stojanovic, N., Studer, R., and Stojanovic, L.: An Approach for the Ranking of Query Results in the Semantic Web. In: Fensel, D., Sycara, K. P., Mylopoulos, J. (eds.): The Semantic Web – ISWC 2003, 2nd Intl. Semantic Web Conf. Lecture Notes in Computer Science, Vol. 2870. Springer Verlag, Berlin Heidelberg New York (2003) 500-516
11. Vallet, D., Fernández, M., and Castells, P.: An Ontology-Based Information Retrieval Model. 2nd European Semantic Web Conference (ESWC 2005). Lecture Notes in Computer Science, Vol. 3532. Springer Verlag, Berlin Heidelberg New York (2005) 455-470