User Modelling on Semi-Structured Documents*

Sanghee Kim[1], Wendy Hall[1] and Andy Keane[2]

[1]Intelligence, Agents, Multimedia Group, Department of Electronics and Computer Science, University of Southampton, U.K., Tel: 44-(0)23-80-59-3256, Fax: 44-(0)23-80-59-2865
{sk98r,wh}@ecs.soton.ac.uk
[2]Computational Engineering and Design Center, School of Engineering Science, University of Southampton, U.K., Tel: 44-(0)23-59-2944, Fax:44-(0)23-80-59-3220
ajk@soton.ac.uk

Technical Report Number: ECSTR–IAM02-001
January 2002
ISBN: Pending

Abstract. We present a new approach that makes use of the embedded structural information of the documents that a user frequently refers to for deriving a personalized concept hierarchy and for identifying user preferences concerning document searching and browsing. In contrast with conventional methods that ignore the distribution of structural elements, our approach accepts semantic clues defined in the structural tags, so that it takes full advantage of the textual and structural information. Formal concept analysis theory is applied to define semantic relationships among the concepts, and reinforcement learning is employed to unobtrusively adapt to individualized information needs. Given a user's query, the personal ontology and user model maximize their knowledge bases to present the relevant documents in a prioritized order, one which the user prefers to browse. We demonstrate the practicability of our approach through two experiments. The former showed significantly improved performance results compared to that of a flat vector model. The second experiment compared the performance accuracy of the structure elements exploited by four users and thereby demonstrated the diversity of user preferences.

Keywords: Personal ontology, Supported browsing, Structured document, Reinforcement Learning, Formal concept analysis

1 Introduction

As it has come to be harder for a user to locate relevant documents quickly, the intelligent document organization system that proactively retrieves, categorizes, and ranks searched documents has become a desirable component of personal desktop applications. Incorporating user models on top of document models demonstrates a clear advantage in that it specifies the salient characteristics of the user that the system can make use of for identifying the user's information needs.

Reviews of current approaches show that embedded structural elements are often ignored, even though these present semantic clues about the kind of underlying document handling that would provide the relevant descriptions of the user and the document. We identify the following two reasons for this. Firstly, conventional text mining methods are designed for unstructured documents insofar as no recognition of structural information is necessary. Secondly, integrating heterogeneous document sources, each of which has its own structural elements and vocabularies, poses a new challenge. Using ontologies for the integration task shows a clear advantage as

* The work presented here has been supported by UTP (University Technology Partnership in Design) program sponsored by Rolls Royce Plc and BAE Systems.

ontologies explicitly define shared vocabularies and structures. Ontology building generally purses shared terminologies among linked objects, so one of the difficulties is that much of the work needs to be done manually rather than automatically. However, we believe that this is not true of a personal ontology, which is more concerned with how to organize the related documents for an effective and fast retrieval given user information needs.

We propose a new approach that makes use of the embedded structural information of the documents that a user frequently refers to for deriving a personalized concept lattice and for identifying user preferences concerning document organization and access patterns. Formal concept analysis theory, which explores the exhaustive combinations of the associated features, is applied to automatically construct a personal ontology. It derives a personalized concept lattice that contextually infers meaningful semantic relationships from related documents. In order to unobtrusively model the user preferences, we base our approach on a reinforcement learning framework so that it can maximize the implicitly available user feedback.

This paper is organized as follows: in section 2, the procedural and theoretical details of personal ontology building are given; section 3 presents descriptions of reinforcement learning based user models. Experimental results with two datasets are reported in section 4 followed by reviews of the current approaches in section 5 and conclusions and directions for future work in section 6.

2 Personal Ontology Building

Personal ontologies proactively organize and categorize the documents that a user frequently refers to for an effective retrieval given individualized information needs. Instead of requiring a user to specify his/her preferred ontology hierarchy, we propose to bootstrap the initial nodes by making use of existing user defined categories, such as file directories, folder hierarchies of e-mail messages or a user's homepage. Not only does this initialization reduce the user's reliance on system interaction, but it also provides good training examples for profiling the user. It is our hypothesis that a user's homepage provides useful structural information about user activities, particularly about a user's professional and personal interests. The development of personal ontologies involves three steps. First, a hierarchy of structural elements is defined and the values of the specified tags from the documents are extracted. Given the structured documents, a set of top-level concepts is derived from the user-defined categories through a two-step category clustering process. Finally, the explicit specifications of relations among the concepts are defined and documented.

2.1 Document Structures

We use XML (eXtensible Markup Language) as the document representation language since it is commonly used for information integration and it is able to represent nested structures. Figure 1 shows a DTD (Data Type Definition), which is used for defining the document structures.

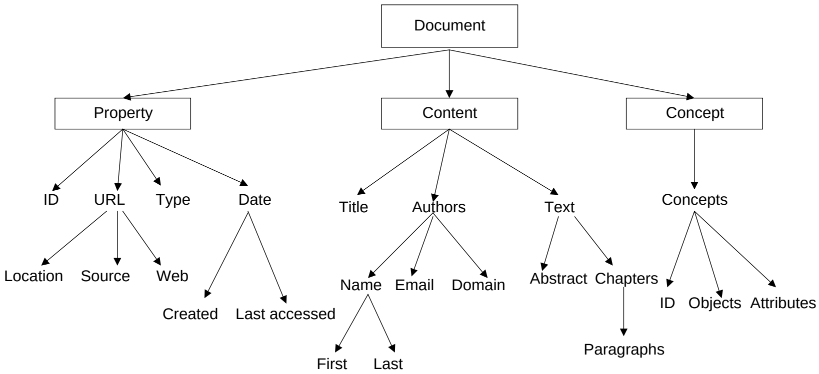

Fig.1. A document is modeled as three parts. The property part is for system-purposes and defines external information about the document. The second part outlines content descriptions extracted from the specified tag in the document. The concept element is appended from a concept lattice.

2.2 Category Clustering

We create a set of top-level (i.e. clusters) in two steps, by decomposing and then grouping the structured documents across user-defined categories. There are two reasons for creating clusters instead of directly taking the existing user-defined categories. Firstly, as the user preferences concerning document organization are prone to change, the coherence of categories is not stable: two categories may have overlapping properties, or a category may be two diverse. Secondly, if the user assigns identical names to the categories created for different information sources, it will be necessary to integrate them to create a new category.

K-means clustering algorithms are simpler and faster than using other hierarchical and unsupervised algorithms for grouping similar documents [11]. Instead of randomly assigning initial K categories, we extend the K-means algorithms to discover a reasonable range of values for K by computing the most non-similar K documents. In the first step, similarities among documents in each category are computed in order to group only similar documents into a sub-category. In the second step, similarities among all sub-categories are compared in order to cluster similar sub-categories into a super-category.

2.3 Concept Lattice - Formal Concept Analysis (FCA)

Given the category hierarchy created in section 2.2, formal concept analysis (FCA) can then be applied to construct a personalized concept hierarchy. FCA is based on applied lattice theory and defines the formal context consisting of a set of concepts, each of which is specified as: (C,A_1), where C is the extent that holds all objects belonging to the concept, and A_1 is the intent that comprises all attributes valid for those objects [3].

In a personal ontology, the object corresponds to a document and the associated attributes are extracted features. Compared to other types of data structures (e.g., a tree, or a graph), using a lattice structure to represent an ontology provides many advantages. Firstly, it is relatively easy to integrate the existing categories with FCA since a sub-lattice can be automatically merged into a more general lattice of categories. Secondly, the addition of new documents creates a new lattice that is derived from the existing lattice. As FCA refines the new lattice to hold only the minimal number of objects needed to describe the new attributes, FCA can compact the lattice size to avoid a combinational explosion [9]. A concept lattice is constructed through an attribute exploration process that extensively investigates the combinations of related attributes to define co-relationships among the concepts. This may be contrasted with conventional information retrieval methods that reduce given information to extract a limited number of attributes and assume no dependency among the attributes. A super or a sub-concept is defined as: $(D,A_1)\subseteq (C,A_2)\Leftrightarrow D\subseteq C$ $(\Leftrightarrow A_2\subseteq A_1)$ meaning that a concept C is a super-concept of D if and only if the attributes (A_2) of C are a subset of the attributes (A_1) of concept D.

A many valued context (MVC) extends FCA to represent attributes that have several values [3]. An MVC consists of a set of objects, a set of many valued attributes, and a set of attribute values. As objects include a number of structural elements, each of which is associated with a set of attributes in the personal ontology, we develop a lattice per element by using MVC. Figure 2 shows an example of such a concept lattice.

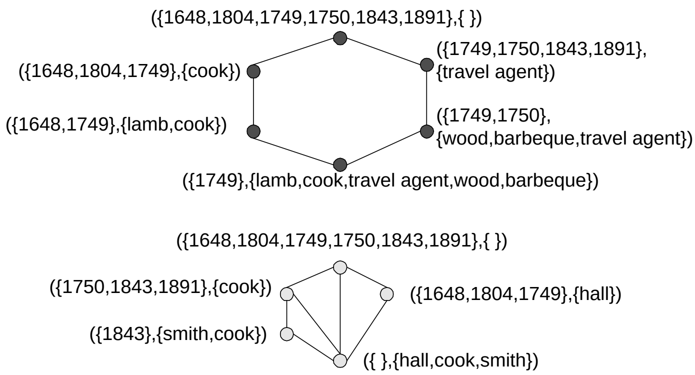

Fig.2. An example of a concept lattice: each concept is denoted as a pair, consisting of a set of objects (i.e., document IDs) and a set of attributes. It shows two lattice, each of which is derived from one of the structural elements. The top lattice and the bottom lattice use the attributes extracted from the abstract and the last name tag, respectively. It also shows the advantage of using the structural elements since an attribute may relate to different objects depending on the context. For instance, the definition of 'cook' in the top lattice is different from the meaning in the bottom lattice which denotes the last name of the related person.

3 Reinforcement Learning for Ranking Structured Documents

User feedback is a useful component for learning about user preferences, and ranking techniques can make use of feedback in order to improve ranking accuracy. Explicit feedback providing, for example, a relevance rating or a sorting into relevant or non-relevant categories is more meaningful compared to implicit feedback. However, since most users are not keen on giving the time necessary to providing such feedback, the ranking algorithms need to take advantage of implicitly observed user behaviors.

We present a new approach, called RLRSD (Reinforcement Learning for Ranking Structured Documents) that is based on a reinforcement-learning framework to effectively learn about user preferences while requiring the minimum of effort by the user. RLRSD also extends conventional document retrieval methods to accommodate user preferences for accessing documents by their structures. That is, we explicitly differentiate terms in a given query, each of which may refer to different objects in the different structural elements. Reinforcement learning (RL) is selected since its computation can be incrementally updated on-line and it directs a learner a future optimal state. A learner in RL receives numeric-valued feedback (i.e., an immediate or future reward) when it takes an action at a given state [5]. The goal of using RL is to discover user preferences without requiring the user's explicit involvement. The only feedback obtained from the user is through observing whether or not the user clicks the specific document, and it is incorporated as the immediate reward which defines the effectiveness of ranking strategies. RL takes into account the feature differences of the ranked documents plus the previously learned user profiles in order to evaluate the value of the next ranking strategy.

User preferences are modeled on two layers, comprising global and local profiles. The global profile represents overall user browsing preferences regarding the distribution of structural elements, each of which specifies its strength by a numeric weight. The local profile defines a set of features associated with importance weights. Formally, the global profile P is a set of local profiles and is defined as:

$$P = \{(P_1, W_1), (P_2, W_2), \ldots, (P_n, W_n)\}\ . \tag{1}$$

where, W_n defines the weight of the local profile P_n which is a set of features:

$$P_n = \{(t_{n1}, w_{n1}), (t_{n2}, w_{n2}), \ldots (t_{nk}, w_{nk})\}\ . \tag{2}$$

where, t_{nk} is the *kth* feature associated with a weight, w_{nk} .

A revision process involves two steps. Firstly, it revises the global preference by using an ε -greedy evaluation method, where ε is a randomly assigned probability [10]. Not only does this method exploit current knowledge to maximize immediate reward, but it also explores as yet unknown features that are possibly relevant to the preferences once in a while. In other words, it selects a set of features that contributes to the reflection of more meaningful user preferences in the future. This selection method was also used for evaluating the impact of four behaviors (i.e., time spent reading, bookmarking, scrolling up and down, following up) observed from a user's Web browsing patterns to identify user interests [7]. Formally, the global preference (i.e., the weight of local profile P_n at state $t+1$) is revised as:

$$W_n^{t+1} = W_n^{t} + C_n / \sum_{n=1}^{N} C_n\ . \tag{3}$$

where, N is the number of local profiles, and C_n is defined as:

$$C_n = \alpha\left[R_n^{t+1} + \gamma \frac{1}{M} \sum_{j=1}^{M} \sum_{i=1}^{K} \left(w_{ni,\,j} - w_{ni,\,j+1}\right)\right]. \tag{4}$$

where, α ($0 \leq \alpha \leq 1$) is a confidence rate indicating that the modeled structures can completely capture user feedback. Generally, α is set to less than 1 since the user may have preferences which are not directly learnable from the documents. *K* is decided by using the ε -greedy evaluation method. M is the number of documents that are retrieved by profile P_n , and γ is a discount factor reflecting that future rewards are less valuable than more immediate rewards. R_n^{t+1} is computed as:

$$R_n^{t+1} = \left(\sum_{j=1}^{M} D_j\right) / S\ . \tag{5}$$

where, D_j is the weight of the *jth* selected document, and starting from the first selected document, its weight is decreased by $1/\log_2(j)$. S is the total number of user-selected documents. In the second step, the local profile, w_{ni}^{t+1} , (i.e., the weight of a feature, t_{ni} , in profile P_n) is revised as:

$$w_{ni}^{t+1} = w_{ni}^{t} + \alpha R_n^{t+1}\ . \tag{6}$$

where, α and R_n^{t+1} are as specified in (4) and (5), respectively.

4 Evaluations

To demonstrate the efficiency of our approach, we experimented with two datasets, each of them with different properties, and thereby drew a range of useful conclusions regarding the proposed approach. With the first dataset, the performance of our approach (RLRSD) was compared to that of a flat vector model (FVM) which represents documents as an unstructured single layer. The second experiment compared the performance accuracy of the structural elements exploited by four users and illustrated the diversity of user preferences.

To measure how well the ranking algorithms perform, precision and recall criteria were used, and by using the recall as an interpolation point, a total of 11 precision values were estimated. However, in contrast with a text classification or filtering task, we had to measure the performances of the ranking algorithms not only by considering the number of correctly retrieved documents, but also by assessing the correctness of the order of the documents. We employed the method used by Yuwono *et al*. to compare four retrieval algorithms in the task of locating relevant Web pages [12]. Starting with the lowest ranked document (*m*) that is relevant to a query, recall is assumed as 1, and precision is measured as the number of correctly ranked documents (*R*) divided by the location of the rank (*m*). Then, we proceeded to the next lowest rank document (m_1). The precision was computed by (*R-1*) divided by (m_1), and the value of *(R-1)* divided by *R* was assigned to the recall.

For the first experiment, one of the authors provided a total of 1405 documents (1148 email messages, 166 bookmarked web pages, 88 texts in postscript format, and 3 web pages from a homepage), collected from 94 categories. These documents were tagged and converted into XML documents using the DTD described in section 2.1. When we applied the clustering method described in section 2.2 to this XML collection, it generated a two-level category hierarchy by firstly decomposing it into 127 sub-categories, and then grouping them into 10 clusters. 1138 concepts were derived from the concept lattice that was developed based on the given 10 clusters. 176 queries were generated, each of which consisted of query terms, and randomly selected structures. We manually examined the entire collection to assign ranking scores to the selected documents for each query. These were fed into the ranking algorithm as user feedback. Figure 3 compares the ranking accuracy of RLRSD to that of FVM.

The approximately 37% higher precision of RLRSD shows the efficiency of using the structural information to assign different weights to a feature that relates to different objects in different structures. Moreover, the improvement (approximately 13%) made by incorporating user feedback into RLRSD is more distinct than that for the FVM (approximately 6%). We suggest that this is mostly due to a term confusion problem caused by the fact that no differentiation of structures is made in FVM. In other words, although FVM received feedback, it did not reflect the different contributions of a feature which relates to different objects to identifying user preferences; instead it assigned a uniform weight to a feature irrespective of its linked objects.

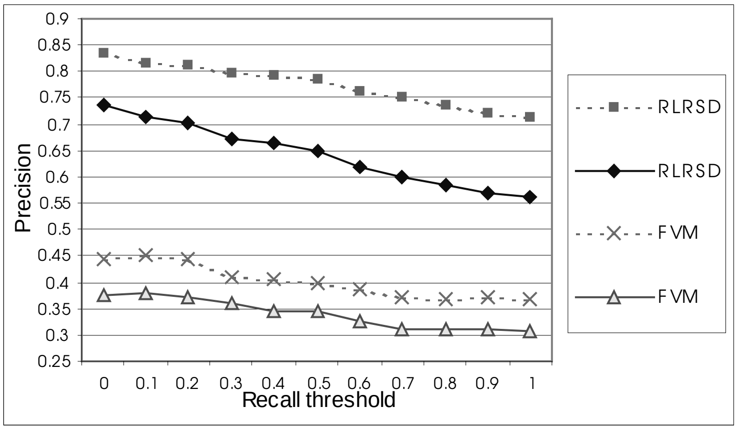

Fig.3. α and γ are set to 0.9 and 0.5, respectively. The dotted line indicates that the specified algorithm received user feedback.

For the second experiment, we downloaded the publicly available cystic fibrosis (CF) dataset. It consists of 1239 XML documents indexed with the term ’cystic fibrosis’ in the National Library of Medicine’s MEDLINE database [8]. One of the advantages of using this dataset is that we can make use of readily available ranked sets of documents associated with simulated queries. That is, it gathered subjective relevance judgments (i.e., highly relevant, marginally relevant, or irrelevant) made by four users against the selected 100 queries. We created an individual relevance score file per user to compare the ranking results by using different structural elements, to which the user referred in deciding the scores. User5 was artificially created by combining the relevance scores produced by the four users. Given the original structure tags, seven elements (title, minor subject, major subject, abstract, reference, citation, and source tag) were used for this experiment. Figure 4 shows the precision curve estimated by each user.

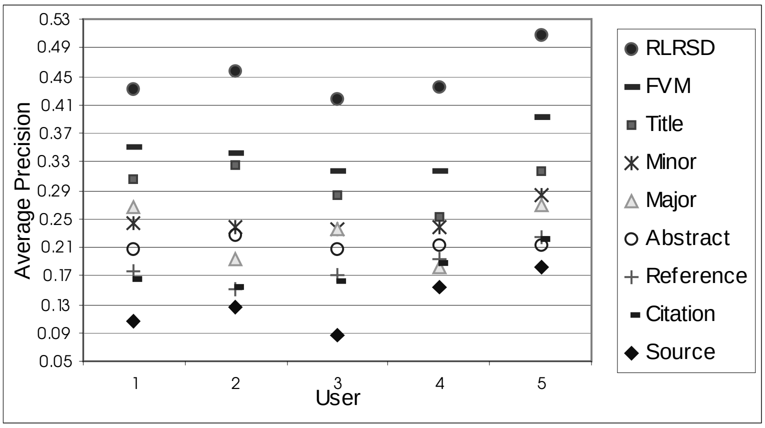

Fig.4. RLRSD outperformed FVM and also achieved a higher performance compared to that of the best-performing structural tag over all users.

User4 shows a unique result in that no specific preferences concerning the seven tags used in deciding scores are observed since the difference between the highest (by title tag) and lowest (by source tag) precision values is rather small compared to those of the other three users. While the other three users are domain experts, user4 is a bibliographer who presumably has better knowledge of citation and reference elements. The relatively higher precision rate of the citation, the reference, and the source tag compared to those of the other three users confirms our finding that user4 uses bibliographic-related tags heavily. While it is easy to notice that the title and the source tag obtained the highest and the lowest results, it is hard to conclude any correlations between the tags and the users that are commonly applicable to all users. This emphasizes the diversity of user preferences.

The performance difference (11%) between RLRSD and FVM on the second experiment was not so significant compared to that (37%) of the first experiment. We base our explanation on the following observation. Dataset2 shared the same structural specification, while dataset1 was originated from heterogeneous sources and so the variations in the distribution of the structural elements are wider than those in dataset2. As RLRSD utilizes both the distribution of structural elements and features, it is clear that RLRSD could not take full advantage of the global profile of dataset2 compared to dataset1. Moreover, while there were only three kinds of feedback (i.e., highly relevant, marginally relevant, or irrelevant) on dataset2, dataset1 used a minimum of one and a maximum of thirty kinds of feedback. As shown in Figure 4, the performance of user5 (who received nine kinds of feedback) obtained higher precision values than those of the other three users. This shows that the ranking algorithms can improve ranking performance when feedback reflects more detailed user preferences.

5 Related Work

The development of a personal ontology is a newly emerging research area which has its roots in the fact that intelligent document organization has become a major concern in personal desktop applications due to the increasing difficulty experienced in coping with large amounts of information. OBIWAN (Ontology Based Informing Web Agent Navigation) maps Web pages into a user's personal ontology for Web navigation support [2]. The mapping is based on a reference ontology provided by Lycos to determine the best-matched node in the ontology by measuring similarity to the retrieved page. In contrast with our approach, the user is expected to define his/her ontology structure, while in our approach this was bootstrapped through formal concept analysis theory. Moreover, OBIWAN poses stringent requirements on the user to submit an ontology that contains at least ten concepts with at least five documents for each concept.

As hyperlinks can be assumed to be a kind of structural information present in hypertext classifying documents, at least two approaches make use of them for the task of Web pages. Compared to the term-frequency based methods, ranking algorithms which are based only on hyperlink information show poor performance [12]. It is argued that though the connectivity between the pages may provide any level of co-relation, it is difficult to extract meaningful semantic relationships due to the diversity of the Web domain. Attardi *et al*. use the embedded link paths together with associated link titles to categorize unseen Web pages. Given a URL path, they evaluate a proper category tree node by matching the titles of the tree to the extracted phrases in the path [1].

Reinforcement learning (RL) has been applied to text mining applications, for instance, for the task of dynamically generating hyperlinks inferred from user browsing patterns, or for an efficient Web page spidering. One of the advantages of using RL is that it allows systems to maximize environmental properties by taking into account the situated constraints by adjusting related variables. Rennie and McCallum applied RL for a domain-specific search spider and reported that RL provided a basis for measuring the utility of actions that give no immediate benefit, but give benefit in the future [6]. One of the experiments reported was to find computer science papers from four university sites, and RL-based algorithms outperformed the breadth-first strategy. Moreover, the RL-based algorithms used many fewer resources in spidering. WebWatcher is a Web navigation support agent that employs RL for recommending the most likely links to the user, given the user's browsing behaviors [4]. It separately approximates the value of each word on a hyperlink represented as a TFIDF vector model.

As document providers can express more contextualised and complex information, structured documents are becoming a major representation language in many areas, e.g., in digital library, e-commerce or Web applications. Structured documents pose new challenges to conventional methods, however, since these were developed for unstructured documents. We believe that user modeling methods also need to be reviewed in order to accommodate user preferences expressed in multiple structure layers. We have demonstrated that it is clearly advantageous to maintain two-layers instead of a single layer for learning about the preferences because different weights can be assigned to a feature that relates to different objects depending on the context.

6 Conclusions and Future Work

We have presented a novel approach (RLRSD) that uses the distribution of structural elements of a document in the development of a personal ontology and requires minimum effort by the user in identifying individualized preferences concerning document organization and browsing.

Since most users are not keen on giving the time necessary to providing their preferences, RLRSD uses readily available user information, for example, the existing user-defined categories to bootstrap its knowledge base about the user preferences. Because these initial user profiles might not consistently reflect the most significant user preferences, RLRSD incrementally revises the profiles according to observed user behaviors.

Through two experiments, RLRSD showed a significant performance improvement for the task of ranking and thereby demonstrated the practicability and the usability of the approach.

We have not yet fully investigated the impact of varying the number of retrieved documents on the performance results by assigning a fixed maximum number to all local profilers in spite of their different strengths. Future work will extend the approach to discover an optimum number.

References

1. Attardi, G., Marco, S., Salvi, D.: Categorization by Context. Int. J. UCS Vol. 4. No. 9. (1998) 719-737
2. Chaffee, J., Gauch, S.: Personal Ontologies for Web Navigation. Proc. Ninth Int. Conf. On Information and Knowledge Management (2000)
3. Ganter, B., Wille, R.: Applied Lattice Theory: Formal Concept Analysis. Preprints, http://wwwbib.mathematik.tu-darmstadt.de/Math-Net/Preprints/Listen/pp97.html (1997)
4. Joachims, T., Freitag, D., Mitchell, T.: WebWatcher: A Tour Guide for the World Wide Web. Proc. Int. Joint Conf. on Artificial Intelligence (1997) 770-775

5. Mitchell, T.: Machine Learning. Int. edn. McGraw-Hill, Singapore (1997)
6. Rennie, J., McCallum, A.: Using Reinforcement Learning to Spider the Web Efficiently. Proc. Sixth Int. Conf. on Machine Learning (1999)
7. Seo, Y., Zhang, B.: Learning User's Preferences by Analyzing Web-Browsing Behaviors. Proc. Fourth Int. Conf. On Autonomous Agents (2000) 381-387
8. Shaw, W., Wood, R., Tibbo, H.: The Cystic Fibrosis Database: Content and Research Opportunities. Int. J. Library and Information Science Research Vol. 13. (1991) 347-366
9. Sowa, J.: Mathematical Background. http://www.bestweb.net/~sowa/misc/mathw.html (2000)
10. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. http://www-anw.cs.umass.edu/~rich/book/the-book.html (1998)
11. Tou, J., Gonzalez, R.: Pattern Recognition Principles. Addison-Wesley, Massachusetts (1974)
12. Yuwono, B., Lee, D.: Search and Ranking Algorithms for Locating Resources on the World Wide Web. Proc. Fifth Int. Conf. on Databases & Systems for Advanced Applications (1997) 41-50