World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:12, 2008

A Weighted-Profiling Using an Ontology Base for Semantic-Based Search

Hikmat A. M. Abd-El-Jaber and Tengku M. T. Sembok

***Abstract*—**The information on the Web increases tremendously. A number of search engines have been developed for searching Web information and retrieving relevant documents that satisfy the inquirers needs. Search engines provide inquirers irrelevant documents among search results, since the search is text-based rather than semantic-based. Information retrieval research area has presented a number of approaches and methodologies such as profiling, feedback, query modification, human-computer interaction, etc for improving search results. Moreover, information retrieval has employed artificial intelligence techniques and strategies such as machine learning heuristics, tuning mechanisms, user and system vocabularies, logical theory, etc for capturing user's preferences and using them for guiding the search based on the semantic analysis rather than syntactic analysis. Although a valuable improvement has been recorded on search results, the survey has shown that still search engines users are not really satisfied with their search results. Using ontologies for semantic-based searching is likely the key solution. Adopting profiling approach and using ontology base characteristics, this work proposes a strategy for finding the exact meaning of the query terms in order to retrieve relevant information according to user needs. The evaluation of conducted experiments has shown the effectiveness of the suggested methodology and conclusion is presented.

***Keywords*—**information retrieval, user profiles, semantic Web, ontology, search engine.

I. INTRODUCTION

THE information on the Web increases tremendously [1] and produced what so called *information overload* [2]. The growth in Web information and users and the reasons behind users' dissatisfaction of information search and retrieval results are discussed in [3], [4]. A survey study has estimated that Web information is viewed by 1.023 billion people worldwide [5]. Therefore, it has been realized that a robust Web-based document retrieval system rather than data retrieval system is needed where both models are different in several aspects as presented in [6], [7].

Consequently, a number of search engines viewed as Web-based Information Retrieval (IR) systems have been developed as tools for searching Web information and finding the relevant documents that satisfy the inquirers needs. Google, for example, is a big name in Web searching which its technology has got underway in 1996. Reference [8] stated that, in the beginning, there was BackRub [9], [10], the service that became Google. Search engines are viewed in [11]–[14] among spectrum of interesting researches. The main components of these search systems namely crawling, indexing and ranking with emphasis on their algorithmic aspects are described in a survey presented by [15]. In the context of Web-based IR systems, the term *relevance* usually associated with documents ranking process. The more and higher appropriate ranking of the retrieved documents according to the user’s needs, the more and higher the relevance of the documents to the user’s preferences. History of the term relevance has been studied in [16] and has been defined by many researchers including [17], [18]. Some research works introduced new ways of measuring relevance [19], [20].

Current Web search systems retrieve topical relevant documents, but not relevant documents, to users. *Topical relevant* document, as defined by [20], is a document relevant to the query, not relevant to user needs. In reality a user issues a query to a search engine, as a result, the search engine provides the user the retrieved documents. The problem is irrelevant pages are presented to the user among this retrieved documents, since the search is text-based rather than semantic-based.

It is believed that a coherent and robust semantic-based Web IR system is needed. This is because the current text-based Web IR systems do not fulfill the user needs for Web searching and have not resolved the irrelevancy and inefficiency issues of retrieving information. Furthermore, these systems do not scale with the Web growth.

Many researchers assigned the problem to the adoption of traditional methods, algorithms, and techniques. For example, Yahoo!, which is one of the biggest search engines, uses subject classification method in categorizing the information and employed human experts for its implementation. Another example is that, some search engines use incoherent automated indexers that have not used agents. Even, those search engines that have used intelligent agents for Web pages crawling, indexing, and ranking, such as Google, have not established yet their consistency and efficiency in retrieving relevant information.

Syntactical analysis of the query terms is one of the primary reasons of this key problem. Indeed, the current Web search

H. A. M. Abd-El-Jaber is with the Al-Zaytoonah Private University of Jordan, Faculty of Information Technology, Computer Science department, P. O. Box 130 Amman 11733 Jordan (phone: 962-77-9701517; fax: 962-6-4291432; e-mail: dr.hekmat@alzaytoonah.edu.jo;hikmat_jaber@yahoo.com).

T. M. T. Sembok is with the National Defense University of Malaysia, Kem Sungai Besi, Kuala Lumpur 57000 Malaysia (e-mail: tmts@upnm.edu.my).

engines are keyword-based engines. The research realized that semantic-analysis of the query terms is needed in order to alleviate this issue; semantic-based search engines able to extract the exact meaning of the query terms intended by the user.

Currently, the information retrieval community attempts to provide some means to move from keyword-based to concept-based information retrieval utilizing ontologies as a reference for conceptual definitions. Reference [21] emphasized that the features of semantics of query terms, the description of document contents, and the user's interpretation of terms must be included in retrieval process to improve search result. Emerging these significant factors in designing an IR system assist producing personalized Web search according to each individual user's needs.

One of the key solutions to this issue is to use the exact meaning of the query in the search process. Having this purpose in mind, the researchers realized the significance of employing user profiles in the search process. For this, the prior work has emphasized on importance of user profiles in personalizing users search for developing effective IR systems. Maintaining an effective interaction between the user and the Web search system requires a flexible dynamic user profile. Therefore, a number of works have been done on profiling that has given a remarkable improvement on the search results.

In utilizing explicit user profiles to represent user's interests, some research works have employed machine learning [22] and knowledge base techniques in artificial intelligence technology [17]. Using software agents, [23] have taken into account both query and user preference (profile) in the IR process through introducing a user profile based on vector space model in the form of a retrieval function defined on basis of similarity function or a distance function between vectors. Other researchers explored XML features [24] and introduced a new relevance measurement for defining such XML-based profile [20]. Some research works attempted using the semantics of query terms by means of a correlations table and its associative tuning mechanism [25] and others utilized learning feature of agent technology [26] and used a learning algorithm to learn long-terms and short-terms (positive and negative) user's interests [27].

Studies and surveys have shown that users during their search are reluctant of providing any type of explicit feedback information even though users' feedback for documents are key factors to achieving better search results. Therefore, there were also a number of works tend to predict the information needs of users implicitly without any extra effort from them.

A variety of approaches and techniques are utilized for implicit profiling construction including: pure browsing history and modified collaborative filtering methods for capturing long term (persistent) and short term (ephemeral) user's preferences [2], clustering methods were user's interests (user profile) which is represented as a weighted concept heirarchy are populated by analyzing the user behavior in terms of the length of the visited Web page and the time spent on it while he is surfing the Web [28], user's browsing histories and user's searching histories for determining the weights of the content of a conceptual user profile represented also as a weighted concept heirarchy [29], and viewing recent Word documents and Internet Explorer Web pages for capturing contexual user interests which is then classified with respect to the Open Directory Project [30] ontology using the vector space model [31].

Although the research works adopted profiling approach and used artificial intelligence, software engineering, data mining, and other applications tools and techniques, for retrieving relevant information, the search engine survey has shown that users are still not satisfied with search results that match their interests and preferences.

Currently, it is believed that the research should start a new applicable direction for utilizing profiles. This direction requires new tools and methodologies that should have the capability of holding the exact meaning of the query terms that express the user needs. Fortunately, the emergence of ontology engineering has given the opportunity to model and develop such tools and methodologies. *Ontological Engineering* refers to the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies [32]. *Ontologies* are concepts tied together with semantic and joint relationships. Ontologies can be used to establish an ontology knowledge which in turn can be used to provide the semantics of its ontologies.

Ontologies and problem solving methods (PSMs) are complementary key factors that have been created to share and reuse knowledge and reasoning behavior across domains and tasks. These two factors have become key tools in developing the Semantic Web because the objective of Semantic Web is to give a well-defined meaning for information which can be achieved by using shared knowledge-components. Ontologies represent static domain knowledge and PSMs will be used inside Semantic Web services that model reasoning processes and deal with that domain knowledge as shown in Fig. 1. An important PSM component is its method ontology because it describes the concepts used by the method on the reasoning process as well as the relationships between such concepts [32].

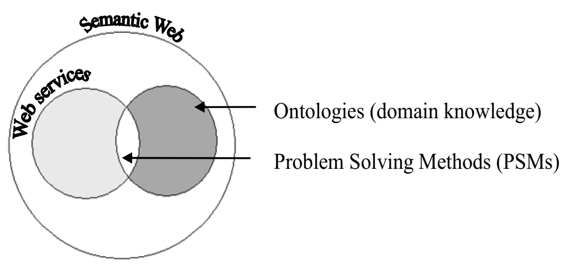

Fig. 1 Ontologies and problem-solving methods in relation with Semantic Web

The current Web content is formatted in HTML for human readers rather than programs. What is needed is information about Web content. The term *Metadata* refers to such information: data about data [33], [34]. Metadata capture part of the *meaning* of data, thus the term *Semantic* in Semantic Web. Semantic Web is an adopted technology to pursue integration, standardization, development of tools, and adoption by users for providing standard structure and semantics of information. The adoption of XML was an important first step for the realization of the semantic Web vision. XML introduces structure to Web documents, thus supporting syntactic interoperability [35].

Search agent in Semantic Web has different characteristics. The researchers should address the accessibility of the information that agent interacts with rather than concentrating on what constitute an agent, is agent intelligent, does the agent think? What characteristics the agent should possess? And so forth. However, in our point of view, it is not agent technology problem rather it is Web content accessibility. An agent is simply a program that does a specific task. This program possesses certain characteristics according to its functionality including logical inference and learning. Web search agent, for example, must be able to process the encountered information in order to browse them, extract the useful ones and offer the result to the user according to his requirements, rather than creating intelligent agent. Therefore, the question arises here is how to make the Web content processable.

Consequently, research community realized the importance of ontology features and capabilities for personalized search and therefore presented a variety of approaches and models to decrease search ambiguity and return relevant results that fits individual user need [36], [37]. For Web search personalization, Web mining area has developed a number of ontological user profiles to render search engines perform more intelligent search and retrieval tasks. In integrating ontology base user profiles into Web searching, [38] have developed Web user profile classified into two diagrams, the data diagram that discover the interest registration data and customer portfolios and the information diagram that discover the interest topics for Web user information needs. Based on user's information search intention and using Topic Ontology based user profile Model (TOM) to catch the user's attention topics, [39] have built user profile called topic ontology constructed from primitives' objects and includes the topic's semantic relationship. After finding out the user's search intent, [39] have adapted Pattern Taxonomy Model (PTM) developed by [40] to distinguish user intent (specificity intent from exhaustivity intent) by analyzing the user feedback and employed a method for assessment of relevance of topic ontology developed by [41] to let the system decide using which relevance function to assess whether the topic is relevant or not.

In [42], an ontology-based user model is proposed to represent user interests by means of personal ontology constructed from user semantic navigation sessions through monitoring his browsing habits. In another work carried by [43], an ontology base user profile is created consisting of concepts annotated with weights calculated based on an accumulated similarity score between the Web pages visited by a user and the concepts in a domain ontology. In [44], the hierarchical relationship among the concepts is also taken into consideration for building the ontological user profile which is updated and the annotations for existing concepts are modified by using spreading activation algorithm. This maintained user context is then utilized for Web search personalization by re-ranking the results returned from a search engine for a given query. In contrast to context model of [44], our approach semantically refines or reformulates the query before posting it into the search engine as we believe this step could assist representing the user context. This semantic-based refinement is done by expanding the initial query to include ontology base concepts (terms) that reflect the exact meaning of the query terms since both the query and the ontological user profile are mapped into the ontology base.

In a previous work, we studied Semantic Web and introduced our perspective of its use in user profiling [45], since we believe that organizing the Web content according to its meaning and extracting new knowledge through automated tools plays a vital role for the advance of knowledge management and efficient information retrieval. This advance in turn alleviates the limitations of the current technology structuring, searching, extracting, maintaining, uncovering, and viewing information and thus improving IR process.

Ontology is emerged recently as a new research area in computer science field. Using ontologies for semantic-based searching is likely the key solution. Adopting a weighted-profiling approach and using ontology base characteristics, this work proposes a strategy for finding the exact meaning of the query terms in order to retrieve relevant information according to user needs.

The objective of this paper is to personalize Web search for providing users relevant retrieved documents based on their issued queries and according to their needs. Therefore, we propose a semantic-based search strategy utilizing profiling associated with an ontology base, in order to shift search engines from location finders to information retrievals.

This paper is organized as follows. Section II suggests an ontology base for guiding profile-based search. A proposed approach for searching the Web semantically based on ontological profiles and query expansion based on the ontology base terms extracted using this approach are presented in section III. Experimental results and evaluation are reported in section IV and conclusion is given in the section V.

II. An Ontology Base for Guiding Profile-Based Search

Semantic-based search adopting profiling approach and using ontology knowledge requires a coherent search system. Developing such a comprehensive system is out of scope of this paper. This section mainly introduces the essential structural components of such a system that could help accomplishing the proposed strategy and improve search process. In our perspective, the search system entails combination of three basic related components. These components are query, profiles, and an ontology base. Compiling such a proposed system in each individual search engine could assist search engine for providing accurate results according to user needs.

A. Query

Queries are formal statements of information needs put to the IR system by users. Expanding or reformulating the issued query is concerned. This query expansion is done by extracting the ontology base concepts that represent the meaning of the query terms and then adding them to the initial query.

B. Profiles

Profiles model can be composed of two types: user profile and system profile. While *user profile* can be viewed as a central referential profile that keeps track of a particular user history and captures his interests, the *system profile* can be viewed as a secondary general profile that often always keeps track of all people history and maintains their preferences. The role of the system profile will be present in the absence of user profile role or when a user profile is unable to provide sufficient information to the search engine for making the decision on what relevant documents that must be retrieved. This means that, when a user issues a query and search engine starts retrieving documents, it should first look at user profile because user profile reflects and preserves the real user desires and if the search engine could not obtain the required information from the user profile then it should look at system profile. Several various cases may occur on searching the Web that make search engine unable to obtain the necessary information from the user profile for accomplishing its task. One case may occur when a user issues a query contains a term not included in the user profile. Another case may arise when a new user just start searching the Web and issues a query for the first time. In this case, user profile cannot provide deterministic information to the search engine for making a correct decision because it has no content since it is not yet constructed. A third case when a user issues a query contains a term having meaning different than its meaning in the existing profile. The aforementioned cases among others require a profile serves as a reference for search engine decision. Such profile is referred to as a system profile. However, this work is not concern of modeling an integrated view of profiles, rather it emphasizes on the semantic-based search strategy adopting profiling and referring to the word 'profile' means either user profile or system profile.

In this work, a user profile that maintains keywords and frequencies is utilized. The keywords represent the user preferences and the frequencies represent the weights of these keywords. I.e. each keyword has a *frequency number* that represent the number of occurrences of that keyword. Such frequencies can be used to provide information on how many times these keywords occur in queries issued by the user during his search history. Assigning weights to profile keywords and mapping these profile keywords into ontology base assist finding the exact meaning of user query terms. Fig. 2 shows the content of user profile. Obviously, a more high frequency number means more preferable and interesting keyword for the user. Adopting ontological weighted-profile, a semantic-based search strategy is proposed to interpret the exact meaning of query terms according to user preferences and extract the ontology base concepts that express this meaning for using them in query expansion to retrieve documents fit user needs.

User profile

Keywords	Frequencies
Keyword 1	frequency number of Keyword 1
Keyword 2	frequency number of Keyword 2
.	.
.	.
Keyword N	frequency number of Keyword N

Fig. 2 Content of user profile

Each frequency number can be converted to its equivalent weight value. Dividing each frequency number by the highest frequency number performs this conversion as illustrated with a simple example in section III. The calculated weights are not contained in user profile, but they are used in ontology base as firing weights for calculating the weights of the rest of ontology base domains as presented in section III. Based on the calculated weights of the entire ontology base domains, the search engine decides what relevant documents that must be retrieved.

Let us take an example illustrates the structure of user profile and how can be constructed in XML format. Suppose a user X issued three queries during his search history; the first query with the keyword 'Java', the second query with the keyword 'Math for Computer Science', and the third query with the keyword 'Information Retrieval'. Moreover, suppose user X issued these three keywords frequently for 12, 8, and 15 times respectively. Interests of user X are captured and maintained as preferences keywords in user profile. The keywords names and frequencies numbers can be represented in user profile as a simple XML format as follows:

```
<user profile>
  <X>
    <keywords>
      <Java>
        <frequency = "12"/>
      </Java>
      <Math for Computer Science>
```

<frequency = "8"/>
</Math for Computer Science>
<Information Retrieval">
<frequency = "15"/>
</Information Retrieval>
<\keywords>
<\X>
<\user profile>

C. *An Ontology Base*

Ontology base or ontology knowledge consists of terms tied together with a particular shared meaning in a hierarchical structure. For example, the keywords 'Pascal' and 'Java' are sharing common understanding since both are computer programming languages. Establishing an ontology base for semantic-based search is a critical issue since it is used as a criterion factor for determining the exact meaning of query terms. Constructing a well ontology base entails two primary things: first, the correct taxonomy of the terms or what is called *pedagogy ontology* [34] and second, the types of relationships between terms in a particular ontology base. For the first requirement, we should be accurate of which term belongs to which domain. For example, the keyword 'Java' should be categorized under 'computer programming languages' domain and 'coffee' domain. A right classification is an important factor for obtaining a correct result. For the second requirement, we should define the essential relationships that join the ontology base terms. For example, synonym relationship can join the two keywords 'Discrete Math' and 'Math for Computer Science'. Establishing a Directed Acyclic Graph (DAG) ontology base for semantic search engine adopted bottom-up approach and involved dynamically is based on two basic relationships: *SubDomain(x, y)* relationship that joins together the ontology base terms (including profile and query terms) in a hierarchical form and *Synonym(x, y)* relationship which defines the equivalence association between the meaning of at least two terms and this assists reducing redundancy and optimizing storage.

For the purpose of presenting the proposed strategy easily in section III, query terms are marked *italic*, profile terms are marked **bold**, and the nodes of a static-domain hierarchy ontology base which will be used as an example as shown in Fig. 3 are unmarked (i.e. normal style). In this simple example, the directed solid arrow is used to indicate the *SubDomain(x, y)* relationship that relates the sub domain x to its domain y, from bottom-up approach, and the undirected dashed line is used to show the *Synonym(x, y)* relationship that may arise to relate two or more terms in the domains hierarchical ontology base. Note that the directed dashed arrow functionality is same as the directed solid arrow functionality, which is used to indicate the *SubDomain* relationship but inferred from the *Synonym* relationship. For example, in the domains hierarchical ontology base of Fig. 3, 'Discrete Math' and 'Math for CS' are synonymous keywords. Therefore, 'Discrete Math' can be a sub domain of 'Computer Science' domain and 'Math for CS' can be sub domain of 'Math' domain.

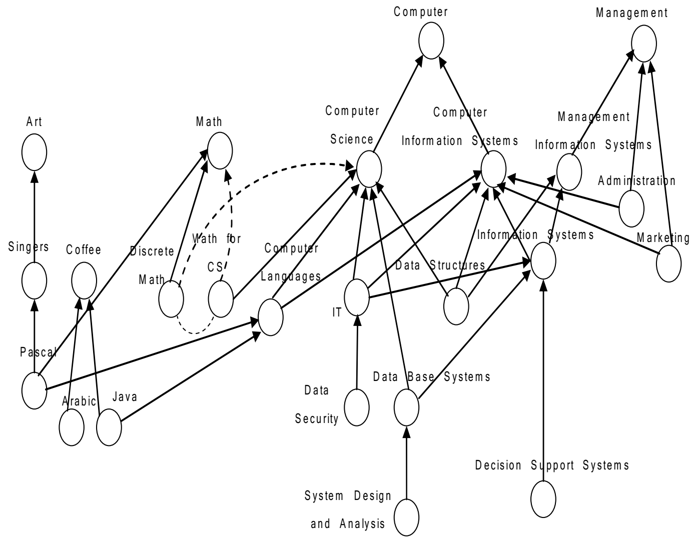

Fig. 3 An example: hierarchical ontology base domains at initial state

III. A Profile-Based Semantic Search Approach

This work adopts profiling approach and uses ontology knowledge for semantic-based searching in order to retrieve relevant information according to user needs. For this, a strategy using a methodology (algorithm) is proposed that receives *query* and **profile** as input and utilizes an associated ontology base for offering best output results as shown in Fig. 4.

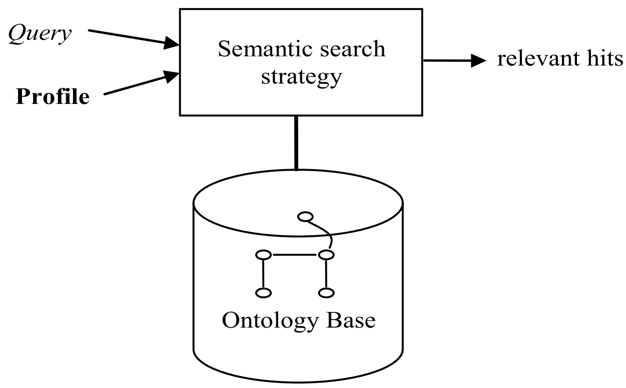

Fig. 4 General view of profiling using ontology base

This section suggests a methodology for semantic search in search engines guided by profiling based on ontology knowledge for pointing out the best upper domain that matches the terms of a user query. In addition, these pointed out best upper domains are used in query expansion by adding their terms to the initial query terms to form the final searchable query.

In a previous work, we presented a bivalent (two-valued decision or on/off) profiling using ontology base for a personalize search [45]. In this work, we present a weighted (multi-valued decision or approximate) profiling using ontology base for a personalize search. The weighted-profiling search strategy is viewed as a general case of the bivalent-profiling search strategy.

A. A Weighted-Profiling Semantic-Search Strategy

The strategy of a weighted-profiling semantic-search is based on multi-valued (approximate) decision for retrieving relevant information according to user needs. These weights of the ontology base domains are calculated based on the confidence weights associated with the user profile keywords. The confidence weight of each profile keyword is computed based on the frequency number assigned to this keyword during search process which represents how it is close to user interests. Moreover, the proposed strategy employs marked **profile** and *query* terms and uses ontology base for accomplishing its task. This strategy consists of four basic processing steps as depicted in Fig. 5.

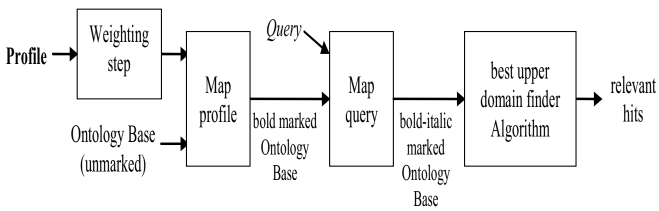

Fig. 5 The processing steps of a weighted-profiling semantic-search strategy

The first step is called *weighting step*. The weighting step calculates the weights of profile keywords and assigns them to the profile keywords in the ontology base. In fact, these calculated weights are firing weights and they represent the base for calculating the weights of the rest ontology base domains. As a result of first step, an ontology base with initial setting of profile keywords weights is produced. The second step maps the profile onto the initially weighted ontology base produced from step one. As a result, marked partially-weighted ontology base is produced in which all weighted profile keywords existing in the ontology base are marked **bold** and the other ontology base domains remain unmarked. The third step maps the query onto the marked partially-weighted ontology base that is produced from step two. Consequently, another marked partially-weighted ontology base is produced in which all query keywords existing in the ontology base are marked *italic*. So, after step three, we will have a hierarchical partially-weighted ontology base with bold and italic marks for profile and query keywords respectively and the rest domains are unmarked. The last step performs an algorithm called NUDA aims to find the best upper domain of the query keywords in the ontology base for providing best results.

For the purpose of describing the proposed strategy clearly, an example illustrates the processes of the steps is given. Assume that a user has a profile contains the keywords '**Pascal**', '**Java**', '**Decision Support Systems**', and '**Administration**' with their assigned frequencies and he issued a query keyword '*IT*' to a search engine for retrieving relevant hits according to his needs as given in Fig. 6. In addition, assume that the search engine contains a hierarchical ontology base as given in Fig. 3. The next subsections explain the processes steps of the strategy based on the assumed profile, query, and ontology base.

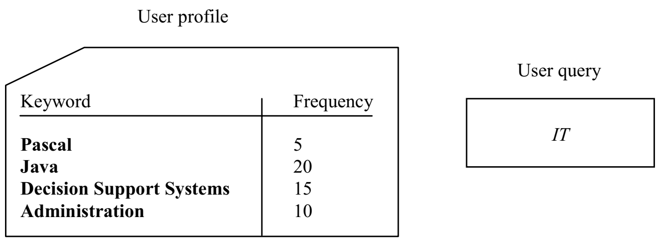

Fig. 6 An example: user profile and user query

1) The weighting step

The frequencies assigned to profile keywords are significant since they express the rate of user interests. The weighting step starts from these frequencies to calculate profile keywords weights. Calculating the weights of the initial keywords (i.e. the profile keywords) is performed by pointing out the highest frequency number and dividing each frequency number by this highest number. Carrying out this weighting step process on the assumed example given in Fig. 6; by dividing each frequency number by 20 since it is the highest frequency number, we get the following initial keywords and their weights:

Keyword	Weight
Pascal	5/20 = 0.25
Java	20/20 = 1.00
Decision Support Systems	15/20 = 0.75
Administration	10/20 = 0.50

These calculated initial weights are utilized by the proposed algorithm (NUDA) for calculating the weights of the remaining ontology base domains.

2) Profile Mapping onto Ontology Base

The second step maps the profile content onto the hierarchical ontology base. This means, the domains in the given ontology base which represent the profile content (i.e. '**Pascal**', '**Java**', '**Decision Support Systems**', and '**Administration**') are marked bold and the rest ontology base domains remain unmarked. This step results a marked ontology base with initial setting of profile keywords weights as shown in Fig. 7. These profile keywords represent firing keys and their weights represent firing weights that will be used for calculating the weights of the rest of ontology base domains.

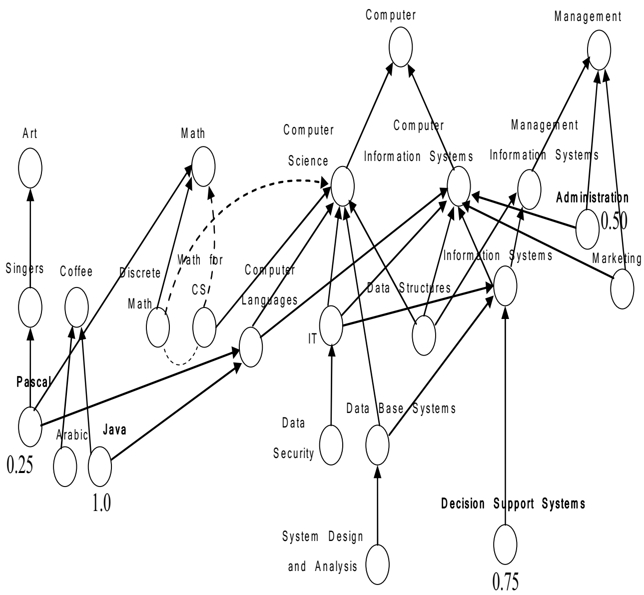

Fig. 7 Initially weighted ontology base with profile keywords marked bold

3) *Query Mapping*

The third step maps the query content onto the initially weighted marked ontology base which is produced from step two. Such mapping marks *italic* the ontology base domains which belong to the query. Therefore, in our given example, the strategy in this step marks italic the query keyword '*IT*' in the ontology base. Consequently, an initially weighted ontology base is produced with profile and query keywords marked bold and italic respectively and the rest domains are unmarked as shown in Fig. 8.

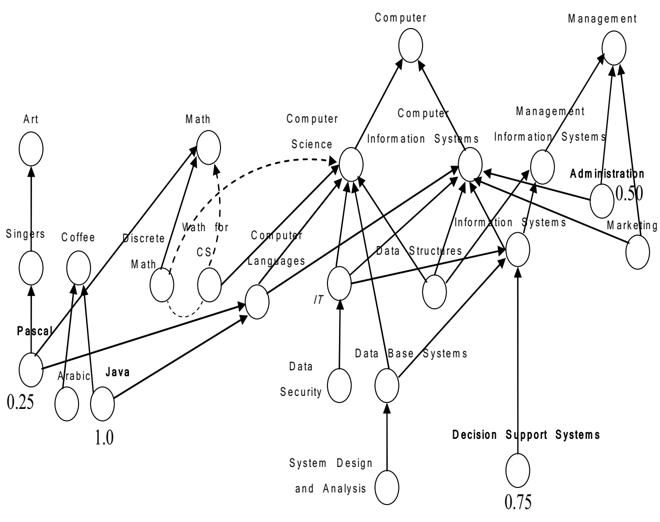

Fig. 8 Initially weighted ontology base with profile and query keywords marked bold and italic respectively

4) *Upper Domain Finding*

The last step performs a fuzzy-like algorithm on the initially weighted marked ontology base to find the best nearest upper domain in the ontology base that expresses the exact meaning of the user query keywords. Finding this best nearest upper domain assists the search engine for searching in this domain and thus providing best results to the user since it is based on semantic search and according to user preferences. To find the best upper domain in the ontology base, we need to calculate the weights of the rest ontology base domains. The weights calculated in the weighting step are assigned only to profile keywords in the ontology base. For calculating the weights of the rest ontology base domains and finding the best nearest upper domain of the query keyword in the ontology base, a *Nearest Upper Domain Algorithm* (NUDA) is proposed. This algorithm calculates the weights starting from the initial firing keys (profile keywords in the ontology base) and excludes from calculation the sub domains in the ontology base that are lower of the query keyword, since it concerns calculating only upper domains. In fact, calculating the weights of lower sub domains of the issued query keyword in the ontology base is meaningless because lower sub domains belong to their upper domains but vice-versa is not true.

The NUDA uses a threshold value 0.5 as a distance factor that helps calculating the weights of the keywords (domains) in the ontology base. Starting from bold marked weighted keywords, NUDA applies the following homogeneous formulas for calculating the weights of the upward and downward keywords:

For calculating weights of upward keywords

Weight of current keyword k can be calculated as follows:

$$\text{Keyword weight} = \frac{1}{2} * \text{Max}\,\{\text{weights of all its children}\}$$

$$Weight_k = \frac{1}{2} * \text{Max}_{\forall\ \text{child}\ i}\,\{weight_i\} \quad (1)$$

For calculating weights of downward keywords

Weight of current keyword k can be calculated as follows:

$$\text{Keyword weight} = \frac{1}{2} * \text{Max}\,\{\text{weights of all its parents}\}$$

$$Weight_k = \frac{1}{2} * \text{Max}_{\forall\ \text{parent}\ j}\,\{weight_j\} \quad (2)$$

For making this point more clear, we take two examples from the ontology base shown in Fig. 9. The first example illustrates calculating the weight of the upward keyword 'Computer Languages' while the second example illustrates calculating the weight of the downward keyword 'Data Structures'.

Example 1: To calculate the weight of the keyword 'Computer Languages':

Weight of the keyword 'Computer Languages'

= 0.5 * Max {weights of all 'Computer Languages' children}
= 0.5 * Max {weight of 'Pascal', weight of 'Java'}
= 0.5 * Max {0.25, 1.0}
= 0.5 * 1.0
= 0.5

Example 2: To calculate the weight of the keyword 'Data Structures':

Weight of the keyword 'Data Structures'
= 0.5 * Max {weights of all 'Data Structures' parents}
= 0.5 * Max { weight of 'Computer Science',
weight of 'Computer Information Systems',
weight of 'Management Information Systems'
}
= 0.5 * Max {0.25, 0.25, 0.1875}
= 0.5 * 0.25
= 0.125

Applying aforementioned formulas for calculating the weights of all ontology base domains, we get a completely weighted marked ontology base as depicted in Fig. 9. For accomplishing this task, the next subsection performs NUDA step-by-step, where each step calculates the weights of the keywords at the same level.

a) Steps of Nearest Upper Domain Algorithm (NUDA)

NUDA starts from the initial firing keys to calculate the weights of the rest of ontology base keywords. In fact, these keys are profile keywords that are mapped onto ontology base after calculating their weights in the weighting step. The keywords that their weights are calculated will become firing keys and will involve in calculating non calculated keywords, and so on. In our ontology base example, the firing keys are '**Pascal**', '**Java**', '**Decision Support Systems**', and '**Administration**' and their corresponding weights are 0.25, 1.0, 0.75, and 0.5 respectively as shown in Fig. 8.

The NUDA first calculates the weights of upward domains. If no more upward domains are left, then NUDA starts calculating the weights of downward domains. The mechanism of calculating upward or downward domains is the same. This mechanism starts from the firing keys and divides the entire ontology base domains into sets of keywords. Each set consists of keywords that are in the same level. Finally, the weights of the keywords are calculated set-by-set, first in the upward direction and then in the downward direction. Although the mechanism is the same, the formula used for calculating upward domains slightly differs from the formula used for calculating downward domains as seen in formulas (1) and (2).

Step 1:

Step 1 calculates the weights of the keywords of the first set which is just one level up from the firing keys. The keywords in this set are 'Singers', 'Coffee', 'Math', 'Computer Languages', 'Computer Information Systems', 'Information Systems', and 'Management'. The weights of these keywords are calculated using the given upward formula (formula 1) as follows:

Weight of the keyword 'Singers'
= 0.5 * Max {0.25} = 0.5 * 0.25 = 0.125
Weight of the keyword 'Coffee'
= 0.5 * Max {null, 1.0} = 0.5 * 1.0 = 0.5
Weight of the keyword 'Math'
= 0.5 * Max {0.25, null, null} = 0.5 * 0.25 = 0.125
Weight of the keyword 'Computer Languages'
= 0.5 * Max {0.25, 1.0} = 0.5 * 1.0 = 0.5
Weight of the keyword 'Computer Information Systems'
= 0.5 * Max {0.5, QK, null, 0.375, null, 0.5}
= 0.5 * 0.5 = 0.25
Weight of the keyword 'Information Systems'
= 0.5 * Max {QK, null, 0.75} = 0.5 * 0.75 = 0.375
Weight of the keyword 'Management'
= 0.5 * Max {null, 0.5, null} = 0.5 * 0.5 = 0.25

In the above calculations, the notation null is a weight value assigned to any keyword that has no weight value at the time of involving it in calculating the weight of other keywords. For instance, in the given ontology base, a null weight value is assigned to the keyword 'Arabic' since it has no weight value at the time of its involvement in calculating the weight of its domain 'Coffee'. Another notation is QK which refers to Query Keyword (here in this example is *IT*) and therefore it has no weight value.

At completion of step 1, we get upward ontology base domains belong to the first set which is one level up from the initial firing keys indicated by their calculated weights as shown in Fig. 9. Moreover, these domains will become firing keys for subsequent calculations.

Step 2:

Step 2 repeats the procedure of step 1 but on different set of keywords. Step 2 handles the second set of keywords which is exactly two levels up from the initial firing keys. This second set includes the keywords: 'Art', 'Computer Science', 'Computer', and 'Management Information Systems'. The weights of these keywords are calculated using the given upward formula (formula 1) as follows:

Weight of the keyword 'Art'
= 0.5 * Max {0.125} = 0.5 * 0.125 = 0.0625
Weight of the keyword 'Computer Science'
= 0.5 * Max {null, null, 0.5, QK, null, null} = 0.5 * 0.5 = 0.25
Weight of the keyword 'Computer'
= 0.5 * Max {0.25, 0.25} = 0.5 * 0.25 = 0.125
Weight of the keyword 'Management Information Systems'
= 0.5 * Max {null, 0.375} = 0.5 * 0.375 = 0.1875

At completion of step 2, we get upward ontology base domains belong to the second set which is two levels up from the initial firing keys indicated by their calculated weights as shown in Fig. 9. In addition, these domains will become firing keys for subsequent calculations. Although, the domains

'Arabic', 'Discrete Math', 'Math for CS', 'Data Base Systems', 'Data Structures', and 'Marketing' are just two units far from the nearest initial firing keys, they are excluded from the second set, because they are downward domains and still upward domains are not completely covered before processing step 2.

Step 3:
All upward domains are covered after completing step 2 processes. So, as we mentioned earlier that if no more upward domains are left then NUDA starts taking the downward domains for calculating their weights. Therefore, step 3 handles the third set of keywords which is in downward direction and includes the keywords: 'Arabic', 'Discrete Math', 'Math for CS', 'Data Base Systems', 'Data Structures', and 'Marketing'. Using the given downward formula (formula 2), weights of the third set keywords are calculated as follows:

Weight of the keyword 'Arabic'
= 0.5 * Max {0.5} = 0.5 * 0.5 = 0.25

Weight of the keyword 'Discrete Math'
= 0.5 * Max {0.125, 0.25} = 0.5 * 0.25 = 0.125

Weight of the keyword 'Math for CS'
= 0.5 * Max {0.125, 0.25} = 0.5 * 0.25 = 0.125

Weight of the keyword 'Data Base Systems'
= 0.5 * Max {0.25, 0.375} = 0.5 * 0.375 = 0.1875

Weight of the keyword 'Data Structures'
= 0.5 * Max {0.25, 0.25, 0.1875} = 0.5 * 0.25 = 0.125

Weight of the keyword 'Marketing'
= 0.5 * Max {0.25, 0.25} = 0.5 * 0.25 = 0.125

Observe that the sub domain 'Discrete Math' has a *SubDomain* relationship with the domain 'Math' (directed solid arrow) and with the domain 'Computer Science' (directed dashed arrow). The relationship *SubDomain(Discrete Math, Computer Science)* originally is not existing, rather it is inferred since 'Discrete Math' and 'Math for CS' are synonymous; means both are related by the relationship *Synonym(Discrete Math, Math for CS)*. Again, the same is true for the relationship *SubDomain(Math for CS, Math)*. It is important to mention that, synonymous keywords are not necessarily having the same weight value. At completion of step 3, we get downward ontology base domains belong to the third set indicated by their calculated weights as shown in Fig. 9. Clearly, these domains will become firing keys for future calculations.

Step 4:
The fourth set contains only one keyword since the only domain left is 'System Design and Analysis'. The weight of this keyword is calculated in this step using the downward formula (formula 2) as follows:

Weight of the keyword 'System Design and Analysis'
= 0.5 * Max {0.1875} = 0.5 * 0.1875 = 0.09375

At completion of step 4, we get the only downward ontology base domain belongs to the forth set indicated by its calculated weight as shown in Fig. 9.

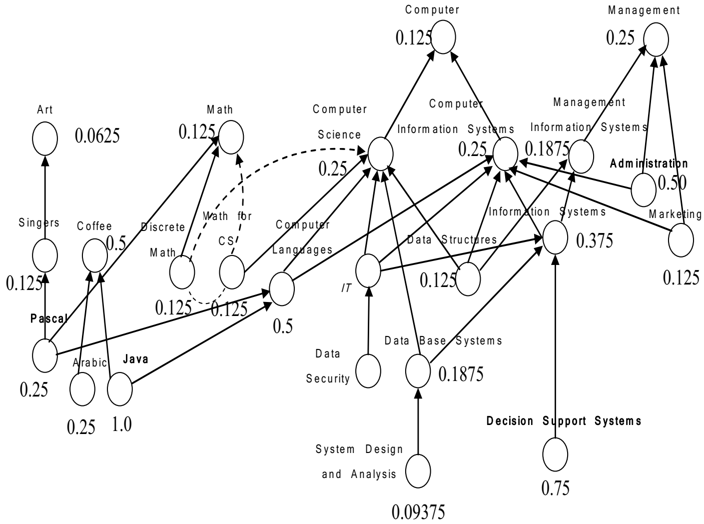

Fig. 9 Step 1 through step 4: weighting all levels of ontology base

With completing the step 4 process, all ontology base domains are labeled with certain weight values as given in Fig. 9, except those are sub domains of the query keyword which are not considered by the proposed strategy. Finally, NUDA performs decision step to find the best upper domain of the query keyword meant by the user.

Decision step:
The resulted labeled ontology base in Fig. 9 shows that there are three upper domains of the query keyword '*IT*'. These upper domains are 'Computer Science', 'Computer Information Systems', and 'Information Systems' with weights 0.25, 0.25, and 0.375 respectively. Clearly, the highest weight value among them represents the best upper domain that reflects how close it is to the meaning of the issued query keyword. Here, the highest weight value is 0.375 which is the value of the domain 'Information Systems' as shown in Fig. 10. Therefore, for the given profile and query, the search engine should retrieve and list the documents that belong only to the domain 'Information Systems' and avoid providing the documents from the two domains 'Computer Science' and 'Computer Information Systems'. We believe that the result is reasonable since the methodology has taken into account the user preferences through mapping profile contents to the ontology base. Such mapping provides a realistic results since analysis of query keywords is semantic-based rather than text-based.

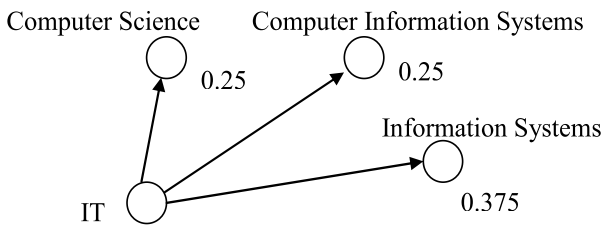

Fig. 10 Decision step: the query keyword and all its related weighted ontology base domains

The proposed strategy attempts finding the relevant documents based on the meaning of the keywords of the issued query and according to user preferences (i.e. profile keywords). Moreover, the strategy modifies the profile after completing its processes. This modification is done depending on the case. In case of the keyword of the issued query does already exist in the profile then the modification is done by incrementing the number of frequency by 1. On the other hand, if the keyword of the issued query does not exist in the profile, then the modification is done by including the query keyword and its frequency number into the profile. The number of frequency of included query keyword for the first time of searching should be set to 1, and each time the user search for this query keyword is increment by 1. For instance, in the presented example, the user issued the query keyword ‘IT’ for the first time; therefore this keyword does not exist in the profile. After implementation of the NUDA, the strategy should modify the profile by including the query keyword ‘IT’ with its associated frequency value 1 into the profile as shown in the Fig. 11. Each time the user issues a query with the keyword ‘IT’, its frequency number will be incremented by 1.

User profile

Keyword	Frequency
Pascal	5
Java	20
Decision Support Systems	15
Administration	10
IT	1

Fig. 11 The modified profile

B. *Query Expansion*

In searching the Web, usually, the initial query may not reveal the user preferences. One way to overcome this problem is to expand the original query. In IR, other methods can be used including filtering and re-ranking. This study adopts query expansion to reflect user interests by simply adding the terms that are extracted from the ontology base to the initial query. This expanded query represents the user desire since it contains terms from the user's interested domain(s). For example, the decision step of the previous section decided that the preferable domain of the query 'IT' is ‘Information Systems’, since it has the highest weight value, therefore, the initial query 'IT' would be expanded to: 'IT' AND ‘Information Systems’.

IV. Experimental Results and Evaluation

An experimental personalized Web software system is built by using JBuilder® 2007. This developed system can be viewed as a primary integral part of any search engine, because it aims to associate the ontology base to the search engine, construct user profile to maintain the user preferences, perform the weighted-profiling strategy including the NUDA algorithm to point out the user preferences from the associated ontology base, and finally use the knowledge information that are extracted from the ontology base for expanding the query.

The system enables the user to create his profile before starting his search sessions. This step is carried out only once for each user. Once user profile is created, the user can start searching the Web frequently by using the search engine. The search engine receives the user query which in turn runs the NUDA to extract the exact meaning of the inputted query terms from the ontology base associated to the search engine. The extracted information is simply the upper domains that express the meaning of the inputted user query terms. Since these extracted upper domains represent the user interests, we employ them in expanding the user query.

Query expansion in this research work relies on the inputted query term. If the inputted query term has shared understanding with several distinct root domains (i.e. it has different meanings for several distinct root domains), then, the query is reformulated by appending the terms of its best upper domain(s) to the initial query term to form the expanded query. On the other hand, if the inputted query term has no shared understanding with other root domains (i.e. all the meanings of the term belongs to just one root domain), then, the query remains unchanged. The expanded query forms the final query which will then be entered in the search engine to retrieve the desired documents.

Let us illustrate how query expansion is performed with a real example of a query inputted into the Google search engine in the real experiment field. Two different users having two distinct domains of interest are chosen. The first user chosen is interested in computer domain, whereas the other user chosen is interested in mathematics domain. Both users are asked to inquire the search engine for the query "what is topology". Two factors have been taken into account for choosing query terms. First, the term should be an ontology base domain. Second, the term should have shared understanding with other domains in the ontology base. The query "what is topology" is chosen to be input by the two users because it satisfies both factors. In one sense, the term 'topology' is an ontology base term. In the other sense, the term 'topology' has common meaning with several distinct domains since it belongs to *mathematics* domain, or to computer network sub domain in *computer* domain, or to geographic information systems sub domain or topological map sub domain in *geography* domain, or to musical ensemble sub domain in *art* and *entertainment* domains, or to geomorphology sub domain in *geography*, *geology* and *space* domains etc. It is clear that, for the first user who is interested in computer domain, the query "what is topology" means 'computer network' whereas for the second user who is interested in mathematics domain, it means topology in 'mathematics'. The proposed strategy is able to extract the exact domain of the query term based on the user profile history. Therefore, when the first user posed the query, the ontology base term 'computer network' is extracted as the best upper domain that reflects his domain of interest. This extracted ontology base term is then added to the initial query to form the final query as "What is topology" + "computer network". For the second user, when he posed the same query (i.e. "What is topology"), the ontology base term 'mathematics' is extracted as the best upper domain that reflects his domain of interest and consequently the final query becomes "What is topology" + "mathematics". For any user of any domain of interest, the reformulated query is finally used for inquiring the search engine. During experiments, users observed and reported that, including user interested domains in the query and excluding uninterested domains, improve the precision of result hits. For example, the precision of the query "What is topology" is improved by 41% when it is expanded to "What is topology" + "computer network" for the first user who is interested in computer domain and 12% when it is expanded to "What is topology" + "mathematics" for the second user who is interested in mathematics domain.

An experimental ACM Topic hierarchy had 1,215 topics collected from Lehigh University (http://swat.cse.lehigh.edu/resources/conftrack/topic/ACMTopics.owl) as a concept hierarchy, since building an ontology base is out of scope of this paper. Initially, the ACM topic hierarchy was intended for computer courses domain. We modify the content of the ACM topic hierarchy to serve the purpose of this work. Modifications have been taken place to include topics (terms) not just from computer science domain but also from other domains such as biology, mathematics, geography, etc. Finally, the experimental concept hierarchy contained 609 terms which are tied in vertical and horizontal dimensions cross domains using subClass relationship.

In this study, the experiments are carried out by extracting the concept hierarchy terms that represent the exact meaning of the query terms for using them in query expansion. Extraction of such useful terms from the hierarchy is done implicitly without any effort from the user. In this context, user profiles assist maintaining the extracted terms.

We evaluate the proposed strategy to examine its effectiveness in retrieving relevant information. Experiments are conducted in a laboratory environment where 10 users interested in three different domains (4 users from computer domain and 3 users from each biology and mathematics domains) are employed to search the Web using Google search engine. Each user according to his interested domain is asked to query the Google search engine twice; the first time just using Google search engine without employing our proposed search strategy while the second time using Google search engine with employing our proposed search strategy. Query terms entered by the users should be selected from the experimental concept hierarchy terms.

The effectiveness of weighted-profiling semantic-search strategy is measured in terms of cut-off points and precision rather than recall points and precision because we cannot really calculate the normal recall points since the number of relevant documents in Google collection is unknown. Cut-off points are made here for the first 150 documents of the search engine hits. Recall that precision is the ratio between the relevant documents retrieved to the total number of retrieved documents. Precision values are calculated at cut-off (the first) 10 documents, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, and 150 documents. In this way, the cut-off points and precision are for a single query. However, to evaluate our retrieval algorithm accurately, we run it for several distinct queries, and an average is used for the cut-off and precision figures.

The relevancies of the retrieved documents are judged by the users by examining how close the document to the inputted query term and how far the document represents the user interested domain at that search session time. We emphasize that the user may alter his interested search domain at any given time. This point has been taken under consideration and accordingly the system should adjust the user's new interested search domain. The users of the experiment tests reported that a considerable number of documents irrelevancy are due to providing a significant rank to the subscription-based commercial or educational Web sites such as business companies, bookstores, research journals, etc.

The experimental trial had been conducted for 40 days by 10 users where each user inputted at least 10 queries. For each query, a cut-off/precision curve is drawn. These drawn curves are averaged to produce the final cut-off/precision as shown in Fig. 12. The figure illustrates that semantic-based IR is better than text-based IR. Experiments recorded that about 23% improvement of weighted-profiling semantic-based search method over the current text-based search methods.

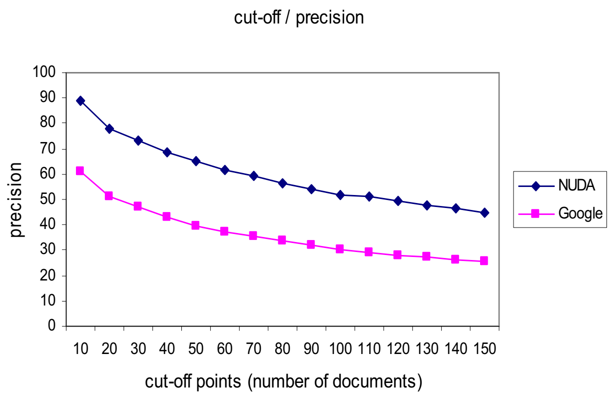

Fig. 12 Precision values at cut-off points for NUDA and Google

V. Conclusion

Integrating ontological user profiles into the processing can be very beneficial for Web search personalization and thus improving the information retrieval effectiveness. Issuing a query to a search engine for retrieving relevant hits according to user preferences can provide better result if the search system can find the exact meaning of the query keywords. This semantic-based searching can be performed by adopting profiling approach and using ontologies. The weighted-profiling (multi-valued decision) semantic-based search shows a considerable improvement than text-based search in terms of search effectiveness. Experiment tests reported that 23% improvement of profiling approach semantic-based search over text-based search. This demonstrates that the user profiles based on the semantic search using ontology base can improve information retrieval performance. For further work, investigating a weighted-profiling semantic-based search vertically within a particular domain after finding out the user domain of interest horizontally can assist providing better search result, since query keyword could have several different meanings even in the same vertical domain. Moreover, adding a well re-ranking method to the proposed search strategy for placing the most relevant documents at the top of the hits list can refine the result. Furthermore, the integration of a system profile that keeps all people view and maintain their interests with a personalized user profile can play a vital role in improving Web search effectiveness.

REFERENCES

[1] Internet Systems Consortium (2006). ISC Internet Domain Survey. [Online]. Available: http://www.isc.org/ops/ds/reports/2006-01

[2] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive web search based on user profile constructed without any effort from users," in *Proc. 13th ACM International Conference on World Wide Web, WWW2004*, pp. 675–684, 2004.

[3] B. Huberman and R. Lukose, "A metasearch engine that learns which search engine to query," *Science* 277, pp. 535–537, 1997.

[4] M. Kobayashi and K. Takeda, "Information Retrieval on the Web," *ACM Computing Surveys (CSUR)*, vol. 32, no. 2, pp. 144–173, 2000.

[5] Internet World Stats: Usage and Population Statistics (2006). Internet Growth Statistics. [Online]. Available: http://www.internetworldstats.com/emarketing.htm

[6] V. C. J. Rijsbergen, *Information Retrieval* (2nd ed.). Computer Laboratory, University of Cambridge. Butter Worths, London, 1979, pp. 1–12. [Online]. Available: http://www.dcs.gla.ac.uk/~iain/keith/

[7] D. C. Blair, "The data-document distinction revisited," *ACM SIGMIS Database*, vol. 37, no. 1, pp. 77–96, 2006.

[8] E. A. Stephen, *The Google Legacy: How Google's Internet Search is Transforming Application Software*. Tetbury, England, 2005, Chapter 3: Google Technology. [Online]. Available: http://www.infonortics.com/publications/google/technology.pdf

[9] Google Corporate Information, *Google Milestones*, 2006, [Online]. Available: http://www.google.com/corporate/history.html

[10] J. Battelle, (2005, August). The Birth of Google, *Wired Magazine*, Issue 13.08. [Online]. Available: http://www.wired.com/wired/archive/13.08/battelle.html

[11] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engines," in *Proc. 7th International World Wide Web Conference, Computer Networks and ISDN Systems*, 1998, vol. 30, no. 1-7, pp. 107–117.

[12] W. Meng, Z. Wu, C. Yu, and Z. Li, "A highly scalable and effective method for metasearch," *ACM Transactions on Information Systems (TOIS)*, vol. 19, no. 3, pp. 310–335, 2001.

[13] K. C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, "Structured databases on the Web: Observations and implications," *ACM SIGMOD*, vol. 33, no. 3, pp. 61–70, 2004.

[14] SearchEngineWatch.com. (2005). Metacrawlers and Metasearch Engines. [Online]. Available: http://searchenginewatch.com/links/article.php/2156241

[15] A. López-Ortiz, "Search engines and Web information retrieval," *Lecture Notes in Computer Science LNCS*, Springer-Verlag Berlin/Heidelberg, vol. 3405/2005, pp. 183–191, 2005.

[16] S. Mizzaro, "Relevance: the whole history," *Journal of the American Society for Information Science (JASIS)*, 48(9), pp. 810–832, 1996.

[17] S. H. Myaeng and R. R. Korfhage, "Towards an intelligent and personalized retrieval system," in *Proc. of the ACM SIGART International Symposium on Methodologies for Intelligent Systems*, Knoxville, Tennessee, United States, pp. 121–129, 1986.

[18] W. Meng, C. Yu, and K. Liu, "Building efficient and effective metasearch engines," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 48–89, 2002.

[19] G. G. Chowdhury, *Introduction to Modern Information Retrieval*. Library Association Publishing, London, 1999.

[20] B. V. Gils and E. D. Schabell, "User-profiles for information retrieval," in *Proc. 15th Belgian-Dutch Conference on Artificial Intelligence (BNAIC'03)*, Nijmegen, Netherlands, 2003. [Online]. Available: http://citeseer.ist.psu.edu/vangils03userprofiles.html

[21] H. B. Styltsvig, "Ontology-based information retrieval," Ph.D. dissertation, Dept. Comp. Sc., Roskilde University, Denmark, 2006.

[22] S. K. Bhatia, J. S. Deogun, and V. V. Raghavan, "User profiles for information retrieval," in *Proc. 6th International Symposium on Methodologies for Intelligent Systems (ISMIS)*, Springer-Verlag Berlin / Heidelberg, 1991, Vol. 542, pp. 102–111.

[23] C. Danilowicz and H. C. Nguyen, "Using user profiles in intelligent information retrieval," in *Proc. 13th International Symposium on Methodologies for Intelligent Systems (ISMIS), LNAI*, Springer-Verlag Berlin/Heidelberg, 2002, Vol. 2366, pp. 223–231.

[24] B. V. Gils, E. Proper, and P. V. Bommel, "Towards a general theory for information supply," in *Proc. 10th International Conference on Human-Computer Interaction*, 2003.

[25] P. M. Chen and F. C. Kuo, "An information retrieval system based on a user profile," *ACM, Journal of Systems and Software*, 54(1), pp. 3–8, 2000.

[26] D. H. Widyantoro, J. Yin, M. Seif El-Nasr, L. Yang, A. Zacchi, and J. Yen, "Alipes: A swift messenger in cyberspace," in *AAAI'99 Spring Symposium on Intelligent Agent in Cyberspace*, pp. 62–67, 1999.

[27] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "An adaptive algorithm for learning changes in user interests," in *Proc. 8th International Conference on Information and Knowledge Management CIKM '99*, 1999, pp. 405–412.

[28] A. Pretschner and S. Gauch, "Ontology based personalized search," in *Proc. 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 1999, pp. 391–398.

[29] S. Gauch, M. Speretta, and A. Pretschner, "Ontology-based user profiles for personalized search," *Integrated Series in Information Systems*, SpringerLink, Vol. 14, pp. 665–694, 2007.

[30] The Open Directory Project (ODP), 2004. [Online] Available: http://dmoz.org

[31] V. Challam, "Ontology-based user profiles for contextual search," M.S. thesis, Kansas Univ., Lawrence, KS, Germany, 2004.

[32] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering* (2nd ed.). Springer-Verlag London Limited, England, 2004, pp. 1–5.

[33] B. Thuraisingham, *XML Databases and the Semantic Web* (1st ed.). CRC Press, USA, 2000, pp. 109–111.

[34] G. Antoniou and G. van Harmelen, *Semantic Web Primer* (1st ed.). The MIT press, Cambridge, Massachusetts, London, 2004, pp. 7–9; 193–194.

[35] J. T. Pollock and R. Hodgson, *Adaptive information: Improving business through semantic interoperability*, Grid Computing, and Enterprise Integration, (Wiley Series in Systems Engineering and Management), Wiley-Interscience, 2004.

[36] A. Singh and K. Nakata, "Hierarchical classification of Web search results using personalized ontologies," in *Proc. 3rd International Conference on Universal Access in Human-Computer Interaction, HCI International 2005*, Las Vegas, NV, 2005.

[37] F. Liu, C. Yu, and W. Meng, "Personalized Web search for improving retrieval effectiveness," *IEEE Transactions on Knowledge and Data Engineering*, 16 (1), pp. 28–40, 2004.

[38] Y. Li and N. Zhong, "Mining Ontology for automatically acquiring Web user information needs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 554–568, 2006.

[39] X. Zhou, S. T. Wu, Y. Li, Y. Xu, R. Lau, and P. Bruza, "Utilizing search intent in topic ontology-based user profile for Web mining," in *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, 2006, pp. 558–564.

[40] S. T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen, "Automatic pattern-taxonomy extraction for Web mining," presented at the 2004 IEEE ACM International Conference on Web Intelligence, WIC, Beijing, China, 2004.

[41] X. Zhou, Y. Li, Y. Xu, and R. Lau, "Relevance assessment of topic ontology," presented at the Fourth International Conference on Active Media Technology. Brisbane, Australia, 2006.

[42] H. Zhang, Y. Song, and H. T. Song, "Construction of ontology-based user model for Web personalization," *User Modeling (UM'07), LNCS*, Springer-Verlag Berlin/Heidelberg, vol. 4511, pp. 67–76, 2007.

[43] J. Trajkova and S. Gauch, "Improving ontology-based user profiles," in *Proc. of the Recherche d'Information Assist e par Ordinateur. RIAO 2004*, Vaucluse, France, 2004, pp. 380–389.

[44] A. Sieg, B. Mobasher, and R. Burke, "Representing context in Web search with ontological user profiles," *CONTEXT 2007, LNAI*, Springer-Verlag Berlin/Heidelberg , vol. 4635, pp. 439–452, 2007.

[45] H. AbdEl-el-Jaber and T. Sembok, "A bivalent-profiling using an ontology base for semantic-based search," JASIST, submitted for publication.