

An Approach to Improve the Representation of the User Model in the Web-Based Systems

Yasser A. Nada

Computer Science Dept., College of Computer Science
and Information Technology,
Taif University, Kingdom of Saudi Arabia.

Khaled M. Fouad

Computer Science Dept., College of Community,
Taif University, Kingdom of Saudi Arabia

Abstract—A major shortcoming of content-based approaches exists in the representation of the user model. Content-based approaches often employ term vectors to represent each user's interest. In doing so, they ignore the semantic relations between terms of the vector space model in which indexed terms are not orthogonal and often have semantic relatedness between one another.

In this paper, we improve the representation of a user model during building user model in content-based approaches by performing these steps. First is the domain concept filtering in which concepts and items of interests are compared to the domain ontology to check the relevant items to our domain using ontology based semantic similarity. Second, is incorporating semantic content into the term vectors. We use word definitions and relations provided by WordNet to perform word sense disambiguation and employ domain-specific concepts as category labels for the semantically enhanced user models. The implicit information pertaining to the user behavior was extracted from click stream data or web usage sessions captured within the web server logs.

Also, our proposed approach aims to update user model, we should analysis user's history query keywords. For a certain keyword, we extract the words which have the semantic relationships with the keyword and add them into the user interest model as nodes according to semantic relationships in the WordNet.

Keywords—User model; Domain ontology; Semantic Similarity; Wordnet.

I. INTRODUCTION

User model [1] is a collection of personal information. The information is stored without adding further description or interpreting this information. It is comparable to a getting-setting mechanism of classes in object-oriented programming, where different parameters are set or retrieved. User model represents cognitive skills, intellectual abilities, intentions, learning styles, preferences and interactions with the system. These properties are stored after assigning them values. These values may be final or change over time.

The Semantic Web [2] “transforms the Web by providing machine understandable and meaningful descriptions of Web resources”. Making the Web content machine understandable, allowing agents and applications to access a variety of heterogeneous resources, processing and integrating the content, and producing added value for the user. Data on the

Web must be defined and linked in a way that can be used for more effective discovery, automation, integration, and reuse across various applications.

The personalization aspects [3] of the user interests or profiles can form a good representation of the learning context, which promises to enhance the usage of learning content. The key knowledge nugget in any personalization strategy for e-learning is an accurate user model. User Modeling is an active research area in e-learning and personalization, especially when abstracting the user away from the problem an abstraction that has, over the years, contributed to the design of more effective e-learning systems. Despite this improvement, the main focus in most systems, for the past decade, has been on models that are “good for all users”, and not for a specific user.

Our proposed approach is to propose improvements in the representation of a user model during building user model in content-based approaches by performing the next steps. First step is domain concept filtering in which concepts and items of interests are compared to the domain ontology to check the relevant items to our learning domain using ontology based semantic similarity. Second step is incorporating semantic content into the term vectors. We use word definitions and relations provided by WordNet to perform word sense disambiguation and employ domain-specific concepts as category labels for the semantically enhanced user models. The implicit information pertaining to the user behavior was extracted from click stream data or web usage sessions captured within the web server logs. The method of representing semantic user model was proposed in [4].

Also, our proposed approach update user model, we should analysis learner's history query keywords. For a certain keyword, we extract the words which have the semantic relationships with the keyword and add them into the user interest model as nodes according to semantic relationships in WordNet. The method of updating user model was proposed in [5, 6].

II. RELATED WORKS

In [7], authors proposed an idea of adaptation using semantic web techniques with reduced cost of user profile acquisition. Cost-effectiveness is achieved by use of distributed hash tables allowing effective store and lookup operation. Actually DHT operations have to be based on unique IDs which can be easily transformed into keys by means of hash

function employed in particular DHT implementation. Such approach is acceptable for rule based adaptation systems which do not require information about similarity amongst user profiles to decide.

A method was proposed [8] for creating hierarchical user profiles using Wikipedia concepts as the vocabulary for describing user interests. Authors proposed a method for distinguishing informational and recreational interests in the profile from the commercial interests. They developed ways of mapping documents to Wikipedia concepts for the purpose of profile generation.

It was presented [9] a framework for content-based retrieval integrating a relevance feedback method with a word sense disambiguation (WSD) strategy based on WordNet for inducing semantic user profiles. Hypothesis of authors is that substituting words with synsets produces a more accurate document representation that could be successfully used by learning algorithms to infer more accurate user profiles. These semantic profiles will contain references to concepts defined in lexicons or ontologies.

In paper [10], authors combines the ontology and concept space, indicates the feature items of user profile with semantic concepts, calculates learner's interest-level to the topic through establishing the word frequency and utilize the suitable calculation methods, mining the concepts within the user's feedback files and the relationship between concepts, combines user's short-term interests and long-term interests to create user profiles model with semantic concept hierarchy tree and embody the drifting of user profile and improves and completes the user profiles model consistently on the related feedback mechanism.

Authors have proposed [11, 12] an approach to personalized query expansion based on a semantic user model. They discussed the representation and construction of the user model which represents individual user's interests by semantic mining from user's resource searching process in order to perceive the semantic relationships between user's interests which are barely considered in traditional user models and to satisfy the requirement of providing personalized service to users in e-Learning systems.

It has been described in [13] a personalized search approach that represents the user profile as a weighted graph of semantically related concepts of predefined ontology, namely the ODP (<http://www.dmoz.org>). The user profile is built by accumulating graph based query profiles in the same search session. We define also a session boundary recognition mechanism that allows using the appropriate user profile to re-rank search results of queries allocated in the same search session.

III. THE PROPOSED APPROACH

A user model is an internal representation of the user's properties. Before a user model can be used it has to be constructed. This process requires many efforts to gather the required information and finally generate a model of the user. The effectiveness of a user profile depends on the information the system delivers to the user. If a large proportion of information is irrelevant, then the system becomes more of an

annoyance than a help. This problem can be seen from another point of view; if the system requires a large degree of customization, then the user will not be willing to use it anymore.

Depending on the content and the amount of information about the user, which is stored in the user profile, a user can be modeled. Thus, the user profile is used to retrieve the needed information to build up a model of the user. The behavior of an adaptive system varies according to the data from the user model and the user profile. Without knowing anything about the user, a system would perform in exactly the same way for all users [1]. Representation of user model [14, 15] is a necessary factor for building effective and accurate adaptive systems. Adaptive systems compare user profiles to some reference profiles or item characteristics in order to predict the user's model in considering items. The outcome of that process depends on the ability to accurately identify and represent the user's model.

The presented approach for constructing a semantically enhanced user model that represents the user's interests from web-log data [16] (web usage logs). The goal of incorporating the semantic content of the web pages to build the semantically enhanced user models is to address the high dimensionality problem and semantic inadequacy of the Vector Space Model [17, 18, 19] on which the initial user model was based, and to map conceptually related terms. To enrich the user model during the user is browsing the pages and navigate the web-based system the user model must be updated. To update user model our proposed approach analyzes user's history query keywords by using WordNet.

To acquire user interests, we must extract the user behavior and visited page address from web-log data. Then we analyze the visited pages to acquire the terms in the pages that can be considered as concepts in the user model. The extracted terms are represented by Vector Space Model [17, 18, 19] that is adapted to our proposed system to achieve effective representations of documents where each document is identified by an n-dimensional feature vector for each dimension corresponds to a distinct term. Each term in a given document vector has an associated weight.

The term vector serves as the initial term-based user model (IUM) upon which we intended to improve. To build a semantically enhanced user model (SUM), we used refined domain-specific concepts. First we obtained a list of domain-specific concepts from domain ontology. Then we performed term-to concept mapping between terms in the initial user model (IT-UM) and domain related concepts based on concept hierarchies in WordNet. The final product is a semantically enhanced user model (SUM) in which terms are mapped to related high-level concepts.

The semantic User Model (SUM) can be updated using user query and WordNet. For a certain keyword, we extract the words which have the semantic relationships with the keyword and add them into the learner interest model as nodes according to semantic relationships in WordNet.

The goal of incorporating the semantic content of the web pages to build the semantically enhanced user models was to

address the high dimensionality problem and semantic inadequacy of the vector space model, on which the initial user model was based, and to map conceptually related terms. To enrich the user model during the user is browsing the pages and navigate the web-based system the user model must be updated. To update user model our proposed approach analyzes user's history query keywords by using WordNet.

The proposed approach architecture is shown in figure 1.

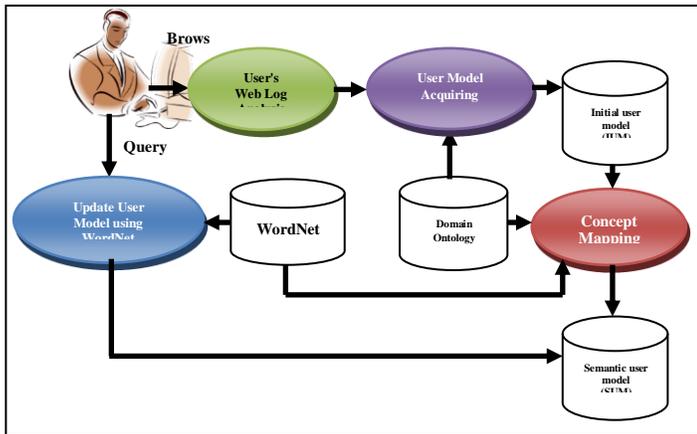


Figure 1: Proposed System Architecture

A. User's Web Log Analysis

Web usage mining [20], the process of discovering patterns from web data using data mining methods, strives to find learner preferences based on the web-logs that reside on servers. Web log [16] records each transaction, which was executed by the browser at each web access. Each line in the log represents a record with the IP address, time and date of the visit, accessed object and referenced object. In such data, we follow sequences in visiting individual pages by the learner, who is, under certain condition, identified by the IP address. In sequences, we can look for learners behavior patterns.

The data from Web logs, in its raw form, is not suitable for the application of usage mining algorithms. The data need to be cleaned and preprocessed. To perform log data analysis, the data pre-processing process must be accomplished. The data pre-processing is the process of cleaning and transforming raw data sets into a form suitable for web mining. The task of the data pre-processing module is therefore, to obtain usable datasets from raw web log files, which, in most cases, contain a considerable amount of incomplete and irrelevant information.

The overall data preparation process [21, 22] is briefly described in figure 2.

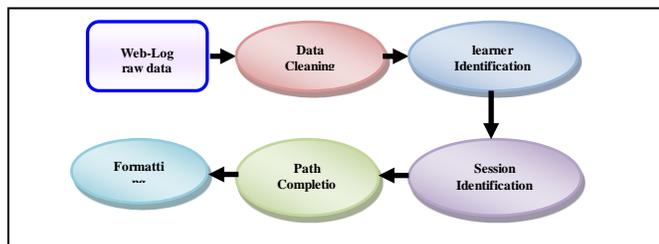


Figure 2: data preparation process for web log

Data Cleaning: to remove accesses to irrelevant items (such as button images), accesses by Web crawlers (i.e. non-human accesses), and failed requests.

Learner Identification: Because web logs are recorded in a sequential manner as they arrive, therefore, records for a specific learner are not necessary recorded in consecutive order rather they could be separated by records from other learners.

Session Identification: To divide pages accessed by each learner into individual sessions. A session is a sequence of pages visited by a learner. We also call it as a usage sequence.

Path Completion: To determine if there are important accesses which are not recorded in the access log due to caching on several levels.

Formatting: Format the data to be readable by data mining systems.

Once web logs are preprocessed, useful web usage patterns may be generated by applying data mining techniques. Table 1 shows a sample of web log data after preprocessing process.

TABLE I. SAMPLE OF WEB LOG DATA

Visit Time	UserId	URL
20090405202122	10	http://www.cs.bu.edu/teaching/
20090405203225	19	http://www.cs.bu.edu/teaching/unix/intro/
20090406081905	10	http://www.cs.bu.edu/teaching/cs113/spring-2000/object/
20090407091215	11	http://www.aw-bc.com/brookshear/
20090407082621	19	http://hortle.ccsu.edu/java5/Notes/chap21/ch21_1.html

The outputs of this step are web based learning materials; that the learner explored and preferred it, and the behavior pattern of the learner. The learner behavior is used to acquiring knowledge requirement for learners based on course ontology.

B. Domain ontology developing based knowledge engineering approach

Ontology engineering is a subfield of knowledge engineering that studies the methods and methodologies for building ontologies. It researches the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tools suite and languages that support them. Knowledge Engineering field usually uses the IEEE 1074-2006 standard [23] as reference criteria. The IEEE 1074-2006 is a standard for developing a software project life cycle processes. It describes the software development process, the activities to be carried out, and techniques that can be used for developing software.

It was proposed [24] a knowledge engineering approach to build domain ontology. Figure 3 shows main steps of the ontology development process.

Identify the purpose and requirement specification concerns to clear identify the ontology purpose, scope and its intended use, that is the competence of the ontology. Ontology acquisition is to capture the domain concepts based on the ontology competence. The relevant domain entities (e.g. concepts, relations, slots, and role) should be identified and organized into hierarchy structure. This phase involves three steps as follows: first, enumerate important concepts and terms in this domain; second, define concepts, properties and

relations of concepts, and organize them into hierarchy structure; third, consider reusing existing ontology.

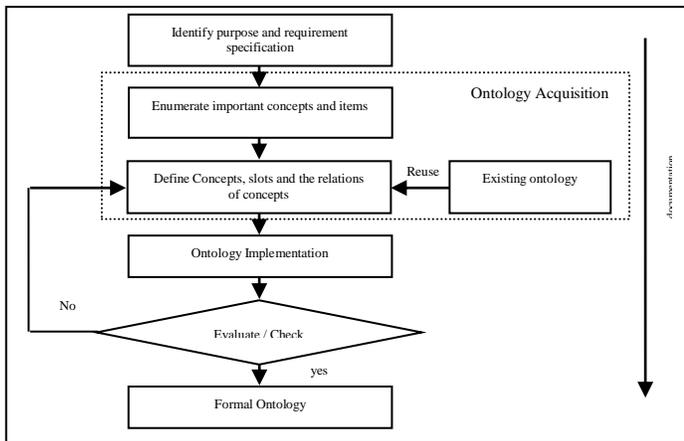


Figure 3 Main steps of the ontology development

Ontology implementation aims to explicit represent the conceptualization captured in a formal language. Evaluation/Check means that the ontology must be evaluated to check whether it satisfies the specification requirements. Documentation means that all the ontology development must be documented, including purposes, requirements, textual descriptions of the conceptualization, and the formal ontology.

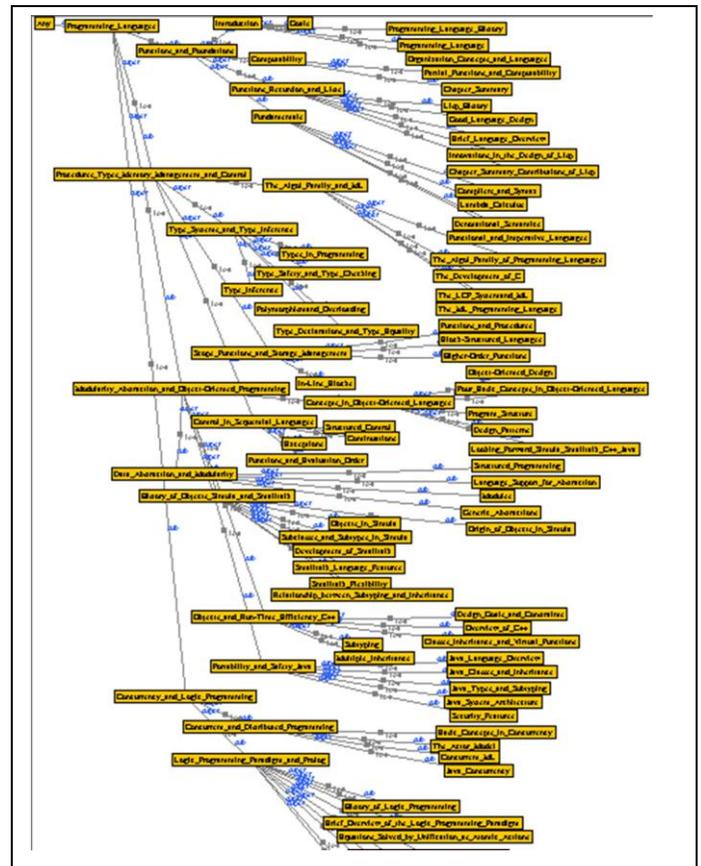
Our domain focuses "programming languages" course. We use Hozo [25] as our ontology editor. Since Hozo is based on an ontological theory of a role-concept, it can distinguish concepts dependent on particular contexts from so-called basic concepts and contribute to building reusable ontologies. A role-concept [24] represents a role which an object plays in a specific context and it is defined with other concepts. On the other hand, a basic-concept does not need other concepts for being defined. An entity of the basic concept that plays a role-concept is called a role-holder. Figure 4 shows part of our domain ontology and the extracted OWL [26] is shown in figure 5.

C. User Model Acquiring

In the proposed system [6, 22], user interest model's knowledge expression uses the thought, which is based on the space vector model's expression method and the domain ontology. This method acquires user's interest was shown in [6, 22]. Figure 5 shows certain steps to acquire user interest.

D. Document Representation

The Vector Space Model [27, 28] is adapted in our proposed system to achieve effective representations of documents. Each document is identified by n-dimensional feature vector for each dimension corresponds to a distinct term. Each term in a given document vector has an associated weight.



```

- <owl:Class rdf:ID="Programming_Languages">
  <rdfs:label>Programming_Languages</rdfs:label>
</owl:Class>
- <owl:Class rdf:ID="Functions_and_Foundations">
  <rdfs:label>Functions_and_Foundations</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Programming_Languages" />
</owl:Class>
- <owl:Class rdf:ID="Introduction">
  <rdfs:label>Introduction</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Functions_and_Foundations" />
</owl:Class>
- <owl:Class rdf:ID="Programming_Language">
  <rdfs:label>Programming_Language</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Introduction" />
</owl:Class>
- <owl:Class rdf:ID="Goals">
  <rdfs:label>Goals</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Introduction" />
</owl:Class>
- <owl:Class rdf:ID="Programming_Language_History">
  <rdfs:label>Programming_Language_History</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Introduction" />
</owl:Class>
- <owl:Class rdf:ID="Organization_Concepts_and_Languages">
  <rdfs:label>Organization_Concepts_and_Languages</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Introduction" />
</owl:Class>
- <owl:Class rdf:ID="Computability">
  <rdfs:label>Computability</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Functions_and_Foundations" />
</owl:Class>
- <owl:Class rdf:ID="Partial_Functions_and_Computability">
  <rdfs:label>Partial_Functions_and_Computability</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Computability" />
</owl:Class>
- <owl:Class rdf:ID="Chapter_Summary">
  <rdfs:label>Chapter_Summary</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Computability" />
</owl:Class>
  
```

Figure 5: Extracted OWL for the Domain Ontology

TABLE III SHOWS THE TERM WEIGHTS IN DIFFERENT DOCUMENTS

Items / DOC	Programming Language	Programming of Lists	Functions and Foundations	Programming of Recursion	Procedures Types	Memory Management and Control	Object Oriented Programming	Structured Programming	Concurrency and Logic Programming	Distributed Programming	Logic Programming	ML Programming Language
Doc1	0.3899	0.8490	0.4622	0.8490	0.5034	0.0000	0.7320	0.8390	0.5034	0.0000	0.5034	0.1830
Doc 2	0.2599	0.2830	0.0000	0.7075	0.6712	0.1830	0.0000	0.0000	0.0000	1.3156	0.3356	0.7320
Doc 3	0.5198	0.8490	0.7703	0.5660	0.0000	0.7320	0.5490	0.8390	0.6712	0.0000	0.0000	0.0000
Doc 4	0.2599	0.5660	0.6162	0.0000	0.0000	0.9150	0.7320	0.8390	0.0000	0.0000	0.0000	0.0000
Doc 5	0.6498	0.9905	0.7703	0.0000	0.5034	0.0000	0.3660	0.8390	1.3425	0.8771	0.0000	0.5490
Doc 6	0.2599	0.8490	0.0000	0.7075	0.8390	0.0000	0.0000	0.0000	0.6712	0.4385	0.5034	1.4641
Doc 7	0.1300	0.7075	0.7703	0.0000	0.0000	0.9150	1.2811	0.8390	1.1747	0.2193	0.0000	0.0000
Doc 8	0.0000	0.0000	0.0000	0.7075	0.0000	0.0000	0.0000	0.5034	1.0068	1.0963	0.6712	0.0000
Doc 9	0.2599	0.7075	0.6162	0.1415	0.0000	0.7320	0.9150	0.8390	0.3356	0.0000	0.1678	0.0000
Doc 10	0.6498	0.0000	0.7703	0.8490	0.0000	1.4641	0.0000	0.0000	0.0000	0.0000	1.0068	0.0000
Doc 11	0.9097	0.0000	0.0000	0.4245	0.5034	0.0000	1.4641	0.1678	0.6712	1.7541	0.5034	0.5490
Doc 12	0.5198	0.0000	0.0000	0.0000	0.6712	0.5490	0.0000	1.1747	1.0068	1.5349	0.0000	0.7320
Doc 13	0.2599	0.4245	0.7703	0.2830	0.3356	0.0000	0.0000	0.8390	0.0000	0.0000	0.3356	0.0000
Doc 14	0.0000	0.2830	0.7703	0.1415	0.1678	0.7320	0.1830	0.0000	0.0000	0.0000	0.1678	0.1830
Doc 15	0.6498	0.8490	1.0784	0.7075	1.1747	0.9150	0.9150	0.0000	0.0000	0.0000	0.8390	0.7320
Doc 16	0.0000	0.5660	0.4622	0.5660	1.0068	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0980

TABLE II SHOWS THE TERM FREQUENCY IN DIFFERENT DOCUMENTS

Items / DOC	Programming Language	Programming of Lists	Functions and Foundations	Programming of Recursion	Procedures Types	Memory Management and Control	Object Oriented Programming	Structured Programming	Concurrency and Logic Programming	Distributed Programming	Logic Programming	ML Programming Language
Doc1	3	6	3	6	3	0	4	5	3	0	3	1
Doc 2	2	2	0	5	4	1	0	0	0	6	2	4
Doc 3	4	6	5	4	0	4	3	5	4	0	0	0
Doc 4	2	4	4	0	0	5	4	5	0	0	0	0
Doc 5	5	7	5	0	3	0	2	5	8	4	0	3
Doc 6	2	6	0	5	5	0	0	0	4	2	3	8
Doc 7	1	5	5	0	0	5	7	5	7	1	0	0
Doc 8	0	0	0	5	0	0	0	3	6	5	4	0
Doc 9	2	5	4	1	0	4	5	5	2	0	1	0
Doc 10	5	0	5	6	0	8	0	0	0	0	6	0
Doc 11	7	0	0	3	3	0	8	1	4	8	3	3
Doc 12	4	0	0	0	4	3	0	7	6	7	0	4
Doc 13	2	3	5	2	2	0	0	5	0	0	2	0
Doc 14	0	2	5	1	1	4	1	0	0	0	1	1
Doc 15	5	6	7	5	7	5	5	0	0	0	5	4
Doc 16	0	4	3	4	6	0	0	0	0	0	0	6
No of Doc's	13	12	11	12	10	9	9	10	10	7	10	9

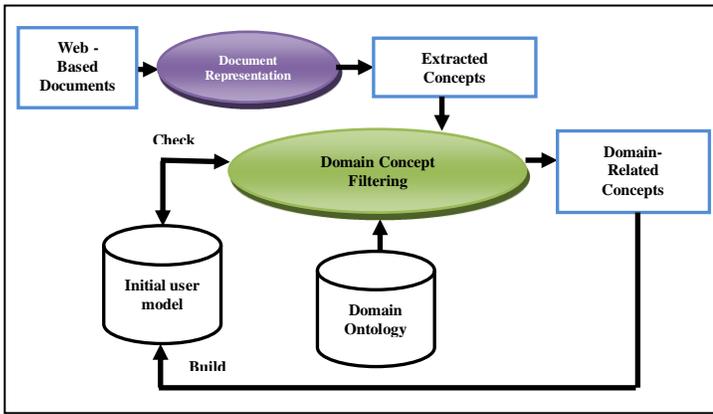


Figure 5: steps to acquire learner interest

The weight is a function of the term frequency, collection frequency and normalization factors. Different weighting approaches may be applied by varying this function. Hence, a document j is represented by the document vector d_j :

$$d_j = (w_{1j}, w_{2j}, \dots, w_{nj})$$

Where, w_{kj} is the weight of the k^{th} term in the document j .

The term frequency reflects the importance of term k within a particular document j . The weighting factor may be global or local. The global weighting factors clarify the importance of a term k within the entire collection of documents, whereas a local weighting factor considers the given document only.

The document keywords were extracted by using a term-frequency-inverse-document-frequency (*tf-idf*) calculation [18, 19], which is a well-established technique in information retrieval. The weight of term k in document j is represented as:

$$w_{kj} = tf_{kj} \times (\log_2^n - \log_2^{df_k} + 1)$$

Where: tf_{kj} = the term k frequency in document j , df_k = number of documents in which term k occurs, n = total number of documents in the collection. Table 2 shows the term frequency in different documents.

The main purpose of this step is to extract interested items in the web page, then get term frequency that reflects the importance of the term. Finally, get the weight of terms in the selected page. The output of this step is the weight of terms in selected page that can be used to build learner interest profile. Table 3 shows a sample of the weighted terms in the documents; that found in table 3.

E. Domain Concept Filtering

This process discovers concepts which represent the learner's interests. These concepts and items are compared to the domain ontology to check the relevant items to the learner profile. The most relevant ones update the learner profile. The items relevance is based on ontology-based semantic similarity where browsed items by a learner on the web are compared to the items from a domain ontology and learner profile. The importance is combined with the semantic similarity to obtain a level of relevance. The page items are processed to identify domain-related words to be added to the learner profile. A bag

of browsed items is obtained via a simple word indexing of the page visited by the learner. We filter out irrelevant words using the list of items extracted from domain ontology. Once domain-related items are identified, we evaluate their relevance to learner's interests.

The selected method was used in [29, 30] to compute semantic similarity function (S) based on a domain ontology. The similarity is estimated for each pair of items where one item is taken from a learner profile, while the other one from a set of browsed items.

The functions S_w is the similarity between synonym sets, S_u is the similarity between features, and S_n is the similarity between semantic neighborhoods between entity classes an of ontology p and b of ontology q , and w_w , w_u , and w_n are the respective weights of the similarity of each specification component.

$$S(a^p, b^q) = w_w \times S_w(a^p, b^q) + w_u \times S_u(a^p, b^q) + w_n \times S_n(a^p, b^q)$$

; For $w_w, w_u, w_n \geq 0$:

Weights assigned to S_w , S_u , and S_n depend on the characteristics of the ontologies.

The similarity measures are defined in terms of a matching process [29, 30]:

$$S(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a, b) |A / B| + (1 - \alpha(a, b)) |B / A|}$$

Where A and B are description sets of classes a and b , i.e., synonym sets, sets of distinguishing features and a set of classes in semantic neighborhood; $(A \cap B)$ and (A / B) represent intersection and difference respectively, $||$ is the cardinality of a set; and α is a function that defines relative importance of non-common characteristics. A set of browsed items that are similar to items from the learner profile is considered as a set of items that can be added to this profile.

IV. BUILDING SEMANTIC USER MODEL USING CONCEPT MAPPING

To overcome these weaknesses of term-based representations, an ontology-based representation [33, 34] using wordnet will be performed. Moreover, by defining an ontology base, which is a set of independent concepts that covers the whole ontology, an ontology-based representation allows the system to use fixed-size document vectors, consisting of one component per base concept.

We present a method based on WordNet [35] that improves traditional vector space model. WordNet is an ontology of cross-lexical references whose design was inspired by the current theories of human linguistic memory. English names, verbs, adjectives, and adverbs are organized in sets of synonyms (synsets), representing the underlying lexical concepts. Sets of synonyms are connected by relations. The basic semantic relation between the words in WordNet is synonymy [36]. Synsets are linked by relations such as specific/generic or hypernym /hyponym (is-a), and meronym/holonym (part-whole). The principal semantic relations supported by WordNet is synonymy: the synset

(synonym set), represents a set of words which are interchangeable in a specific context. WordNet [36] consists of over 115,000 concepts (synsets in WordNet) and about 150,000 lexical entries (words in WordNet). This representation requires two more stages: a) the “mapping” of terms into concepts and the choice of the “merging” strategy, and b) the application of a disambiguation strategy.

The purpose of this step is to identify WordNet concepts that correspond to document words [31]. Concept identification is based on the overlap of the local context of the analyzed word with every corresponding WordNet entry. The entry which maximizes the overlap is selected as a possible sense of the analyzed word. The concept identification architecture for the terms in the initial user model is given in figure 6.

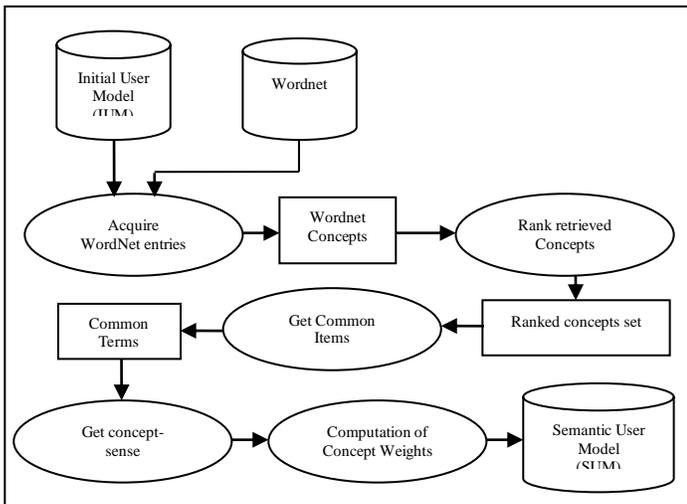


Figure 6: Semantic User Model using Concept Mapping

We use WordNet categories [32] to map all the stemmed words in all documents into their lexical categories. For example, the word “dog” and “cat” both belong to the same category “noun.animal”. Some words also has multiple categories like word “Washington” has 3 categories (noun.location, noun.group, noun.person) because it can be the name of the American president, the city place, or a group in the concept of capital. Some word disambiguation techniques are used to remove the resulting noise added by multiple categories mapping which are disambiguation by context and concept map.

A. The Weight of Concept Computation

The concepts in documents are identified as a set of terms that have identified or synonym relationships, i.e., synsets in the WordNet ontology. Then, the concept frequencies Cf_c are calculated based on term frequency tf_t as follows:

$$Cf_c = \sum_{t_m \in r(c)} tf_{tm}$$

Where $r(c)$ is the set of different terms that belong to concept C . Note that WordNet returns an ordered list of synsets based on a term. The ordering is supposed to reflect how common it is that the term is related to the concept in standard

English language. More common term meanings are listed before less common ones. The authors in [33, 34] have showed that using the first synset as the identified concept for a term can improve the clustering performance more than that of using all the synsets to calculate concept frequencies.

Hypernyms of concepts can represent such concepts up to a certain level of generality. The concept frequencies are updated as follows:

$$hf_c = \sum_{b \in H(c,r)} Cf_b$$

Where $H(c,r)$ is the set of concepts C_H , which are all the concepts within r levels of hypernym concepts of c .

In WordNet, is obtained by gathering all the synsets that are hypernym concepts of synset c within r levels. In particular, $H(c,\infty)$ returns all the hypernym concepts of c and $H(c,0)$ returns just c .

The weight of each concept c in document d is computed as follows:

$$wh_c = hf_c \times idf_c$$

Where idf_c is the inverted document frequency of concept c by counting how many documents in which concept c appears as the weight of each term t in the document d .

The weights of the concepts after mapping the items in table 3 is shown in table 4 after computing the concepts weights.

V. UPDATE USER MODEL USING WORDNET

During the user is working through the web based learning system, user interests' change quite often, and users are reluctant to specify all adjustments and modifications of their intents and interests. Therefore, techniques that leverage implicit approaches for gathering information about users are highly desired to update the user interests that are often not been fixed.

In order to update user interest [6, 37], first of all, we should analysis user's history query keywords. For a certain keyword, we extract the words which have the semantic relationships with the keyword and add them into the user interest model as nodes according to semantic relationships in WordNet.

With new words added constantly, user is always interested in the kind of the words with a higher score which standard for some type of knowledge. We must constantly, update the user model after the users enter the new specific keywords. User model is updated by the new keywords. The incremental updating strategy is used here, and gives the related words the different score according to the relations which reflect their importance of different words in order to render the interestingness of the words. As a result, the words that are more frequent have a higher score. Because of history keywords have the order, the keywords which are inquired later always have more meaning than the keywords which are inquired earlier; it need multiply a factor of attenuation β when increasing the score. Because the keywords are added

constantly and the scale of the user model becomes bigger, some old nodes must be removed in order to reduce user interest model.

The main steps of this method can be described as follows:

1) *If a new keyword is found in the original user model, we increase the score of the related nodes directly. That is, the node is given by five score after multiplying a factor of attenuation β . If it is not found, we must create a new word node and give it five score.*

2) *Finding the following three relations between new keywords and inputted words in the user model based on the WordNet:*

a) *Synonymous relations: obtain the synonym set and insert every synonym into the original user model in turn. If the synonym is found in the original user model, we increase the score of the related nodes directly. That is, the node is given by four score after multiplying a factor of attenuation β . Otherwise, create a new word node with four score and add a new undirected edge labeled synonym relation.*

b) *Hyponym or Hypernym relations: obtain the hyponym or hypernym set and insert every word into the original user model in turn. If the word is found in the original user model, we increase the score of the related nodes directly. That is, the node is given by two score after multiplying a factor of attenuation β . Otherwise, create a new word node with two score and add a new directed edge labeled hyponym or hypernym relation.*

c) *Meronym or Holonym relations: obtain the meronym or holonym set and insert every word into the original user model in turn. If the word is found in the original user model, we increase the score of the related nodes directly. That is, the node is given by one score after multiplying a factor of attenuation β . Otherwise, create a new word node with one score and add a new directed edge labeled meronym or holonym relation.*

d) *In order to reduce user interest model, the nodes which have the lower score must be removed after some time.*

CONCLUSION

We have presented in this paper a novel approach for conceptual document indexing. Our contribution concerns two main aspects. The first one consists on a concept-representation approach of the initial user model items based on the use of WordNet. The approach is not new but, we proposed new techniques to identify concepts and to weight them. In addition to the semantic representation approach to build the semantic user model, we proposed approach to update the user model using the Wordnet.

REFERENCES

- [1] F. Christoph, (2005). User Modeling and User Profiling in Adaptive E-learning Systems, Master's Thesis At Graz University of Technology.
- [2] P. Apple Wai and S. Horace.(2007), Educational Ontologies Construction for Personalized Learning on the Web, Studies in Computational Intelligence (SCI) 62, 47–82, Springer-Verlag Berlin Heidelberg.
- [3] Z. Leyla, R. Elizabeth, W. Robert. (2009), The Effectiveness of Personalization in Delivering E-learning Classes, 2009 Second International Conferences on Advances in Computer-Human Interactions, 978-0-7695-3529-6/09, IEEE.
- [4] A. Palakorn, H. Hyoil, N. Olfa, J. Roberta. (2007), Semantically Enhanced User Modeling, SAC'07, March 11-15, 2007, Seoul, Korea., ACM 1-59593-480-4/07/0003.
- [5] H. Changqin, J. Ying, D. Rulin. (2010), A semantic web-based personalized learning service supported by on-line course resources, Networked Computing (INC), 2010 6th International Conference, Print ISBN: 978-1-4244-6986-4, IEEE.
- [6] H. Hany, F. Khaled. (2010). Semantic web based Approach to learn and update Learner Profile in Adaptive E-Learning, Al-Azhar Engineering Eleventh International Conference, December 23-26.
- [7] K. Tom, J. Ivan. (2008), Semantic User Profile Acquisition and Sharing, International Conference on Computer Systems and Technologies – CompSysTech'08, Gabrovo, Bulgaria, June 12-13, 2008 , ACM 2008 ISBN: 978-954-9641-52-3/08/06.
- [8] R. Krishnan and K. Komal. (2009), Creating User Profiles Using Wikipedia, A.H.F. Laender et al. (Eds.): ER 2009, LNCS 5829, pp. 415–427, 2009. Springer-Verlag Berlin Heidelberg.
- [9] M. Degemmis, P. Lops, and G. Semeraro. (2006), WordNet-Based Word Sense Disambiguation for Learning User Profiles, EWMF/KDO 2005, LNAI 4289, pp. 18–33, Springer-Verlag Berlin Heidelberg.
- [10] W. Cuncun, Chongben , T. Hengsong. (2009), A Personalized Model for Ontology-driven User Profiles Mining, 2009 International Symposium on Intelligent Ubiquitous Computing and Education, DOI 10.1109/IUCE.2009.128, 978-0-7695-3619-4/09, IEEE.
- [11] L. Xiaojian, C. Shihong. (2009), Personalized Query Expansion Based on Semantic User Model in e-Learning System, 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, DOI 10.1109/FSKD.2009.787, 978-0-7695-3735-1/09, IEEE.
- [12] L. Xiaojian, C. Shihong. (2009), Research on Personalized User Model Based on Semantic Mining from Educational Resources Searching Process, 2009 International Joint Conference on Artificial Intelligence, DOI 10.1109/JCAI.2009.67, 978-0-7695-3615-6/09, IEEE.
- [13] D. Mariam, T. Lynda, B. Mohand. (2009), A Session Based Personalized Search Using An Ontological User Profile, SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A., ACM 978-1-60558-166-8/09/03.
- [14] Z. Ning, L. Yuefeng. (2003). Ontology-Based Web Mining Model: Representation of User Profiles. Proceeding of the IEEE/WIC International Conference on Web Intelligence (WI'03), 0-7695-1932-6/03, IEEE.
- [15] S. Ahu, M. Bamshad, B. Robin. (2007). Ontological User Profiles for Representing Context in Web Search. 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Workshops. 0-7695-3028-1, IEEE.
- [16] <http://www.w3.org/TR/WD-logfile.html>.
- [17] F. Christoph. (2005), User Modeling and User Profiling in Adaptive E-learning Systems, Master's Thesis at Graz University of Technology, Copyright 2005 by Christoph Fr'oschl.
- [18] L. Xiaojian, C. Shihong. (2009), Personalized Query Expansion Based on Semantic User Model in e-Learning System, 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, DOI 10.1109/FSKD.2009.787, 978-0-7695-3735-1/09, IEEE.
- [19] A. Palakorn, H. Hyoil, N. Olfa, J. Roberta. (2007), Semantically Enhanced User Modeling, SAC'07, March 11-15, 2007, Seoul, Korea., ACM 1-59593-480-4/07/0003.
- [20] G. PAOLO, F. SILVIA. (2009). Applied Data Mining for Business and Industry Second Edition, Chapter 6: Describing website visitors, ISBN: 978-0-470-05887-9 (Pbk), John Wiley.
- [21] K. Natheer, C. Chien-Chung. (2005). Web Usage Mining Using Rough Sets, NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society, 0-7803-9187-X/05, IEEE.
- [22] F. Khaled, M. Hany, M. Nagdy. (2011). Semantic Web supporting Adaptive E-Learning to build and represent Learner Model. The Second International Conference of E-learning and Distance Education (eLI 2011) – Riyadh 2011.

- [23] IEEE Standard for Developing Software Life Cycle Processes, IEEE Computer Society, New York (USA), April 26, 1997.
- [24] H. YUN, J. XU, M. J. XIONG. (2009). Development of Domain Ontology for E-learning Course, 978-1-4244-3930-0/09, IEEE.
- [25] K. Kozaki, Y. Kitamura, M. Ikeda, and R. Mizoguchi. (2002). Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of Role” and “Relationship”, Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002), Sigüenza, Spain, October 1-4, 2002, pp.213-218.
- [26] <http://www.w3.org/TR/owl-ref/>.
- [27] B. Qiu, W. Zhao. (2009). Student Model in Adaptive Learning System based on Semantic Web, 2009 First International Workshop on Education Technology and Computer Science, 978-0-7695-3557-9/09, IEEE, DOI 10.1109/ETCS.2009.466.
- [28] P. Jianguo, Z. Bofeng, W. Shufeng, W. Gengfeng, and W. Daming. (2007). Ontology Based User Profiling in Personalized Information Service Agent, Seventh International Conference on Computer and Information Technology, 0-7695-2983-6/07, IEEE.
- [29] R. Marek and K. Sayed. (2009). Updating User Profile using Ontology-based Semantic Similarity, FUZZ_IEEE 2009, Korea, August 20-24, 978-1-4244-3597-5, IEEE.
- [30] R. M. Andrea, E. Max J. (2003). Determining Semantic Similarity among Entity Classes from Different Ontologies, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 15, NO. 2, MARCH/APRIL, 1041-4347/03, IEEE.
- [31] B. Fatiha, B. Mohand, T. Lynda, D. Mariam. (2010). Using WordNet for Concept-Based Document Indexing in Information Retrieval, SEMAPRO: The Fourth International Conference on Advances in Semantic Processing, Pages: 151 to 157, IARIA.
- [32] A. Abdelmalek, E. Zakaria, S. Michel, M. Mimoun. (2008). WordNet-based and N-Grams-based Document Clustering: A Comparative Study, Third International Conference on Broadband Communications, Information Technology & Biomedical Applications, 978-0-7695-3453-4/08, IEEE.
- [33] D. Mauro, C. Celia, G. Andrea. (2010). An Ontological Representation of Documents and Queries for Information Retrieval Systems, IEA/AIE 2010, Part II, LNAI 6097, pp. 555–564, Springer-Verlag Berlin Heidelberg.
- [34] C. Celia, G. Andrea. (2006). An Ontology-Based Method for User Model Acquisition. In: Ma, Z. (ed.) Soft computing in ontologies and semantic Web. Studies in fuzziness and soft computing, pp. 211–227., Springer-Verlag Berlin Heidelberg.
- [35] F. Christiane. (2010). WordNet. In R. Poli et al. Theory and Applications of Ontology: Computer Applications, (pp. 231-243). 231-243, DOI: 10.1007/978-90-481-8847-5_10, Springer Science+Business Media B.V.
- [36] A. Abdelmalek Amine, E. Zakaria, and S. Michel. (2010). Evaluation of Text Clustering Methods Using WordNet. The International Arab Journal of Information Technology, Vol. 7, No. 4.
- [37] S. Yan, L. Yun, L. Luan, C. Ling. (2009). A Personalized Search Results Ranking Method Based on WordNet, 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 978-0-7695-3735-1/09, IEEE.
- [38] Khaled M. Fouad, Shehab Gamalel-Din, Mofreh A. Hogo, Nagdy M. Nagdy. Adaptive E-Learning System based on Semantic Web and Fuzzy Clustering (IJCISIS).

AUTHORS PROFILE



DR. Yasser A. Nada Was born in Ismailia, Egypt, in 1968. He received the BSc degree in pure Mathematics and Computer Sciences in 1989 and MSc degree for his work in computer science in 2003, all from the Faculty of Science, Suez Canal University, Egypt. In 2007, he received his Ph.D. in Computer Science from the Faculty of Science, Suez Canal University, Egypt. From September 2007 until now, he worked as a lecturer of computer science, Faculty of Computers and Information Systems Taif University, KSA. His research interests include Expert Systems, Artificial Intelligence, Semantic Web Object Oriented Programming, Computer Vision, and Genetic.



Khaled M. Fouad received his Master degree of AI and expert systems. He is currently a PhD candidate in the faculty of engineering AlAzhar University in Egypt. He is working now as lecturer in Taif University in Kingdom of Saudi Arabia (KSA) and is assistant researcher in Central Laboratory of Agriculture Expert Systems (CLAES) in Egypt. His current research interests focus on Semantic Web and Expert Systems.