Research Journal of Applied Sciences, Engineering and Technology 4(15): 2334-2342, 2012
ISSN: 2040-7467

Submitted: January 08, 2012 Accepted: March 15, 2012 Published: August 01, 2012

An Intelligent Technique for Extracting Subjects from User Profile Using ODP Ontology-Driven Reasoning

[1]Mina Minazadeh, [2]Kambiz Badie and [3]Mir Mohsen Pedram
[1]Department of Computer Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran
[2]Information Technology Research Faculty, Telecommunication Research Center, Tehran, Iran
[3]Computer Engineering Department, Faculty of Engineering, Tarbiat Moallem University, Karaj/Tehran, Iran

Abstract: Nowadays, the amount of available information, especially on the Web, is increasing. In this field, the role of user modeling and personalized information access is obviously vital. The traditional techniques like BOW (Bags of words) limit recommendations to the words which have been stored in the profile. In other words, the news items, which semantically relate to the users interests, can't be recognized and recommended to the users. Besides, BOW technique suffers from the curse of dimensionality, thus computational burden reduction is an essential task to efficiently handle a large number of terms in practical applications. This study focuses on the problem of choosing a representation of documents that can be suitable to induce concept-based user profiles as well as to support a content-based retrieval process. In this study, a new approach has been proposed to construct a ranked semantic user profile through extracting the related subjects. The new items can be recommended through collecting information from the user's selections, based on existing domain ontology ODP. The efficiency of the proposed technique has been shown by embedding it into an intelligent aggregator, RSS (RSS is acronym of " Really Simple Syndication) feed reader, which has been trained and evaluated by different and heterogeneous users. The results in experimental session show that the incoming news item which semantically relates to the profile gets highly recommended to the user despite its excluding of common words in the profile.

Keywords: Component, ontology, ranking, semantic profile, user interest

INTRODUCTION

Due to the impressive growth of the availability of text data, there has been a growing interest in augmenting traditional information filtering and retrieval approaches with Machine Learning (ML) techniques inducing a structured model of a user's interests, the user profile, from text documents (Burke, 2002). These methods typically require users to label documents by assigning a relevance score and automatically infer profiles exploited in the filtering/retrieval process.

There are information access requests, like "interesting technology news", that cannot be answered through straightforward matching of the interesting words and documents represented by some keywords. In order to find relevant information in these problematic information scenarios, a possible solution is to develop methods which are able to analyze documents already deemed by the user as interesting in to discover relevant concepts to store in his personal profile. It should be noted that, keyword-based approaches are unable to capture the *semantics* of the user interests. They are only driven by a string-matching operation, i.e., If a string is found in both the profile and the document, a match is made and the document is then considered as relevant. String matching suffers from problems of *polysemy*, the presence of multiple meanings for one word and *synonymy*, multiple words having the same meaning. Due to synonymy, relevant information might be missed if the profile does not contain the exact keywords occurring in the documents, while wrong documents might be deemed as relevant because of the occurrence of words with multiple meanings. One of the well-used algorithms in content-based method is Bags of Words algorithm, which stores only the words of the interesting documents then recommends just the items which have the exact stored words in the profile. Therefore the items which semantically relate to the user's interests without including the common words are not recommended. Alternative methods are not able to learn more accurate profiles to capture concepts expressing users' interests from relevant documents. Then in this study a user-

Corresponding Author: Mina Minazadeh, Department of Computer Engineering, Islamic Azad University, Science & Research Branch, Tehran, Iran

profiling technique named BOC (Bags of Categories), which recognizes the interesting subject instead of the words during the training sessions is proposed. With the proposed technique the weights of the categories in the news vectors are become fully substantial for finding the main subjects of the news. It means, the categories with high frequencies help us find the relative subjects in the related news. Thus a number of unrelated subjects may get omitted in the storing phase. Besides Assigning rank to each category in the proposed technique would show a persistent interest for each category in the profile.

This problem has been addressed in our study. The proposed system constructs a semantic profile by extracting the categories with the manual ontology and exploits user profiles learned by a content-based system to improve the recommendation algorithm according to the user's interests. After initial training sessions, aggregated contents in the profile are ranked based on their weights. Results of experiments have shown that the proposed algorithm largely improves the information sorted by BOW algorithm.

Anoverviewof recommendation technique: As we know recommender systems have the effect of guiding users in a personalized way to interesting or useful objects in a large space of possible options (Mladenic, 1999). Recommender algorithms use input about a customer's interests to generate a list of recommended items. Also the recommendation can be understood as a filtering process in which the filter passes only the contents relevant to each individual user.

Systems implementing the content-based recommendation approach, analyze a set of documents, usually textual descriptions of the items previously rated by an individual user and build a model or profile of user interests based on the features of the objects rated by that user Middleton *et al*. (2001). Each type of filtering methods has its own weaknesses and strengths (Middleton *et al*., 2001; Shardanand and Maes, 1995). The profile is exploited to recommend new items of interest. The user's preferences are acquired by monitoring his/her behavior when navigating on the web, applying automatic learning techniques associated with an ontological representation (Lee, 2001). Since the user is reluctant to provide information about his personal interests explicitly, as in personalized Google, MyYahoo and InfoQuest, implicit feedback has recently attracted much attention in the user profile modeling (Middleton *et al*., 2001; Joachims, 2002). Recommender systems like "*LETIZIA*" (Lieberman, 2003) and "*JITIR*" (Rhodes, 1997) are adaptive systems that exploit information which are collected from emails or pages viewed by the user to represent the short term user context as his current intention and propose proactively to the user, relevant information according to his current task. More recent approaches aim to model the user profile precisely, while some works use only the user's feedback to build the user profile as a set of class vectors (Mc Gowan, 2003) or term relations (Shen *et al*., 2002; Koutrika and Ioannidis, 2005; Sieg *et al*., 2005) Use domain ontology as an additional source of evidence to build a semantic representation of the user profile. NectaRSS is a content-based recommended system which is designed to rank newly arrived information according to an automatically elaborated user profile and the user profile finally takes the form of the usual Bag of Words (Samper and Castillo, 2008).

In Pazzani and Billsus (1997) learning user profiles has been suggested as Bayesian classifiers. System supports users in document searching by maintaining user profiles which store both interests and explicit *dis*interests (Asnicar and Tasso, 1997). A sense-based representation to build a user profile as a semantic network whose nodes represent senses of the words in documents requested by the user is exploited (Magnini and Strapparava, 2001) Adopts a Rocchio relevance feedback method to create and update user personal models that are directly compared to determine similar users for collaborative recommendations. In Rodriguez *et al*. (1997) Word Net is used to enhance neural network learning algorithms.

In this study, an algorithm has been proposed for constructing a semantic profile by embedding it into an RSS aggregator for news recommendation.

METHODOLOGY

Overall procedure: The pieces of information which our system retrieves are referred to as news items, each of which is composed of a headline, a hyperlink to its content and a summary. Information aggregators usually show the headline hyperlinked to the content and the summary and the hyperlink is a unique ID for the news item. It is assumed that user's click on an item in the aggregator, corresponds to an interested topic, thus the categories of the words which belong to the interesting items, are recognized and stored in the user profile. Here "word categories" is used as the possible subjects of the news items. Word categories provide more information about the content of a document rather than words do.

Figure 1 shows the procedural view of the BOC algorithm. The gray parts show our main goal in the proposed technique and the dashed line shows online training, which means the training will be continued after the initial sessions during the test.

In the initial training sessions, the news items are randomly suggested to the user during 4 sessions and the profiles are incrementally built based on the content of the choices in training sessions. After the initial training phase, the new incoming items are scored and then recommended based on the user profile.

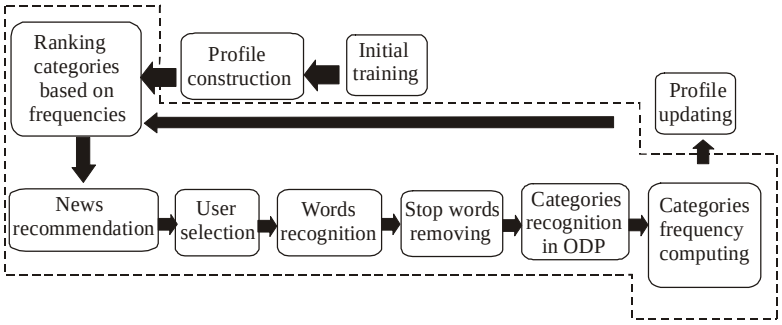

Fig. 1: BOC algorithm procedure

Extracting significant concepts from user profile: In order to find relevant news items for each user, a possible solution could be to develop methods for discovering concepts that characterize documents, which the user has already read as interesting ones. Traditional keyword-based approaches are unable to capture the semantics of the user interests. They are primarily driven by a string-matching operation: If a string, or some morphological variant, is found in both the profile and the document, a match is made and the document is considered relevant. String matching suffers from the following deficiencies:

- Polysemy, the presence of multiple meanings for one word (e.g., the noun "Bat" as a nocturnal mouse like mammal, or squash racket)
- Synonymy, multiple words have the same meaning (e.g., the verbs "make", "manufacture" and "produce" all refer to the production of items)

The result is that, due to synonymy, relevant information can be missed if the profile does not contain the exact keywords in the documents, while due to polysemy, wrong documents could be deemed relevant. These problems call for alternative methods able to learn more accurate profiles that capture concepts expressing user interests from relevant documents.

In this phase of our approach, the user profile is built in an implicit way: the user will not have to take any additional action such as explicit feedbacks to build her/his profile, more precisely the profile will be constructed automatically according to the user navigation history by the news items headlines which are presented to her/him.

As it was expressed, the proposed, BOC runs in three main steps:

- Representing a keyword derived from the interesting items. In our study the stop words are left out and stemming is performed on the text.
- After user selection, the categories of the words which represent the semantics of the user interests, must be provided, thus user context keyword are mapped on to the ODP ontology.
- We represent the user context by disambiguation mapped concepts. This phase could take advantage to recommend new items with high semantic relevance compared to the user profile, because of omitting some irrelevant categories.

These steps are described in the following with more details:

- **Keyword extraction:** In the first step the words are extracted by using a simple program which has been implemented by C# language. The original sentence is tokenize and then reduction to lemmas (for example, verbs are turned to their base form) is performed. Afterwards stop words are removed from the sentences based on the provided list which contains 600 different stop words.
- **Mapping the user context keyword on the reference ontology:** Once we had the user context keyword, we map it onto the ontology in order to extract the most relevant concepts. These mapped concepts are used later to represent concepts of depth three of the user context. There are many domains of ontology created manually and designed to organize web content such as yahoo (http: //www.yahoo.com), Magellan (http://www.mckinley.com) and the open directory project (http://www.dmoz.org). Since August 31, 2007, the Open Directory Project (ODP) is a manual edited directory of millions URLs that have been categorized into thousands categories and also is the most widely distributed database of Web content classified by humans. We have used it to get a concept-based representation of the user context. Indeed, storing the categories has two main advantages:

  Since most of the words are categorized the same, a considerable dimensional reduction in vector space model will be achieved.
  The proposed system can present more items which are semantically concerned to the user's interests. Supposing the user is interested in language

programming topics, if the profile has been constructed using keyword-base method, the recommendation is limited to the stored words without attention to the semantics of the words, but as we will see in the following, the proposed algorithm can suggest new items based on semantics. As it was explained, we represent the user context with general depth three related concepts issued from the ontology. As an example the category of word "Vb" in ODP database is: "Computers: Programming: Languages: Visual Basic: VBScript" that depth three is sufficient for our approach. Thus "Computers: Programming: Languages" is considered for the word "Vb" as in Fig. 2. The procedure for getting the representation of the ODP concepts is explained in details in Rodriguez *et al*. (1997). ODP databases have been restored to a SqlServer for the RSS reader proposes, so the processes are all done offline.

- **Disambiguation mapped concepts:** In this subsection, the user context will be represented with general depth of three. Afterwards some irrelevant categories can be found by mapping words to the ODP concepts and recognizing the categories for each word. These kinds of categories which cause ambiguities of the concepts must be denoted and eliminated from encountered subjects. The used method of disambiguation is based on the categories weights in the news items. More precisely, the encountered categories with higher frequencies show the main subject of the selected items. And against some other ones with lower frequencies are the categories with a negligible importance or even they are irrelevant. As an example suppose the news: "Lua has beenreleased, Lua is a dynamically-typed scripting language offering object-oriented capabilities, designed for extending applications." has been selected by the user. Table 1 shows some founded categories for the news from ODP which the depth of three for each category has been highlighted. Afterwards, category weight must be computed. Here the user's selection reveals his interests then the only criterion for category weight is the category frequency w^n_{ck} which is defined as follows:

$$w^n_{ck} = \frac{tf^{\,n}_{ck}}{\sum_{i=1}^{m} f^{\,n}_{ci}} \quad (1)$$

where, w^n_{ck} is the frequency of the category C_k in both headline and summary of the news between all found categories and *m* is the total number of categories in the news which the user has chosen. Table 2 also shows the computed weights. Consequently, some of unrelated categories can be eliminated before storing them in the profile. As it shows the most relevant category has the highest weight, which defines the main subject for the selected news, so "Computers: Programming: Language" category will be stored in the profile with the highest weight for the example. It should be mentioned that the proposed technique has found the category "Computers: Programming: Language" for some new incoming words like "Lua", which keyword-based algorithms are unable to recognize it that which subject it belongs to.

Ranking the profile contents: After training sessions and generating the profile, the profile will be processed in order to rank the contents. As it was explained, profile contents ranking are affected by the frequency of the categories. Weight of categories is defined as follows:

$$w^p_{ck} = \frac{tf^{\,p}_{ck}}{\sum_{i=1}^{m} tf^{\,p}_{ci}} \quad (2)$$

where, w^p_{ck} is the frequency of the category C_k in the profile and *m*is the total number of categories in the profile. Both the news items and the user profile will be represented using the vector space model, thus we Define the user profile P which has been generated at the end of the training as follows:

$$P = (r^P_{c1}\, r^P_{c2,}\, r^P_{c2}\, r^P_{c3}, \ldots)$$

where, r^P_{c1} is the rank of category C_k in the profile, which in fact is the computed weight for each category in BOC and the vector of news item, is defined in a similar way, i.e.,

$$N = (w^n_{c1,}\, w^n_{c2,}\, w^n_{c3,} \ldots)$$

As a result, a higher frequency of a category means a higher rank which affects more on recommendation.

Updating the user profile: The user profile is updated based on the next user's selection. If the categories of the selected items don't exist in the generated profile, they will be stored as the previous ones, otherwise their ranks are increased. The profile update is as follows:
Upgrade : (P_i, P_{i+1})

- Search the new incoming category c_k in the profile.
- If c_k was found in the profile go to 3 otherwise go to 4.
- Set weight of $c_k : w^p_{ck} = \left(\frac{tf^{\,p_i, p_{i+1}}_{ck}}{\sum_{i=1}^{m} tf^{\,p}_{ci}}\right)$
- Add c_k as a new category to the profile
- Go to 1 for other categories.

User profile upgrading phase, shows that the system is continually trained and improved during the test sessions. In other words, training is not stopping, because profile is greatly enriched during the sessions. In fact,

Table 1:Extracted categories from odp ontology for selected news

"*LUA*" 's categories
(Computers: Programming: Languages:Lua: Tools) (Computers: Programming: Languages:Lua) (World: Deutsch: Computer: Programmieren: Sprachen: Lua) (World: Japanese: コンピュータ: プログラミング: 言語: Lua) (World: Português: Ciência: Astronomia…)
"*Release*" 's categories
(News: Breaking News: Official Press Releases) (Computers: Software: Beta Releases) (Shopping: Weddings: Reception: Tosses and Releases: Butterflies) (Computers: Security: Advisories and Patches: Vendor Releases) (Computers: Internet: Web Design and Development: Promotion: Press Release Services) (Shopping: Health: Alternative: Life Extension: Human Growth Hormone: Releasers) (Recreation: Outdoors: Wildlife: Rehabilitation) (Business: Arts and Entertainment: Music: Labels: Specialty: Dance: Techno) (Arts: Movies: News and Media…..)
"*Dynamically-typed*" 's categories
(Computers: Programming: Languages: C++: Templates) (Computers: Programming: Languages: Clean)(Computers: Programming: Languages: Data Structured) (Computers: Programming: Languages:ElastiC) Computers: Programming: Languages: Java: Extensions: Groovy: FAQs, Help and Tutorials) Computers: Programming: Languages:Lua)
"*Scripting*" 's categories
(Computers: Programming: Languages: PHP: Scripts) (Arts: Writers Resources: Screenwriting: Script Consulting) (Computers: Programming: Languages: JavaScript: Scripts) (Computers: Programming: Languages: JavaScript: FAQs, Help and Tutorials) (Computers: Programming: Languages: PHP: Scripts: Frameworks) (Computers: Programming: Languages: PHP: Scripts: Content Management: Joomla: Extensions and Templates) (World: Japanese: コンピュータ: プログラミング: 言語: JavaScript
"*Language*" 's categories
(Computers: Programming: Languages: Fortran: Source Code: Statistics and Econometrics) Computers: Programming: Languages: PHP: Scripts) (Computers: Programming: Languages: PHP: Scripts: Content Management) Computers: Programming: Languages: Delphi: Components) (Arts: Education: Language Arts: English: Academic Departments) (Computers: Programming: Languages: Comparison and Review) (Computers: Programming: Languages: C++: Class Libraries) (Computers: Programming: Languages: JavaScript: Scripts) Reference: Dictionaries: World Languages: E: English) Arts: Education: Language Arts: English: English as a Second Language: Language Schools: Europe: United Kingdom: England: London…)
"*Object-Oriented* " 's categories
(Computers: Programming: Languages: Object-Oriented: Class-based) (Computers: Software: Operating Systems: Object-Oriented: Open Source) (Computers: Software: Operating Systems: Object-Oriented) (Computers: Programming: Languages: PHP: Scripts: Frameworks) (Computers: Programming: Languages: Object-Oriented) Computers: Programming: Languages: Scripting: Object-Oriented) (Computers: Programming: Methodologies: Object-Oriented: Criticism) (Computers: Software: Operating Systems: Object-Oriented: Java) (Computers: Programming: Compilers:Lexer and Parser Generators) (Computers: Software: Operating Systems: Object-Oriented: Syllable) (Science: Math: Software) (Business: Information Technology: Employment: Resumes: Programming)
"*Capabilities* " 's categories
(Business: Electronics and Electrical: Contract Manufacturing: Printed Circuit Boards: Fabrication) (Business: Industrial Goods and Services: Casting, Molding,) (Machining: Machine Shops: Metal Fabricators) (Computers: Programming: Languages: C++: Class Libraries) (Computers: Security: Firewalls: Products) (Computers: Software: Industry-Specific: Salon Management) (Games: Video Games: Shooter: Q: Quake Series: Quake: Modifications and Add-Ons: Engines: Quakeworld) (Computers: Software: Internet: Clients: Usenet: Windows) (Regional: Europe: United Kingdom: England: West Yorkshire: Leeds: Business and Economy: Industrial: Engineering) (Regional: North America: United States: Kansas: Localities: W: Wichita: Business and Economy: Industrial) (Regional: Europe: United Kingdom: Business and Economy: Event Planning: Management Companies…)
"*Designed* " 's categories
(Computers: Internet: Web Design and Development: Designers: Basic Service: W) (Computers: Internet: Web Design and Development: Designers: Full Service: I) (Computers: Internet: Web Design and Development: Designers: Full Service: N) (World: Italiano: Affari: Edilizia: Progettazione e Design: Architetti) (Computers: Internet: Web Design and Development: Designers: Basic Service: A) (Computers: Internet: Web Design and Development: Designers: Full Service: M) (Business: Business Services: Design: Graphic Design: Designers: Multi-Discipline: Europe: United Kingdom: England)
"*Extending* " 's categories
(Regional: North America: United States: Texas: Localities: H: Houston: Travel and Tourism: Lodging: Hotels and Motels: Extended Stay) (Regional: North America: United States: Texas: Localities: S: San Antonio: Travel and Tourism: Lodging: Hotels and Motels: Extended StayNews: Extended Coverage)

Table 1: (countiuesd)

(Computers: Hardware: Peripherals: Switching Devices) (Regional: North America: United States: Texas: Localities: A: Austin: Travel and Tourism: Lodging: Hotels and Motels) (Computers: Programming: Compilers:Lexer and Parser Generators) (Computers: Programmig: Languages: C++: Class Libraries: STL) (Computers: Programming: Languages: Prolog: Implementations) (Home: Family: Family Websites: B)
***"Application "* 's categories**
(Science: Math: Applications: Publications: Journals) (Computers: Software: Enterprise Application Integration) (Science: Math: Applications: Events: Past Events) (Science: Math: Applications: Mathematical Biology: Events: Past Events) (Computers: Internet: On the Web: Web Applications: Content Management) (Computers: Internet: On the Web: Web Applications: Photo Sharing) (Computers: Internet: On the Web: Web Applications: Bookmark Managers) (Business: E-Commerce: Developers:Database Applications) (Computers: Internet: On the Web: Web Applications: Photo Sharing: Image Hosting) (Computers: Internet: On the Web: Web Applications: Storage) (Science: Technology: Electronics: Reference: Application Notes and Data Sheets) (Computers: Open Source: Software: Internet: Web Applications)

Table 2:Category weight in the for selected news

Category	Weight
Computers: Programming: Language	0.35
Computers: Internet: Web Design and Development: Promotion	0.088
Computers: Software: Operating Systems	0.04
Regional: North America: United States:	0.03
Regional: Europe: United Kingdom:	0.03
Computers: Programming: Compilers	0.019
Arts: Education: Language Arts:	0.019
Business: Arts and Entertainment: Music:	0.019
World: Deutsch: Computer	0.009
World: Japanese:	0.009
コンピュータ:プログラミング	
Œ¾Œê	
World: Português: Ciência	0.009
News: Breaking News: Official Press Releases	0.009
Computers: Software: Beta Releases	0.009
Shopping: Weddings: Reception	0.009
Computers: Security: Advisories and Patches:	0.009
Shopping: Health: Alternative	0.009
Recreation: Outdoors: Wildlife	0.009
Arts: Writers Resources: Screenwriting	0.009
Computers: Software: Databases	0.009
Science: Math: Software	0.009
Business: Information Technology: Employment:	0.009
Business: Electronics and Electrical: Contract Manufacturing:	0.009
Business: Industrial Goods and Services: Casting, Molding, Machining	0.009
Computers: Security: Firewalls:	0.009
Games: Video Games: Shooter:	0.009
Shopping: Vehicles: Autos: Warranty	0.009
Computers: Hardware: Peripherals:	0.009
Business: E-Commerce: Developers:	0.009

online training is continually performed, while the process of profile upgrading is offline.

News recommendation: After training sessions and computing ranks of categories, new items can be recommended. So the vectors of new items and profiles are constructed based on these ranks. Afterwards, in order to compute the similarity between news items and the user profile, we will compare the corresponding characteristic vector N = ($w^n_{c1,}$ $w^n_{c2,}$ $w^n_{c3,}$ …) with the user profile P = (r^P_{c1} $r^P_{c2,}$ r^P_{c2} r^P_{c3} ,…). The similarity between the user profile P and the characteristic vector news items n, is calculated by applying the cosine measure (Salton, 1989):

$$Similarity(P, n) = \frac{P.n}{|P|.|n|} = \frac{\sum_{i=1}^{m} r_{ci}^{p} r_{ci}^{n}}{\sqrt{\sum_{i=1}^{m} (r_{ci}^{p})^2 \cdot \sum_{i=1}^{m} (r_{ci}^{n})^2}} \quad (3)$$

where, r_{ci}^{p} shows ranks of categories in the profile and r_{Ci}^{n} shows the ranks of the categories in the news item.

Consequently, the incoming news item which includes common high rank categories within the profile is highly recommended to the user though it excludes the common words with the profile.

EXPERIMENTAL RESULTS

The well-known Average of Similarity and R-Precision measures, which are defined later, have been used to check the validity of the proposed technique. Values obtained during experiments and conclusions are stated below.

Average weight of a set of news and maximum mean weight: In each session the certain number T of news items offered and the user must choose them based on his interests, which named E(T). Then, the Average of Similarity or $\overline{\upsilon(E(T))}$ is computed for the set of news selected by the user in that session. On the other hand, it can be computed a Maximum Average Similarity value or $\overline{\upsilon_{\max}(T)}$ for that set of news items. It is obtained when the chosen news (N) are the same to the first N news offered by the proposed system in a given session. The $\overline{\upsilon_{\max}(T)}$ value is computed automatically by the system. To quantify the relationship between the value $\overline{\upsilon(E(T))}$ of the news items chosen by the user and the value $\overline{\upsilon_{\max}(T)}$, the rate C_D is defined as:

$$C_D = \frac{\overline{\upsilon(S(T))}}{\upsilon_{\max}(T)} \quad (5)$$

R-precision: A simple summary value for a set of news items ranked according to their similarity can be generated by computing the precision in the R-th position of the ordered list, with R being the number of news chosen by the user among those offered by the system. Hence, the R-Precision measure is defined as:

$$RP(i) = \frac{posR(E(T))}{card(E(T_i)} \quad (6)$$

where,, card ($E(T_i)$ is the number of selected items among all recommend items and *posR* ($E(T)$)is the number of chosen news between first R news.

Evaluation of the suggested system with different users: In training sessions, a lot of items are presented to the user from 100 news sources. The user profile is initially empty and is constructed incrementally during the run. The proposed system has been tested by 6 different and heterogeneous users and every one of the 6 voluntary users has attended in 4 training and then 10 experimental sessions.

Evaluation of the results: In this subsection, BOC has been compared with BOW. In fact the sources of the news RSS are identical in both algorithms, but the incoming news is ranked based on the ranked user profiles which the proposed algorithm has constructed and the similarity measure has been computed. Hence after ranking recommended items based on their similarities, the news must be shown to the users. The results of the experiments have been shown in Fig. 2.

As we know, some of the news items have common words with the profile. Therefore, finding the related news items based on these words, is easily done, nonetheless there are some problems like polysemy, synonymy and worthless words in the news that decrease the efficiency of recommending items to some extent. Thus outputting some related news items in BOW is so hard, especially when the incoming news items relate semantically to the stored profile, without including the basic common words. This problem is what we have focused and addressed in this study. The main result in this part of evaluation is due to one reason: Storing and using the word's categories instead of the main words in a semantic profile actually improves recommendation system as there commended news items are more semantically relevant to the user's interests.

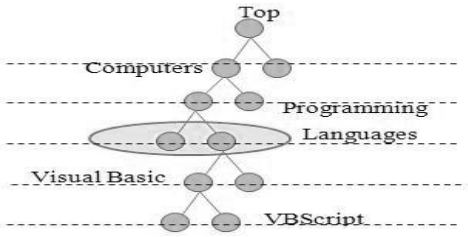

Fig. 2: Depth-three concept in ODP

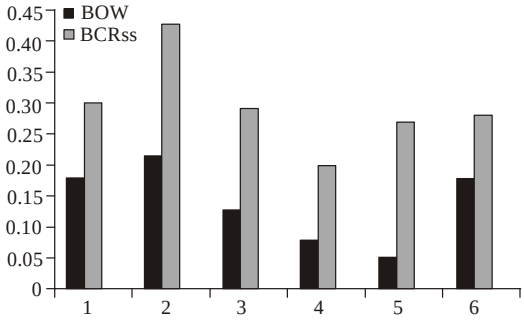

Fig. 3 : Avarage of r-precision average measure along of 10 experimental sessions, when the cosine measure is used to score the news for BOC and BOW

As the experimental results show, the overall performance of the proposed technique is better.

C_D and RP metrics measure the ability of a recommender system to produce a recommended ordering of items that matches the user's interests. These measures range from 0 to 1. The adoption of both metrics gives us the possibility to evaluate whether the systems are able to recommend good items or not and how these items have been ranked. For example, even if the top R items ranked by the systems were relevant and user selects R items, RP metric might give the highest value because the best item are actually in top rank.

Figure 3 compares Average R-Precision for BOW and BOC, which shows the average of R-Precision for 10 experimental sessions for all 6 users. This figure is also presenting the best user's results when the incoming news items semantically relate to the profiles. More precisely incoming news items include many words, which some of them conceptually relate to the user profile. It means that classical keyword based algorithm cannot recognize these kinds of relations. The best visited result is the recommendations for user #2 which has the RP measure higher than 0.4 in BOC, where,as was nearly 0.23 in BOW. The better results for user #2 are probably related to his own attention in choosing news.

Figure 4 shows the two users's results, which the user #2 has totally the best and user #4 has the worst results with CD measure. It indicates that the system gradually improves its news items recommendations for user #2 and user #4, during the sessions and it also shows that the

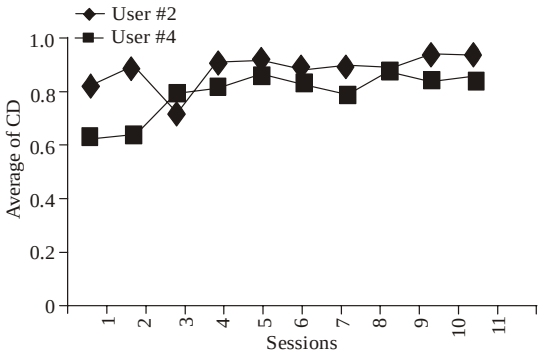

Fig. 4: Results obtained by the user #2 and by the user #4 for the CD value throughout 10 experimental sessions. It is observed a favorable evolution of the CD

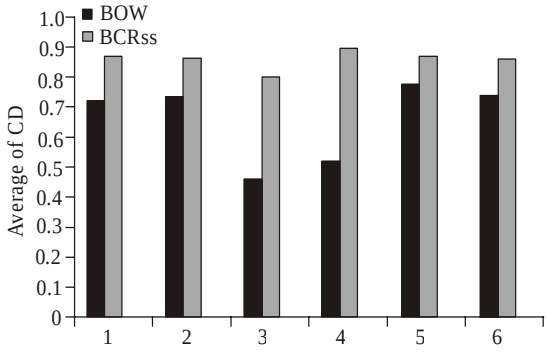

Fig. 5: C_D average measures for 6 users along 10 experimental sessions, using the cosine measure to score news by BOC and BOW

ranking offered by BOC highly matches the user's interests in the last sessions. It should be mentioned that the best quality recommendations were offered to the user #2 and the poorest ones were offered to the user #4 in all 10 sessions, which act as the upper and lower bounds in performance, thus the rest of users' results are placed between the two ones.

Figure 5 shows the average C_D for the 6 users. As it can be seen, the average of C_D along 10 sessions, are better for all users in the BOC than ones in the BOW.

One of the most important deficiencies of BOW is the high dimensions, where,as the proposed algorithm considerably reduces it, because of storing categories instead of the words in BOC phase. According to the experiments after 20 training and testing sessions, BOW profile contains 5170 number of words, where,as BOC has 3765 categories of the words. The experimental results in Fig. 6 show that BOC results are higher than BOW before session 15, but after that when the system trained, BOC has a constant growth where,as BOW has an ascending growth. This is perfectly normal because the words in the news are increasingly stored in the profile, while the categories are common between the news items,

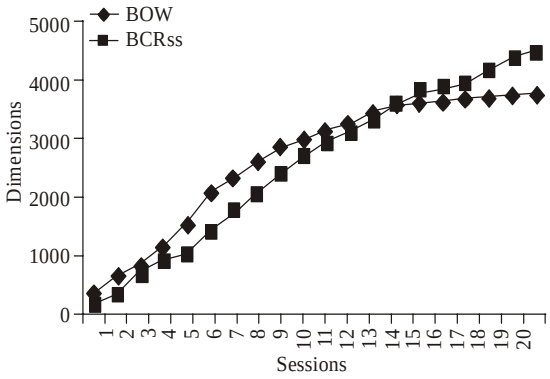

Fig. 6: Profile vector by BOC and BOW techniques

so it naturally cause a reduction in computing the similarity between the vectors. It should be mentioned that, this differs for the different users. In other words it depends on the number of user selections in each session.

CONCLUSION AND FUTURE WORKS

Recommender systems facilitate the natural social recommendation behavior and alleviate the pressure of information overload. Keyword-based techniques are unable to capture the *semantics* of the user interests and thus limit the recommended items. In order to overcome these limitations, construction of a ranked semantic profile through extracting related subjects has been proposed which:

- Stores words categories in the profile instead of the words, to reduce the dimension of the vector space.
- Serves to find some incoming items, semantically related to the user's interests.
- Ranks the contents of the profile to make an ordered profile based on the user's interest.

The proposed technique works in two main phases:

- Extracting subject from user profile based on ODP ontology driven reasoning,
- Ranking the categories in the profile based on their frequencies.

According to the experimental results, the incoming news items which semantically relate to the user profile are highly recommended to the user, though they exclude the interested words. In these conditions BOC algorithm is more effective than keyword-based algorithm BOW.

Finally, it should be mentioned that the program implementation for the proposed technique has been installed on the client side, which causes high user's

information security, as well as enough space to save data and offline processing on the client side. As a future work, a new ranking algorithm can be considered to yield better performance in profile ranking phase.

REFERENCES

Asnicar, F. and C. Tasso, 1997. A prototype of user model-based intelligent agent for documentation filtering and navigation in the word wide web. In Proceeding of 1st Int. Workshop on adaptive systems and user modeling on the WWW, pp: 2-5.

Burke, R., 2002. Hybrid recommender systems: Survey and experiments. User Model. User-Adap., 12(4): 331-370.

Joachims, T., 2002. Optimizing search engines using clickthrough data. In Proceedings of SIGKDD, pp: 133-142.

Koutrika, G. and Y. Ioannidis, 2005. A Unified User Profile Framework for Query Disambiguation and Personalization. Proceedings of Workshopon New Technologies for Personalize, pp: 44-53.

Lee, W.S., 2001. Collaborative Learning for Recommender Systems. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp: 314-321.

Lieberman, H., 2003. Autonomous Interface Agents. In ACM Conference on Human-ComputerInterface, ACM Press, New Yoork, pp: 66-74.

Magnini, B. and C. Strapparava, 2001. Improving user modelling with content-based techniques. In Proc. 8th International Conference User Modeling, 21: 74-83.

Mc Gowan, J.P., 2003. A multiple model approach to personalized information access. M.Sc. Thesis, in Computer Science, Computer Science Department, University College Dublin.

Middleton, S., D. De Roure and N. Shadbolt, 2001. Capturing Knowledge of user Preferences: Ontologies in Recommendersystems. In: Proceedings of the 1st International Conference on Knowledge Capture, Victoria, BC, Canada, pp: 100-107.

Mladenic, D., 1999. Text-learning and related intelligent agents: A survey. IEEE Intell. Syst., 14(4): 44-54.

Pazzani, M. and D. Billsus, 1997. Learning and revising user profiles: The identification of interesting web sites. Mach. Learn., 27(3): 313-331.

Rhodes, B., 1997. Just-In-Time Information Retrieval. 39, pp: 685-704.

Rodriguez, M.D., B. Gomez-Hidalgo, J.M. and B. Diaz-Agudo, 1997. Using wordnet to complement training information in text categorization. In 2nd International Conference on Recent Advances in NLP, pp: 150-157.

Salton, G., 1989. Automatic Text Processing, the Transformation, Analysis and Retrieval of Information by Computer. Reading, Addison-Wesley, MA.

Samper, J. and A. Castillo, 2008. NectaRSS, an intelligent RSS feed reader. J. Netw. Comput. Appl. Archive, 31(4): 793-806.

Shardanand, U. and P. Maes, 1995. Social Information Filtering: Algorithms for Automating/Word of Mouth. In: Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, 1: 210-217.

Shen, X., B. Tan and C. Zhai, 2002. Ucair: Capturing and exploiting context for personalizedsearch, Proceedings of the Burke, R.: Hybrid recommender systems: Survey and experiments. User Model. User-Adap., 12(4): 331-370.

Sieg, A., B. Mobasher, R. Burke, G. Prabu and S. Lytinen, 2005. Representing user informationcontext with ontologies. Proceedings of HCI International Conference.