Journal of Computer Science 7 (4): 573-581, 2011
ISSN 1549-3636

An Ontology Based Approach to Implement the Online Recommendation System

[1]Vijayakumar Mohanraj and [2]Muthaial Chandrasekaran
[1]Department of Information Technology, Sona College of Technology, Salem-636005, India
[2]Department of Electronics and Communication Engineering, GCE, Salem, India

Abstract: Problem statement: Every web user has different intent when accessing the information on website. The primary goal of recommendation system is to anticipate the user intent and recommend the web pages that contain user expected information. Effective recommendation of web pages involves two important challenges: accurately identifying the user intent and predict the result show that novel web usage mining method and ontological concept scoring algorithm based on website domain ontological profile helps the recommendation system imminent navigation pattern in such a way that it provides required content while users browse the predicted navigation. **Approach:** We present a ontology based approach to implement recommendation system that involves applying innovative web usage mining on log system to discover all possible imminent navigation patterns of current user and resolve any uncertainties in discovering the navigation pattern by applying ontological concept based similarity comparison and scoring algorithm. **Results:** result show that novel web usage mining method and ontological concept scoring algorithm based on website domain ontological profile helps the recommendation system to predict and present most relevant navigation pattern to users. **Conclusion:** our recommendation system confirms that ontology based approach should be used to ensure excellent accuracy in predicting and capturing future navigation pattern of web user.

Key words: Web usage mining, recommendation system, domain ontology, information retrieval, ontology based approach, Greatest Common Subsequence (GCS), Live Session Window (LSW), Imminent Browsing Pattern (IBP)

INTRODUCTION

With the increasing popularity of Web, we have witnessed explosive growth in the data accessible on the web. It become mandatory for all commercial sites to provide personalized access of information based on user intent. Otherwise, user is loaded with burden of information. Users might not find the more relevant information during their website visit. Thus, online recommendation system has attracted interest in research community to capture future navigation of user (AlMurtadha *et al*., 2010) and personalize the user's web experience.

The aim of any recommendation system is to assist users to find needed information in the easiest way via helping them to capture pages otherwise they might not find the relevant information during their visit. Web Usage Mining and Web Structure Mining (Kumar and Singh, 2010) are playing a crucial role to implement such recommendation system. Web Usage Mining (Kosala and Blockeel, 2010) is the discovery of web server access pattern. It can be used to obtain some interesting pattern about user behavior from usage Logs. Recently, all the ecommerce sites focus on providing personalized access to user on their site with the implementation of recommendation system (Baraglia and Silversti, 2004).We consider three essential elements that collectively play a critical role in personalized access of website. These three independent elements are user short term information such as user's current navigation pattern, semantic knowledge about the website domain being investigated and the user's navigation profile captured with the web usage mining as long term interest.

Recently, a number of studies have been proposed to capture the forthcoming navigation pattern of web users. We have conducted investigation on different WUM system and architecture that can be matched with our proposed system. Analog is one of the first web usage mining systems. It consist of online and offline

Corresponding Author: Vijayakumar Mohanraj, Department of Information Technology, Sona College of Technology, Salem-636005, India

component. The offline component builds session clusters by analyzing user navigation pattern recorded in the log file. In the online component part, active user session is classified according to the generated model. The classification identifies the pages that match with the active session and returns the requested page with list of suggestion. Clustering approach of system affected by several limitation especially scalability and accuracy. There is variety of clustering algorithms available for usage. Each approach could have different type of cluster [Exclusive (K-Means), Overlapping (Fuzzy C Means) and Hierarchical].It's difficult to compare the performance of algorithm on large dataset like web log. In addition, Clustering approach used in all recommender systems needs to be back up by good classification method. Analog did not have the proper classification approach over the overlapping cluster.

Another WUM system called SUGGEST (Baraglia Silversti, 2007) provides useful information to optimize web server performance and make easier for the web user navigation. SUGGEST adopts a two level architecture composed by an offline creation of historical knowledge and online component that understands the user's behaviour. SUGGEST uses the markov model for calculating the probability of a page the web user visit in future after visiting pages in the same session. This system uses the high order markov model to improve the accuracy. However, the system can't be used for web site made up of large number of pages due to high space complexity. The limitation of system might be (a) the memory required to store web server pages is quadratic in the number of pages. It is a sever limitation in the huge web site. (b) SUGGEST does not permit us to manage web site made up of pages dynamically generated. A Web personalizer system provides dynamic recommendation, as a list of hyperlinks to users. In the Web personalizer system, analysis is based on the usage data combined with structure formed by the hyperlinks of site. Aggregated usage profile is obtained by applying data mining technology [i.e., clustering, association rule] in pre-processing phase. In this phase web server logs are converted into cluster made up of set of pages with the common usage characteristic. The online phase considers the active user session in order to find match among user activities and discover usage profile. Matching entries are then used to compute a set of recommendations which will be inserted into the last requested page as a list of hypertext links. Webpersonalizer is good example two tier architecture for personalized system. However the accuracy of the Webpersonalizer is affected by association rule mining used for discovery of frequent item set in web log data. The main problem with the association rule mining method is discovery of contradictory association rules. As a result of inconsistent rules, predicting accuracy of system is degraded. Even the non redundant association rule mining algorithm does not help the system because of the web log data nature where the number of page hits is so high.

MATERIALS AND METHODS

Our survey reveals that there is a race for finding architecture concept (Baraglia and Silversti, 2007) that helps to improve the accuracy of online recommendation system in capturing imminent navigation pattern of online users. In our study, we proposed the novel architecture for online recommendation system that uses the novel user navigation classification approach and website ontology concept scoring algorithm for improving accuracy. The website ontology concept score is updated based on the user's ongoing behavior. Our result shows that ontology concept score update propagation method is really contributes to online recommendation system in achieving excellent accuracy in the prediction.

Different components of Ontology based online Recommendation System Architecture: Our goal is to utilize the user context to personalize the web access by predicting the future navigation of that user based on the novel classification and ontological concept. Figure 1 displays our ontology based approach to implement the online recommendation system. The functionality of each component in our proposed Recommendation system architecture is explained in the following subsection.

Log system: The main role of the log system is to prepare the usage data of web site for the web usage mining. When user access information on web pages, all the accessed information are recorded in the log file. These log files are pre processed and given as input to the navigation pattern modelling. The pre-processing of web logs is very important functionality and also time demanding. It comprises of four different tasks (1) Data collection: A flat file was constructed from original weblog file. Each record of the file consists of time, ip address, name, requested resource (URL) and HTTP Status code. (2) Data Cleaning: In this step, we perform the removal of all the data tracked in web log that are useless for mining purpose such as Navigation session performed by robots and web spider. (3) Session Identification and reconstruction: it involves (i) Identifying the different users session from usually very poor information available in log files and (ii) Reconstructing the user's navigation path within the identified session.

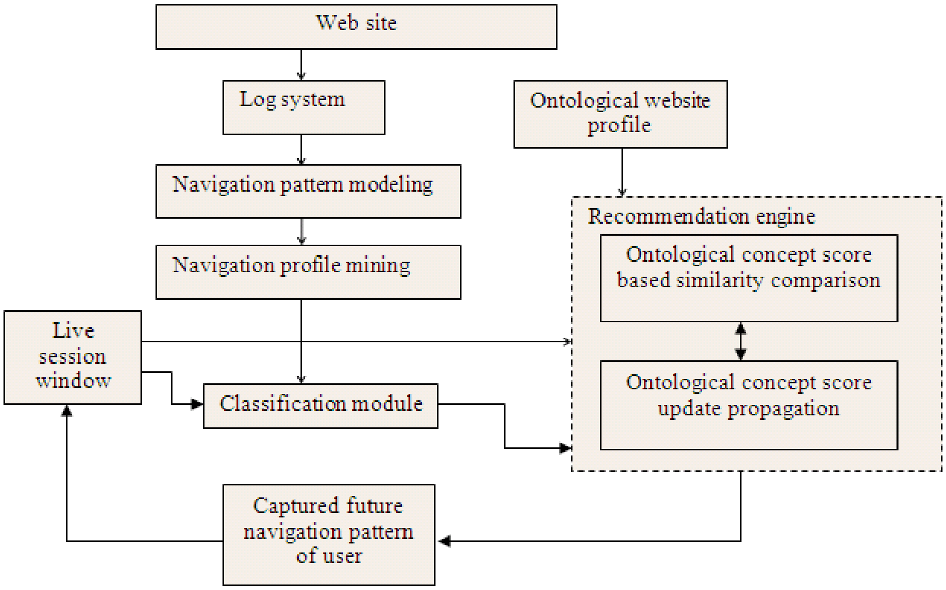
Web site
Log system
Ontological website profile
Navigation pattern modeling
Recommendation engine
Ontological concept score based similarity comparison
Navigation profile mining
Live session window
Ontological concept score update propagation
Classification module
Captured future navigation pattern of user

Fig. 1: Components of ontology based online recommendation system architecture

(4) Content and Structure Retrieving: Mostly all WUM uses the visited URL as the poor source of information. They do not convey any information about actual page. We employ the content based information to enrich web log data in the form of maintaining ontology for each web page. This information is used by the recommendation system to choose the best classification.

Navigation pattern modeling: A Log system provides the web pages accessed information across various sessions which are modeled as undirected Graph G = (V, E). The set Vertex (V) identifies the different web pages hosted on the web server model. The edge weights are determined by the following Eq. 1:

$$WP_{ij} = \frac{C_{ij}}{Max\{C_i, C_j\}} \qquad (1)$$

Where:

C_{ij} = The number of session containing both pages i and j

C_i and C_j = Respectively the number of sessions containing only pages i or page j

Dividing by the maximum between single occurrences of two pages has the effect of reducing the relative importance of links involving index pages. Such pages are those that generally do not contain useful content and are used only a starting point for a browsing session.

The edge weights (WP_{ij}) are kept in the adjacency matrix WA where each entry WA_{ij} contains the value computed according to equation (1). To limit the number of edge in such graph, elements of WA_{ij} whose value is less than a threshold are known to be less correlated and thus discarded.

Navigation profile mining: We perform clustering algorithm which aims to group session into clusters. These patterns will be further used to facilitate the user profiling process of the system. We apply a graph partitioning algorithm to find groups of strongly correlated pages by partitioning the graph according to its connected components. Clusters are formed by starting from a Vertex a DFS on the graph induced by WM is applied to search for the connected component reachable from this vertex. Once the component has been found, the algorithm checks if there are any nodes not considered in the visit. If it so, it means that a previously connected has been split and therefore, it needs to be identified. To do this, DFS is again performed by starting from one of the nodes not visited.

Classification module: The critical component of our system is classification module. Main inputs for the Classification module are:

- Navigation pattern profile: It consists of clusters of web pages formed in the Navigation profile mining of our system

- Live session window: A Sequence LSW={lwp_1,lwp_2,....,lwp_m} is the set of web pages in live session window where m is the size of the current active session window

Input: Live Session Window LSW = {lwp_1, lwp2… lwpm} and Navigation Profiles contains n clusters with set of web pages NP = {NP_1, NP2, … , NPn}
Output: Navigation Profile that contain possible forthcoming browsing pattern of user along with profitable score

Algorithm 1:

1. Live Session Window (LSW) is given as input to the system.LSW is the set of web page visited by user in the live session. LSW is represented as {LWP_1,$LWP_{2,}$,...,LWP_n) where 'n' is the size of session window.
2. Initially assign the arbitrary profitable score of value 100 to each Agent (A_i). The score is denoted as score (A_i).This score is updated by the respective Agent on the discovery of subsequence.
3. For each Agent on its assigned cluster of Navigation Profile does the following:
   i. Each Ai executes the Greatest Common subsequence Detection (described in next section) on its cluster in respect to Live Session Window (LSW) which produces the highest degree of GCD as the discovered subsequence which could be the candidate of online user's imminent browsing pattern.
   ii. Discovered sequence is denoted as IBP = {IBP_1, $IBP_{2,}$,., IBPn}.Each Agent updates its initial score using Eq. 2 and with the help of adjacency matrix WA built in the Navigation pattern modeling where each entry WA_{ij} contains the value computed according to Eq. 1:

$$\Delta\text{score}(A_i) = \text{score}(A_i) + \sum_{i=1}^{n}\sum_{j=1}^{m} WA_{IBPiLWPj} \qquad (2)$$

   Where $WA_{IBPiLWPj}$ = Value in Adjacency matrix between the each page in Imminent Browsing Pattern (IBP) discovered by Agent and pages in the Live Session Window (LSW).

   iii. After each Ai executed the steps i and ii, Agent sends its updated score and discovered subsequences.
4. After receiving the profitable scores from each agent (A_i), it selects the first 3 High scored Agent's output. The scores are denoted as PS_1, PS2 and PS_3.
5. Module computes the absolute difference between the PS_1, PS2 and PS_3 to find the closeness. IF $|PS_1\text{-}PS_2|$ or $|PS_1\text{-}PS_3|$ or $|PS_1\text{-}PS_2|$ $\leq\beta$ Where β is Uncertain Profitable Threshold value then There is race between discovered subsequence of PS_1 or PS_2 or PS_3 to become an imminent browsing pattern of web user. This can be resolved with the help of Recommendation Engine ELSE Agent chooses the PS_1's sequence as best discovered subsequence.
6. Classification module reports the predicted forthcoming browsing pattern as PS_1 Or Competing discovered subsequence as input to Recommendation Engine in the case of competition.
7. Suppose, if the next user activity in live session window different from the suggested captured list then the system has to restart once again to classify the new user activities.

Based on the above algorithm and upon receiving URL's in the form of Live Session Window, Our classification module of our system acts on the Navigation profiles by executing the novel Greatest Common Subsequence Detection (discussed in subsection) to discover the subsequences which may be considered as possible imminent browsing pattern of web user. This module submits the profitable score along the discovered subsequence to the recommendation engine. When the recommendation engine receives the profitable score along the sub sequences, it starts to decide the best profitable source of navigation profile. In the case of close race between the sub sequences, Ontology concept score based Similarity Comparison plays a crucial role in selecting the right cluster. The main objective of Recommendation Engine is to test the each discovered navigation pattern against already built ontology concept profile and choose the web pages which are similar to current ontology profile. Finally, our system suggests the output of Recommendation Engine as the imminent browsing pattern of web user. The above algorithm 1 depicts the working of classification module to capture the possible forthcoming browsing pattern of user in our system architecture.

Greatest common subsequence detection: Every Agent initiated by classification module should perform the similarity comparison between set of pages in Live Session Window and web pages in the cluster to discover the subsequence that could be the forthcoming browsing pattern of online user. It's clear that every agent has to perform some kind of pattern matching. In the pattern matching, comparing the similarity between the two sequences $\vec{a}$ and $\vec{b}$ are fundamental problem. One of the fundamental problem is to determine the Greatest Common Subsequence (GCS) between $\vec{a}$ and $\vec{b}$. The GCS is a String comparison metric that measures the subsequence of maximum length common to both the sequences. Main objective of Agent (A_i) is to find the Greatest Common Subsequence among the sequence of paths in the form of page visits A= $\{a_1,a_2,a_3,a_4,......a_m)$, B=$\{b_1,b_2,b_3,b_4,...b_n)$.

Theorem 1: Let a= $\{a_1,a_2,a_3,a_4,......a_m)$ and b=$\{b_1,b_2,b_3,b_4,...b_n)$ be the sequences and Let c=$\{c_1,c_2,c_3,...c_k\}$ be any GCD of a and b:

- If $a_m=b_n$ then $c_k=a_m=b_n$ and c_{n-1} is a GCD of a_{n-1} and b_{n-1}
- If $a_m \neq b_n$ then $c_k \neq a_m$ implies that c is a GCD of a_{m-1} and b
- If $a_m \neq b_n$ then $c_k \neq b_n$ implies that c is a GCD of a and b_{n-1}

We have implemented the GCD with added module that outputs the subsequence of indices of the two sequences that match in getting the Greatest Common subsequence If A = $\{wp_1,wp_2,wp_3,wp_5,wp_4,wp_2,wp_1\}$ and B =$\{wp_2,wp_4,wp_3,wp_1,wp_2,wp_1\}$. Their GCD is GCD= $\{wp_2, wp_4, wp_2, wp_1)$. This Greatest Common Subsequence (GCS) detection method is used in classification module by agents to discover the navigation profile that contains possible future browsing pattern. It finds the navigation profiles for which LSW is the Greatest Common Subsequence. All the discovered navigation profiles which have closeness between them will send to the Recommendation Engine.

In the case of Uncertainty, Recommendation Engine is used to find the similarity between web pages of LSW and web pages of discovered navigation profile using the ontological concept. After finding the similar web pages in Navigation profile in respect to the web pages of LSW, it performs the ranking of similar web pages based on the concept score and recommends those to online user as imminent browsing pattern of online user.

Ontological website profile: Ontology is an explicit specification of concepts and relationship that can exist between them. When the knowledge of a website domain is represented in declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects and the desirable relationships among them are reflected in representational vocabulary with which a knowledge-based program represents knowledge.

The personalization is improved by performing similarity comparison between the web pages of discovered navigation profile and the web pages in current live session window based on the ontological concept. All the discovered similar pages are ranked according to the concept score and recommended to online user as imminent browsing pattern.

Our recommendation model for user is represented as an instance of our reference website domain ontology in which concepts are annotated by concept score derived and updated implicitly based on the user's web site access behavior. We call this as Ontological website profile. In our approach, the purpose of using an ontology website profile is to identify topics that might be of interest to the current online user. Therefore, we define our ontology as a hierarchy of topics, where the topics are utilized for the classification and categorization of web pages. The hierarchical relationship among the concept is taken into considerations for building the ontological website profile as we update the annotated concept score for existing concepts using the ontological update propagation. Ontological website profile is an annotated instance of reference website domain ontology. Each concept in the profile is annotated with concept score which has initial value of one. As the user interacts with the system by selecting or viewing new document, the ontological profile is updated and the annotations for existing concepts are modified by update propagation. Thus, the user context is maintained and updated incrementally based on user's ongoing behavior.

Our current implementation uses the Sona Directory Project, which is organized into hierarchy of topics and web pages that belong to these topics. We utilize the web pages as training data for the representation of the concepts in the reference ontology. The textual information that can get extracted from web pages explains the semantics of the concepts and is learned as we build term vector representation (Castells *et al*., 2007) for the concepts. We create an aggregate representation of the reference ontology by computing a term vector $\vec{n}$ for each concept n in the concept hierarchy. Each concept vector represents, in aggregate form, all individual document indexed under the concept, as well as its entire sub concepts. A global dictionary of terms is extracted from documents indexed under each concept. A stop list is used to remove high frequency, but semantically non-relevant terms from the content.

Each document d in the training data is represented as a term vector $\vec{d}$ = {w_1,w_2,…, w_k}, where each term weight (W_i) is computed using term frequency and inverse document frequency. The term count in the given document is simply the number of times a given terms appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j. Thus we have the term frequency, defined as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (3)$$

where, $n_{i,j}$ is the number of occurrences of the considered term (t_i) in document d_j and the denominator is the sum of number of occurrences of all terms in document d_j, that is, the size of the document | d_j | . The inverse document frequency is a measure of the general importance of the term obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm and quotient):

$$idf_i = \log \frac{|D|}{|\langle d : t_i \in d \rangle|} \quad (4)$$

With $|D|$:cardinality of D, or the total number of documents in the corpus $|\langle d : t_i \in d \rangle|$:Number of documents where the term t_i appears.

Then:

$$W_i = tf_{i,j} * idf_{i,j} \quad (5)$$

We further normalize each document vector, so that $\vec{d}$ represents a term with unit length. Our Sona directory Project is the website for users to browse and download study and entertainment materials, The concept scores for the concept are updated with update propagation using an input term vector which is built from Live Session Window. Each node is a pair (C_j, $CS(C_j)$) where C_j is a concept in the reference ontology and $CS(C_j)$ is the concept score annotation for that concept. The input term vector represents the active interaction of the user, such as web pages visited in the current session.

Concept update propagation module: This module performs the incremental updation of concept score based on the user's ongoing behavior. Therefore, the ontological profile is treated as semantic network and the concept scores are updated based on instigation values. In our approach, we use update propagation, depicted in Algorithm 2, for the sole purpose of maintaining concept scores. The algorithm has an initial set of concepts from ontological profile. These concepts are assigned an initial instigation value.

The main idea is to instigate other concepts following a set of weighted relations during propagation and at the end obtain set of concepts and their respective instigation. As any given concept propagates its instigation to its neighbors, the weight of the relation between the origin concept and destination concept plays an important role in the amount of instigation that is passed through the network. Thus, a one-time computation of the weights for the relations in the network is needed.

Since the nodes are organized into concept hierarchy derived from the domain ontology, we use the measure of containment to find weights for the relations between each concept and all of its sub concept. The containment weight produces ranges of values between zero and one such that a value of zero indicates no overlap between the two nodes whereas a value of one indicates complete overlap.

The weight of the relation W_{ij} for the concept i and one of its sub concept j is computed as:

$$W_{ij} = \frac{\overrightarrow{n_i} . \overrightarrow{n_j}}{\overrightarrow{n_i} . \overrightarrow{n_i}} \quad (6)$$

where, $\overrightarrow{n_i}$ is term vector for concept i and $\overrightarrow{n_j}$ is the term vector for sub concept j. After the completion computation, once again we process weights to ensure the total sum of weights of relations between a concept and its entire sub concept equals to 1. The main input for the Algorithm 2 is set of web pages accessed in the current session window. The algorithm considers in turn each of the documents assumed to represent the current intent of the user. In all the iteration, the initial instigation value for each concept is reset to zero. We compute a term vector for each document wp_i in LSW and compare the term vector for wp_i with the term vectors for each concept C_j in ontology concepts using a cosine similarity measure. It is a measure of similarity between two vectors by finding the cosine of the angle between them, used to compare documents:

$$\cos(\theta)\ \frac{D.C}{\|D.C\|} = \frac{\sum_{i=1}^{n} D_i \times C_i}{\sqrt{\sum_{i=1}^{n}(D_i)^2} \times \sqrt{\sum_{i=1}^{n}(C_i)^2}} \qquad (7)$$

Input: Ontological profile with concept score and set of web pages in the live session window

Output: Ontological profile with updated instigation value CONCEPTS = $\{C_1, \ldots, C_n\}$, Concepts with concept score $CS(C_j)$,LSW = $\{wp_1, \ldots, wp_n\}$, Live Session Window containing web documents accessed by user in the current session

Algorithm 2:

```
foreach $d_i \in$ LSW do
      Initialize the priority Queue;
      foreach $C_j \in$ CONCEPTS do
       Cj. Instigation=0
      end
      foreach $C_j \in$ CONCEPTS do
       Calculate Sim($wp_i$,$C_j$);
       if Sim($wp_i$,$C_j$) > 0 then
             Cj. Instigation = CS( $C_j$ ) * Sim($wp_i$,$C_j$);
             priorityQueue.Add($C_j$);

       else
             Cj. Instigation = 0;
      End;
While priorityQueue.Count> 0 do
            Sort priority Queue in descending order of instigation value
            $C_S$ = priority Queue (0);
            priorityQueue.Dequeue($C_s$);
            if passRestriction($C_s$) then
            linkedConcept= GetLinkedConcept(Cs)
            foreach $C_k$ in linked concepts do
             $C_k$. Instigation + = $C_k$. Instigation *
             $C_k$.weight;
             priorityQueue.Add($C_k$)
            end
            end
      end
```

Those concepts with a similarity score, Sim(wp_i,Cj) greater than are added to priority queue in descending order based on concept's instigation value. The instigation value for concept C_j is assigned to CS(C_j) * Sim(wp_i,C_j). The concept with the highest instigation value is then removed from the queue and processed; if the current concept passed through restriction i.e. (concept with instigation value greater than the threshold), it propagates the instigation value to its neighbors. The amount of instigation that is propagated to its neighbor concept node is determined using the weight of the relation between the concepts according to the equation (6). The neighboring concepts which are activated and are not currently in the priority queue are added to the queue, which is then reordered. The process repeats until there are no further concepts to be processed in the priority queue. The neighbors for the propagating concepts are considered to be the linked concepts. We can ensure the algorithm process each edge only once by iterating over the linked concepts only one time.

After the execution of Algorithm 2, each concept in website ontology profile has updated instigation value according to online user intent. This instigation value is used to update the concept score First the resulting instigation value is added to the existing concept score. The concept scores for all the concepts are then treated as vector, which is normalized to predefined constant length k. The effect of normalization is to prevent the concept score from continuously escalating throughout the network. As the online user expresses interest in one set of concept, the concept score for the other concept has to decrease. The concepts in the ontological profile are updated with normalized concept score.

Ontology concept score based similarity comparison: This is very important functional component in Recommendation Engine. According to the Algorithm 3, we are performing similarity comparison between the web pages of discovered navigation profile and the web pages in current live session window based on the ontological profile built in the previous step. All the discovered similar pages are ranked according to the concept score and recommended to online user as imminent browsing pattern of that user. Each webpage wp_i in the predicted navigation profile is compared with the ontological concept C_j of web pages in Live Session Window using the Sim(wp_i,C_j) . For that a term vector is computed for each document $wp_i \in$ NP, where NP is set of web pages predicted in the classification module. To calculate the rank score for each document, first the similarity of the document with the concepts and respective concept score is used. Once all documents in the predicted navigation profile have been processed and added to Queue, all the web pages in the queue are sorted in descending order with respect to the rank score. Finally, our proposed recommendation engine suggests the web pages in the queue as captured future navigation pattern to user.

Input: Ontological profile with concept score and set of web pages visited in the live session window LSW = $\{wp_1, \ldots, wp_n\}$ and Predicted Navigation profile from classification module NP = $\{wp_1, .., wp_k\}$

Output: Ranked and capture imminent browsing pattern of online user

CONCEPTS= $\{C_1, C_2, \ldots, C_n\}$ of web pages in live session window

$CS(C_j)$, Concept score ; NP = $\{ wp_1, \ldots, wp_k \}$; LSW = $\{ wp_1, \ldots, wp_k\}$

Algorithm 3:

```
foreach $wp_i \in$ NP do
    foreach $C_j \in$ CONCEPTS do
        Calculate $Sim(wp_i,C_j)$
        If $Sim(wp_i,C_j) > 0$ then
         CapturedPriorityQueue.Add($wp_i$)
         rankscore($wp_i$) = rankscore($wp_i$) + CS($C_j$) *
              Sim($wp_i$,$C_j$)
    end; end;
```

Arrange the captured web pages of CapturedPriorityQueue in Descending order with respect to Rankscore and Recommend the Ranked web pages as user's imminent browsing pages

RESULTS AND DISCUSSION

Our proposed ontology based approach to implement online recommendation system is tested on the web log dataset of Sona Project Directory. Our Sona Directory Project is the website for users to browse and download study and entertainment materials this site is deployed in the IBM's Web sphere Application Server. All the algorithms are implemented in the Java language. Our system is tested on the weblog entries of users over a period of 12 weeks. There are approximately 92,745 entries in the log file. For our input dataset, Navigation pattern mining has produced the navigation profile output which consists of 22 clusters. The performance of the proposed system is analyzed based on the two metrics. They are namely Accuracy and Coverage.

Accuracy of ontology based recommendation system: Accuracy measure is defined as a degree to which captured imminent browsing pattern as suggested by the system matches with the actual browsing pattern of user. It is given by Eq. 8. The Fig. 2 depicts the comparison of our recommendation system with the other two system namely Web personalizer and SUGGEST:

$$\text{Accuracy} = \frac{\left|P(IBP_{np}, LSW) \cap \text{Original}_{np}\right|}{\left|P(IBP_{np}, LSW)\right|} \qquad (8)$$

Where:

LSW = Live session window

$P(IBP_{np}, LSW)$ = PREDICTED imminent browsing pattern of user

Orignal_{np} = Original Navigational pattern of user

Coverage of ontology based recommendation system: Coverage measure is defined as the ability of our system to produce all page views that are most likely visited by the user. It is given by:

$$\text{Coverage} = \frac{\left|P(IBP_{np}, LSW) \cap \text{Original}_{np}\right|}{\left|\text{Original}_{np}\right|} \qquad (9)$$

The Fig. 3 depicts the coverage of our system as Live Session Window (LSW) size increases.

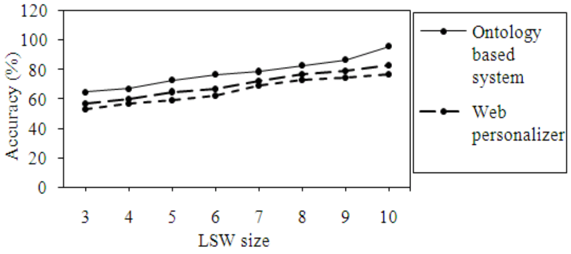

Fig. 2: Accuracy of our Ontology based online recommendation system Vs others

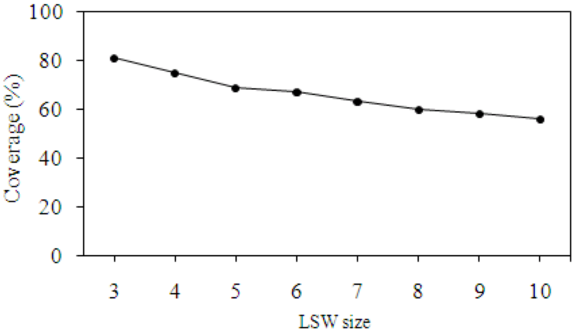

Fig. 3: Coverage of our ontology based online recommendation system

CONCLUSION

We presented ontology based recommendation framework for identifying the user intent and predict future browsing pattern of online user using novel web usage mining method. In our system, we have proposed the Greatest Common Subsequence Detection method to classify user and capture the imminent browsing pattern of user. In addition, Recommendation Engine uses the ontological profile based similarity algorithm to choose the best in the predicted subsequence and thereby completely avoids the misclassification in finding the forthcoming browsing pattern compared with other system. In addition, it ranks the predicted web pages based on the rank score and recommend the predicted web pages to user. In future, we try to propose a system that will allow us to monitor ontological profile over time to ensure the incremental updates to concept scores accurately reflects the changes in user interest.

REFERENCES

AlMurtadha, Y.M., M.N.B. Sulaiman, N. Mustapha and N.I. Udzir, 2010. Mining web navigation profiles for recommendation system. Inform. Technol. J., 9: 790-796.

Baraglia, R. and F. Silversti, 2004. An online recommender system for large web sites. Proceedings of the International Conference on Web Intelligence, Sept. 20-24, Beijing, China, pp: 199-205.

Baraglia, R. and F. Silversti, 2007. Dynamic personalization of web sites without user intervention. Commun. ACM, 50: 63-67. DOI: 10.1145/1216016.1216022

Castells, P., M. Fernandez and D. Vallet, 2007. An adaptation of the vector-space model for ontology-based information retrieval. IEEE Trans. Knowl. Data Eng., 19: 261-272.

Kosala, R. and H. Blockeel, 2010. Web mining research: A survey. ACM SIGKDD Exp., 2: 1-15. DOI: 10.1145/360402.360406

Kumar, P.R. and A.K. Singh, 2010. Web structure mining: Exploring hyperlinks and algorithms for information retrieval. Am. J. Applied Sci., 7: 840-845. DOI: 10.3844/ajassp.2010.840.845