An Auto-Recommender Based Intelligent E-Learning System

Abdallah Gomah **Samir Abdel Rahman** **Amr Badr** **Ibrahim Farag**

Cairo University, Faculty of Computers & Information, Cairo, Egypt

Summary

Internet has been universally recognized as a medium for network-enabled transfer of information and knowledge in various areas. Taking advantage of the continuously improving, web-based learning systems plays an important role for self-learning. Nevertheless, learning systems do not generally adapt to learners' profiles. Learners have to spend a lot of time before reaching the learning goal that is compatible with their knowledge background. In this paper, we propose architecture for intelligent E-Learning system architecture which helps learners taking advantage of the web massive contents through the use of an automated assistant (a recommender system). The proposed recommender system is based on a novel feature reduction technique with the goal of being able to adapt according to the learner's interest in the general domain without being specific to a single domain.

Key words:

Auto Recommender, E-Learning, Artificial Intelligence and Tutoring Systems.

1. Introduction

Learning is an important mechanism for organizations of any kind to enhance the skills of their [1].

With the increasing use of telecommunication technologies, the Internet has been widely recognized as a valuable and inexpensive medium for transfer of skills, information, and knowledge.

E-Learning technology, which delivers educational material electronically via the Internet, has been widely used in both academic education and corporative training [2].

In recent years there has been an explosion in the number of publicly available web pages ranging from well-structured web pages to very poorly designed ones. Also it is common to get a huge number of web pages that seem similar to what you are interested in, only to discover that they are not. With so many sites, it is easy to get lost, so, there is a need for automatic assistant systems to help people navigating the internet and reaching their preferred web pages.

Web users usually face a problem in finding web pages related to their real interests rather than web pages related to a single term or sentence, or related to a general category.

We are in need to have an auto recommender to guide learners and collect valuable materials from the internet.

1.1. E-Learning

The meaning of the letter 'e' is vast and encompasses many fields – from astronomy to video games. Used in technology, 'e' means electronic. E-learning, then, is e-(lectronic) learning, just as e-mail is e-(lectronic) mail. The 'e' represents the means by which we receive or access learning – electronically, typically on the Web (online) via a Web browser [3].

1.2. Recommender Systems

Recommender systems have proven to be an important response to the information overload problem by providing users with more proactive and personalized information services. Recommender systems usually track user's behavior and collect information about items the user seems to be interested in so that they can build a model of what user likes.

The recommendation process is usually done using a user profile (set of ratings, interests, preferences, demographical info, etc…), then based on the profile new items are compared against it and a decision is issued either by saying whether this item is recommended or not or by giving a rating value to the item to be recommended.

Recommender systems have become an important research area since the appearance of the collaborative filtering in the mid-1990s. The interest of this area is still high. Practical applications try to help users to deal with information overload and provide personalized recommendations, content, and services to them.

2. Motivation

Research on e-learning has gained more and more attention thanks to the recent explosive use of the Internet. However, the majority of current web-based learning systems are closed learning environments, where courses and materials are fixed and the only dynamic aspect is the organization of the material that can be adapted to allow a relatively individualized learning environment.

With the vast amount of web pages appearing daily and the explosive growth of information available on the net, it is difficult for a user to reach the information s/he needs

Manuscript received January 5, 2011
Manuscript revised January 20, 2011

while browsing the internet as s/he will find many non-relevant content.

There has been a big need of a **recommender** system for an intelligent e-learning system that can help users to browse different web pages with diversity in topics and domains, and provide a lot of alternative resources to the learners helping them make good use of the course they are obtaining. The provided resources should be provided automatically without the interaction of the learner based on his learning progress.

3. Related Work

Many attempts have been made to solve the information need problem, many recommender systems have been proposed, and here we investigate some examples of those systems.

3.1. Syskill & Webert

Syskill & Webert [4] is a software agent that learns a profile of a user’s interest, and uses this profile to identify interesting web pages. It learns a separate profile for each topic of each user. There are two modes for using Syskill & Webert:

1. In the first mode it learns a profile from the user’s ratings of pages and uses this profile to suggest other pages accessible from an index page. Once the user profile has been learned, the profile can be used to determine whether the user would be interested in another page. This decision is made by analyzing the HTML source of a page, Syskill & Webert annotates each link on the index page with an icon indicating the user’s rating or its prediction of the user’s rating.
2. In the second mode, Syskill & Webert tries queries for LYCOS (www.lycos.com) based on the user profile; this is useful to find pages that might be interesting for a user anywhere on the Web (provided the pages have been indexed by LYCOS). The user profile contains information about two types of words that occur in pages that have been rated. First, it contains words that occur in the most number of pages that have been rated "hot". The second set of words are those whose presence in an HTML file helps discriminate pages that are rated hot from other pages. Since LYCOS cannot accept long queries, query generation is limited to 6 words of the first type and 7 words of the second type.

3.2. METIOREW

METIOREW [5] works in an objective oriented way where an objective expresses an information need. Each objective is represented by a model. This means that for each user we can have several models depending on the different information needs. With this representation the same user will be able to work in different sessions with different objectives. The final objective of METIOREW is to find the most relevant Web pages for the current user’s objective. The pages will come from Web robot search, supervised navigation and collaborative retrieval. METIOREW is considered to be both a content and collaborative based recommending system.

3.3. Letizia

Letizia [6] is an agent that assists a user browsing the World Wide Web. As the user uses a conventional Web browser such as Netscape, the agent tracks user behavior and attempts to anticipate items of interest by doing concurrent, autonomous exploration of links from the user's current position. The agent tracks the user's browsing behavior (following links, initiating searches and requests for help) and tries to anticipate what items may be of interest to the user. It uses a simple set of heuristics to model what the user's browsing behavior might be. Upon request, it can display a page containing its current recommendations, which the user can choose either to follow or to return to the conventional browsing activity.

3.4. WebWatcher

WebWatcher [7] works as a tour guide which accompanies users while browsing a certain collection of pages, and interactively suggests appropriate hyperlinks (according to user's interest), and improve the quality of its suggestions by learning and observing its user's actions. The user can communicate with the system and give feedback.

WebWatcher tries to suggest appropriate links in a specific page based on user interest and other information it learns from the page. It calculates link quality based on the page and user interest. A link quality is interpreted as the probability that a user will select this link.

3.5 Web Browser Intelligence (WBI)

WBI [8] is a multi-agent framework system for assisting web browsing. It organizes agents on a user's machine to observe user actions, proactively offer assistance, modify web documents, and perform new functions. WBI can annotate hyperlinks with network speed information, record pages viewed for later access, and provide shortcut links for common paths. In this way, WBI personalizes a user's web experience by joining personal information with global information to effectively tailor what the user sees.

4. The Proposed System Architecture

The proposed system consists of number of modules and components cooperating and coordinating with each other to give a better navigation experience for the course and recommends new materials for learner. The architecture of the proposed system was designed in three layers: presentation, logic and data layer (see Fig. 1):

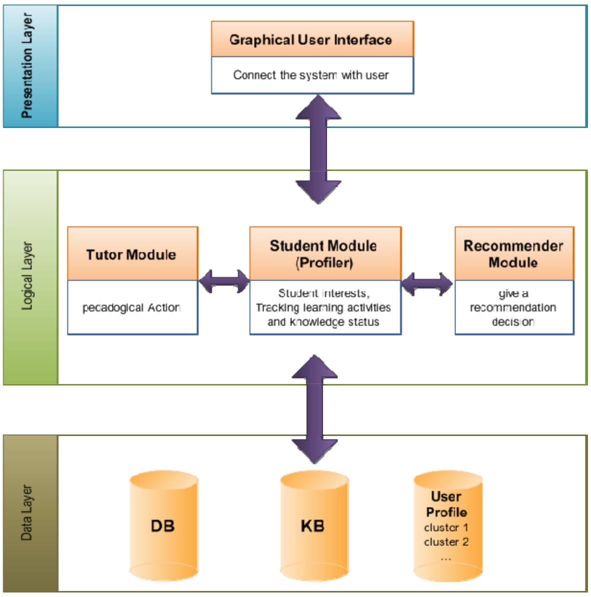
Presentation Layer
Graphical User Interface
Connect the system with user
Logical Layer
Tutor Module
pecadogical Action
Student Module (Profiler)
Student interests, Tracking learning activities and knowledge status
Recommender Module
give a recommendation decision
Data Layer
DB
KB
User Profile
cluster 1
cluster 2
...

Fig. 1 Proposed system architecure.

4.1. Presentation Layer

This layer contains the graphical user interface that connects the system with the user in a Web environment. This user interface makes use of AJAX (Asynchronous JavaScript and XML) technology which enhances user interface and increases user experience with the system.

4.2. Business Logic Layer

It refers to the business rules and validations required in the application, such as educational activities of the tutor model and monitoring of learning activities related to knowledge, skills, goal achievement and satisfaction and progress activity and tracking student interest and giving appropriate recommendation decision, This layer consists of the following modules:

4.2.1. Tutor Module

The Tutor module uses information from the student model to determine what aspects of the domain knowledge should be presented to learner. This information, for example, may be new material, a review of previous topics, or feedback on the current topic. The tutor module contains a planner with rules. The planner generates the plans for teaching each item of subject matter, which needs pedagogical decisions. The tutor module monitors the student’s actions and adapts the system responses to the student.

4.2.2. Student Module

This module consists of two parts the **first part** is responsible for modeling the individual characteristics of the student, among then one of the most important is the instant individual knowledge about the domain.

The **second part** called **profiler** which builds a model of user interest based on the pages he visits and the rating given to those pages (either explicitly or implicitly). This module is needed to be active all the time a user navigates the web. It will be able to add new information and modify existing information in the user profile while the user is visiting new web pages without interrupting his actions. Also, it must adapt the profile according to changes in user interests.

The **profiler** uses a clustering algorithm to build the profile, every time a new page needed to be added to the profile, the clustering algorithm will process it and add it to the nearest cluster, or create a new one for this new page.

The profiler has to adapt the profile according to changes in user interests over time; this is done by removing clusters with no more interest in recent time periods.

4.2.3. Recommender Module

Recommender module fetches all pages behind the currently shown web page. In contrast with the GUI module (from Presentation Layer) which just fetches a page to be shown to the user, the recommender module fetches pages silently without showing them to the user.

After fetching a page the recommender module analyzes its content using the user profile to indicate whether it is interesting or not by comparing its content against the information stored in the user profile, then asks the GUI module to annotate every visible link in the page with either being interesting or not.

Recommender module decides whether a page is considered to be interesting or not. A page URL is sent to this module, then the recommender module grabs the page content and grabs all pages behind all links appearing in that page. Normal feature extraction techniques are applied on each page and a model is built for every extracted page. Extracted features are compared against the user profile by calculating the similarity between extracted features and each cluster. If a cluster is found to be similar to that page, then the recommender module will indicate that a page is interesting, otherwise a page is considered not to be interesting.

Similarity calculation is also based on the similarity histogram clustering technique with some changes. The following steps are used to check whether a page is considered to be interesting or not:

1. Traverse all clusters in the user profile.
   - Calculate the histogram ratio for the current cluster, let it be HR_{old}.
   - Simulate adding the document D to the current cluster.
   - Calculate the histogram ratio for the current cluster, let it be HR_{new}.
   - If ($HR_{new} \le HR_{old}$) then highlight that this page is interesting and exit.
2. If the current document D is not found to be interesting for all clusters, then highlight that the page as not interesting.

4.3. Data Layer

This layer allows easy access and manipulation of information stored in databases, and user (learner) profiles.

5. Conclusion

The majority of current web-based learning systems are closed learning environments, where courses and materials are fixed and the only dynamic aspect is the organization of the material that can be adapted to allow a relatively individualized learning environment.

Web recommender systems are important to facilitate web usage and decrease time and effort needed to reach information needed by a user. Many researches and projects have aimed to address this topic, but in general most work was conducted on controlled domains or specific websites which prevents developed systems from being applied on general web page browsing.

Three-Tier client server architecture (presentation, logic and data) is represented to handle the implementation in distributed environments.

A recommender system that looks for all links in a web page and grabs the contents behind those pages then give recommendation decisions based on profiles automatically built by inspecting and analyzing web pages browsed by a user and the feedback given about those web pages either implicitly or explicitly.

Acknowledgments

Insert acknowledgment, if any.

References

[1] http://www.cdac.in/html/pdf/session6.1.pdf

[2] Zhang, D. (2004). Virtual Mentor and The LBA System - Towards Building An Interactive, Personalized, and Intelligent E-Learning Environment. Journal of Computer Information Systems, 35-43.

[3] http://www.brandon-hall.com/publications/elearning101/elearning101.shtml

[4] M. J. Pazzani, J. M. (1996). Syskill & webert: Identifying interesting web sites. AAAI/IAAI'96, 54-61.

[5] D. Bueno, R. C. (2001). METIOREW: An Objective Oriented Content Based and Collaborative Recommending System. In Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia (pp. 310-314). ACM Press.

[6] Lieberman., H. (1995). Letizia: An agent that assists web browsing. IJCAI'95 Morgan Kaufmann, 924-929.

[7] T. Joachims, D. F. (1997). Web Watcher: a tour guide for the World Wide Web. In Proceedings of the 15th International Joint Conference on Artificial Intelligence (pp. 770-777). IJCAI'97.

[8] R. Barrett, P. P. (1997). How to personalize the web. In Proceedings of the ACM Conference on Human Factors in Computing Systems. Atlanta, GA: CHI '97.

Abdallah Gomah received the B.Sc. in Computer Science from Faculty of Computers & Information, Cairo University in 2003. He loves computer programming especially in C++. On 2002 he achieved the 2nd place in ACM Arab and North African Regional contest. Now he is studying to achieve the master degree in computer science in the field of e-learning.