

The Pyramid Collaborative Filtering Method: Toward an Efficient E-Course

Sofiane A. Kiared¹, Mohammed A. Razek², Claude Frasson¹

¹Département d'informatique et de recherche opérationnelle
Université de Montréal C.P. 6128, Succ. Centre-ville Montréal,
Québec Canada H3C 3J7

[\[kiaredas, frasson\]@iro.umontreal.ca](mailto:{kiaredas, frasson}@iro.umontreal.ca)

² Faculty of Science, Math. and Computer Science Department
Azhar University Cairo, Egypt
abdelram@sci-azhar.edu.eg

Abstract.

Web-based applications with very diverse learners fail because they fail to satisfy various needs. Some people use collaborative filtering methods to analyze learners' profiles and provide recommendation to a new learners, but this methods provides false recommendations from beginners. We present a new method, which provides recommendations that depend on the credibility rather than the number of learners. We have designed, implemented, and tested what we call the Intelligent E-Course Agent (IECA). Our evaluation experiment shows that our approach greatly improves learners' knowledge and therefore presents a course that is more closely related to their needs.

1. Introduction

Recently, research teams have investigated ways of solving problems for learners who use hypermedia systems and have proposed several ways to overcome them. A recommendation system might work. It would try to personalize students' needs by building up information about their likes and dislikes, what succeeds and what fails [Lynch 2001]. It would rely on two techniques: content-based filtering (CB) and collaborative filtering (CF) [Breese 1998, Herlocker 2000]. Collaborative filtering techniques information that is based on the opinions of those whose needs and preferences are like those of the learner. But this method does not take into account their credibility.

To clarify our point of view, consider a web-based educational system called Annaba. Every day, more than 100 learners use it. Each chooses topics to study. Suppose that units U_i , and U_j of topic C_k were selected 1000 times and 500 times respectively. The existing collaborative filtering method would recommend U_i , but U_j would be better. We suggest that many learners were still beginners and therefore choose the wrong unit. In this paper, we suggest a new selection method, one that depends on the credibility of the learners. Suppose now that the credibility of learners who select U_i , and U_j are 0.3 and 0.7 respectively. The selection value for U_i , and U_j would become $1000*0.3$ and $500*0.7$, which would equal 300 and 350 respectively. In that case, Annaba would recommend the correct unit: U_j .

In this paper, we will modify a Pyramid Collaborative Filtering Model (PCFA) [Razek et al. 2004] for filtering and recommending a unit rather than a learner. PCFA has four levels.

Moving from one to another depends on three filtering techniques: domain-model filtering, user-model filtering, and credibility-model filtering.

Our underlying hypothesis is that we can return a unit that has been selected by a learner who satisfies the following conditions: (1) help from someone with extensive knowledge of the concept being taught, (2) someone whose behavior and learning style are like those of the learner, and (3) someone whose credibility guarantees his choice (i.e., we can depend on his unit). To satisfy the third condition, we follow use a PCFA. We need answer two questions: Is this a unit that the learner needs? Does it contain information that he can understand? Based on what we learned by asking those three questions, we have designed, implemented, and tested what we call an Intelligent E-Course Agent (IECA).

We have organized the rest of this paper as follows. Section 2 briefly describes some related work; section 3 shows how the IECA method works; section 4 presents an overview of our system and IECA architecture; section 5 presents some experimental results; and section 6 suggests future projects.

2. Related Work

Before proposing a new architecture, I find it necessary to discuss some available ones. Adaptive technologies will probably new ways to offer efficient web-based education systems [Brusilovsky 2000]. But the main ones, which still use the learner-style concept, fall into the following three categories [Brusilovsky 1998] [Papanikolaou et al. 2004]: Adaptive presentation and curriculum sequencing; adaptive navigation support; and adaptive collaboration support.

2.1. Adaptive presentation & and curriculum sequencing

These technologies aim at adapting learning style. Adaptive-presentation technologies adapt the content of user interfaces to the user's goals, knowledge, and other information that the user model stores. Curriculum-sequencing technology (also known as instructional-planning technology) offers to a learner the most suitable individually planned unit through the learning materials (the optimal path) [Brusilovsky 1998] [Papanikolaou et al. 2004]. Examples in this category are CITS [Razek et al. 2003a], Arthur and CS383 [Papanikolaou et al. 2004], ACE and INSPIRE [Papanikolaou et al 2004].

CITS uses learning-machine techniques to identify the learners' preferred learning style. The system tracks their behaviors and adapts the presentation accordingly. It also searches the web and returns documents that are related to the topic [Razek et al. 2003b]. Arthur and CS383 adopt alternative styles of instruction by developing several types of educational material and using several media for each particular section of the course [Papanikolaou et al. 2004]. The system that we developed uses almost the same approach. In ACE and INSPIRE, for example, we adapted sequence of resources according to learning styles by following a variety of instructional strategies [Papanikolaou et al. 2004].

2.2. Adaptive-navigation support

The goal of adaptive-navigation-support technology is to help learners navigate in hyperspace by changing the appearance of each visible link. In other words, it helps learners to find their paths by adapting link presentation to their goals, knowledge, and other characteristics. The most popular of these techniques are direct guidance, sorting, hiding, annotating, and generating

(which is the newest and most popular technology in the context of e-commerce) [Brusilovsky 2003].

Examples include ELM-ART [Brusilovsky 1998] [Weber et al. 2001] and InterBook [Brusilovsky 2003] [Brusilovsky et al. 1998] for direct guidance and adaptive annotation in the context of web-based education. InterBook and Hy-SOM [Brusilovsky 2003] [Kayama et al. 1999] proposed the most popular kind of link generation (dynamic recommendation of relevant links). ELM-ART proposes an additional method: generating links for similarity-based navigation. An example of that in the Knowledge Sea system can be found in [Brusilovsky 2003] and [Brusilovsky et al. 2002].

2.3. Adaptive-collaboration support

Adaptive-collaboration support is a new way to enhance the quality of web-based education. It involves communication between several learners (social interaction) and potentially collaboration. The aim is to use a system's knowledge of many users (stored in user models) in forming a matching group of learners [Papanikolaou et al. 2004] [Brusilovsky 1998].

3. IECA Methodology

Although the content-based approach [Lynch 2001] studies the contents of recommended items, collaborative filtering [Breese 1998] treats learners as a community. Using content-based and collaborative-filtering techniques to recommend units is an interesting and challenging application, but they consider neither the credibility of learners nor their learning styles. We have applied the PCFA to the collection of books. This paper focuses on only two levels of PCFA: domain-model filtering and user-model filtering.

3.1. Problem statement

Following [Razek et.al. 2003], a course could contain several concepts. Each would consist of units: background, definition, problems, examples, and exercises. Let C be an online course. Suppose that C contains k chapters and that each chapter can be divided into n units:

$$C = \{U_{11}, U_{12}, \dots, U_{1n}, U_{21}, \dots, U_{2n}, \dots, U_{i1}, \dots, U_{ij}, \dots, U_{in}, \dots, U_{k1}, \dots, U_{kn}\}$$

If we each unit has its own exercises, learner L must pass them before moving on to the next unit. Suppose that L fails to pass U_{23} , for some reason. Maybe he cannot understand the preceding or following units, as shown here:

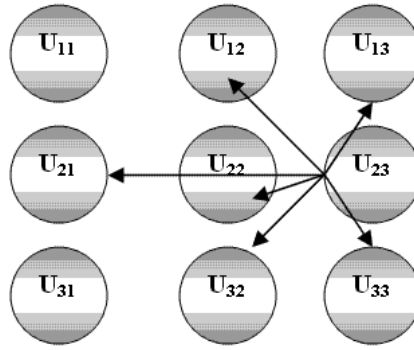


Fig. 1. Interdependence of units

Our system would switch him to a suitable unit, which would help him understand U_{23} . In this sense, we would apply the PCF technique in order to

- find units that share the same concept,
- find a subset of these units that meets L 's learning style, and
- choose one unit of this subset, one that has a greater credibility, which means that we must calculate the amount of credibility for each unit.

We would return the unit of a learner, therefore, who satisfies three conditions: he knows a lot about the concept being taught is like L_n in behavior and learning style and we can depend on his unit because of its credibility). To satisfy the later, we would use PCF [Razek et al. 2004]. This method suggests three questions. Does the learner need this unit? Can the learner understand it? And can we guarantee that this unit will meet his needs? To answer that question, we would need to find best learner and recommend his unit. So, we would apply PCF on a units set.

3.2. Identifying the suitable unit for a learner

In this section, we will discuss all stages in the process of identifying the suitable unit algorithm (as shown in Figure 2). To some extent, the algorithm will identify the helper (who is also a learner) who has almost the same characteristics (i.e. learning style) of the learner and followed at least a part of the course. The first step shows how to find a helper with information that L_n needs.

Step 1: Dominant meaning filtering

- For each learner $L_u \in L, L_u \neq L_n$, compute the dominant meaning similarity $S(Q, C_h)$ between the new learner's concept Q and others concepts C_h that L_u has visited.

- Suppose that a set of the C_h concept's dominant meanings is $\{w_1, \dots, w_m\}$ and that the dominant meaning set of the query Q is $\{q_1, \dots, q_s\}$. We can evaluate the dominant meaning similarity $S(Q, C_h)$:

$$S(Q, C_h) = \frac{1}{s} \sum_{i=1}^s \frac{1}{m} \left[\sum_{j=1}^m \Theta(w_i, q_j) \right], \quad (1)$$

where

$$\Theta(w_i, q_j) = \begin{cases} 1 & w_i = v_j \\ 0 & w_i \neq v_j \end{cases}$$

- List learners for whom the dominant meaning similarity value is higher than the threshold fixed for the concept Q .
- Let us call this $List_q$.

Step 2: User model filtering

- For each learner $L_u \in List_q$, compute the similarity of user behaviors $B_{u,v}^q$ as follows:

After participating, we could have a visiting vector for each user. Visiting vector $V_q = (v_i^q)_{i=1}^l$ represents the atomic units in concept Q , which have been visited by user v . Component v_i^q is equal to zero if v does not visit it and equal to one if he or she does. Therefore, we can compute the similarity between users v and u as follows:

$$B(u, v) = \frac{1}{D} \sum_{k=1}^D B_{v,u}^k, \quad (2)$$

Where D is the number of concepts, and

$$B_{v,u}^k = \frac{\sum_{i=1}^l v_i^k u_i^k}{\sqrt{\sum_{i=1}^l (v_i^k)^2 \sum_{i=1}^l (u_i^k)^2}}$$

- For each learner $L_u \in List_q$, compute the learning style similarity $LS_{u,v}$. Following [Razek et al.2002], the GA distinguishes several learning styles (LS): visual V ; auditory A ; kinesthetic K ; visual-kinesthetic VK ; visual-auditory VA ; and visual-auditory-kinesthetic VAK .

$$LS_{u,v} = \begin{cases} 1 & LS_u = LS_v \\ \frac{1}{2} & LS_u \in \{V, A, K\} \& LS_v \in \{VK, VAKA\} \\ \frac{2}{3} & LS_u \in \{VK, VAKA\} \& LS_v \in \{VKA\} \\ \frac{1}{3} & LS_u \in \{V, A, K\} \& LS_v \in \{VKA\} \end{cases} \quad (3)$$

- For each learner $L_u \in List_q$, compute the similarity degree between user's modeling,

$$UMD(u, v) = \frac{1}{2} (B_{u,v}^q + LS_{u,v}).$$

- Keep the list of learners for whom the dominant-meaning-similarity value is higher than the threshold fixed for concept Q . Let us call this $List_{q2}$.

Step 3: Credibility filtering

- For each learner $L_u \in List_{q2}$, compute the credibility of learner Ω_u . The credibility Ω_u [Razek et al. 2004] is the dependability degree of learners on the information presented by helpers during a learning session.
- Thus, the best helper for learner L_v is one who has the greatest value for Ω_u ; let that be L_h .
- Therefore the algorithm will return the unit of the learner L_h .
- If learner L_u is a new participant in the system, then copy the entire unit of helper L_h to that of L_u . Otherwise, copy the file unit except parts of the course [that L_u has] already followed.

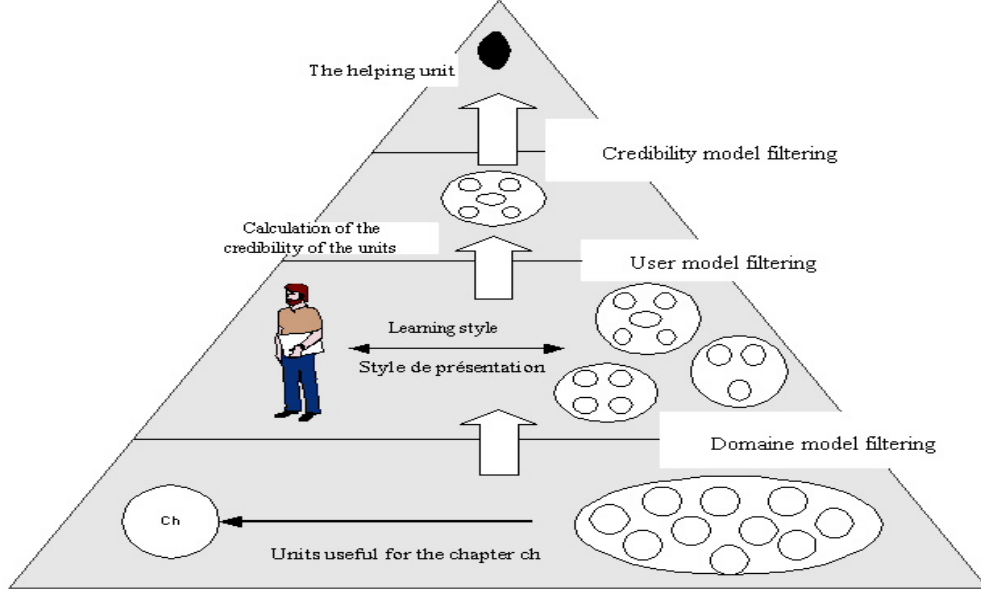
Note: the credibility here is for the unit, not the learner. Therefore, we suppose that the unit's credibility is associated with that of the person who has visited it. Suppose that N_{13} learners have visited unit U_{13} . Say N_{13}^s of them have succeeded and N_{13}^f failed. Therefore, the credibility of N_{13} is divided into two types: positive and negative.

Positive credibility C_{ij}^s is the average number of learners who have visited U_{ij} and success times the summation of their credibilities. C_{ij}^s is computed as

$$C_{ij}^s = \left(\frac{N_{ij}^s}{N_{ij}} \right) \sum_{k=1, L_k \in \{N_{ij}^s\}} C(L_k)$$

Fig. 2. The pyramidal approach of the IECA

Negative credibility C_{ij}^f is defined as the average number of learners who have visited U_{ij} and failed times the summation of their credibilities. C_{ij}^f is computed as follows:



$$C_{ij}^f = \left(\frac{N_{ij}^f}{N_{ij}} \right) \sum_{k=1, L_k \in \{N_{ij}^f\}} C(L_k)$$

Therefore, compute the credibility of U_{ij} as follows:

$$C_{ij} = \begin{cases} C_{ij}^s - C_{ij}^f & C_{ij}^s > C_{ij}^f \\ 0 & C_{ij}^s \leq C_{ij}^f \end{cases}$$

4. System Overview

This section shows how IECA has implemented, demonstrate IECA architecture, and present a scenario for a searching session.

4.1. The IECA Architecture

The first time that learners use the system, it allows them to enter their interests (username, password, subjects that they prefer, and so on). Then, it asks questions to predict their learning styles. IECA architecture contains two sides, that of the client and that of the server, as shown in figure 3. The former presents user interfaces where learners can interact and use IECA's

features. The latter contains the IECA, which interacts with user profiles, a dominant-meaning dictionary, and the course unit (which is stored as an xml file).

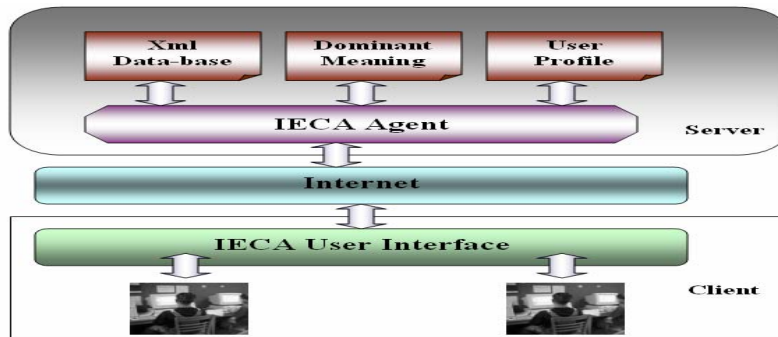


Fig. 3. The IECA architecture

4.2. IECA Scenario

To implement the system, we used ASP .NET technology -- Microsoft C# .NET framework 1.1, XML, and an Access database (for the collaborative part only) -- along with IIS as a web server. Conceptually, the general model has these components: sign-up and registration, learning-style test, identifying a suitable unit for a learner, lesson, and collaborative unit.

Sign-up and Registration

The new learner signs up by using the following registration form (see Figure 4) in order to create a personal profile. Each profile stores personal information: last name, first name, login (known as static information), and information about learning style, dominant-meaning words, and behavior (dynamic information). IECA uses this information to identify suitable units by applying the pyramidal collaborative filtering algorithm. During the course session, the system updates this profile by recording progress. The learner may change this information at any time

The screenshot shows a web browser window titled 'Register - Microsoft Internet Explorer'. The page header includes the 'Université de Montréal' logo and navigation links for 'new user', 'Learning Style Test', and 'Disconnection'. The main content area is titled 'Registration' and contains a 'Profile Information' section. On the left, there is a photo of a man with a cap and buttons for 'Update Picture', 'Edit', and 'Save'. On the right, there are input fields for:

- Login: Diaa
- Password: masked with dots
- Confirm password: masked with dots
- e-Mail: diaa@iro.umontreal.ca
- First Name: Diaa
- Last Name: Diaa
- Sex: Radio buttons for Male (selected) and Female
- Address: C.P. 6128, succursale Centre-ville
- City: Montréal
- Province: QC
- Code Postal: H3C 3J7
- Quality: Radio buttons for Beginning (selected) and Expert

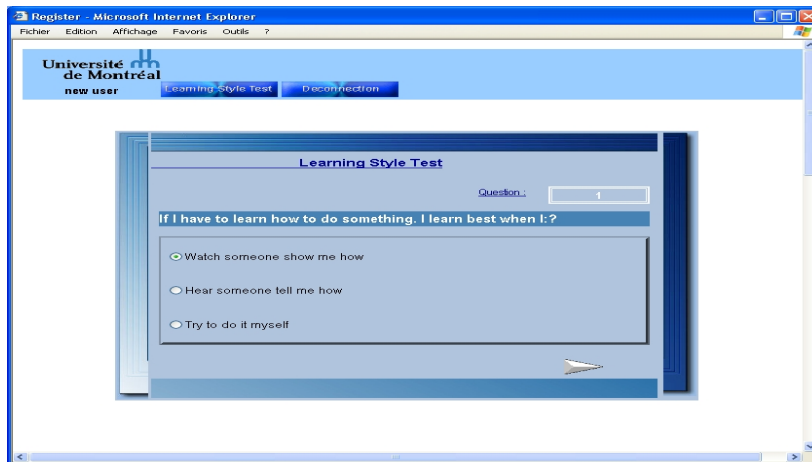
by editing it.

Fig. 4. Sign-up and registration

Learning-style test

Immediately after registration, the system opens a session. At this stage, the learner cannot begin the course. Only two operations are available: passing the learning-style test or disconnecting from the system (see Figure 5).

According to [Yammine et al. 2004], the learning-style test consists of questions. The IECA uses the answers to identify a specific the learning style. There are seven distinct styles: visual, auditory, kinesthetic, visual-auditory, visual-kinesthetic, auditory-kinesthetic, and visual-auditory-kinesthetic. Having identified a learning style, the IECA saves this information as part of the learner profile and creates a learner unit based on the course's template unit (xml file),



which is available for each learning style.

Fig. 5. Learning-style test

Learning Session

Our application focuses on courses. Once they see the course summary (Figure 6), the learners can begin a course by selecting either the last section completed before signing off or, for new learners, the first section.

The example that we use for our application is a course on the Internet. It consists of seven chapters, each having one to three sections. Each section provides information on Internet. To monitor progress, we have included a multiple-choice test at the end of each section. A learner who passes with a grade of 50% will move on to the next section. Otherwise, the IECA will provide assistance with the helping unit.

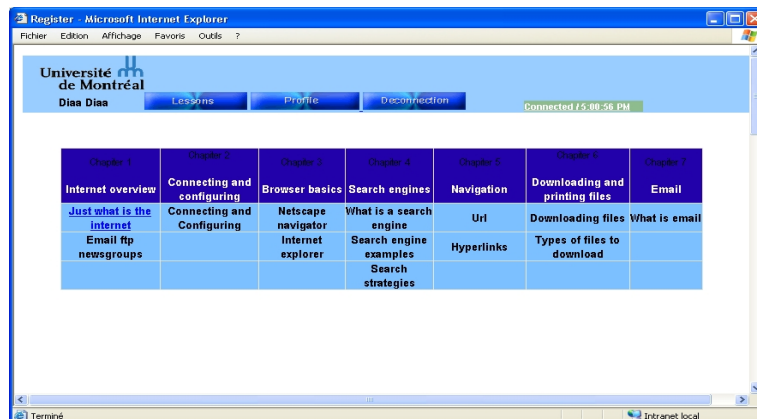


Fig. 6. The course synopsis

5. System evaluation

We come now to the results of our experiment. Our aim, of course, was to measure the effectiveness of our system.

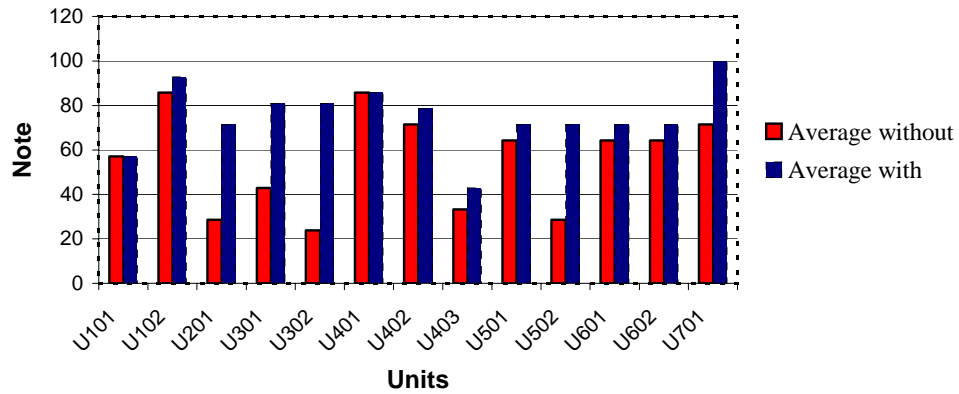
We selected two groups of 30 learners. Learners from the one called NASSG followed the course without the assistance of an IECA. Those from the one called ASSG did. Table 1 presents the results for both. This table contains four folders: the unit's name, the number of questions, the averages of correct answers by unit, and the averages of correct answers by question. Figure 7 presents the averages for each group by course unit.

Table 1 Experiment results

Units	No. Of questions	The average of the correct answer per unit		The average of the correct answer per unit	
		NASSG	ASSG	NASSG	ASSG
U101	Q101_1	57.14	57.14	57.14	71.43
U102	Q102_2	85.71	92.86	85.71	100.00
	Q102_2			80.00	80.00
	Q201_1			52.38	90.48
U201	Q201_2	28.57	71.43	20.00	100.00
	Q201_3			0.00	20.00
	Q301_1			57.14	90.48
U301	Q301_2	42.86	80.95	80.00	80.00
	Q301_3			20.00	80.00
	Q302_1			42.86	95.24
U302	Q302_2	23.81	80.95	40.00	80.00
	Q302_3			0.00	60.00
U401	Q401_1	85.71	85.71	85.71	85.71
U402	Q402_1	71.43	78.57	85.71	100.00
	Q402_2			40.00	40.00
U403	Q403_1	33.33	42.86	28.57	57.14
U501	Q501_1	64.29	71.43	57.14	64.29
	Q501_2			80.00	80.00
U502	Q502_1	28.57	78.57	28.57	85.71
	Q502_2			40.00	60.00
U601	Q601_1	64.29	71.43	57.14	71.43
	Q601_2			80.00	80.00
U602	Q602_1	64.29	71.43	42.86	50.00

		The average of the correct answer per unit		The average of the correct answer per unit		
	Q602_2			80.00	100.00	
	Q701_1			78.57	100.00	
U701	Q701_2	71.43	100.00	100.00	100.00	
	Q701_3			60.00	100.00	
	Q701_4				40.00	100.00

The two histograms indicate that ASSG did better than NASSG. Our system offers learners a considerable advantage, but the two groups have the same contribution at units U₁₀₁ and U₄₀₁. Figure 8 shows, by contrast, that the group with help from our system did than the one without help.



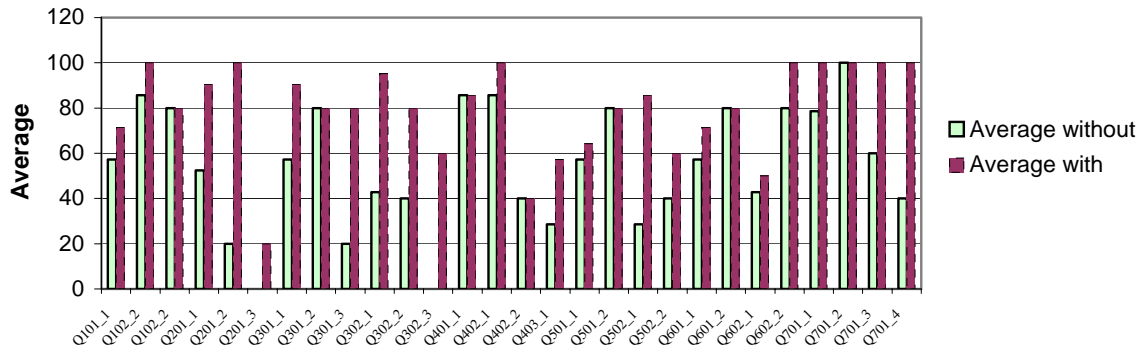


Fig. 7. Average of correct answers per unit

6. Conclusion and Future work

We have described the use of PCF to provide learners with the best units available in hyperspace. We based our proposed intelligent-agent system on a three level PCF, which uses dominant-meaning filtering at the lower level, user-model filtering at the middle level and credibility filtering at the upper. This technique answers the following questions: Does the learner need this unit? Can he understand the information? Our experiments show that this method greatly improves e-course performance in terms of both efficiency and credibility. Our future work will focus on how to improve testing results by conducting more experiments with more learners.

7. References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. *Int. J. Digit. Libr.* 1 (1997) 108–121.

8. [Brusilovsky 2000] Brusilovsky P. 2000. "Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education Intelligent Tutoring Systems." Lecture Notes in Computer Science, 1839 (2000): 1-7. Paper presented at the 5th International Conference on Intelligent Tutoring Systems, ITS 2000, 19-23 June in Montreal.
9. [Brusilovsky 2003] Brusilovsky, P.: 2003. "Adaptive Navigation Support in Educational Hypermedia: The Role of Student Knowledge Level and the Case for Meta-Adaptation." British Journal of Educational Technology 34. 4 (2003): 487-497.
10. [Brusilovsky 1998] Brusilovsky, P. 1998. "Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies." Paper presented at the 4th International Conference on Intelligent Tutoring Systems (ITS '98), in San Antonio .
11. [Brusilovsky et al 2002] Brusilovsky P. and R. Rizzo. 2002. "Map-Based Horizontal Navigation in Educational Hypertext." Journal of Digital Information 3 1.
12. [Brusilovsky et al 1998] Brusilovsky P, J. Eklund, and E. Schwarz. 1998. "Web-Based Education for All: A Tool for Developing Adaptive Courseware." Computer Networks and ISDN Systems (Proceeding of Seventh International World Wide Web Conference, 14-18 April. 30 (1-7), 291-300.
13. [Herlocker 2000] Herlocker, J.L., J.A. Konstan, and J. Riedl. 2000. "Explaining Collaborative Filtering Recommendations." Paper presented at the ACM 2000 Conference on Computer Supported Cooperative Work.
14. [Kayama et al 1999] Kayama, M. and T. Okamoto. 1999. "Hy-SOM: The Semantic Map Framework Applied on an Example Case of Navigation." In Advanced Research in Computers and Communications in Education. Frontiers of Artificial Intelligence and Applications, Vol. 2, Amsterdam: IOS Press, 252-259.
15. [Lynch 2001] Lynch, C. 2001. "Personalization and Recommender Systems in the Larger Context: New Directions and Research Questions." Paper presented at the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries, Dublin, Ireland, 18-20 June.
16. [Papanikolaou et al 2004] Papanikolaou, K.A. and M. Grigoriadou. 2004. "Accommodating Learning Style Characteristics in Adaptive Educational Hypermedia Systems." Paper presented at the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, 23-26 August, at the Eindhoven University of Technology in Eindhoven.
17. [Razek 2002] Razek, M.A., C. Frasson, and M. Kaltenbach. 2002. "Using a Machine Learning Approach to Support an Intelligent Collaborative Multi-Agent System." Paper presented at TICE 13-15 November, in Lyon.
18. [Razek et al. 2003a] Razek, M.A., C. Frasson, and M. Kaltenbach. 2003. "A Context-Based Information Agent for Supporting Intelligent Distance Learning Environments." Paper presented at the 12th International World Wide Web Conference, 20-24 May, in Budapest.

19. [Razek et al. 2003b] Razek, M.A., C. Frasson, and M. Kaltenbach. 2003. "Using Adaptability to Create an Automatically Adaptive Course Presentation on the Web." Paper accepted for IEEE ICALT, 9-11 July, in Athens.
20. [Razek et al. 2004] Razek, M.A., C. Frasson, and M. Kaltenbach. 2004. "A New Filtering Model towards an Intelligent Guide Agent." Paper presented at the 17th International FLAIRS Conference, 17-19 May, in Miami Beach.
21. [Weber et al 2001] Weber G and Brusilovsky P. 2001. "ELM-ART: An adaptive versatile system for Webbased instruction." International Journal of Artificial Intelligence in Education 12. 4 (2001): 351-384.
22. [Yammine et al. 2004] Yammine, K., M. Razek, E. Aïmeur, and C. Frasson. 2004. "Discovering Intelligent Agent: A Tool for Helping Student Searching a Library." Paper presented at the 7th International Conference on Intelligent Tutoring Systems (ITS-04), September in Maceió, Brazil.