

Pyramid Collaborative Filtering Technique for an Intelligent Autonomous Guide Agent

Mohammed A. Razeq,^{1,*} Claude Frasson,^{2,†} Marc Kaltenbach^{2,‡}

¹*Mathematics and Computer Science Department, Faculty of Science,
Al-azhar University, Cairo, Naser City 11886, Egypt*

²*Computer Science Department and Operational Research, University of
Montreal, C.P. 6128, Succ. Centre-ville Montreal, Quebec H3C 3J7, Canada*

This article presents an autonomous guide agent that can observe a community of learners on the web, interpret the learners' inputs, and then assess their sharing. The goal of this agent is to find a reliable helper (tutor or other learner) to assist a learner in solving his task. Despite the growing number of Internet users, the ability to find helpers is still a challenging and important problem. Although helpers could have much useful information about courses to be taught, many learners fail to understand their presentations. For that, the agent must be able to deal autonomously with the following challenges: Do helpers have information that the learners need? Will helpers present information that learners can understand? And can we guarantee that these helpers will collaborate effectively with learners? We have developed a new filtering framework, called a pyramid collaborative filtering model, to whittle the number of helpers down to just one. We have proposed four levels for the pyramid. Moving from one level to another depends on three filtering techniques: domain model filtering, user model filtering, and credibility model filtering. A new technique is filtering according to helpers' credibilities. Our experiments show that this method greatly improves filtering effectiveness. © 2007 Wiley Periodicals, Inc.

1. INTRODUCTION

Diaa is a student at Al-Azhar University in Egypt. He registers in a course on data structures and studies a specific concept: "queue." Unfamiliar with it, he desperately wants someone to help him. Maybe he can find another student in his class. With that in mind, Diaa turns to a recommender system. Even though most recommender systems have their advantages, they are ineffective for this purpose.

* Author to whom all correspondence should be addressed: e-mail: abdelram@iro.umontreal.ca.

† e-mail: frasson@iro.umontreal.ca.

‡ e-mail: kaltenba@iro.umontreal.ca.

To solve this problem and others, the Confidence Intelligent Tutoring System¹ (CITS) provides an intelligent e-learning for a community of online learners. For sound collaboration, the CITS searches its online learners and recommends a reliable helper to establish a learning session with Diaa.

A major part of this article tries to answer the following questions: Does the helper have information that Diaa needs? Will he or she present information that Diaa can understand? And can we guarantee that he or she will collaborate effectively with Diaa? We propose a new filtering framework called a Pyramid Collaborative Filtering (PCF) model to gradually diminish the number of helpers to just one. The proposed pyramid has four levels. Moving from one level to another depends on three filtering techniques: domain model filtering, user model filtering, and credibility model filtering.

To answer the first question, we represent domain knowledge as a hierarchy of concepts. Each concept consists of some words that reflect its dominant meaning. Each of those is linked with some chunks to define it. Moreover, this concept is associated with learners who have visited it. Knowing Diaa's concept, we can get a list of helpers. The dominant meanings definition is known as "the set of keywords that best fit an intended meaning of a target word."² This technique sees a query as a target meaning plus some words that fall within the range of that meaning. It freezes the target meaning, which is called a master word, and adds or removes some slave words, which clarify the target meaning.

To answer the second question, we create a new filtering method. Based on helpers' common learning styles, dominant meaning, and behaviors, this method can divide the list of helpers into subgroups. Each group includes people who might have behaviors and learning styles like that of Diaa. As a result, we can identify the subgroup to which Diaa belongs, and they can present information in a way that Diaa understands.

To answer the third question, we filter this subgroup, in turn, to recommend the best collaborator for Diaa. There are two well-known methods of doing this: collaborative filtering (CF) and content-based (CB) systems. CF systems build user profiles of user ratings of available concepts.³ They would use similarities among users' profiles to figure out which one is similar to Diaa's. It recommends this user to Diaa. CB systems would compare Diaa's concept content with those of other users to find who has the most similar one and then recommend him or her to Diaa.

These methods rely on ratings from users, however, and do not consider their credibilities. As a result, when unreliable users recommend bad concepts, it pushes the system to recommend incorrect items. The greater the credibility and knowledge of helpers, the more successful the collaborative learning will be. We define "credibility" as the dependability degree of the learners on the information presented by helpers during a learning session. It would be changed according to the helper's level of knowledge, learning styles, and goals. Our goal is to find an efficient way of calculating credibility. For that, we represent the interactions among a community of online learners as a social network problem.

To implement these solutions, we need an autonomous recommender agent. It must observe conversation, rate the inputs from learners, and calculate the

credibility of each learner. In that case, it could identify a reliable learner to interact with Diaa. We describe an autonomous guide agent (AGUA), one that observes discussions during a collaborative learning session and rates the inputs from learners. We represent interactions among a community of online learners as a social network problem. Based on the solution of this problem, the AGUA can group learners according to their credibilities and thus recommend a reliable helper. The AGUA has been fully implemented and integrated into CITS.

The rest of this article is organized as follows. Section 2 discusses pending problems to create an autonomous guide agent and gives a brief introduction to previous work. Section 3 briefly describes the overall structure of the AGUA. Section 4 discusses the role of the pyramid collaborative filtering model. In Section 5, we show how to represent experiments, the data set, experiment algorithms, and the evaluation framework. Section 6 presents the results of experiments conducted to test our methods. Section 7 concludes the article.

2. RELATED WORK

A great deal of work has been done on building systems that use filtering to recommend an item or a person. Two approaches are particularly important in this context: collaborative filtering and matchmaking systems.

Two approaches to collaborative filtering systems, in turn, are prevalent: collaborative filtering (CF) and content-based (CB) methods. CF systems build user profiles of user ratings of available concepts. They use the similarities among user profiles to figure out which one is most similar to that of the requester. Using positive training examples, Pazzani³ represented a user profile as a vector of weighted words. For predicting, he applied CF to the userratings matrix. In Fab,⁴ the relevance feedback of the user is used to classify a personal filter. Moreover, another filter is related to this topic. Each document is classified by the topic filter that is used to classify the document related to the user relevance feedback. In another approach,⁵ CB and CF approaches are allowed to create distinct recommendations and therefore to merge their prediction directly. Horting⁶ is an alternative, graph-based technique in which nodes are users; edges between nodes indicate the degree of similarity between two users. Predictions are produced by walking the graph to nearby nodes and combining the opinions of the nearby users. Our method differs from these by using the credibilities of users rather than the ratings from users.

Match⁷ is a matchmaking system that allows users to find agents for needed services. It can store various types of advertisement coming from various applications. Paolucci et al.⁷ proposed an agent capability description language, called Language for Advertisement and Request for Knowledge Sharing (LARKS), which allows for advertising, requesting, and matching agent capabilities. LARKS produces three types of match: exact match (most accurate), plug-in match (less accurate but more useful), and relaxed match (least accurate). The matchmaking process of LARKS has a good trade-off between performance and quality.

In LARKS, comparisons apply on words in the context slot of considered specifications. In contrast, the comparison in our approach depends on the

dominant meaning words by which the filter can recognize that there is a semantic distance between the words in a pair (computer, notebook), and that there is no closer distance between the words in a pair (computer, book). For effective filtering, we have created a new classification method called a Pyramid Collaborative Filtering Model (PCFM). It depends on three classification techniques: domain filtering, user model filtering, and credibility filtering. The next subsection sheds light on the guide agent framework.

3. AUTONOMOUS GUIDE AGENT (AGUA) FRAMEWORK

To explain the scenario supported by the CITS, we need to specify a reliable helper who can meet Diaa's needs. In fact, we must find a way to answer the following question: what type of knowledge is useful in calculating credibility? A diagrammatic summary of the guide agent is shown in Figure 1. The framework of AGUA has three components: communication between user profiles and learning styles to save any changes in the learner behavior, calculation of credibility value, and interaction with the CITS user interface to post recommended helpers.

We turn now to learner classification roles and how we can represent social activities in a community of online learners.

4. PYRAMID COLLABORATIVE FILTERING

Learners' collaboration is the key to learners' classification. This group contains N learners $\Pi = \{L_v\}_{v=1}^N$. It is frequently hard to recognize who knows whom, who prefers to collaborate with whom, how learners collaborate, and how they come to know each other. This work suggests three questions. We claim that

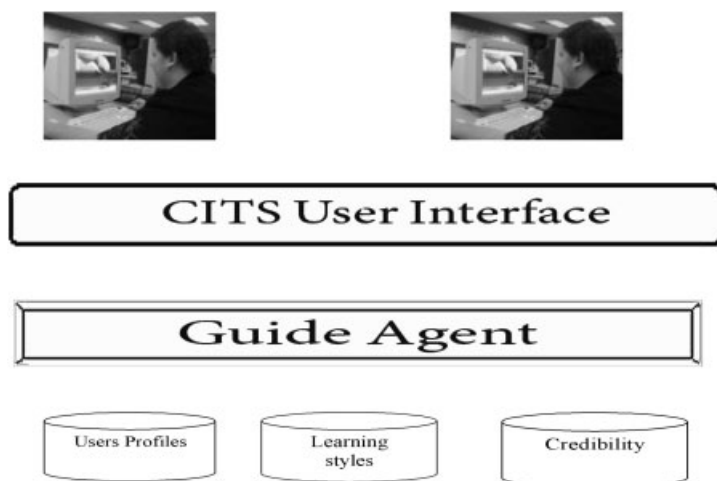


Figure 1. Guide agent framework.

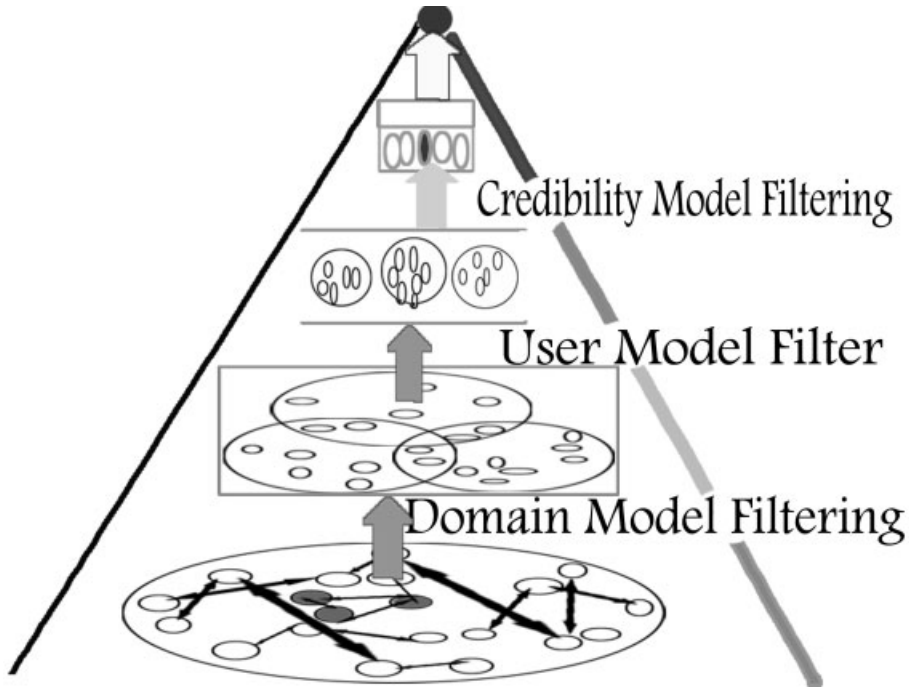


Figure 2. Pyramid collaborative filtering model.

answering them can overcome the difficulties of collaboration. These questions are as follows: Does the helper have information that a learner needs? Will he or she present information so that the learner can understand it? And can we guarantee that this helper will collaborate effectively with the learner?

We propose a four-layer classification as shown in Figure 2. Moving from one layer to another depends, in turn, on three classification techniques:

- domain model filtering
- user model filtering
- credibility model filtering.

The next section illustrates how to do this.

4.1. Domain Model Filtering

In this section, we show how the AGUA finds a helper who has information that the learner needs. We use a hierarchical classification technique in order to classify learners according to their domain knowledge.

Decomposing a classification problem, it would allow for competence in both learning and representation. Basically, each subproblem is smaller than the original one, and it is sometimes possible to use a much smaller set of features for

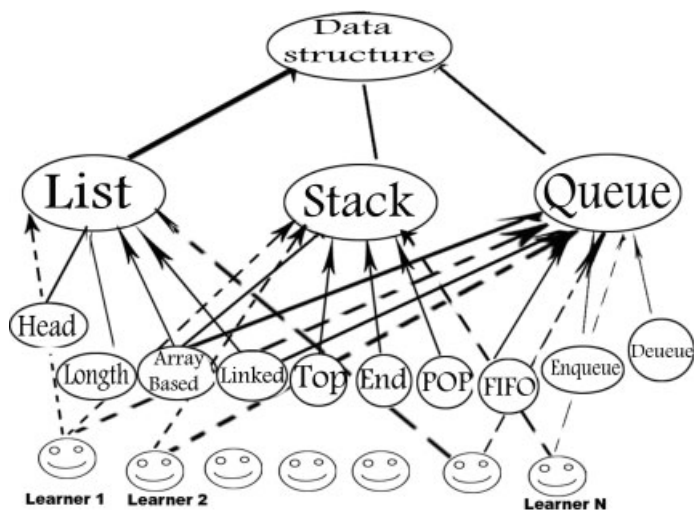


Figure 3. Hierarchical domain classification.

each.⁸ Each concept consists of some dominant meanings, in our approach, and each of those is associated with learners who have visited it. Knowing Diaa's concept, therefore, we can get a list of helpers.

Figure 3 shows a course on data structure associated with learners. The hierarchy has three top-level categories ("List," "Stack," and "Queue") and from four to five subcategories within each ("List/Head," "List/Length," "List/Array-Based," "List/Linked," "Stack/Array-Based," "Stack/Linked," and so on). In our hierarchical model, the word "List" would be very discriminating at the first level. At the third level, more specialized dominant meanings could be used as features within the second "List" category. The same features could be used by more than one category, such as "Array-Based" might be a useful feature (dominant meaning) for the categories "List/Array-Based," "Stack/Array-Based," and "Queue/Array-Based"; however, each feature might have one or more definitions. These definitions are associated with the features by a link to a document in the knowledge base.

In the next subsection, we explain how to extract learners that share the main concept of a discussion through a learning session.

4.1.1. Extract Helpers Who Share the Same Concept

We use the latter graph to extract the learners who share the same concept of any discussion. For example, suppose that the dominant meaning set in a learner's user profile is $E = \{\text{Linked}, \text{Top}, \text{FIFO}\}$. The problem is to find which concept probably represents these words. Therefore, we can extract the associated learners with this concept. It is obvious that traversing the graph is an important issue and certainly to be taken into account. To traverse the graph, we use the hill climbing

- (1) Put Search List = [Starting point];
- (2) If Starting point = w_r then C_r = Starting point, Exit successfully and return C_r ;
- (3) while Search List $\neq \emptyset$ do begin
 - Remove the leftmost child from Search List, call it X;
 - If X = w_r then C_r = parent(X), Exit successfully and return C_r ;
 - If not begin
 - Generate children and edges of X.
 - For each child of X
 - Calculate the edge heuristic value $H(E_i) = P_{i,j}$;
 - Sort children related to $H(E_i)$ as decreasing order;
 - Add sorted children to the front of the Search List;
- (4) If the goal has been found, announce success and return the sharing learners list $L(C_r)$.

Figure 4. Learner's extracting algorithm (requested word w_r).

search algorithm with some modifications.² If we traverse the graph in Figure 3 to look for the main concept of the set of words E , we would get the set of corresponding concepts as $C(E) = \{\text{List, Stack, Stack, Queue}\}$. We observe that the concept "Stack" is repeated twice, therefore, it will be considered as the main concept of the discussion. As a result, the sharing learners list of it may be represented as $L(\text{"Stack"}) = \{L_1, L_2, \dots, L_N\}$.

In our algorithm, initially Search List consists of generated children that we intend to consider during the search. Suppose that a child is searched after it is taken out of Search List. After being opened, it can be expanded and removed. The proposed algorithm ends when a request word is extracted from Search List (success) or when we try to extract a child while it is empty (failure) or, in some cases, when we generate a goal state (success). The input of our traversal algorithm is a requested word w_r and the output would be the sharing learners list $L(C_r)$. The pseudocode for this algorithm is shown in Figure 4.

Consider this algorithm of the directed graph in Figure 3. At each iteration, it removes the first element from the Search List. If it meets the requested word, the algorithm returns its parents (which led to the concept). If the first element is not a requested word, the algorithm generates its children and then applies heuristic evaluation $P_{i,j}$ to its adages. These states (children) are sorted in decreasing order according to the heuristic values before being inserted at the top of the Search List. This brings the best state in the front of the Search List.

4.2. Dominant Meaning Filtering

For an effective classification of the helper's knowledge level, we associate each of them with their visited concepts. If any user studies a concept, to be more precise, he or she will be linked with it. Moreover, the dominant meanings of that concept will be associated with his or her user profile. As a result, whenever the guide agent knows the concept that a learner needs, it can predict a list of users. The prediction depends on similarity to the active learner. This similarity is calculated according to the common dominant meaning appearing in the user profile. Using this similarity, we implement a domain classification algorithm to group the helpers.

We use the dominant meaning space between a concept and a user profile to measure the closeness between them. Once dominant meaning space has been computed, a threshold must be set for the minimum acceptable value. We connect helpers with the most closely related concept.

We suggest the following measure to compute the distance between user profile and concept. Suppose that C_h is a concept that a learner L_v is trying to understand. Suppose also that a set of the concept's dominant meanings is $\{w_1, \dots, w_m\}$, and that the dominant meaning set of the L_v 's user profile is $\{v_1, \dots, v_s\}$. At the moment, the problem is to evaluate users so that their profiles have the highest degree of similarity to concept C . Here is how we can evaluate the dominant meaning similarity $S(L_v, C_h)$ between user L_v and concept C_h :

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

where

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

Some disagreements could emerge here. Why do we not compute the similarity between user profiles rather than between use profiles and concepts? To answer that question, we can say the underlying idea is to find a helper who has more knowledge than the learner and could therefore answer his or her questions. The next section shows how a guide agent divides this list into subgroups of learners who have common learning styles, behaviors, and goals.

4.3. User Model Filtering

In e-learning, where both helpers and learners are separated geographically, user modeling is one of the most important challenges. Through a learning session, the discussion is derived according to the helper's own point of view, which depends on his or her learning styles and behaviors. As a result, the learner will find it hard to understand. To avoid this situation, we classify helpers according to their learning styles and behaviors. We must know the learner's style, his or her behavior, and the course contents to establish learning level, propose, or capability. The next section shows how we can do that. The purpose of this section is to group helpers according to user modeling. The algorithm we use can be summed up as follows:

- Compute the similarity between user behaviors $B_{u,v}$.
- Calculate the similarity between users' learning styles $LS_{u,v}$.
- Compute the user modeling degree $UMD(u, v)$ as follows:

$$UMD(u, v) = \frac{1}{2} (\lambda LS_{u,v} + (1 - \lambda) B_{u,v}) \quad (1)$$

where $UMD(u, v)$, $LS_{u,v}$, and Bu, v represent the similarity degree between the user's modeling, learning style, and behavior, respectively. To simplify the experimental results, we choose $\lambda = \frac{1}{2}$.

Next, we show how to calculate those degrees.

4.3.1. User Behaviors Similarity

To keep track of users' behaviors, we represent domain knowledge as a hierarchy of concepts. There are some dominant meanings to define and to join each concept. As we just mentioned in Subsection 4.1, these dominant meanings consist of a set of meanings that best fit a concept or that reflect the particular view of a specific learner. In our approach,⁹ the course consists of concepts. Each concept consists of five fragments: background, definition, problems, examples, and exercises. For each fragment, moreover, there are links to the three chunks that define it. And each chunk consists of specific units such as text, images, audio, and video.

After participating, we could have a visiting vector for each user. Visiting vector $V_k = (v_i^k)_{i=1}^l$ represents the atomic units in concept C_h , which have been visited by user v . Component v_i^k 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

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

4.3.2. Learning Style Similarity

Following Ref. 10, the GA distinguishes several learning styles (ls): visual V , auditory A , kinesthetic K , visual and kinesthetic VK , visual and auditory VA , and visual and auditory and kinesthetic VAK :

$$LS_{u,v} = \begin{cases} \mu & LS_u = LS_v \\ \mu/2 & LS_u \in \{V, A, K\} \text{ \& } LS_v \in \{VK, VA, KA\} \\ \mu/3 & LS_u \in \{VK, VA, KA\} \text{ \& } LS_v \in \{VKA\} \\ \mu/3 & LS_u \in \{V, A, K\} \text{ \& } LS_v \in \{VKA\} \end{cases} \quad (3)$$

To make the experimental results simple, we choose $\mu = 1$.

Therefore, from formulas (2) and (3), we can get the value of formula (1). In the next section, we illustrate the role of credibility value and show how to compute it.

4.4. Credibility Model Filtering

To illustrate why we use learner credibility to evaluate the significance of a segment, consider learner L_j , who recommends segment Γ_h . In fact, however, many segments are better than that one. We attribute this mistake to the learner's credibility. If he or she has good performances and enough credibility, therefore, his or her recommendation will rank at or near the top. A learner might have much background knowledge, but his or her credibility as a tutor might become doubtful if his or her answer to another learner's question is inadequate. Actually, credibility depends not on what the learner believes about himself or herself but rather on what other people believe about him or her. In a sense, the credibility of a response is based not on the sender but on the receiver or observer.

If we observe communication among learners, we can represent the problem of how to calculate the credibility of each learner as a collaboration network problem. As shown in Figure 5, that is a complete directed graph $G = (L, E)$, where L is the set of nodes of G and E the set of directed edges. Each edge has a nonnegative length. Each node represents a learner. For any two learners, L_i and L_j , there are three kinds of the direct link, L_{ij} , L_{ji} , and L_{ii} , as shown in Figure 5. Link L_{ij}

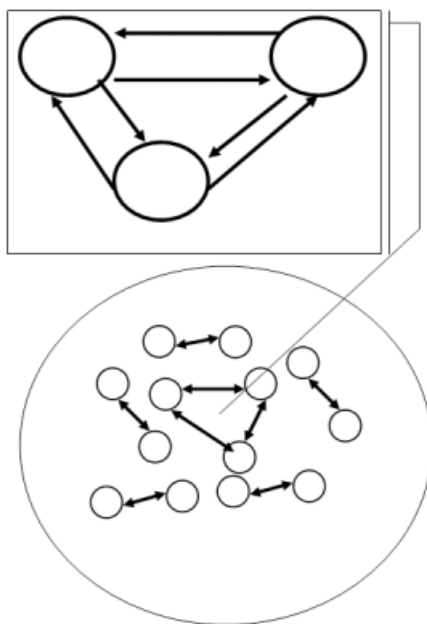


Figure 5. Collaboration network graph.

gives the collaboration space from learner L_i to learner L_j , and L_{ii} represents the learner's level of knowledge. In other words, it represents the amount of information that learner L_i has.

Therefore, we can use collaboration value L_{ij} as the number of questions, answers, and other forms of help that learner L_i has offered to learner L_j plus the number of questions that learner L_i has received from learner L_j .

Using the above graph in Figure 5 as an example, we can represent the credibility problem as a matrix:

$$CM = \begin{pmatrix} L_{11} & \dots & L_{1N} \\ \cdot & \cdot & \cdot \\ L_{N1} & \dots & L_{NN} \end{pmatrix}$$

Using this matrix, we can calculate the credibility of learner L_v as follows:

$$\Omega_v = L_{vv} + \sum_{f=1, f \neq v}^N L_{vf} - \sum_{f=1, f \neq v}^N L_{fv}$$

In the next section, we present our experiments and results.

5. EXPERIMENTS

In this section, we illustrate the data set, metric, and methodology for evaluating variants of our pyramid filtering algorithms.

5.1. Data Set

We use the IFETS Discussion List^a to collect a set of messages in order to evaluate the performance of our algorithm. The IFETS discussion is provided by the International Forum of Educational Technology and Society, which is a subgroup of the IEEE Learning Technology Task Force.^b There are two kinds of discussion in the IFETS: formal and informal ones. Formal discussions are topic based and occur for 1 to 2 weeks. There are moderators who conclude and summarize most of the individual estimations about the suggested topic; their summaries become visible in the forum's electronic journal.^c Informal discussions happen daily. Any participant can submit new topics or questions to the forum. Each day, users discuss several topics related to educational technology. We accumulated a collection of messages from February 27, 2002, to July 31, 2003, and considered users who had submitted two or more messages. Table I presents the collection features and indicates the number of messages, the number of users, the number of topics discussed, and the average number of messages per week.

^a<http://ifets.ieee.org/discussions/discuss.html>.

^b<http://lttf.ieee.org/>.

^c<http://ifets.ieee.org/periodical/>.

Table I. Collection used for the experiment.

Collection	Description
Number of messages	502
Number of users	95
Number of topics (concepts)	5

We used 80% of the collection as a training set and 20% as a test set. All messages have already been classified manually into five concepts.

5.2. Evaluation Methodology

To evaluate the accuracy of a prediction algorithm, researchers used two main approaches: statistical accuracy metric and decision support metric.¹¹ Furthermore, we suggested a new statistical metric called Precision Probability Value (PPV).

To measure statistical accuracy, we used the mean absolute error (MAE) metric—defined as the average absolute difference between the predicted and actual provided users list. We used MAE and PPV¹² to account for the prediction experiments, because they seem to be a simplest measure of overall error.

5.2.1. Mean Absolute Error

We used MAE¹³ for evaluating the accuracy of our filtering algorithm. This metric assesses the accuracy of a prediction by comparing predicted users with actual provided users. In our experiments, we computed the MAE on the test set for each concept and then averaged the results over the set of test users. Suppose that the total number of users who have participated in experiments is N , and the predicted and actual provided users list are $\{L_i\}_{i=1}^{N_s}$ and $\{L_j\}_{j=1}^{N_p}$, respectively. The MAE function for the concept C_h is given by

$$MAE(C_h) = \frac{|N_p - N_c|}{N}$$

where

$$N_c = \sum_{i=1}^{N_s} \left[\sum_{j=1}^{N_p} \Theta(L_i, L_j) \right], \quad \Theta(l_i, L_j) = \begin{cases} 1 & w_i = v_j \\ 0 & w_i \neq v_j \end{cases}$$

5.2.2. Precision Probability Value

PPV evaluates the accuracy of a system by finding the numerical scores of the corrected users predicted compared to the actual provided users. Moreover, it takes into consideration the entire number of the filtering list, which contains correct and incorrect users N_s . The PPV is computed for each concept C_h :

$$PPV(C_h) = \frac{N_c}{N_s} \times \frac{N_c}{N_p}$$

where N_c represents the correct predicted list and N_p represents the actual provided list.

5.2.3. Experimental Procedure

The experiments were conducted in two stages: training and test. For the former, we built a hierarchical domain classification of 80% of the collection.²

Training Stage. In the training stage, our hierarchy consisted of five concepts: Theory Of Learning (TOL), Teaching Strategies (TS), e-learning (EL), Capturing Knowledge (CK), and Educational Adaptive Hypermedia (EAH). For the comparative experiments, we constructed the dominant meaning sets of these concepts.² We associated users with each concept that they used in its discussion. Moreover, we linked each user with their dominant meaning set. In the same sense, we have used 80% of the collection to find the threshold of dominant filtering (μ) and credibility filtering (ν), respectively. The Training Stage Algorithm is summed up as follows:

- Build concepts of the hierarchical domain classification.
- Build dominant meaning vectors of these concepts.
- Compute dominant meaning vectors for each user.
- Associate each user with the corresponding concept.
- Calculate μ and ν as needed.

Test Stage. The test stage was to conduct three tests. We implemented three methods as described in Ref. 2 and tested them on our data sets. For each similarity algorithm, we implemented the algorithm to generate the prediction. The first test was applied on the basis of the pyramid model. Our purpose was to clarify the effectiveness of using a dominant meaning rather than keywords in predicting a group of users who have some specific concepts in common. The intent of using the Test1 Algorithm is to group the learners who have knowledge C_h in a set called $C[h]$.

Test1 Algorithm

- Index the collection.
- For each concept $C_h \in \{C_i\}_{i=1}^5$
 - For each user $L_v \in \{L_j\}_{j=1}^N$
 - if $S(L_v, C_h) \geq \mu$ then $L_v \rightarrow C[h]$.

We could not apply the second level filtering on the IFETS list, because we could not find a way to evaluate learning styles and behaviors. Consequently, we tested only the improvement of applying credibility filtering on the predicted list coming from the previous test. The algorithm is as follows.

Table II. Experiments results.

	Theory of learning	Teaching strategies	E-learning	Capturing knowledge	Educational adaptive hypermedia
Actual provided list	17	33	73	27	27
Entire predicted list using dominant meaning	15	25	60	20	17
Correct predicted list using dominant meaning	10	20	55	18	16
Entire predicted list using keywords	15	22	50	25	25
Correct predicted list using keywords	7	15	25	12	10
Entire predicted list using credibility filtering	15	27	65	25	25
Correct predicted list using credibility filtering	13	25	60	22	21

Test2 Algorithm $[\{C[h]\}_{j=1}^5, \{L_j\}_{j=1}^N]$

- Construct the credibility matrix for all learners $\{L_j\}_{j=1}^N$.
- For each user $L_v \in \{C[h]\}_{j=1}^5$
 - Compute the credibility value Ω_v or each user
 - if $\Omega_v \geq \nu$ then $L_v \rightarrow CC[h]$.

6. EXPERIMENTAL RESULTS

In this section, we discuss a series of experimental results conducted on the first and third levels of the pyramid filtering model. As mentioned above, we ran these experiments on our training data and used a test set to compute MAE and PPV. Table II shows the results. It indicates for each concept the number of users on the actual provided list (i.e., those who actually participated in the concept discussion), the entire predicted list, using dominant meaning approach (including correct and incorrect users), the correct predicted list, using dominant meaning approach (including correct users only), and so on.

The average percentage of correct users predicted by both experiments is directly proportional to the number of the users who participated. The following subsections try to clarify and analyze our conclusions.

6.1. Dominant Meaning Versus Keyword

The goal of this experiment was to demonstrate the effectiveness of the dominant meaning approach in a prediction task at the base of the pyramid. We treat this task as a text-categorizing problem. Melville et al.¹⁴ and Mitchell¹⁵ think of user profiles as bags of words. In contrast, we think of them as sets of dominant words. These words represent the dominant meaning of concepts that the user knows well. This experiment presents a comparison between the two approaches. The experiment is evaluated by both measures: MAE and PPV. Figure 6 shows experimental MAE results and Figure 7 PPV results.

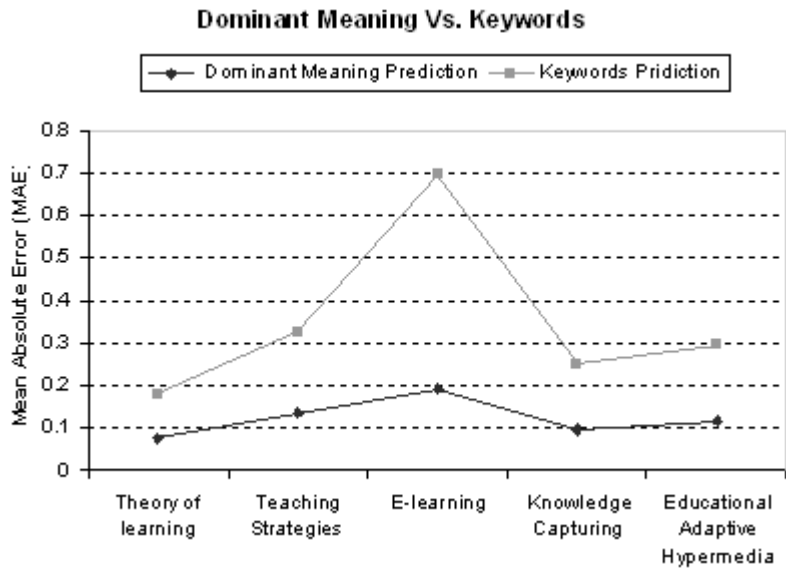


Figure 6. Mean absolute errors (MAE).

They show performance improvement, when the list of users is predicted by using dominant meaning rather than bags of keywords.

As can be seen in Figure 6, our approach performed better than the other one. Although the dominant meaning approach had a bad MAE at “E-learning,”

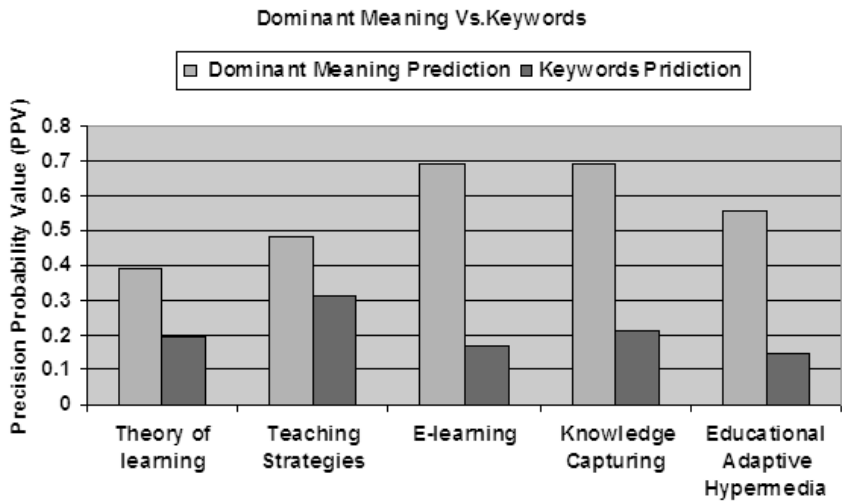


Figure 7. Precision probability value (PPV) measure.

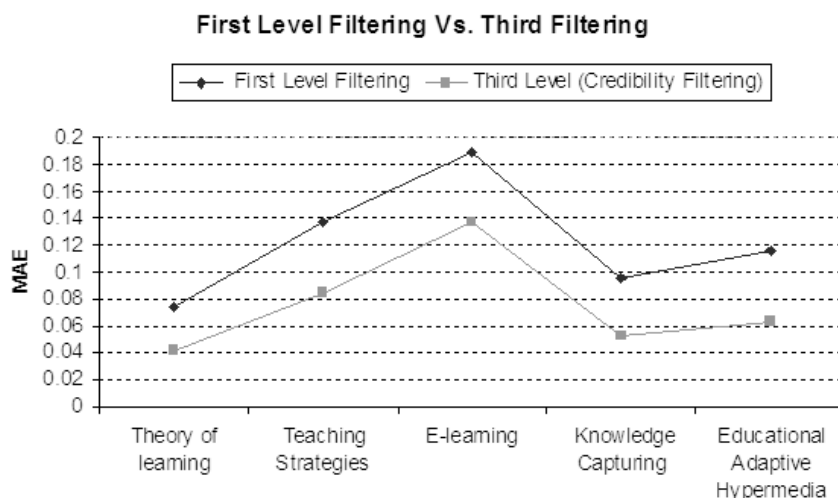


Figure 8. MAE for first level versus credibility filtering.

it still performed better than the other. The error of the keywords approach was directly proportional to the number of users. Meanwhile, the dominant meaning approach remained consistent. In Figure 7, we show that the level of precision made possible by our dominant meaning approach is higher than that of the other approach.

In the next subsection, we present another experiment for validating the third level performance of our pyramid model.

6.2. The Effectiveness of the Credibility Filtering

To the best of our knowledge, none of the currently used filtering algorithms takes into consideration the credibility of users. They are based only on the rating capabilities of users. Based on user credibility, we conducted an experiment to illustrate its effectiveness on the prediction task. The plot of the MAE and PPV of the first filtering and the credibility of the filtering algorithm are depicted in Figures 8 and 9, respectively.

The figures show that our credibility method yields more performance improvement according to both measures: MAE and PPV. The level of improvement changes at two measures. This changing depends on the number of users. According to the MAE, the error increases directly with respect to the number of users.

Although, the MAE gives a clear explanation, it fails to explain why the average percentage of correct users predicted by both experiments is directly proportional to the number of users who participated. On the other hand, precision is directly proportional to the number of users.

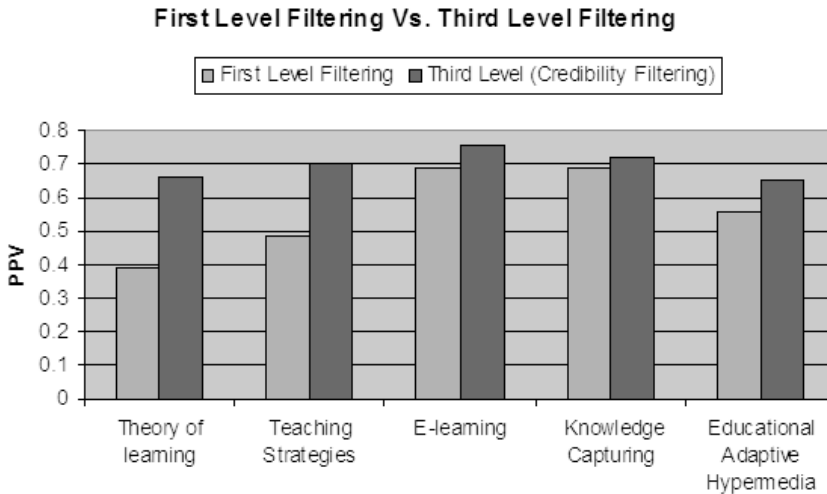


Figure 9. PPV for first level versus credibility filtering.

7. CONCLUSION

This article describes a novel model for processing online users according to their credibilities and profiles. Our approach satisfies the desire of many users to find helpers in specific domains. We have developed a pyramid collaborative filtering model. It is based on dominant meaning and users' behaviors and credibilities. Based on the proposed model, we have implemented a guide agent, which can gather helpers and recommend reliable helpers. The experimental results testify to the significant potential of our approach. These results show that filtering using dominant meaning and credibility significantly outperforms current filtering algorithms.

Even though the pyramid collaborative filtering model performs consistently better than others, the difference in performance is not very large. The performance of our system can be raised, however, by using the methods described above. We need more experiments that compare ways of combining learning styles and user behaviors, which are described above.

References

1. Razek MA, Frasson C, Kaltenbach M. A confidence agent: Toward more effective intelligent distance learning environments. In: Proc ICMLA'02, Las Vegas, NV, USA, June 24–27, 2002. pp 187–193.
2. Razek MA, Frasson C, Kaltenbach M. A context-based information agent for supporting intelligent distance learning environments. In: Proc 12th Int World Wide Web Conf, WWW2003, Budapest, Hungary, May 20–24, 2003.
3. Pazzani MJ. A framework for collaborative, content-based and demographic filtering. *Artif Intell Rev* 1999;13:393–408.

4. Balabanovic M, Shoham Y. Fab: Content-based, collaborative recommendation. *Commun Assoc Comput Mach* 1997;40:66–72.
5. Cotter P, Smyth B. PTV: Intelligent personalized TV guides. In: *Proc 12th Conf on Innovative Applications of Artificial Intelligence*; 2000. pp 957–964.
6. Aggarwal CC, Wolf JL, Wu K, Yu PS. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In: *Proc ACM KDD'99 Conf*, San Diego, CA; 1999. pp 201–212.
7. Paolucci M, Niu Z, Sycara K, Domashnev C, Owens S, Van Velsen M. Matchmaking to support intelligent agents for portfolio management. In: *Proc Seventeenth Nat Conf on Artificial Intelligence and 12th Conf on Innovative Applications of Artificial Intelligence*, July 30–August 3, 2000, Austin, TX, USA. Menlo Park, CA: AAAI Press/The MIT Press; 2000. pp 1125–1126.
8. Koller D, Sahami M. Hierarchically classifying documents using very few words. In: *Proc 14th Int Conf on Machine Learning (ICML97)*, 1997. pp 170–178.
9. Razeq MA, Frasson C, Kaltenbach M. Web course self-adaptation. In: *Proc IEEE/WIC Int Conf on Intelligent Agent Technology (IAT 2003)*, Halifax, Canada, October 13–17, 2003. pp 614–617.
10. Razeq MA, Frasson C, Kaltenbach M. Using machine learning approach to support intelligent collaborative multi-agent system. In: *Proc Int Conf on Technology of Information and Communication in Education for Engineering and Industry, TICE 2002*, Lyon, France, 2002. pp 45–50.
11. Herlocker J, Konstan J, Borchers A, Riedl J. An algorithmic framework for performing collaborative filtering. In: *SIGIR 99: Proc 22nd Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval*, 1999. pp 230–237.
12. Razeq MA, Frasson C, Kaltenbach M. Building an effective groupware system. In: *Proc IEEE/ITCC 2004 Int Conf on Information Technology*, Las Vegas, NV, USA, April 5–7, 2004. pp 202–206.
13. Razeq MA, Frasson C, Kaltenbach M. A new filtering model towards an intelligent guide agent. In: *Proc AAAI/FLAIRS-2004, the 17th Int FLAIRS Conf*, Miami Beach, FL, USA. Menlo Park, CA: AAAI Press; 2004. pp 442–449.
14. Melville P, Mooney RJ, Nagarajan R. Content-boosted collaborative filtering for improved recommendations. In: *Proc 18th Nat Conf on Artificial Intelligence (AAAI 02)*, Edmonton, Canada. Menlo Park, CA: AAAI Press; 2002. pp 187–192.
15. Mitchell TM. *Machine learning*. New York: McGraw-Hill; 1997.