A New Filtering Model Towards An Intelligent Guide Agent

Mohammed A. Razek, Claude Frasson, Marc Kaltenbach
Computer Science and Operations Research Dept.
University of Montreal
C.P. 6128, Succ. Centre-villeMontral, Qubec Canada H3C 3J7
{abdelram, frasson, kaltenba}@iro.umontreal.ca

Abstract

In E-learning systems, where both helpers (tutors) and learners are separated geographically, finding a reliable helper is one of the most important challenges. Although helpers could have a lot of useful information about courses to be taught, many learners fail to understand their presentations. A major part of this paper deals with the following challenges: do helpers have information that the learners need? Will helpers present information so that learners can understand? And can we guarantee that these helpers will collaborate effectively with learners? A new technique is filtering according to helpers' credibilities. We define ”credibility” as the dependability degree of the learners on the information presented by helpers during a learning session. We propose a guide agent, based on the pyramid model, which can group helpers. This makes it possible to recommend reliable ones. Furthermore, we developed a new statistical metric called Precision Probability Value. We have used this metric to measure statistical accuracies rather than the mean absolute error.

Introduction

Confidence Intelligent Tutoring System (CITS) (Razek, Frasson, & Kaltenbach 2002) is an E-learning system for computer supported intelligent distance learning environment. To elucidate the scenario supported by the CITS, suppose that learner A needs to discuss a specific concept, say ”queue,” in a course on data structure. To achieve this, the CITS tries to find a second learner, B, with similar interests but more knowledge. Suppose also that A prefers to begin with a written concept, lots of explanation, and does not like drawing. B, on the other hand, prefers to begin with drawn figures, lots of examples, and does not like written concepts. The problem comes when B explains something to A. He or she discusses it from his or her own point of view, which is less dependent on written concepts. Consequently, A will find it hard to understand. In this situation, the CITS, based on a machine learning technique, would predict their learning styles and thus adapt the presentation to suit both A and B.

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 a Guide Agent (GUA), one that observes discussions during a collaborative learning session and rate 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 GUA can group learners according to their credibilities and thus recommend a reliable helper. The GUA has been fully implemented and integrated into the CITS.

The GUA tries to overcome the following challenges: Does the helper B have information that A needs? Will he or she present information as A can understand? And can we guarantee that he or she will collaborate effectively with A? We propose a new Filtering framework called a Pyramid Collaborative Filtering Model (PCFM) to gradually diminish the number of helpers to just one. The proposed pyramid has four levels. Moving from a level to another de-pends 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 dominant meanings, and each of those is linked with some chunks to define it. Moreover, this concept is associated with learners who have visited it. Knowing the concept of the learner A, we can get a list of helpers.

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

To answer the third question, we filter this subgroup, in turn, to recommend the best collaborator for A. 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(Pazzani 1999). They would use similarities among users' profiles to figure out which one is similar to A's. It recommends this user to A. CB systems would compare the concept content of A with those of other users to find who has the most similar one and then recommends him or her to A.

These methods rely on ratings from users, however, and do not consider their credibilities. As a result, when unreliable users recommend bad concepts; therefore, 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.

The rest of this paper is organized as follows. Section 2 gives a brief introduction to previous work. Section 3 briefly describes the characteristics of the guide agent. Section 4 discusses the role of the classifying learners. Section 5 presents the results of experiments conducted to test our methods. And section 6 concludes the paper.

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 particular 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, Pazzani (Pazzani 1999) represented user profile as a vector of weighted words. For predicting, he applied CF to the user-ratings matrix. In Fab (Balabanovic & Shoham 1997), the relevance feedback of user is used to classify a personal filter. Moreover, another filter is related to this topic. Each document is classified by the topic filter. It is sent to a personal filter of a user ; therefore, it classifies it related to the user relevance feedback. In another approach, Cotter et al. (Cotter & Smyth 2000), CB and CF approaches are allowed to create distinct recommendations and therefore to merge their prediction directly. Horting (Aggarwal *et al.* 1999) 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.

Match (Paolucci, Niu, & K 2000) is a matchmaking system that allows users to find agents for needed services. It can store various types of advertisement coming from various applications. Sycara et al. (Paolucci, Niu, & K 2000) 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 word 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 CITS system and its functions.

Guide Agent Framework

To elucidate 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 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 GUA has three components:

- Communication between user profiles and learning styles to save any changes in the learner behavior.
- Calculation credibility value.
- Interaction with CITS user interface to post recommended helpers.

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

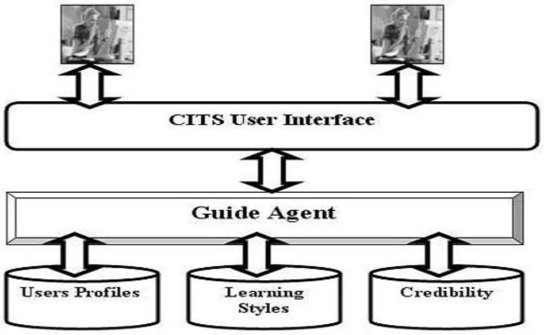


Figure 1: Guide Agent Framework

Classifying Learners

Learners’ collaboration is the key to learners’ classification. To represent the collaboration problem, suppose that group of on-line learners Π participates in the CITS. This group contains N learners $\Pi = \{L_i\}_{i=1}^{N}$. It is often 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 answering them can overcome the difficulties of collaboration. These questions are so follows: Does the helper have the information that a learner needs? Will he or she present information that the learner can understand? And can we guarantee that this helper will collaborate effectively with the learner?

We propose a four layers 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,and
- credibility Filtering

The next section illustrates how to do that.

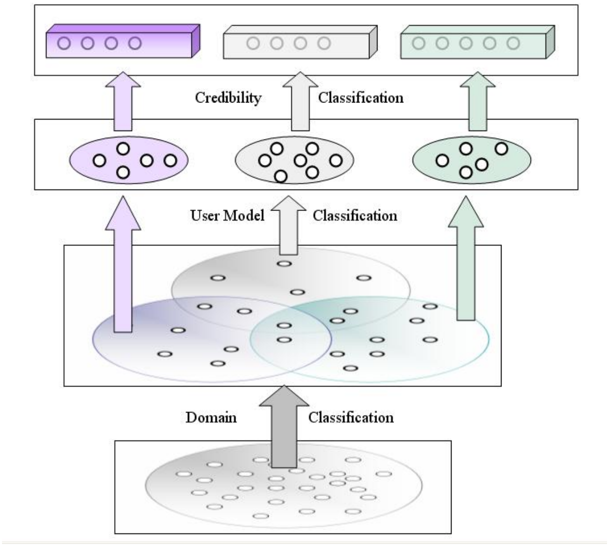


Figure 2: Pyramid Collaborative Filtering Model

Domain Model Classification

In this section, we show how the GUD 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.

Figure 3 shows a course of 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.

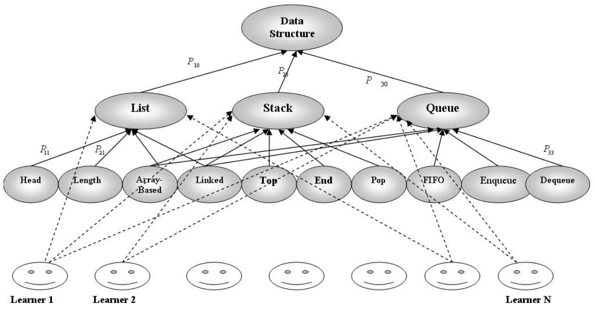


Figure 3: Hierarchical Domain Classification

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 user profile. Using this similarity, we implement a domain classification algorithm to group the helpers.

Dominant Meaning Filtering We use the dominant meaning space between a concept and a user profile to measure the closeness between them. At the moment, the problem is to evaluate users that their profiles have the highest degree of similarity to concept C_h. Here is how we can evaluate the dominant meaning similarity $S(L_v, C_h)$ between user L_v and concept C_h:

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

where

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

where C_h is a concept that learner L_v is trying to understand. Also, $\{w_1, ..., w_m\}$ is the set of the concept’s dominant meanings, and that the dominant meaning set of the L_v’s user profile is $\{v_1, ..., v_s\}$.

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 in 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:

- Calculate the similarity between users' learning styles $LS_{u,v}$.
- Compute the similarity between user behaviors $B^k_{u,v}$.
- Compute the User Modeling Degree (UMD) between two users u, v with respect to the concept $C_k$as follows:

$$UMD(u,v) = (\lambda LS_{u,v} + (1-\lambda)B^k_{u,v}), \qquad (2)$$

where $UMD(u,v)$, $LS_{u,v}$, and $B_{u,v}$ represents the similarity degree between the user's modeling, learning style, and behavior respectively. To simple the experimental results, we choose $\lambda = 1/2$. the calculation of these degrees are showed in details at (Razek, Frasson, & Kaltenbach 2004).

Credibility Model Filtering 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. Using this representation, we can represent the credibility problem as a matrix (Razek, Frasson, & Kaltenbach 2004):

$$\mathbf{CM} = \begin{pmatrix} L_{11} & L_{12} & \dots \\ L_{21} & L_{22} & \dots \\ \vdots & \vdots & L_{NN} \end{pmatrix} \qquad (3)$$

where, link L_{ij} 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 this matrix, we can calculate the credibility of learner L_i as follows:

$$\Omega_v = L_{vv} + \sum_{f=1, f\neq v}^{N} L_{vf} - \sum_{f=1, f\neq v}^{N} L_{fv} \qquad (4)$$

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

Experiments and results

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

Data Set

We use IFETS Discussion List[1] to collect a set of messages in order to evaluate the performance of our algorithm. The IFETS discussion is provided by International Forum of Educational Technology & Society, which is a subgroup of IEEE Learning Technology Task Force[2]. There are two kinds of discussion in the IFETS: formal and informal ones. Formal discussions are topic-based and occur for one to two 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[3]. 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 27 Feb 2002 to 31 July 2003 and considered users who had submitted 2 or more messages. Table 1 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. We used 80% of the collection as a training set and 20% as a test set. All messages have already been classified manually into 5 concepts.

Table 1: Collection used for experiment

Collection	Description
Number of messages	502
Number of users	95
Number of topic (concepts)	5
Average messages per week	15

Evaluation Methodology

To evaluate the accuracy of a prediction algorithm, researchers used two main approaches: statistical accuracy metric and decision support metric (Herlocker *et al.* 1999). Furthermore, we suggested a new statistical metric called Precision Probability Value. To measure statistical accuracy, we used the mean absolute error (MAE) metric-defined as the average absolute difference between predicted and actual provide list. We used MAE and PPV to account prediction experiments, because they seem to be a simplest measure of overall error.

Mean Absolute Error We used Mean Absolute Error (MAE), a statistical accuracy metric, 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:

$$MAE(C_k) = \frac{\|N_p - N_c\|}{N} \qquad (5)$$

where

$$N_c = \sum_{i=1}^{N_s} \Big[\sum_{j=1}^{N_p} \Theta(L_i, L_j) \Big]$$

, and

[1] http://ifets.ieee.org/discussions/discuss.html

[2] http://lttf.ieee.org/

[3] http://ifets.ieee.org/periodical/

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

Precision Probability Value 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 .

$$PPV(C_k) = \frac{N_c}{N_s} \times \frac{N_c}{N_p} \quad (6)$$

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

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, using the method proposed in (Razek, Frasson, & Kaltenbach 2003).

Training Stage In the training stage, our hierarchy consisted of 5 concepts: the future of learning; theory of learning; teaching strategies; e-learning; and capturing knowledge. For the comparative experiments, we constructed the dominant meanings of these concepts. We associated users with each concept that they used in its discussion. In the same sense, we applied the same technique at (Razek, Frasson, & Kaltenbach 2003)on the collection to find the threshold of dominant filtering (μ) and credibility filtering (ν)respectively. The algorithm is summed up as follows: **Training Stage Algorithm***:*

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

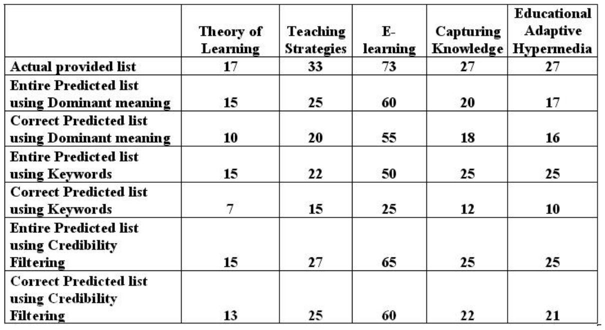
Test Stage

The test stage was to conduct three tests. We implemented three methods as mentioned before, 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 bases 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 in common some specific concepts.

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.

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 Mean Absolute Error (MAE) and Precision Probability Value (PPV). Figure 4 shows the results. It indicates for each concept the number of users on the actual provided list (i.e. those who actually participated on 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.



	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

Figure 4: Experiments Results

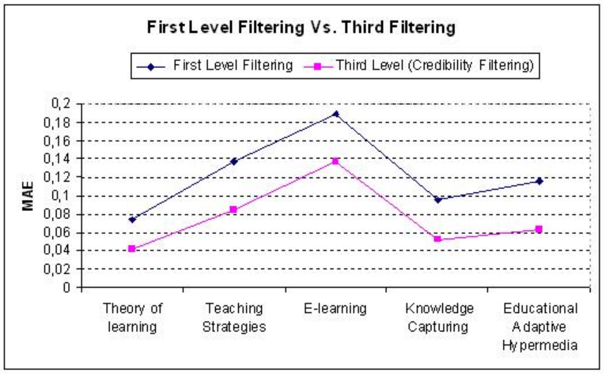


Figure 5: MAE for First level vs. Credibility Filtering

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.

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

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 prediction task. The plot of the MAE and PPV of the first filtering and the credibility of the filtering algorithm are depicted in Figure 5 and Figure 6 respectively.

The figures show that our credibility method yields more performance improvement according to both mea-

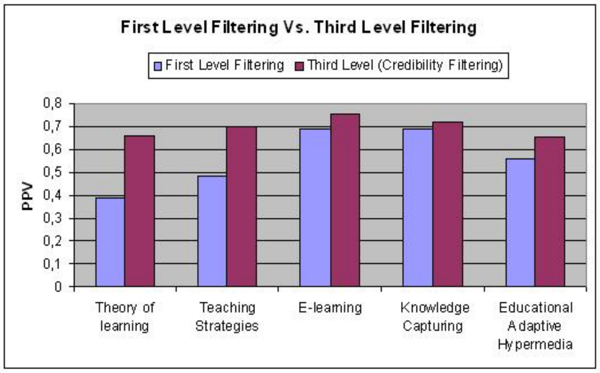


Figure 6: PPV for First level vs. Credibility Filtering

sures: 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.

Conclusions

In this paper, we present the methodology and functionalities of the guide agent. based on the pyramid collaborative filtering model, it can gather helpers and recommend reliable helpers.

Furthermore, we developed a new statistical metric called Precision Probability Value. We have used this metric to measure statistical accuracies rather than the mean absolute error. 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

Aggarwal, C. C.; Wolf, J. L.; Wu, K.; and YU, P. S. 1999. *Horting Hatches an Egg: A New Graph-theoretic Approach to Collaborative Filtering.* San Diego, CA. pp. 201-212.: In Proc. of the ACM KDD'99 Conference.

Balabanovic, M., and Shoham, Y. 1997. *Fab: Content-based, collaborative recommendation.* 40(3),pp.6672: Communications the Association of Computing Machinery.

Cotter, P., and Smyth, B. 2000. *PTV: Intelligent personalized TV guides*. pp.957964: In Twelfth Conference on Innovative applications of Artificial Intelligence.

Herlocker, J.; Konstan, J.; Borchers, A.; and Riedl, J. 1999. *An algorithmic framework for performing collaborative filtering*. pp.230237: In SIGIR 99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Paolucci, M.; Niu, Z.; and K, K. S. 2000. *Matchmaking to Support Intelligent Agents for Portfolio Management*. In Proc. of the AAAI2000.

Pazzani, M. J. 1999. *A framework for collaborative, content-based and demographic filtering*. 13(5-6):pp. 393-408: Artificial Intelligence Review.

Razek, M. A.; Frasson, C.; and Kaltenbach, M. 2002. *A Confidence Agent: Toward More Effective Intelligent Distance Learning Environments*. Las Vegas, USA, June 24-27: ICMLA'02.

Razek, M. A.; Frasson, C.; and Kaltenbach, M. 2003. *A Context-Based Information Agent for Supporting Intelligent Distance Learning Environments*. Budapest, Hungary, 2003: Twelfth International World Wide Web Conference, WWW2003: 20-24 May.

Razek, M. A.; Frasson, C.; and Kaltenbach, M. 2004. *Building an Effective Groupware System*. Las Vegas, NV, USA: IEEE/ITCC 2004 International Conference on Information Technology, April 5 -7.