(438) ENTER Proceedings 2002
ENTER 2002 Innsbruck

Knowledge Bases and User Profiling in Travel and Hospitality Recommender Systems

Joaquin Delgado,[a] and
Richard Davidson[b]

[a] TripleHop Technologies, Inc.
New York, NY, USA
joaquin@triplehop.com

[b] Ski Marketing Corporation.
Houston, TX, USA
richardd@ski-europe.com

Abstract

Recommender systems for the travel and hospitality industries attempt to emulate offline travel agents by providing users with knowledgeable travel suggestions. The ultimate goal is to help the user in the travel planning phase trough offering a comfortable understanding of the options and also giving a select set of alternatives. This paper presents a novel approach for constructing such systems: a) creating a domain specific dialog model, b) semi-automatically building a knowledge base of ratings for the items of interest (i.e. destinations, airfare, hotel, vacation packages), and c) generating personalized recommendations ordered by relevancy. Items of interest are selected to best fit the needs of travelers, based on their individuality, interests and preferences. Explicit and tacit user feedback, as well as the extrapolation of individual user interests through attribute-based collaborative filtering, allows the system to learn rich profiles and refine its knowledge base, generating ever-improving recommendations. Empirical results confirm the hypothesis that recommender systems tend to accelerate the decision-making process by showing an improvement in look-to-buy ratios of up to 4.95 times, when compared to normal purchases on a ski travel e-commerce site.

Keywords: artificial intelligence, virtual sales agent, recommender systems, user modeling, collaborative filtering, knowledge-based systems.

1 Introduction

The decision to visit a destination, select a vacation package or pick a hotel typically relies on the information available to the tourist (Rita, 2000; Vogt and Fesenmaier, 1998). Processing the large amount of information available from online sources, such as destination marketing and travel websites, can become a complex and time-consuming process. Internet travel sites have been extremely efficient at enabling e-commerce transactions, allowing users to directly buy, with minimum human intervention, transportation and hospitality services. However, the same sites have been relatively poor at aiding the user during the planning phase, most of the time assuming the

users know exactly where they want to go, what to do and where to stay. It is not uncommon that users perform extensive destination-based research elsewhere before actually making the purchase on-line. The direct result of this, among other reasons, is low conversion rates (look-to-book ratios). Non-commoditized travel products, such as packaged travel, vacations, cruises and escorted tours introduce an additional level of complexity when offered online. Traditionally, travel suppliers that offer such services could expect lower conversion rates compared to the ones of a typical "agency-type" online travel services that mainly deal with just air or hotel.
Recommender systems for travel and hospitality attempt to emulate the user interactivity of offline travel agents. They provide users with knowledgeable travel suggestions. They are very effective customer relationship management decision-support tools, providing customized travel recommendations to best fit the needs of individual travelers, based on their individuality, interests and preferences. In addition to improving look-to-book ratios, recommendation technology enhances the customer experience and increases customer loyalty, resulting in higher transactional revenues.
The paper explains how recommender systems for travel and hospitality can be built. It is organized as follows: in chapter 1 and 2 we explain the basics of recommender systems technology and the difficulties of applying it directly to problems in the travel and hospitality domain; chapter 3 describes the TripMatcher application, explaining the details about the knowledge base construction processes, the underling matching algorithms and the feedback mechanisms present in the system; in chapter 4 and 5, we present a case study along with the achieved results; finally in chapter 6 and 7, we discuss similar systems and research, present our conclusion and future work.

2 How Do Recommender Systems really Work?

Seminal works in information agents research (P. Maes 1994; U. Shardanand and P. Maes, 1995) describe *Recommendation Agents* in terms off their filtering techniques:

1. Feature-based filtering (also called Content-based filtering);
2. Automatic Collaborative filtering (ACF);
3. Constraint-based filtering (CSP; as for constraint satisfaction programming).

Traditionally, recommender systems (Resnick, Paul and Hal R. Varian, 1997) are often referred to as "collaborative filtering systems" which assist and augment the transfer of recommendations between members of a community. A typical system collects preferences or opinions from individual users, then aggregates and transfers those recommendations to other members of the community.
In recent years, interest in recommender systems has dramatically increased, driven primarily by demand for Internet personalization applications. Many major e-commerce web sites are currently using recommender systems to personalize their content layout and target sales. Most notably, Amazon.com, the largest online retailer, makes extensive use of recommender system technology.
These systems usually track purchasing behavior and make predictions based upon correlations performed on numeric ratings, which are implicitly and explicitly captured from users to the items being recommended. These ratings are usually normalized and

stored in a User-Item rating matrix (see Figure 1) that tends to be very sparse, with each row-vector representing a user profile and each column-vector an item profile. As observed in equations 1 and 2, parson-*r* correlation between user profiles can be used as a similarity measure to discover "like-minded" users, acting as a weighting factor for interest prediction. Then, only the items with higher predicted value are presented, as a recommendation, to the user.

$$
\begin{array}{c c}
 & \begin{array}{cccc} I_1 & I_2 & \cdots & I_n \end{array} \\
\begin{array}{c} U_1 \\ U_2 \\ \vdots \\ U_m \end{array} &
\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}
\end{array}
$$

Fig. 1. User-Item Rating Matrix

$$prediction(U_{active}, I_j) = \overline{U}_{active} + \sum_{i}^{m} similarity(\vec{U}_{active}, \vec{U}_i) \times (a_{i,j} - \overline{U}) \quad (1)$$

where $\overline{U}_x = \frac{\sum_{i=1}^{n} a_{x,i}}{n}$

$$similarity\ (\vec{U}_x, \vec{U}_r) = \frac{\sum_{i=1}^{n} (a_{x,i} - \overline{a}_{x,i})(a_{r,i} - \overline{a}_{r,i})}{\sqrt{\sum_{i=1}^{n} (a_{x,i} - \overline{a}_{x,i})^2 \sum_{i=1}^{n} (a_{r,i} - \overline{a}_{r,i})^2}} \quad (2)$$

where $\overline{a}_{j,i} = \frac{\sum_{i=1}^{n} a_{j,i}}{n}; j \in \{x, r\}$

Collaborative filtering techniques, however, fall short when trying to predict recommendations for complex products, such as travel purchases, in which the automation of "word of mouth", by itself, is not sufficient to support the decision-making process. To recommend a destination or a hotel, just because similar people preferred it in previous occasions, makes less sense for travel and hospitality than for more taste-based commodities such as movies, music and books[1]. Several attempts to use clustering techniques and combine collaborative filtering and content-based filtering in other domains can be found in the literature (Ungar, L.H. and D.P. Foster 1998; Delgado and Ishii 1998; Pazzani, M. - in press). However, in travel and leisure, other factors, such as seasonality, distance, trip-specific settings and individual interests and activities, must be considered when generating meaningful recommendations, thus combining the three original techniques for recommendation. The prediction problem naturally escalates, as not just historical rating data but also contextual and preference data need to be weighted and combined in a final *matching function*.

[1] Recommender systems were originally applied to suggest books, movies and CDs (i.e. Amazon.com, www.moviecritic.com, CDNow.com, www.launch.com)

3 TripMatcher: A Recommender for Travel & Hospitality

In many cases, website visitors, who may have made the decision to travel to a general destination (the Caribbean, Europe, someplace warm, etc), have not selected a specific destination or have not planned what to do while they are at their selected destination. In the traditional call center business model, travel providers are forced to hire experts with in-depth knowledge of the destinations they offer in order to sell their products successfully. Large amounts of money are ineffectively spent in a sales effort to assist customers find the destination that best matches their requirements and preferences, before the booking process can even begin.

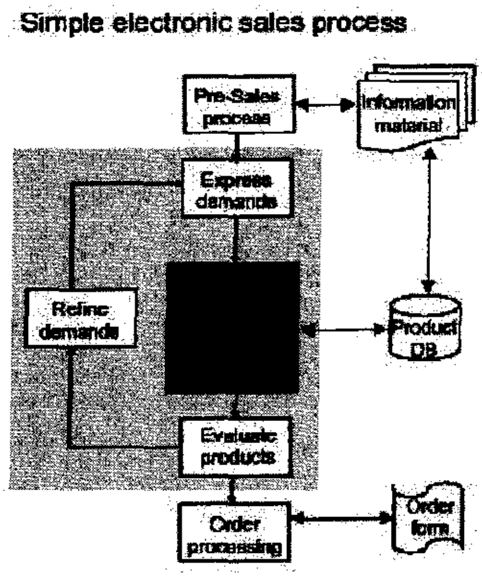

Fig. 2. The role of recommendations in an electronic sales process

In simple terms, the recommendation engine acts as an experienced online salesperson (see Figure 2) that interacts with the customers, learns their preferences, and responds with highly targeted, relevant information and personalized recommendations, supported by rich, original and customized content. Rather than hoping customers find travel products buried amidst tons of offerings, the travel providers can proactively offer destinations, resorts, itineraries and products that best match their customers' preferences.

In order to replicate the in-depth knowledge needed to generate credible recommendations and explanations from start, these systems need to be bootstrapped by acquiring a knowledge base of the destination(s) and/or travel products. This can be achieved either by mining textual content, proprietary to the travel provider, or having it developed by outside professional researchers to provide a comprehensive range of options and travel solutions for the potential traveler. For each destination, the knowledge base comprises both quantified evaluation criteria and deep text content. The technology then *interacts* with customers and provides highly targeted, relevant information and personalized recommendations. It helps them quickly find a destination, itinerary, tour, cruise, local event or experience that best fits their preferences, interests, objectives, pace, lifestyle and budget.

3.1 Building the Knowledge Base

There are several key concepts introduced in the design of the knowledge base, but perhaps the most relevant is the generic, multi-level tree *Domain Model* or *Search Taxonomy*, which also impacts the graphical user interface (GUI).

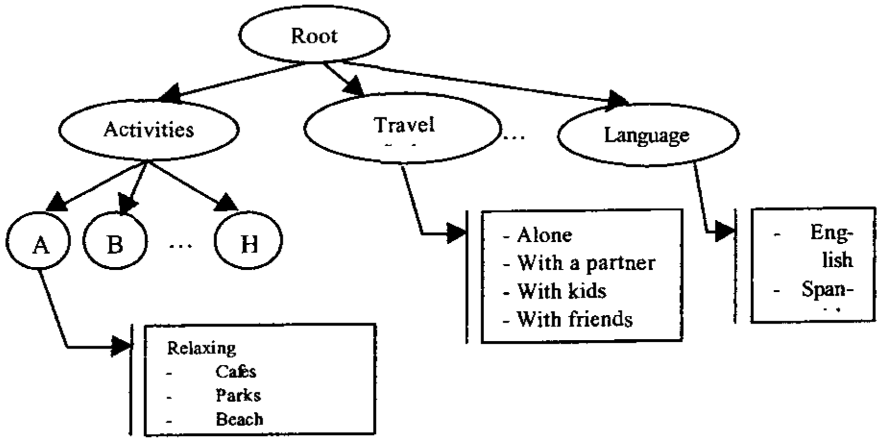

Fig. 3. Domain Model or Search Taxonomy

In the model, depicted in Figure 3, each node represents an attribute used as a criterion in the decision-making process. Attached to the node is a label, and in some cases text, to be used as a question that is displayed in the interface dialog. A node may also contain the type of interface menu object to be used for its children (i.e. multi-criteria check-boxes, combo-box radio buttons, single criteria radio buttons, pull-down menus), used to dynamically generate the interface (See Figure 4).

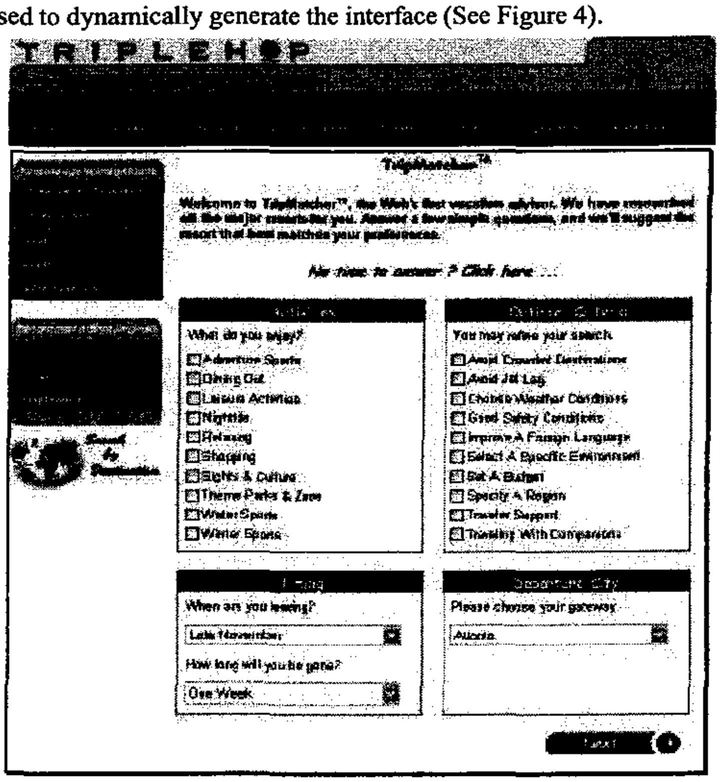

Fig. 4. TripMatcher's dynamically-generated interface.

During the actual session, the navigation of the tree determines the actual interface dialog, engaging the user into a simple question-and-answer conversation. The user can stop this process at any level, requesting recommendation with whatever input he or she has given to that point in time. Recommendations can be obtained without any input if there is already a model of the user (i.e. returning user – see "*No time to answer? Click here*" in Figure 4).

The domain model should not be confused with a directory or travel *taxonomy*, shown in Figure 5, typically organized as a browsable/searchable hierarchy (Fodness, Dale, and Brian Murray, 1998).

- Continent or World Region
  - Country
    - Region or State
    - City or Destination
      - Products
        - Airfare
        - Hotels
        - Vacation Packages
        - Etc.

Fig. 5. Item Taxonomy

In order to bootstrap the system and provide dynamically generated content to support the destination and/or product recommendations, a mapping between the search criteria and nodes in the item hierarchy has to be established.

There are basically two complementary ways of building this map, which constitutes the heart of the knowledge base. First, content and ratings are provided by experts. Second, ratings are automatically-generated through text mining of product descriptions in electronic format.

3.1.1 Destination Content and Ratings, provided by Experts

Like an experienced travel agent, TripMatcher supports its destination recommendations with rich, original and customized content. TripMatcher includes a knowledge base of over 400 destinations, developed by 250 professional researchers worldwide to provide a comprehensive range of options and travel solutions.

For each destination, travel experts have ranked 88 activities (Fesenmaier, D. R., and S.R. Lieber, 1988), during different times of the year. Interests from museums to snorkeling to children's recreational activities get ranked in order of relevance according to user preferences. If the travel client does not offer travel products to a specific destination, these destinations are excluded from the knowledge base of this specific client. TripMatcher never recommends something the client does not sell.

For example, if the user says that she wants to snorkel and scuba dive for a week in February, leaving from Los Angeles, TripMatcher may recommend Maui as a good destination, and will support this recommendation by providing a rating and a few paragraphs about snorkeling and scuba diving in Maui at that time of the year. Subsequently, TripMatcher will recommend the clients' hotels, flights or vacation packages, or any travel product that they may offer for Maui in February.

3.1.2 Deriving Ratings on Products by Mining Textual Data

We believe recommending specific products is the key to increasing sales at any e-commerce site. The goal is to achieve this without the need for time-consuming and error-prone manual indexing. These errors are almost impossible to avoid when dealing with a large and constantly changing inventory.

TripMatcher's indexing and classification technology automatically processes travel products descriptions (hereafter described as "document"), ranks them against the search taxonomy, and suggests the packages, tours, tickets or hotels that the user is the most likely to buy, thus providing the ability to deliver the "last mile" of the recommendation. Based on the same technology, our engine can identify and recommend activities, attractions and events that best match a user profile, without the need for a human or human-assisted indexing, rating or classification.

Documents are represented within the vector space model[2]. Automatic rating is achieved by expanding a query for each concept in the search taxonomy so that the relevance of each document to each node in the domain tree can be assessed. Attached to each node, there is a list of keywords that is combined with positive and negative adjectives to yield an objective rating for each exposed document by means of a proprietary weighting scheme similar to that of term frequency – inverse document frequency or TFIDF (Salton, G. 1989). For example, the criterion *dining out*, under activities in the search taxonomy, would expand to "dining, gourmet, restaurant, entrée, hors d'oeuvre..." Then used in combination with positive adjectives, such as "fine, good, great, tasty, delicious, yummy, appetizing, delectable, scrumptious" and negative adjectives, such as "bad, worst, unappealing, disgusting, not fresh, bland, overcooked", the rating algorithm generates a value in [0,1] for each one of the elements being evaluated (i.e. descriptions about hotels, vacation packages, things-to-do, etc.).

The detailed algorithms and procedures used to obtain ratings on individual travel products are beyond the scope of this paper.

3.1.3 Self-corrective Statistical Rating Systems

The knowledge base, which consists of ratings on items of interest created both automatically and manually, is not exempt from errors. This is principally due to the subjective nature of human evaluations, and to the fact that automatic text classification is still an active area of research (Dumais, S. T., et al. 1998; T. Joachims 2001). Additionally, automatically-generated ratings, though objective, can only be as good as the descriptive power of the content it processes. For these reasons, we have created a feedback mechanism that is able to correct the assigned ratings depending on the selected criteria, the user performance and user reactions after being presented with the list of recommendations. The main actions we log in each session are:

- Positive implicit feedback:
    - printing an item
    - intending to book a reservation
    - actual purchase

---

[2] The vector space model treats documents as a bag-of-words and is most commonly used in information retrieval and text mining.

- Negative implicit feedback
    - None of the positive feedback occurs after viewing an item.
- Explicit feedback
    - rating an item

For any explicit rating at session t, we directly receive a feedback rating in [-1,1]. For each positive implicit action, we assign a feedback rating in (0,1] and for each implicit negative feedback a feedback rating in [-1,0). Every time an action is performed on item i, we add a micro-value to the predicted rating of all attributes selected during the session. This micro-value will depend on the feedback given and the order of the item in the result list of recommendations. The predicted rating $\hat{r}_{i,a}^{t+1}$ on item i for attribute a, is recursively calculated as follows:

$$\hat{r}_{i,a}^{t+1} = \hat{r}_{i,a}^{t} + \left(\frac{\beta}{\beta + (j_{\max} - j)^2} \right) \frac{f}{2} \qquad (3)$$

Where f is the feedback value, j is the descendent order, $j_{\max}$ is the last index in the recommendation list and β is a parameter in (0,1], that eventually determines the micro-incremental ratio. Initially the predicted rate $\hat{r}_{i,a}^{t}$ is same as the base rate $r_{i,a}$, already present in the knowledge base (i.e. $\hat{r}_{i,a}^{0} \leftarrow r_{i,a}$). When the predicted rating exceeds a threshold d, typically an increment of more that 0.25 compared to the base rating, and the collected data is statistically significant[3], we proceed to assign the predicted rating to the base rating (i.e. $r_{i,a} \leftarrow \hat{r}_{i,a}^{t}$ when $|\hat{r}_{i,a}^{t} - r_{i,a}| > \delta$), closing the loop.

3.2 User Modeling and Attribute-based Collaborative Filtering

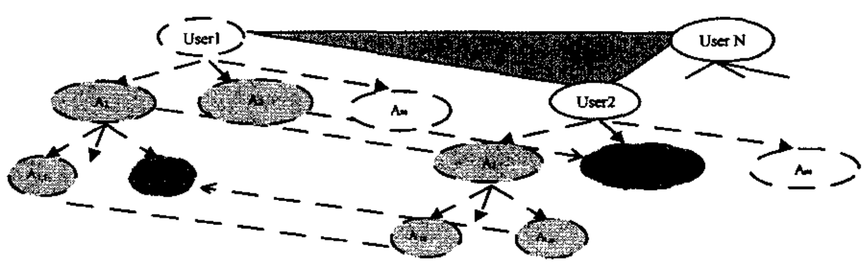

Fig. 6. Interest Extrapolation through Attribute-based Collaborative Filtering

Our objective is to build user models that will allow us to match their preferences with items of interest. We are also interested in looking at selections done by "similar users" and extrapolating profile values in order to produce new insights into the recommendation process. The idea is not to eliminate candidate items, but rather favor those that have high values in predicted criterions.

For each user, we have a user profile as shown in Figure 6. Note that the actual tree structure depends on the domain model. The profile only contains information about the attributes the user has "visited" (nodes in light grey, Figure 6). The profile is a

---

[3] We use standard ?^{2} test.

collection of two values [0,1], assigned to each node of the tree: a short-term memory (STM) and a long-term memory (LTM) values. The long-term memory is calculated as the ratio of the number of times a user has selected an attribute, divided by the total number of searches performed by the user. The STM of a given attribute is assigned 1 if an attribute has been selected by the user. It begins to decay at a constant factor $\beta \in [0,1]$ if it doesn't continue to be selected in the following query-sessions the user performs, as shown in the following equation:

$$STM^{(t+1)} = STM^{(t)} \times \beta \tag{4}$$

The final calculated value for each node is called the *attribute-interest ratio*, defined as the average between the LTM and STM. Consequently, one can recognize the attributes in which the user has expressed more interest by verifying the attribute-interest ratio (the closer to 1, the more interest shown). The concepts behind LTM and STM are that of the user's long-term interest and short-term interest ratios on a particular attribute.

Now we are ready to perform collaborative filtering in order to predict the interest on attributes the user has not shown explicit interest yet (i.e., unvisited nodes in blue, Figure 6). This process is called *attribute-based collaborative filtering*, since it uses information about the similarity between users in order to predict the interest on an attribute the user has not even thought of yet, thus providing recommendations with new insights.

First, we decide for which node we should predict an attribute-interest ratio, among those that our target user U has not visited in the past. This is done by selecting the most popular "visited" node among those that belong to the profile of similar users. The similarity function among users is pre-calculated offline by observing the overlap and the attribute values of each pair of profile trees in the domain space. The basic equations for this tree similarity between a user U and a user i is:

$$sim(U,i) = \frac{\sum_{j=1}^{M} \hbar_j^{\,2} v_{U,j} v_{i,j}}{\sqrt{\sum_{j=1}^{M} (\hbar_j v_{U,j})^2 \sum_{j=1}^{N} (\hbar_j v_{i,j})^2}} \tag{5}$$

where $v_{U,j}$, $v_{i,j}$ are the attribute-interest ratio of user U and user i for attribute j respectively, and and $\hbar_j$ is a factor that depends on the depth of the attribute j and the maximum depth of the domain tree. M is the number of common attributes registered in both user profiles. Once we have selected a target attribute j, its predicted LTM value for user U, $\widehat{v}_{U,j}$, is calculated as:

$$\widehat{v}_{U,j} = \frac{\sum_{i=1}^{N} sim(U,i)(v_{i,j})}{\sum_{i=1}^{N} sim(U,i)} \tag{6}$$

where N is the number of users in the database with non-zero similarity that have an attribute-interest ratio for j. The STM for j is set to one every time it is selected as target, but decays in the same way as other attribute's STMs, if not selected for prediction. If the user visits a predicted node in the future, the event-based information overwrites whatever prediction has been done.

Still, these predictions and user profiling can be sensitive to errors in the interaction with the system or changes in user interest. In order to account for these scenarios, the system provides ways to correct the captured interest about certain attributes by letting the user explicitly update his profile. This is done through a graphical interface that allows the user correct the "amount of interest" the system has learned for a given attribute and set it to its actual value, overwriting the interest-attribute ratio already registered in the system, thus producing more accurate results.

3.3 The Matching Engine

The Matching Engine implements multiple filtering techniques such as Constraint-based, Content-based and Collaborative information filtering (Delgado 2000). It was built modularly in order to be flexible and extensible. The output of the engine is a finite (generally small) ordered list of recommended items (i.e. destinations and/or hotels and/or packages, etc.) out of a large universe of possible selections that best satisfies the end user's request. It also produces explanations for the recommendations through dynamically-created content and ratings for each important criterion. For the ME to work, at least one of the following inputs is needed:

- A single or compounded[4] user profile
- A request for recommendations (inputted preferences/criteria through the dynamically generated GUI)

The system also has to have static or real-time access to:

- A knowledge base of time-dependent/independent attribute-based item ratings.
- Contextual information such as flying/driving time between cities, time-zones, average temperatures, etc.

The high level steps of the algorithm are explained below and in the flowchart shown in Figure 7.

1. **Front-end Generation**: The database-driven interface is generated and the user selects search options. For example, the user might decide to look up activities and select *winter sports*, or to go even deeper and select the type of sport he would be interested in (*e.g., snowmobiling*).
2. **Update User Profile**: All inputted search criteria are logged and saved into the user profile and preference databases.
3. **Context Filters**: Eliminates undesired items by applying domain-specific rules called context rules, calculating for the remaining items a normalized context matching value [0,1], which averages the score on each rule. The individual scores will depend on the contextual information used in each rule. For example, the "vacation length" rule eliminates destinations that are out of reach from the selected departure city, given the length of the trip (a weekend, a week, a month, etc.) The score for each item will have a value inversely proportional to the flying time to that destination.

---

[4] A compound profile is an average of two or more individual profiles used to represent group interests.

4. **Content Filters:** Receives the reduced list out of the context filters and eliminates undesired items by looking at the time dependencies and numerical values of each inputted criteria for each item and applying threshold rules defined by the system administrator through the domain modeling tool. For each remaining item, it calculates a normalized content-matching value that will depend on the numeric values in the rating knowledge base, the user profile and the hierarchy (model) of the search criteria.

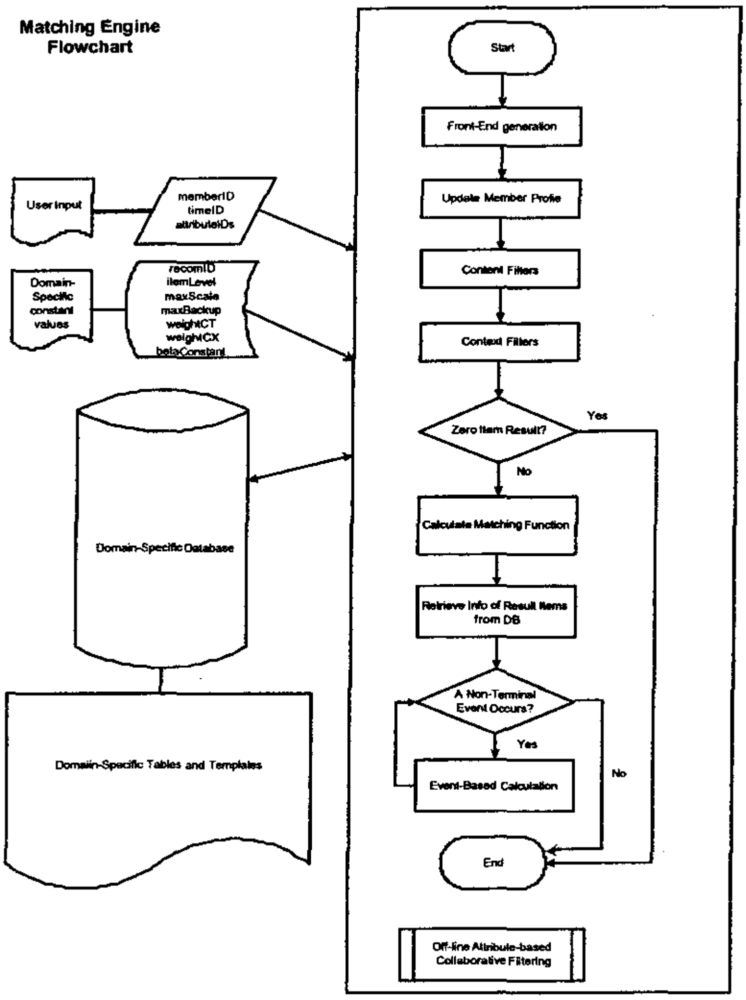

Fig. 7. Matching Engine Flowchart

5. **Calculate Matching Function**: This is the core function of the system. It generates a matching percentage for each item that still remains in the list. When calculating the matching percentages, the current search options and the learned user profiles are both taken into account. The matching percentage for each item I is calculated using the following weighted average that combines the attribute-

interest ratio $v_{U,j}$, a factor $\hbar_j$ that depends on the depth of the attribute j and the maximum depth of the domain tree and the rating $r_{I,j}$ that item I has on attribute j:

$$Match(U,I) = \frac{\sum_{j=1}^{M} v_{U,j} \hbar_j r_{I,j}}{\sum_{j=1}^{M} v_{U,j} \hbar_j} \times 100 \qquad (7)$$

6. **Retrieve Result Items Information from the Database**: The sorted list of items is presented to the user. Both the criteria selection and the actions on the result list are logged and later processed in order to update the user preference model. A link to additional text for each item can be aggregated around the attribute structure and shown to the user, together with an explanation of the recommendation through the expert's ratings for the selected item.
7. **Event-Based Calculation**: Once feedback has been received through the various types of events defined by the system administrator, the predicted ratings are updated as explained in section 3.1.3.
8. **Offline Attribute-based Collaborative Filtering**: Collaborative filtering is performed at the attribute level to predict the weight of attributes the user has not explicitly shown interest in as explained in section 3.2. Along with the feedback process (click-stream analysis), this is a "behind the scenes" learning mechanism that primarily affects the user profiles and the way they are modeled.

Finally, each filter/rater can be controlled, if required, by modifying its individual weight through the use of the administrative tool. These weights determine the effect of this module in the overall matching calculation. Further layers of filters/raters can be easily added to accommodate new dynamic rules that are not covered by the existing components. This new layer would receive as input the item being examined, returning whether it should or not be eliminated from the result list and, in later case, the normalized rate (in [0,1]) that this item deserves. In this way, the logic of the filter/rater is left to the programmer, whose job is to implement efficient business rules and has access to more company-specific data. A call needs to be added from the original code. A weight and order are assigned to this filter in order to affect the end matching results. An example of this would be to implement a filter that considers dynamic pricing data as an input to rate the suggested items (destinations/flights, packages or hotels) in order to favor some over the others in the final recommendation.

4 Ski-Europe.com: A Case Study

Founded in 1985 as a traditional tour operator, Ski Marketing Corporation's Ski-Europe.com is an e-commerce site specializing in winter ski vacations from North America to Europe. Interactive Week recently featured Ski Marketing Corporation on an "Interactive 500" list of fastest growing Internet companies.
The rapid growth of site traffic made it difficult to respond to inquiries in a timely fashion. With support costs rising, it was quickly becoming inefficient to manage the time-consuming process of assisting customers with personal recommendations and first-stage assistance.

SkiMatcher, the customized TripleHop TripMatcher recommendation engine, helps Ski Marketing Corporation and its affiliates more effectively manage business by automatically generating personalized recommendations (see Figure 8) based on a visitor's needs and preferences.

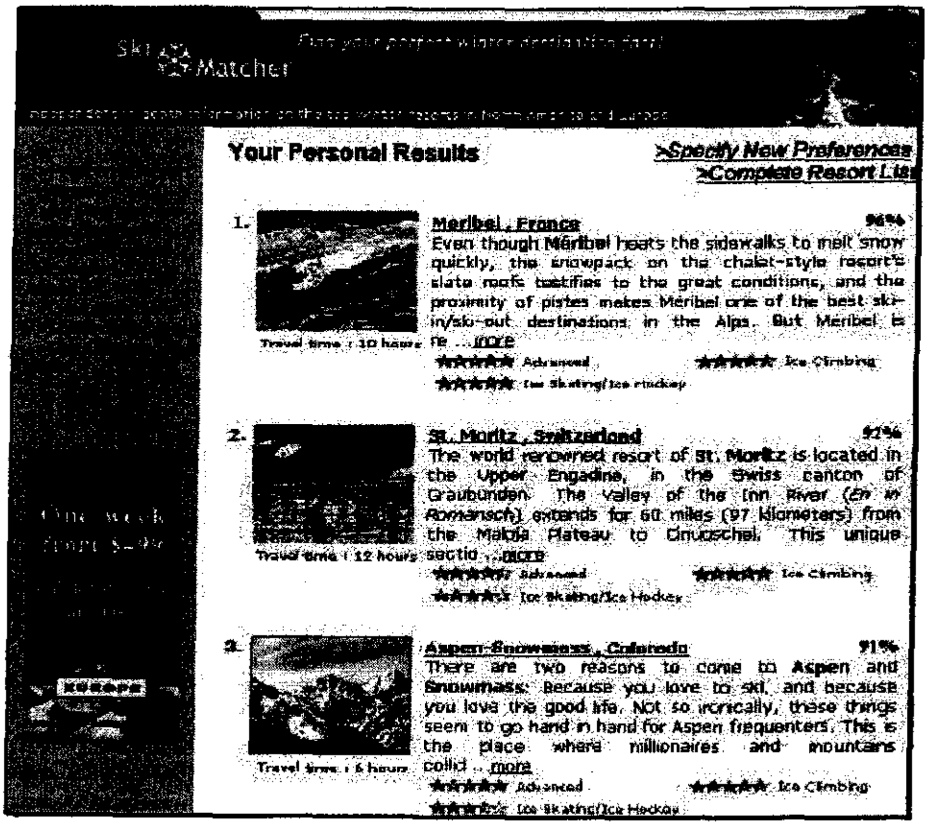

Fig. 8. SkiMatcher Recommendation Page

5 Results

Table 1. User Comments

While it cannot entirely replace a knowledgeable and experienced destination specialist, SkiMatcher interacts with web visitors in a similar manner: learning the visitors' requirements and preferences, and responding with personalized relevant suggestions supported by customized original destination information.

Jo Olsen "I have been planning a Euro ski trip for a couple months. I chose three resorts. My choices were in the top four of the system's choices! This is only a small test but the outcome bodes well for your matching system."
Adam Rosenberg "I think the site is a vast improvement. Love the SkiMatcher. We'd already planned to go to Cortina and Val Gardena this year, and when we typed in what we were looking for they came up in the top 6!"
Dan Vellieux "I like the format. It provides the ability to limit the search to locations and criteria that may suit the traveler. Of course the real value is in the results of the search".

As evidenced by the conversion data (Table 2), people who use the SkiMatcher are much more likely to request assistance in purchasing their travel arrangements. For example, in the month of August, there were a total of 15,560 unique visitors at Ski-Europe.com, out of which 1,673 used SkiMatcher. Out of those members that used SkiMatcher, 143 completed a request for proposal (RFP) indicating a conversion rate of 8.55%. When compared to the 2.61% conversion rate observed on those sessions that did not used the recommender system the increase in conversion is in the order of 327% or 3.27 times the conversion observed by non SkiMatcher users. The difference in conversion reported was consistent, yielding an average of 414% of comparative increase, over a period of four months.

Table 2. Summary of results over 4 months, year 2001

	July	Aug	Sep	Oct
Unique Visitors				
SkiEurope Total	10,714	15,560	18,317	24,416
SkiMatcher Users	1,027	1,673	1,878	2,558
Not SkiMatcher Users	9,687	13,887	16,439	21,858
RFP's Generated				
SkiEurope Total	272	506	445	641
SkiMatcher Users	75	143	161	229
Non SkiMatcher Users	197	363	284	412
Conversion				
SkiEurope Total	2.54%	3.25%	2.43%	2.63%
SkiMatcher Users	7.30%	8.55%	8.57%	8.95%
Non SkiMatcher Users	2.03%	2.61%	1.73%	1.88%
Increase in Conversion				
SkiMatcher / Non SkiMatcher	359%	327%	496%	475%

The SkiMatcher has become an attraction to SkiEurope's website; users' comments have been overwhelming favorable. See Table 1 for some examples of comments made by users of SkiMatcher.
In addition, it has been possible to secure placement of co-branded versions of the SkiMatcher on a number of major travel and recreation websites that receive high traffic counts, thereby leveraging SkiEurope's visibility and e-commerce opportunities.

6 Related Work

Case-Based Personal Travel Agent (Waszkiewicz, Cunningham & Byrne, 1999) helps user plan and book travel by identifying similar cases. A very similar approach, based on case-based reasoning (CBR), is being used in the *Intelligent Recommendation for Tourism Destination Decision Making* project (http://dietorecs.itc.it [November, 26th 2001]) of the Electronic Commerce and Tourism Research Laboratory- Istituto Trentino di Cultura, Italy (eCTRL). Both systems seem to produce recommendation only at the destination level. The offer no hint on how they would use CBR to recommend specific travel products, though we believe the framework can be extended. TripMatcher differs from them substantially, as it deploys information filtering and text mining algorithms instead of case-based reasoning, and is able to produce recommendations down to the product level.
Collaborative filtering was recently proposed as a solution for travel destination bundling (Hwang, Yeong-Hyeon and Daniel R. Fesenmaier. 2001). *TravelPlan* (Camacho D., Borrajo D., Molina J.M., 2000) presents a multi-agent systems (MAS) and information brokering approach. The example given in the paper is a travel planner matches user preferences with travel options (air, land, sea) when the destination and other restrictions are known. Both are very interesting works that explore the application of agents and recommendation technologies to more complex multi-destination/multi-channel travel planning.
Besides TripleHop Technologies, other commercial ventures have developed destination based recommender systems, or alike. Particularly interesting are a) VacationCoach.com, which provides a rule-based engine for destination recommendations and b) 10best.com, a location-driven personalized travel guide provider. Neither of them seems to be using sophisticated filtering mechanisms nor does either have the capability of generating recommendations of travel products sold by their clients or other providers.

7 Conclusion and Future Work

Recommender systems for travel and hospitality are potentially powerful tools to increase conversion rates of travel e-commerce sites as well as for destination research and marketing (http://www.eyefortravel.com/index.asp?news=19888 [August, 23rd 2001].). Through this paper, we have emphasized the complexity of these systems. We have suggested a hybrid approach for information filtering and matching implemented into our system called TripMatcher. Through the analysis of quantitative and qualitative usage data, provided by Ski Marketing Corporation (http://www.SkiEurope.com [November, 26th 2001]), we could test the effectiveness of the system.

Anyone who wishes to build travel recommender systems must understand the underlying factors that affect the decision-making process. Once these factors are determined and modeled, destination and product knowledge bases have to be constructed via input from experts and mining of textual descriptions. This is a lengthy and costly process that few companies and institutions are willing to undertake. For this reason, companies such as TripleHop Technologies, base their business model on licensing software that already includes a manually-built reusable destination based knowledge base. It also provides filtering and text mining technologies that allow their systems to expand to the product level and become even more accurate with continuous use.

We are now looking towards developing recommender systems work on wireless devices and instant messaging platforms. These recommender systems will filter and deliver location-based information and recommendations of activities and events that can be downloaded into PDA's and smart phones, used then as travel companions.

References

Camacho D., Borrajo D., Molina J.M. (2000) TravelPlan: A Multiagent System to Solve Web Electronic Travel Problems. *Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000).* Workshop Agent-based Recommender Systems (WARS 2000). Barcelona,, Spain. June, 2000

Delgado, J. and Ishii, N. (1998), Content+Collaboration=Recommendation, *AAAI Workshop on Recommender Systems*, 1998. Published as Technical Report WS-08-98, AAAI Press.

Delgado, Joaquin (2000). Agent-based Recommender Systems and Information Filtering on the Internet; *PhD. Thesis. Nagoya Institute of Technology*, March 2000.

Dumais, S. T., Platt, J. Heckerman, D. and Sahami M. Inductive algorithms and representations for text categorization. *Proceedings of the Seventh International Conference in Information and Knowledge Management* (*CIKM'98*), 148-155.

Fesenmaier, D. R., and S.R. Lieber. (1988). Destination diversification as an indicator of activity compatibility: An exploratory Analysis. Leisure Studies, 10: 167-168.

Fodness, Dale, and Brian Murray. (1998). A typology of tourist information searchstrategies. *Journal of Travel Research*, 37: 108-119.

Hwang, Yeong-Hyeon and Daniel R. Fesenmaier. (2001). Collaborative filtering: Strategies for travel destination bundling. In Payline J. Sheldon, Karl W. Wöber, and Daniel R. Fesenmaier (eds.) *Information and Communication Technologies in Tourism 2001: the proceedings of the international conference of ENTER 2001*. Montreal, Canada, pp 167-175.

Joachims, T. A Statistical Learning Model of Text Classification with Support Vector Machines. *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR)*, ACM, 2001.

Maes, Pattie. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM*, Vol. 37, No.7,pp. 31-40, 146, ACM Press, July 1994.

Pazzani, M. (in press). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review.*

Resnick, Paul and Hal R.Varian (1997). *Recommender Systems.* Special section in Communications of the ACM, Vol. 40, No. 3; March 1997

Rita, Paulo. (2000). Web marketing tourism destinations. *Proceedings of the 8thEuropean Conference on Information System*, Vienna, Austria.

Salton, G. (1989) *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley 1989.

Shardanand, U. and Maes, P. (1995), Social Information Filtering: Algorithms for Automating 'Word of Mouth' , *Proceedings of the CHI-95 Conference,* Denver, CO, ACM Press, May 1995.

Ungar, L.H. and D.P. Foster (1998) Clustering Methods for Collaborative Filtering, *AAAI Workshop on Recommendation Systems*, 1998. Published as Technical Report WS-08-98, AAAI Press.

Vogt, C. A. and D.R. Fesenmaier. (1998). Expanding the functional information search model. *Annals of Tourism Research*, 25(3): 551-578.

Waszkiewicz, P., Cunningham, P., Byrne, C., (1999) Case-based User Profiling in a Personal Travel Assistant, in *Proceedings of UM'99, 7th International Conference on User Modeling*, Banff, Canada, June, 1999. Springer-Verlag.