Learning User Profiles for Content-Based Filtering in e-Commerce

Abbattista F., Degemmis M., Fanizzi N., Licchelli O.,
Lops P., Semeraro G., and Zambetta, F.
{fabio, degemmis, fanizzi, licchelli, lops, semeraro, zambetta}@di.uniba.it

Abstract. The recent evolution of e-commerce emphasized the need for more and more receptive services to the unique and individual requests of users. Personalization became an important business strategy in Business to Consumer commerce, where a user explicitly wants the e-commerce site to consider her own information such as preferences in order to improve access to relevant products.
In this work, we present a personalization component that uses supervised machine learning to induce a classifier able to discriminate between interesting and uninteresting items for the user. The prototype system exploits a content-based technique, which makes use of textual annotations usually describing the products offered by e-commerce sites. Experimental results demonstrate the effectiveness of the method and encourage the integration of the prototype in the personalization module developed in the COGITO project, which aims at improving consumer-supplier relationships in future e-commerce using advanced technologies.

1 Introduction

The COGITO project (IST 1999-13347) is based on *intelligent personalized agents* which represent virtual assistants or advisors (also visually) by modeling their ability to support customers. There are many possible applications for such virtual assistants. They could instruct customers in the use of a web site, point out new offers, help to sift through products, and other support. There have already been some efforts made in developing chat robots ("chatterbots") based on expert systems [7].

A *chatterbot* is a software system capable of engaging in conversation (in written form) with a user, often entertaining the user with some "smalltalk" – sometimes accompanied by cartoons expressing emotions. In most applications, chatterbots are used as *guides* who can show the user around on a web site. This can be a stereotyped "guided tour" allowing only few deviations; however, this concept has to be abandoned when the web site is too large to be explored by navigation, or contains too many offers. This is the case in e-commerce applications, where pages are generated on demand by retrieving data from a product database and assembling the result into HTML pages, usually hit lists of searches.

Virtual assistants must be capable of flexible behavior if they are to be acceptable to users on a long-term basis [2, 4, 8]. This means that, in addition to some of the abilities already available (e.g., help question answering controlled by simple event-action rules), a further reaching *dialogue management* will be needed to help accomplish two major goals.

Whereas an increase in general dialogue intelligence can be achieved by elaborate rule sets, the naturalness of the dialogue depends on the degree in which the system is able of adaptivity to individual users, whether it is able to learn about their preferences and

attitudes during the dialogue, and memorize them for later use. For this purpose, we have included learning mechanisms that extract permanent features of a given user from the dialogue (of course, the user must consent to this, and will be given an opportunity to inspect and change the data). The resulting *user profiles* will be further analyzed to automatically extract usage patterns from the data given about user communities. This helps content providers to tailor their offers to the customers' needs, and can be used to generate assumptions about new users, when they start to converse with the system. Published research to date [5, 23] shows that a further development of personalized interfaces into more flexible dialogue-oriented interfaces could increase the acceptance of such personalized agents.

2 Personalization in e-commerce

In the COGITO project, user personalization is mainly performed by the Profile Extractor module, which is responsible for the user profile generation.

By user profile we mean all the information about a user, extracted from the information collected when he logs to a web site, in order to take into account her needs, wishes, and interests. Roughly, a user profile is a structured representation of the user's needs through which a retrieval system should act upon one or more goals based on that profile in order to autonomously pursue the goals posed by the user. It is quite obvious that a user profile modeling process requires two steps (which constitute the user profile modeling methodology). It has to be decided:

- *what* has to be represented and
- *how* this information is effectively represented.

Generally, the information stored in a user profile can be conceptually categorized in seven classes, according to the source it has been collected from: *Registration Data, Question&Answer, Legacy Data, Past History, 3rd Party, Current Activity, Open Adapter*.

A user profile is given as a list of attribute-value pairs, in which each attribute is assigned with the proper value on the ground of the specific user it refers to. Each attribute-value pair represents a feature of that user. The list of attributes must be finite as well as the possible values related to each attribute. Examples of attributes in that list are: LAST NAME, FIRST NAME, AGE, ADDRESS, JOB, ANNUAL INCOME, PREFERENCES, etc. The attribute list is the same for all the users.

These attributes or features can be divided into three categories:

- *Explicit*, whose values are given by the user herself (*Registration Data* or *Q & A*).
- *Existing, i.e. that* can be collected from existing applications, such as register systems (e.g., ADDRESS, JOB).
- *Implicit*, elicited from the behavior of the user, through the history of her navigation or just from the current one.

For our purposes, a computational customer profile will be useful to describe univocally a user that accesses to the web site.

Considering the previous kind of features, the most common approach to build a customer profile mixes three different techniques [18]. In the first one, the buyers have to fill an initial form that asks for typical information (such as the customer's gender and year of birth), and some specific information (such as product categories of interest among the list of categories available in the store). Since only a limited amount of information can be acquired in this way (customers might not be able or willing neither to fill large forms nor to provide personal details and preferences), the approach usually followed is to present the customer with a limited number of fields and to let her decide which fields she is willing to fill.

The second one exploits demographic profiles (available on the market) that give detailed and readily available information on the different categories of buyers, and can also be used to make predictions about consumer's interests, preferences and behavior.

The third technique dynamically updates the user model by considering data (e.g. purchases made, number of visits, etc.) recorded on past visits to the store.

These three techniques complement each other, allowing one to obtain a more complete customer model. Moreover, the integration of these three techniques leads to a less intrusive system: users are not required to provide information about preferences, tastes, etc. but they actively participate in the definition of their profiles.

In the COGITO project, the profiling module has been implemented through machine learning techniques that enable the generation of user profiles starting from data collected in log files of the past user interactions with the BOL web site, an on-line media shop specialized in books.

2.1 Profile Extractor Module

The Profile Extractor (Figure 1) is the module that allows for the classification of users accessing the COGITO project through machine learning techniques.

During a session, user dialogues with the web agent are stored in log files. The Dialog Analyzer module receives the log files of past sessions and processes them in order to produce a Structured Dialogue History, representing user interests and preferences.

The goal of the Profile Extractor is to identify, from data stored in the Structured Dialogue History, the main features that are necessary to produce a user profile. The Profile Extractor module is further made up of four sub-modules:

- The *XML I/O Wrapper*, whose aim is to extract from the Structured Dialogue History the most relevant parts of the dialogue and to transform them into a set of examples capable of being processed from the other sub-modules.
- The *Rules Manager*, implemented through one of the WEKA [25] classifiers. During a learning session, each example of the dialogue history, representing a single user feature vector, must be pre-classified by a human expert. The WEKA package processes training examples and induces rules for extracting user features from further unclassified examples, to be used by the *Profile Manager* module.
- The *Community Manager*, implemented through a clustering algorithm (unsupervised learning) available in WEKA. This sub-module groups usage sessions in order to infer some usage patterns that can be exploited for understanding trends in the system exploitation for further market studies and to group single users to form user communities, sharing the same interests and preferences [17].
- The *Profile Manager*, that performs the profiling task, according to the set of rules induced by the *Rules Manager* and the user history. Once a user accesses the system, her history is retrieved in the Structured Dialogue History repository and her characteristic features are singled out, according to the rules that fired. Hence, the rest of the dialogue can benefit of knowing standard information about her interests, her community, etc.

As mentioned above, dialogue files of user sessions are decomposed to extract facts about information needs, attitudes towards items (e.g. desires), known items, etc. This step produces a *model* of the user representing her interests and background in the dialogues.

Supervised machine learning techniques are used for analyzing a number of dialogues of an individual user. The aim is to induce a set of rules, expressed in the same representation language. Such rules can be regarded as the core of an *extractor*, which is able to generate user models from new unclassified incoming structured user logs.

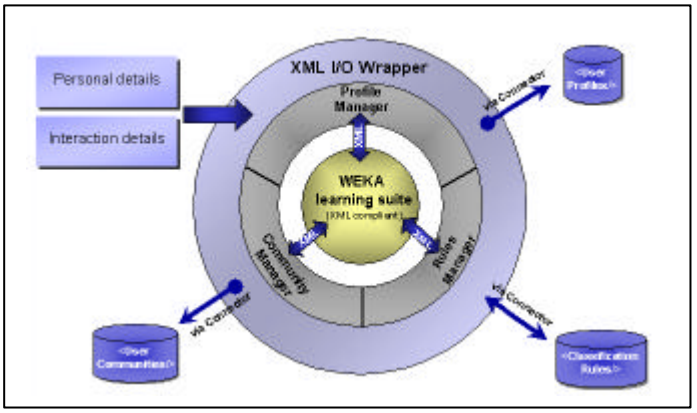

Figure 1: The Architecture of the Profile Extractor.

The Profile Extractor can be seen as an intelligent component capable of automatically assigning a customer to a specific class in order to improve the system usability. This component should help users to accomplish their goals easier (through explanations and suitable interaction modalities performed by the agent). As a consequence, one of the main problems concerns the definition of meaningful classes and the identification of the features that properly describe each of them and characterize the corresponding kind of interaction. In the system, the classes we considered are the book categories.

Thus the main function provided is to automatically assign each buyer to these predefined classes on the ground of information drawn from real sessions (*interaction modeling*). By examining the dialogue histories, it is possible to extract some characteristics that are useful for recognizing the buyer. We observed that most of the characteristics that were identified turned out to be application dependent, while only few of them seem to be system dependent.

For instance, relevant characteristics are those concerning the way users exploit the capabilities of the search engine of the web site, such as date and time of session beginning, number and frequencies of searches performed on a certain category, number and frequencies of purchases performed on a certain category, etc. This information constitutes examples exploited to train the learning system in order to induce a set of rules [15].

After the training phase, the interaction of any user that accesses the web site through a client will generate/update a dialogue history file. This file will be exploited to provide a new example that the Profile Extractor will classify on the ground of the rules inferred. In this way it is possible to create a personal profile of each customer, which contains information about her interests, tastes, preferences. The system is capable of tracking user behavior evolution, so customer profiles may change across multiple interactions.

2.2 Item Recommender

The profiles inferred by the COGITO system simply contains the book categories preferred by a user. No more details about her preferences in each category are included. Our intention was to enhance the profiles in order to achieve more precise book recommendations. Thus, we decide to adopt content-based book recommending by applying automated text categorization methods to semi-structured text [14]. Our current prototype, called Item Recommender (ITR), uses information extraction techniques to obtain book information from the web pages of the BOL web site and store them in a local database. Then, books selected from several categories are rated by different users in order to provide the system with training examples. ITR uses a Bayesian learning algorithm [12] to induce a single probabilistic model of a book category.

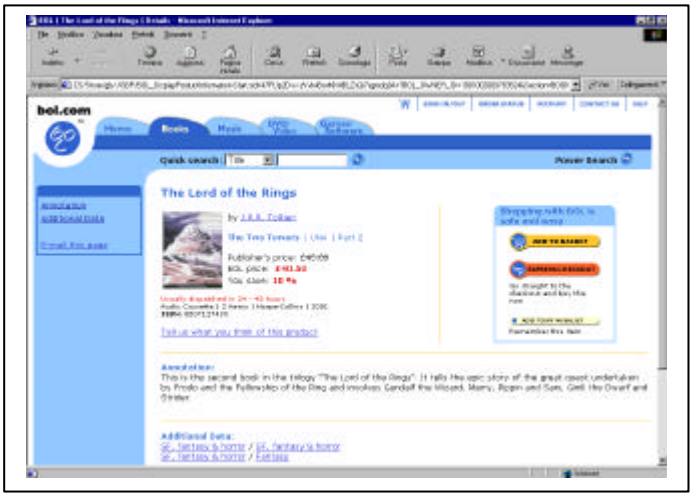

Figure 2: A web page at uk.bol.com

The system learns a classifier able to discriminate between interesting and uninteresting items for each book category preferred by a user.

The first step performed by the system is to build the dataset of training instances by a simple Item Extractor module that downloads web pages containing book description, obtained by submitting some queries to the BOL search engine (Figure 2).

Each instance is described by a set of *slots*. Each slot is a textual field corresponding to a specific feature of a book. The current slots utilized by ITR are: *title*, *authors* and *textual annotation*. The module uses a simple pattern-matcher to analyze the document and to extract a set of strings, the *tokens* to fill each slot. The system also eliminates stopwords and applies stemming. The text in each slot is processed using a *bag of words (BOW) model*: the text is seen as a collection of words, taking into account their occurrences in the original text. Thus, each example is represented as a vector of BOW (one for each slot). The complete extraction process is depicted in Figure 3.

Users select and rate a set of training books according to their preferences, providing a discrete rating (from 1 to 10) for each selected title; the rating from 1 to 5 is interpreted as negative and the rating from 6 to 10 as positive.

The goal of the classification algorithm is to predict the probability that a book would be rated as positive rather than negative: the system performs a probabilistic binary categorization task. A document is represented as an ordered sequence of word events belonging to the same vocabulary V.

The posterior probability of a class, given a document d_i is calculated as follows:

$$P(c_j \mid d_i) = \frac{P(c_j)}{P(d_i)} P(d_i \mid c_j)$$

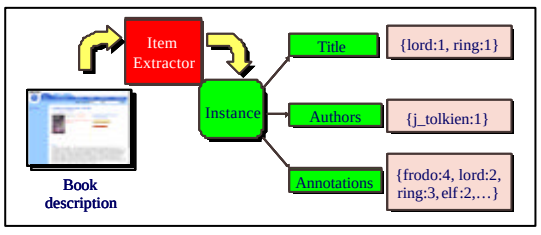

Figure 3: The BOW extraction process

In our problem, we have only 2 classes: c_1 represents the positive class and c_0 the negative one. Since books are represented as a vector of documents, one for each BOW, the posterior category probabilities for a book d_i are computed using:

$$P(c_j \mid d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|d_{im}|} P(a_{imk} \mid c_j, s_m)$$

where $S = \{s_1, s_2, \ldots, s_{|S|}\}$ is the set of slots, d_{im} is the document in the slot s_m of the instance d_i, and a_{imk} is the k^{th} word in the slot d_m.

If b_{im} is the BOW in the slot s_m of the document d_i, and n_{imk} is the number of occurrences of the token t_k in b_{im}, we obtain:

$$P(c_j \mid d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|V|} P(t_k \mid c_j, s_m)^{n_{imk}}$$

The prior probabilities of the classes and the conditional probabilities of the terms are estimated from the training set *TR*, where instances are weighted according to user ratings. If r is the user rating for a book, the two weights, one for each class, are computed as follows:

$$a_{i1} = \frac{r-1}{9}; \qquad a_{i0} = 1 - a_{i1}$$

Thus, the model parameters are estimated as:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} a_{ij}}{|TR|}$$

$$\hat{P}(t_k \mid c_j, s_m) = \frac{O(t_k, c_j, s_m)}{L(c_j, s_m)}$$

$$O(t_k, c_j, s_m) = \sum_{i=1}^{|TR|} a_{ij} n_{imk}; \qquad L(c_j, s_m) = \sum_{i=1}^{|TR|} a_{ij} \mid b_{im} \mid$$

where n_{imk} is the number of occurrences of the term t_k in the slot s_m of the i^{th} example and the $L(c_j,s_m)$ denotes the total weighted length of the slot s_m in the class c_j.

This approach allows the refinement of the profiles by including words most indicative of user preferences for each preferred book category the system was trained on. An example of profile obtained by rating books about "Computer and Internet" is given in Figure 4. The features are ranked according to a *strength* measuring the discriminatory power of a word in classifying a book.

3 Experimental results

This section presents the experimental evaluation conducted in order to test the ITR prototype. We performed two different experiments: The first consisted in observing the accuracy of the predictions made by the system. After the training phase, a number of metrics were used to measure its performance on the test data.

Figure 4: An example of ITR user profile

The goal of the second experiment was to evaluate the combining of the COGITO profiles with the ITR ones to form a more specific user profile, represented by preferred categories integrated with their specific keywords discovered by ITR. We are persuaded that an improvement of the recommendations can be achieved by means of this approach. Thus, in this experiment, a comparison between two different kinds of user profiles is performed.

For both the experiments, 5 book categories at *uk.bol.com* were selected. For each one of the 5 categories, the system has been trained by a specific user interested in that category and having a COGITO profile. After the extraction phase from the site, a local database of book descriptions was built. The collected data are summarized in Table 1.

Table 1: Data Information

Category	*Book descriptions*	*Books with annotation*	*Avg. annotation length*	*User Id*
Computer & internet	5414	4190 (77%)	42,39	User1
Fiction & literature	6099	3378 (55%)	35,54	User2
Travel	3179	1541 (48%)	28,29	User3
Business	5527	3668 (66%)	42,04	User4
SF, horror & fantasy	667	484 (72%)	22,33	User5
Total	20886	13261		

Next, each user selected a set of books by searching for particular authors, titles or performing a simple keyword search. Approximately 90 books are rated by each user, according to her interests in the training category. Data distribution is shown in Figure 5.

At the end of the rating procedure, a dataset of roughly 450 classified instances is obtained. It was analyzed by means of a 10-fold cross-validation, that is, the dataset is divided randomly into 10 blocks of near-equal size and distribution of class values. Then, each part is held out in turn and the learning scheme is trained on the remaining nine-tenths and tested on the hold-out block. Several metrics were used in the testing phase:

Recall (Re): is defined as the fraction of positive examples classified as positive;

Precision (Pr): is defined as the fraction of examples classified as positive that are actually positive.

F-measure (F): is defined as a weighted average of *Pr* and *Re*;

Normalized Distance-based Performance Measure (NDPM): is the distance between the ranking imposed by the user ratings and the ranking predicted by the system. Values range from 0 (agreement) to 1 (disagreement) [26].

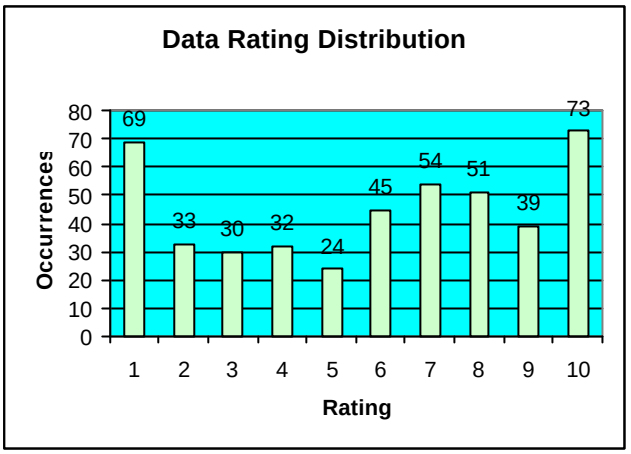

Figure 5: Data Rating Distribution

Rank Correlation (Rs): Spearman's rank correlation is a statistic measure used to establish whether there is any correlation between two sets of data. Its value falls between -1 and 1. A correlation coefficient of 0,3 to 0,6 is considered as moderate and above 0,6 is considered strong.

Error (E): is calculated as an average of the absolute difference between the user ratings and those predicted by the system.

All results of the experiment are reported in Table 2.

Table 2: 10-fold cross validation results

Category	Pr	Re	F	NDPM	Rs	E
Computer & internet	0,8500	0,5476	0,6660	0,3241	0,5499	0,3498
Fiction & literature	0,5971	0,7033	0,6459	0,4458	0,0676	0,3489
Travel	0,8100	0,8900	0,8481	0,3322	0,4683	0,2885
Business	0,7364	0,6800	0,7070	0,3741	0,3466	0,3576
SF, horror & fantasy	0,4695	0,7833	0,5871	0,3583	0,3970	0,4105
Avg.	0,6926	0,7209	0,6909	0,3670	0,3659	0,3611

Values of *Pr*, *Re* and *F* provide evidence that the system produces accurate recommendations. NDPM is fairly consistent, while looking at *Rs* we observe that there is at least a moderate correlation for each category.

The second experiment consisted in asking each user for submitting 3 different queries to ITR. Then, a feedback is given to the system by rating the 20 top ranked books in the result set. The experiment has been modelled on the basis of two different scenarios.

In the first scenario, books are ranked according to the COGITO profile, whereas in the second scenario the ranking is performed using the COGITO profile integrated with the ITR one. For both scenarios feedback evaluation results are given in Table 3.

For pairwise comparison of methods, the non-parametric Wilcoxon signed rank test is used [16], since the number of independent trials is relatively low and does not justify the application of a parametric test, such as the t-test. The Wilcoxon signed rank test reduces measurement data to ordinal data by replacing the differences between measurements with ranks. The statistic W is obtained by adding together the ranks and is used to determine the winner. Under the null hypothesis that the two populations have the same distribution, we would expect the ranks of the plus and minus differences to be evenly distributed. If the hypothesis is false, we would expect W to be large (either positively or negatively). In this experiment, the test is adopted in order to evaluate the difference between the effectiveness

of the different profiles by means of the metrics pointed out in Table 3, requiring a significance level $p < 0.05$.

Table 3: Results of the comparison between the profiles

User	Query	Pr		NDPM		Rs	
		Sc. 1	Sc. 2	Sc. 1	Sc. 2	Sc. 1	Sc. 2
1	Java	0,50	0,90	0,594	0,423	-0,288	0,300
1	Graphics	0,30	0,70	0,465	0,328	0,156	0,490
1	Security	0,80	0,75	0,636	0,410	-0,412	0,278
2	Realism	0,35	0,50	0,421	0,400	0,258	0,329
2	romanticism	0,60	0,55	0,505	0,636	-0,053	-0,362
2	Science fiction	0,65	0,55	0,468	0,476	0,042	0,109
3	Islands	0,65	0,90	0,600	0,536	-0,288	-0,136
3	Guides	0,40	0,60	0,539	0,694	-0,130	-0,581
3	restaurants	0,30	0,35	0,505	0,415	0,037	0,338
4	Business manager	0,35	0,60	0,513	0,494	-0,074	0,018
4	enterprise solution	0,20	0,30	0,365	0,292	0,405	0,595
4	investment	0,50	0,70	0,547	0,605	-0,118	-0,312
5	s_king	0,30	0,60	0,589	0,197	-0,261	0,806
5	Space	0,10	0,40	0,447	0,184	0,178	0,839
5	King	0,70	1,00	0,550	0,326	-0,154	0,517
	Avg.	0,45	0,63	0,516	0,428	-0,047	0,215
	W=	103		-74		72	

On the basis of the values of the W statistic calculated above, we can deduce that there is a consistent statistically-significant difference in performance among the two different profiles.

4 Related work

Various learning approaches have been applied to discover user preferences (and to construct user profiles) to make personal recommendations. We have already introduced in Section 2.2 the text categorization method adopted by Mooney and Roy [14] in their LIBRA system, that makes content-based book recommendations exploiting the product descriptions found in Amazon.com, using a naïve Bayes text classifier, as in our system. A similar approach is adopted by the system named Syskill & Webert [18], which tracks the users browsing to formulate user profiles. The system identifies informative words from Web pages to use as boolean features and learns a naïve Bayesian classifier to distinguish interesting Web pages on a particular topic from uninteresting ones. A different profile is learnt for each topic. In [3] a statistical approach based on traditional term frequency and inverse document frequency is used to recommend Web pages.

A reinforcement learning method is applied by Personal WebWatcher [9], a content-based system that recommends web-page hyperlinks by comparing them with a history of previous pages visited by the user. The system generates a profile made up of bag of words for each page visited during previous browsing sessions. Hyperlinks on new pages can then be compared to this profile and ranked accordingly. A more recent system, News Dude [6], learns about user interests in daily news stories using a multi-strategy learning approach to induce user models that represent short-term and long-term interests separately.

The new generation of Web personalization tools is attempting to incorporate techniques for pattern discovery in Web usage data. Web usage systems run a number of data mining algorithms on usage or clickstream data gathered from Web sites in order to

discover user profiles. For example, the WebPersonalizer system described in [13] provides a list of recommended hypertext links to a user while browsing a Web site. Profiles are derived from Web server logs and are represented as weighted collections of URIs. The discovery of navigation patterns in Web logs has also been studied in [22] with the aim of inspecting the behavior of users in a web site. The patterns are interpreted differently on the basis of the user role (customer and non-customer) and are exploited to perform a comparative analysis of the navigation behavior in order to improve the site efficiency in turning non-customers into customers.Data mining methods are also used by the 1:1Pro (One-to-One Profiling) system [1], that builds personal profiles based on customer transactional data that are analyzed for discovering a set of rules capturing individual customer behavior.

On the other hand, an evolutionary approach is adopted by Lee et al. [10] in designing a learning agent able to model customer interests for DVD film recommendations. The system maintains a profile for each customer obtained by monitoring her activities during the navigation of a movie site and by recording the contents she has read. The features used to describe a product are the keywords associated with a film, available from the on-line database. The collected data are used by the evolutionary mechanism to learn a model of prediction for a customer that is stored in her profile and used for further recommendations.

One complex common problem for a recommender system is the *cold-start* problem, where recommendations are required for new items or users for whom little or no information has yet been acquired. In fact, to be able to make accurate predictions, the system must first learn the user preferences from the ratings that she makes. If the system does not show quick progress, a user may lose patience and stop using the system. Schein et al. [21] propose a probabilistic model that combines content and collaborative information by using expectation maximization learning to fit the model to the data. Another recent approach [11] exploits ontologies to investigate how domain knowledge can help in the acquisition of user preferences. Ontologies are used to complement the behavioral information held within recommender systems, by providing some initial knowledge about users and their domains of interest. In [19] different techniques are analyzed to select the sequence of items that each new user has to rate. These techniques include, for example, the use of information theory to decide on the items that will give the most value to the system. Others authors have integrated agents into a collaborative filtering environment to extract user preference information transparently [24]. This method has the advantage of collecting implicit information in addition to explicitly provided ratings, and seems a very promising approach.

A paper by Schafer et al. [20] presents a detailed taxonomy and examples of recommender systems used in e-commerce applications and how they can provide one-to-one personalization and at the same time can capture customer loyalty.

5 Conclusions

In this paper we evaluated a simple approach, based on the naive Bayes machine learning method, to build user profiles for a content-based book recommender system. We presented a prototype system, called Item Recommender, able to refine user profiles by adding a list of words to each book category preferred by a specific user. Our goal was to integrate the prototype in an already existing personalization system, the Profile Extractor, which employs machine learning techniques to infer the book categories preferred by a user, and stores them in a user profile.

Experiments on book collections belonging to different categories confirm that the integrated profile improves the quality of the recommendations suggested by the system.

Thus, we can conclude that the use of the integrated profiles has a significant positive effect, encouraging us to incorporate the ITR prototype in the Profile Extractor module to obtain more specific user profiles.

In the future, we plan to tackle the cold-start problem according to an approach based on conversational agents, as experimented in the COGITO project. At present, the system is able to provide personalized predictions only if a profile of the user is available. A conversational agent could minimize the new user effort requested to a new user by getting her to the fun part, while still learning information useful to make good recommendations. We intend to apply information extraction techniques to discover information about a new user from the dialogue she had with the agent. Moreover, we are evaluating the possibility of using ontologies in capturing knowledge of user preferences, in order to get profiles that refer explicitly to concepts of a standard ontology, and not just a list of words.

Acnowledgments

The authors would like to thank Daniele Capursi for his enthusiasm and dedication in designing and developing the ITR system.

References

[1] Adomavicius G. and Tuzhilin A. Using Data Mining Methods to Build Customer Profiles, IEEE Computer, vol. 34 num. 2 (2001) 74-82.

[2] Andrè E., Rist T., and Muller J., Integrating Reactive and Scripted Behaviors in a Life-Like Presentation Agent, Proceedings of the II International Conference on Autonomous Agents, Minneapolis (1998) 261-268.

[3] Balabanovic M. and Shoham Y., Fab: Content-Based Collaborative Recommendation, Communication of the ACM, vol. 40 num. 3 (1997) 66-72.

[4] Ball G., Ling D., Kurlander D., Miller J., Pugh D., Skelly T., Stankosky A., Thiel D., Dantzich M. van, and Wax T., Lifelike Computer Characters: the Persona Project at Microsoft Research, in: Bradshaw J.M. (ed.), Software Agents, AAAI/MIT Press, (1997) 191-222.

[5] Belkin N. J., Cool C., Stein A., and Thiel U., Cases, Scripts, and Information Seeking Strategies: On the Design of Interactive Information Retrieval Systems, Expert Systems and Applications, vol. 9, num. 3 (1995) 379–395.

[6] Billsus D., Pazzani M.J., A Hybrid User Model for News Story Classification, Proceedings of the VII International Conference on User Modeling, Banff, Canada (1999) 99-108.

[7] Cassell J. C., Pelachaud C., Badler N. I., Steedman M., Achorn B., Becket T., Dourville B., Prevost S., and Stone M., Animated Conversation: Rule-based Generation of Facial Expression, Gesture and Spoken Intonation for Multiple Conversational Agents, Proceedings of Siggraph'94, Orlando, USA, (1994) 413-420.

[8] Hayes-Roth B., Johnson V., Gent R. van and Wescourt K., Staffing the Web with Interactive Characters, Communications of the ACM, vol. 43 num. 3 (1999) 103-105.

[9] Joachims T., Freitag D., and Mitchell T. Webwatcher: A Tour Guide for the World Wide Web, Proceedings of the XV International Joint Conference on Artificial Intelligence, Nagoya, Japan (1997) 770-775.

[10] Lee W.-P., Liu C.-H., Lu C.-C., Intelligent Agent-based Systems for Personalized Recommendations in Internet Commerce, Expert Systems with Applications, vol. 22 num. 4 (2002) 275-284.

[11] Middleton S., Alani H., Shadboldt N. R., and De Roure D. C. (2002) Exploiting Synergy between Ontologies and Recommender Systems, Proceedings of the Semantic Web Workshop, Hawaii, USA, (2002).

[12] Mitchell T., Machine Learning, McGraw-Hill, New York (1997).

[13] Mobasher B., Cooley R., and Srivastava J., Automatic Personalization Based on Web Usage Mining. Communications of the ACM, vol. 43 num. 8 (2000) 142-151.

[14] Mooney R. J. and Roy L., Content-Based Book Recommending Using Learning for Text Categorization, Proceedings of the V ACM Conference on Digital Libraries, San Antonio, USA, (2000) 195-204.

[15] Moustakis V. S. and Herrmann J., Where Do Machine Learning and Human-Computer Interaction Meet?, Applied Artificial Intelligence, vol. 11 num. 7-8 (1997) 595-609.

[16] Orkin M., Drogin R., Vital Statistics, McGraw Hill, New York (1990).

[17] Paliouras G., Papatheodorou C., Karakaletsis V., Spyropoulos C., and Malaveta V., Learning User Communities for Improving the Service of Information Providers, Lecture Notes in Computer Science, n. 1513, Springer-Verlag, Berlin (1998) 367-384.

[18] Pazzani M. and Billsus D., Learning and Revising User Profiles: The Identification of Interesting Web Sites, Machine Learning, vol. 27, num. 3 (1997) 313-331.

[19] Rashid A. M., Albert I, Cosley D., Lam S. K., McNee S., Kostan J. A., and Riedl J., Getting to Know You: Learning New User Preferences in Recommender Systems, Proceedings of the 2002 International Conference of Intelligent User Interfaces, San Francisco, USA, (2002).

[20] Schafer J. B., Konstan J., and Riedl J., Electronic Commerce Recommender Applications, Journal of Data Mining and Knowledge Discovery, vol. 5 num. 1-2 (2001) 115-152.

[21] Schein A. I., Popescul A., Ungar L. H., Methods and Metrics for Cold-Start Recommendations, Proceedings of the XXV Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland (2002).

[22] Spiliopoulou M., Pohle C., and Faulstich L.C., Improving the Effectiveness of a Web Site with Web Usage Mining, Proceedings of the Workshop on Web Usage Analysis and User Profiling, WEBKDD '99, San Diego, USA, (1999) 51-56.

[23] Stein A., Gulla J. A., and Thiel U., User-Tailored Planning of Mixed Initiative Information-Seeking Dialogues, User Modeling and User-Adapted Interaction, vol. 8 num. 1-2 (1999) 133-166.

[24] Wasfi A. M. A., Collecting User Access Patterns for Building User Profiles and Collaborative Filtering, Proceedings of the 1999 International Conference on Intelligent User Interfaces, Los Angeles, USA, (1999) 57-64.

[25] Witten I. H. and Frank E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers, San Francisco, USA (1999).

[26] Yao Y. Y., Measuring Retrieval Effectiveness Based on User Preference of Documents, Journal of the American Society for Information Science, vol.46 num. 2 (1995) 133-145.