Generating User Profiles for Information Agents

Daniela Lis Godoy

ISISTAN Research Institute, UNICEN University
Campus Universitario (CP 7000), Tandil, Bs. As., Argentina
TEL/FAX +54 (2293) 440363 - {dgodoy}@exa.unicen.edu.ar
Also CONICET

Abstract

The advent of the World Wide Web and its constant growing have transformed the search for information into a time-consuming task. Intelligent information agents have emerged as a solution to this problem. These agents learn users' interests and model them into user profiles in order to assist users by discovering, retrieving and summarizing information on behalf of them. This work is focused on the construction of user profiles for information agents starting from observation of users' readings and behavior in the Web. Existing approaches have attacked partially this problem, treating the user profiling task either as a classification problem from the machine learning point of view or as a pure keywords analysis problem from the information retrieval point of view. However, user profiling embrace a number of additional aspects that are not currently addressed in these approaches, such as modeling topic of interest with different levels of abstraction or modeling of contextual information about topics. In this work we propose a user profiling technique to be used in the development of intelligent information agents that deal with these aspects.

1 Introduction

Intelligent agents assisting users with different tasks on the Internet have proved to be a valuable means to cope with the overloading of information produced by the proliferation of on-line sources. Personal assistants recommending Web pages, filtering news articles from newsgroups or processing email messages on behalf of users are examples of this kind of agents [6].

To provide effective assistance with these tasks, agents need to capture users' interests in an unobtrusive way by extracting them automatically through monitoring users' behavior (e.g. when they read a web page, move a mail message between folders, etc.). The modeling of these interests enables the agent to predict users' information preferences on advance and supply personalized advice.

There is a number of possibilities for representing users' interests related to information management tasks. The degree of detail of the resultant model, that conforms a user profile, determines the level of assistance that could be reached by an agent. As an example, the simplest approach that uses a set of keywords as the representation of one particular interest is not sufficient as it is not rich enough to capture users' interests with the expected precision.

In this work we propose a technique for user profiling to be applied in the development of intelligent information agents. The main objective of this technique is to guide developers in the construction of agents assisting users dealing with texts in the Web. Profiles resulting from the application of this technique could be use for a wide range of tasks, such as generate personal newspaper, help users to browse the Web, searching the Web to pro-actively to recommend relevant material, etc.

In order to validate our user profiling technique we developed an search agent based on it. This agent assists users to find interesting documents in the Web. It carries out a parallel search in the most popular Web search engines and filters their result, listing to the user a reduced number of documents with high probability of being relevant to him. This agent will allows us to evaluate the effectiveness of the user profiling technique and to compare it with current approaches for the same task.

This article is organized as follows. Section 2 presents the technique to model users' interests. The Web pages representation model adopted in this work is described in Section 2.1. How documents are analyzed to detect topics of interest and the way they are organized into a hierarchy are explained in Sections 2.2 and 2.3 respectively. An intelligent agent that assists Web search based on this technique is presented in Section 3. Finally, conclusions are discussed in Section 4.

2 Modeling Users' Interests

We propose in this work a technique that allows agents to capture users' topics of interest and detect hierarchical relationship underlying these topics which is based on Textual Case-Based Reasoning (TCBR), an specialization of Case-Based Reasoning (CBR) for document management. CBR is a problem solving paradigm that reuses solutions of previous experiences, which are named cases [4]. The term TCBR was subsequently coined for situations where these experiences are given in textual documents [5].

A case-based reasoner remembers problem-solving situations as cases. Then, it retrieves relevant cases (the ones matching the current problem) and adapts their solutions to solve new situations. These solutions could be complex, like the description of a treatment for a given disease, or simple like the category in which a concept or problem fits into. In the last perspective, CBR is applied to accomplish a classification task, i.e. find the correct class for an unclassified case. The class of the most similar past case becomes the solution to the classification problem.

For our goal, TCBR is used to dynamically classify new documents according to their topic inside those interesting for a user. Since each user could potentially have different topics of interest, they need to be obtained on the fly by the case-based reasoner. Our assumption is that topics can be obtained by similarity and frequency analysis of user readings. To accomplish both kinds of analysis, the readings of a particular user are represented as textual cases in the context of TCBR. Using this approach previous read documents can help to categorize new ones into specific categories or topics of interest assuming that similar documents share the same topic. In the next sections we explain how textual cases are obtained from Web pages and how they are grouped by similarity to obtain users' topics of interest.

2.1 Web pages Representation as Textual Cases

In the context of CBR a case contains the specific knowledge that describes a particular situation [4]. The main parts of a case are the description of a problem that has been solved, the description of its solution itself and the feedback got from the user for that solution. Words in the content of a Web page permit to describe a particular situation in our topic classification problem. The solutions to this kind of textual cases are specific topic definitions. In this way when a new document appears with similar distribution of words into its content compared to another document already seen, the agent can deduce that both documents are about the same topic.

To reflect the importance of each word in a document representation a weight is associated with each of them in the case as the result of a function $weigth(w_j, d_i) = tf_{ij} + \Delta_{ij}$, where tf_{ij} is the frequency of a word w_j in the document d_i and Δ_{ij} an additive factor defined in terms of several word characteristics in the document. The value of Δ_{ij} is calculated taking into account the word location inside the document HTML structure (e.g. words in the title are more important than words in the document body) and the word style (e.g. bold, italic. etc.).

Previous to the document representation as cases, non-informative words such as prepositions, conjunctions, pronouns, very common verbs, etc. are removed using a standard stop-word list. Words present in this list are excluded from the document representation since they are assumed topic independent words that appear with similar frequency in the majority of documents. After stop-words removal a stemming algorithm is applied to the remaining words. This is a process of linguistic normalization in which the variant forms of a word are reduced to a common one [7].

The solution for our textual cases is the topic which they belong among the topics of interest for the current user. Topics predictions are made starting from document contents, pages URL (i.e. pages belonging to the same site are probably about a same topic at a general level) and contextual information stored within cases.

2.2 Identifying Topics of Interest for a User

A topic of interest within a user profile is extensionally defined by the group of cases that have the same value as case solution. As mentioned in previous section the solution of a textual case is assigned according to its similarity with other cases already classified.

This comparison of cases is performed through a number of dimensions that describe them (i.e. content, URL and contextual information). A similarity function *sim* is defined for each one of these dimensions, being the most important the one that measures the similarity between relevant word lists. This similarity is calculated by the inner product with cosine normalization [8]:

$$sim(v_i^E, v_j^R) = \cos(\alpha) = \frac{\sum_{k=1}^{n} w_{ik} \cdot w_{jk}}{\sqrt{\sum_{k=1}^{r} w_{ik}^2} \cdot \sqrt{\sum_{k=1}^{r} w_{jk}^2}} \tag{1}$$

where v_i and v_j are the respective lists of words or vectors, and w_{ik} and w_{jk} the weights of the word k in each vector. This similarity function measures the cosine of the angle α between the vectors representing both documents.

A numerical evaluation function that combines the matching of each dimension with the importance value assigned to that dimension, is used to obtain the global similarity between the entry case (C^E) and the retrieved one (C^R). The function used in our technique is the following:

$$S(C^E, C^R) = \frac{\sum_{i=1}^{n} \left(w_i \cdot sim_i(f_i^E, f_i^R) \right)}{\sum_{i=1}^{n} w_i} \tag{2}$$

where w_i is the importance of the dimension i, sim_i a similarity function for this dimension, and f_i^E, f_i^C are the values for the feature f_i in both cases. If the similarity value obtained from S is higher than a given threshold, the cases are considered similar and then we can conclude that both cases are about the same user topic of interest.

2.3 Hierarchy of Topics

A personalized hierarchy of increasing specificity is used to organize the topics a user is interested in. This topic hierarchy could be seen like a tree. Each internal node in the tree holds features shared by their child nodes and the cases below it in the way of a classifier to the topic. Items without those features live in or below their sibling nodes and leaf nodes hold cases themselves.

The topic hierarchy needs to be built by an agent automatically starting from the scratch. To do this, as soon as new cases appear describing user's interests (e.g. the agent detects interesting reading for the user), they are grouped by similarity into the user profile. Each of these groups represents a very specific topic of interest for that user.

Then, a general inductive process automatically builds a classifier for this topic or category c_i by observing the characteristics of a set of cases that have been classified under c_i. A novel document should also have this characteristics in order to belong to c_i. A classifier for a category is composed of a function $F_i : d_j \rightarrow [0,1]$ that, given a document d_j, returns a number between 0 and 1 that represents the evidence for the fact that d_j should be classified under c_i. This function also has a threshold τ_i such that $F_i(d_j) \geq \tau_i$ is interpreted as a decision to classifying d_j under c_i, while $F_i(d_j) < \tau_i$ is interpreted as a decision of not classify d_j under c_i.

Once a classifier is built representing a generic topic in the hierarchy, new cases belonging to this topic (those with $F_i(d_j) \geq \tau_i$) are placed below it and new groups will be created. From these groups new classifiers will be obtained and added like child nodes of the first classifier, defining a hierarchy of them. Cases that do not belong to any topic (those with $F_i(d_j) < \tau_i$) in one level of the tree will be placed in this level inside the group of cases that correspond or will create a new topic of interest.

We use in this work linear classifiers that represent a category or topic like a vector $c_i =< w_{1i}, ..., w_{ri} >$where w_{ji} is the weight associated with the word j in the category i and the F_i function associate a each classifier is the cosine similarity measure shown in Equation 1. As was proved in [3] a very small number of features need to be included in the classifier in order to get an accurate document classification [3].

These features are selected using the document frequency measure over a group of cases. The document frequency $\#T_r(t_k)$ of a term t_k is the number of documents (textual cases in the same group) in which the term occurs. This value is calculated for each unique term that appears in the cases on the group and those terms whose $\#T_r(t_k)$ was higher than a given threshold will be constitute the classifier for that group or user topic of interest.

3 The *PersonalSearcher* Agent

The technique described in previous section was applied to the development of the *PersonalSearcher* agent [2], that assists users to find interesting documents in the Web. The agent carries out a parallel search in the most popular Web search engines and filters the resultant list of documents according to the users' interests or preferences.

Instead of receiving a big number of document, most of them irrelevant, as usually append with traditional search engines, a user gets a reduced number of documents with high probability of being relevant to him . Each agent, instance of *PersonalSearcher*, learns a model of preferences and topics of interest for his associated user based on observation of user browsing in the Web.

For each reading in the standard browser the agent observes a set of indicators in order to estimate the interest of the user in that Web page. This process is called implicit feedback since it can be obtained from the user without disturbing his normal behavior or distracting him to ask explicit evaluations for each visited page. These indicators are the time consumed in reading (with relation to its length), the amount of scrolling in a page and whether it was added to the list of bookmarks.

Web pages classified as interesting by this means are recorded as textual cases in the user profile. The agent deals with these cases in order to learn its salient characteristics that allows it to deduce the topics treated on them. At the same time, it organizes these topics building a topic hierarchy, which determines the user profile such as it was previously explained.

The agent operates over this profile in order to assist during Web search. Users interact with their *PersonalSearcher* expressing their information needs by keywords as usual. In turn, the agent posts these queries to the most popular search engines in the Web (Altavista, Infoseek, Excite, etc.) getting a set of documents that covers a wide portion of the Web.

The relevance degree of each document in relation to the user profile is computed by the agent to determine the convenience of suggesting the document to the user for a future reading. This process involves classifying the document into the hierarchy and looking for the case in the current level that presents the higher similarity level with it. Only documents that surpass a given similarity threshold as regards to the most similar case in the profile are sent back to the user as a result to his query.

Experimental results with this agent proved not only the high accuracy of agent suggestions, but also a steadily improvement on accuracy as the number of documents incorporated to the user profile grows. Also some preliminary results on the comparison of hierarchies have proved have proved that the collaboration among agents is possible [1].

4 Conclusions

This work has presented a technique to categorize documents according to a personalized hierarchy of topics of interest, which constitutes a contribution to user modeling in intelligent agents development. We have also presented an experience with an agent using our technique for Web search. Experiences with this agent have demonstrated that out technique can be successfully applied to this domain. As a future work remains to research their adaptability for tasks that involve other kinds of texts like e-mail messages, news articles, etc. The comparison between users profile based on this technique is also an open issue that needs to be treated in order to enable collaborative work among user communities.

References

[1] G. Giménez Lugo, A. Amandi, J. Sichman, and D. Godoy. Enriching Information Agents' Knowledge by Ontology Comparison: A Case Study. In F. Garijo, J. Riquelme Santos, and M. Toro, editors, *Advances in Artificial Intelligence - IBERAMIA 2002, 8th Ibero-American Conference on AI*, volume 2527 of *Lecture Notes in Computer Science*, pages 546–555. Springer Verlag, 2002.

[2] D. Godoy and A. Amandi. PersonalSearcher: An intelligent agent for searching web pages. In Maria Carolina Monard and Jaime Simão Sichman, editors, *Proceeding of the International Joint Conference IBERAMIA'2000/SBIA'2000*, volume 1952 of *LNAI*, pages 43–52. Springer-Verlag, November 2000.

[3] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In Douglas H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97)*, pages 170–178. Morgan Kaufmann Publishers, 1997.

[4] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[5] M. Lenz, A. Hübner, and M. Kunze. Textual CBR. *Case-Based Reasoning Technology, From Foundations to Applications*, 1400, 1998.

[6] D. Mladenic. Text-Learning and Related Intelligent Agents: A Survey. *IEEE Intelligent Systems*, 14(4):44–54, 1999.

[7] M. Porter. An algorithm for suffix stripping. program. *Program*, 14(3):130–137, 1980.

[8] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.