ANALYZING WEB CHAT MESSAGES FOR RECOMMENDING ITEMS FROM A DIGITAL LIBRARY

Stanley Loh
Catholic University of Pelotas – UCPEL, Brasil
Lutheran University of Brazil – ULBRA, Brasil

Ramiro Saldaña, Daniel Licthnow, Thyago Borges, Roberto Rodrigues,

Gabriel Simões, Leonardo Albernaz Amaral, Tiago Primo
Catholic University of Pelotas - UCPEL, Brasil

Keywords: text mining, recommendation, ontologies, chat, textual messages

Abstract: This work presents a recommender system that analyzes textual messages sent during a communication session in a private Web chat, identifies the context of each message and recommends items from a Digital Library. Recommendations are directly made to users in the chat screen and are decided by a software system through a proactive paradigm, without any request of the users. A domain ontology, containing concepts and a controlled vocabulary, is used to identify subjects in textual messages and to automatically classify items of the Digital Library.

1 INTRODUCTION

According to NONAKA & TAKEUCHI (1995), the majority of the organizational knowledge comes from interactions between people. People tend to reuse solutions from other persons in order to gain productivity.

People communicate using synchronous interactions (e.g., exchange of messages in a chat), asynchronous interactions (e.g., electronic mailing lists or forums), direct contact (e.g., two persons talking) or indirect contact (when someone stores knowledge and others can retrieve this knowledge in a remote place or time).

Recommender systems can help in processes of knowledge exchange and acquisition. A recommender system is a software whose main goal is to aid in the social process of indicating or receiving indication about what options are better suited in a special case for a certain individual (RESNICK & VARIAN, 1997). The main goal is to locate information sources related to a person's interest or need (MONTANER ET AL., 2002).

Recommendations are broadly used in electronic commerce for suggesting products or providing information about products and services, helping people to decide in the shopping process (LAWRENCE ET AL., 2001) (SCHAFER ET AL., 2001).

Recommender systems are proactive devices, because they can supply information without people having to search, query or look for it. The offered gain is that people do not need to request information, but a software system decides what and when to suggest. This kind of system is especially useful when there are many options to choose and users have little information about those options.

This work presents a recommender system to support people when using a Web chat for exchanging knowledge. Recommendations are made by the system during the online discussion, analyzing the context of the textual messages exchanged by the chat participants. To decide what

Loh S., Saldaña R., Licthnow D., Borges T., Rodrigues R., Simões G., Albernaz Amaral L. and Primo T. (2004).
ANALYZING WEB CHAT MESSAGES FOR RECOMMENDING ITEMS FROM A DIGITAL LIBRARY.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 41-48
DOI: 10.5220/0002627900410048

to recommend, the system uses an ontology and a Digital Library.

The paper is structured as follows. Section 2 discusses the proactive strategy used by recommender systems to retrieve information. Section 3 presents related works about recommendations. Section 4 describes the proposed system in details, including the functionality of each component of the system. Finally, section 6 presents concluding remarks and future work.

2 RECOMMENDATIONS: A PROACTIVE STRATEGY FOR INFORMATION RETRIEVAL

There are information retrieval systems to help people to find information in digital libraries (SPARCK-JONES & WILLET, 1997). Most of these systems demand people to state an information need as a query (in a formal language).

Taylor (*apud* OARD & MARCHIONINI, 1996) defines 4 types of information need:

- visceral need: when the need is not consciously perceived;
- conscious need: when the user perceives his/her need and knows what he/she wants;
- formalized need: when the user expresses his/her need in a formal way;
- compromising need: when the need is represented in the system.

For using information retrieval systems, the user has to formalize his/her information need (*formalized or compromising need*). The problem is that people are not able to specify what they need in formal languages, because that is exactly what is missing. Belkin and others (BELKIN ET AL., 1997) define the information need as an Anomalous State of Knowledge (ASK). Thus, any information need is inherently hard to specify.

Another problem is that sometimes people do not have a explicit knowledge (awareness) about what they need (*visceral need*).

CHOUDHURY & SAMPLER (1997) identified 2 different processes for information acquisition: reactive and proactive. In the first case, the user knows what he/she needs and is able to specify what is looking for. In the second case, the proactive one, the user does not have a specific goal and the purpose is to explore or monitor some situation, in order to discover something new. The typical characterization of this situation is when the user asks "*tell me something new or useful*". Proactive strategies are suited for users with *visceral needs*.

An example of a proactive strategy is the work of SWANSON & SMALHEISER (1997). They use common words to relate texts and find analogies. Their work has reached success discovering new hypothesis in the medicine area.

Traditional information retrieval systems use the reactive strategy. They consider that users have a clear definition of what they are looking for and that users are able to specify this using a formal language (for example, keywords and Boolean operators).

Some recent researches are dealing with proactive strategies for information retrieval through the use of recommendations.

Recommender systems help users with conscious or visceral needs. In the first case (*conscious need*), recommendations help the user to find what he/she wants without having to specify or formalize a query. A software system automatically identifies the user's need, goal or interest and searches for items that may be useful. This identification is made by analyzing the user's actions (as for example, navigation across a web site).

In the second case (*visceral need*), a recommender system may help the user by finding useful and new items, without any intervention of the user. Characteristics of the user (his/her profile) or the user's history (items bought, read, borrowed) may be analyzed in order to identify the user's interesting areas or his/her background knowledge. After that, the software system can look for new items inside that context.

The recommender system presented in this paper uses a proactive strategy for help users in these two cases. Analyzing messages sent by the user, the software system can identify the user's interest (a possible information need) and then select items from a digital library to suggest to this user.

By other side, when one user sends a message, all other users may receive recommendations, because the system admits that the participants of the discussion are also interested in the same subject. Of course, the system first analyzes the user's profile for not recommending twice the same item or for not recommending basic items for advanced users. This way, the user, even without making any action, may receive a suggestion.

3 RELATED WORK ABOUT RECOMMENDATIONS

Recently, recommender systems are being used to support knowledge acquisition. BRUSILOVSKY (1996) discuss applications of adaptive hypermedia systems (a kind of recommender systems) in educational environments, to support students in learning processes.

According to TERVEEN & HILL (2001), there are 4 kinds of recommender systems. *Content-based systems* use only customers' preferences. Items to be recommended are chosen from those similar to the ones related to the customer, for example, products in stock that are classified in the same section of the products bought by the customer. *Recommendation support systems* do not make automatically recommendations but help people to produce recommendations. *Social data mining systems* discover preferences analyzing records from social activities, like messages in newsgroups, citations in scientific papers, usage logs of a system, peer-to-peer services (like exchange of music and documents), etc. *Collaborative filtering systems* do not consider the content of the items but the similarity among people and the items related to them; the goal is to find people with similar preferences and make cross-recommendations.

GroupLens system uses collaborative filters to help people to find useful information (RESNICK ET AL., 1994). The technique collects user's feedback to select new articles that can be interesting to the user.

TERVEEN & HILL (2001) discuss the PHOAKS system, which extracts addresses of Web pages from messages in the Usenet newsgroup for future recommendations.

Other system (proposed by Donath et al. *apud* TERVEEN & HILL, 2001) analyzes messages in the Usenet and other chats intending to later recommend group of messages according to some attributes (for example, presence of certain themes or discussions with greater number of participations).

The TAPESTRY system allows people to evaluate messages or news and it allows an user to retrieve items based on content or in collaborative evaluations (GOLDBERG ET AL., 1992). Unfortunately, this system does not act in proactive way, because users have to retrieve items using queries.

KOMOSINSKI ET AL. (2000) presents a system that identifies terms in messages of a chat and gives to the participants the definition of the terms during the interaction.

The system presented in the current paper is different from the others because it analyzes the messages exchanged in the chat and identifies the context of the discussion for then selecting items from the digital library to recommend particularly to each user.

4 DESCRIPTION OF THE SYSTEM

The goal of the present recommendation system is to provide people with useful information during a collaboration session. To do that, the system analyzes textual messages sent by users when interacting in a private Web chat, identifies subjects/themes/concepts inside the messages and recommends items cataloged in a digital library, previously classified in the same subjects. The Digital Library contains electronic documents.

A Text Mining module analyzes each message. The words present in the message are compared against the domain ontology. After, it passes the concept identified to the Recommender module, that looks in the library for items to suggest.

According to the classification of TERVEEN & HILL (2001), the system is a *content-based recommender system* because the context of the messages is matched against the content of items in the database. The system is also a *social system* because it analyzes messages exchanged in a Web chat. And the system uses *collaborative filtering* because the database is created by people, especially the digital library, where users of the community can upload items into.

One difference of the proposed system from others is that it is not necessary to store a profile for a user to use the system and receive recommendations. Messages sent by users are enough for the system deciding what recommend.

In the next sections, each component of the system is described in details. All the system was implemented with PHP, Javascript and MySQL.

4.1 The Web Chat

The chat works like traditional ones in the Web. The difference is that it is specially constructed for this system and it is not open to non-registered users. Thus, users have to be authenticated for using the system. There is no limit for the number of persons interacting at the same time.

At the moment, only one chat channel is allowed. Thus a discussion session concerns all messages sent during a day. In the future, this restriction will be eliminated.

4.2 Message Analysis

The main component of the system is a Text Mining Module. It works as a *sniffer*, examining each message sent in the chat. This module is responsible for identifying subjects in the messages. Subjects are identified by comparing words present in the message against terms defined in the ontology. Generic terms like prepositions (called *stopwords*) will be discarded. Each message is compared online against all concepts in the ontology. The concepts identified in the message represent user's interests and will be forwarded to the Recommender Module.

The text mining method employed in this system (a kind of classification task) was first presented in LOH ET AL. (2000). Instead of using Natural Language Processing (NLP) to analyze syntax and semantics, the method is based on probabilistic techniques: subjects can be identified by cues. Using a *fuzzy* reasoning about the cues found in a text, it is possible to calculate the likelihood of a subject being present in that text.

The algorithm is based on Rocchio's and Bayes' algorithms (ROCCHIO, 1966; RAGAS & KOSTER, 1998; LEWIS, 1998), since it uses a prototype-like vector to represent texts and concepts. The method evaluates the relationship between a text and a concept of the ontology using a similarity function that calculates the distance between the two vectors. The vectors representing texts and concepts are composed by a list of terms with a weight associated to each term. In the case of texts, the weight represents the relative frequency of the term in the text (number of occurrences divided by the total number of terms in the text). And the weight in the concept vector represents the probability of the term being present in a text of that subject. The next section (the ontology) describes how concept weights are defined.

The text mining method compares the vector representing the text of a message against vectors representing concepts in the ontology. The comparison between the two vectors is done through a *fuzzy* reasoning process, following (ZADEH, 1973) and (NAKANISHI et al., 1993). In the comparison method, weights of common terms (those present in both vectors) are multiplied. The overall sum of these products, limited to 1, is the degree of relation between the text and the concept, meaning the relative probability of the concept presence in the text or that the text holds the concept with a specific degree of importance. The decision concerning if a concept is present or not depends then on the threshold used to cut off undesirable degrees. This threshold is previously set by an expert.

The method is based on the relevancy index proposed in (RILOFF & LEHNERT, 1994) whose definition is "*a collection of features that, together, reliably predict a relevant event description*". Some terms may indicate the presence of a subject with a degree of certainty. Therefore the *fuzzy* reasoning process must evaluate the likelihood of a concept being present in a text, analyzing the strength of its indicators. The process is like an abductive reasoning. According to GULLA (1997), in a deduction, if "A → B" and "A is truth" then we can infer "B is truth". In abduction, if "A → B" and "B is truth" then "A is a probable cause for B being truth". This means that if words describing a concept <u>c</u> appear in a text, there is a high probability of that the concept <u>c</u> will be present in that text.

When two or more concepts are identified in the same message, the degrees of relationship between the message and the concepts are used to form a ranking. Only the top concept in the ranking is considered. If two concepts are identified with the same degree, but one is "father" of the second in the ontology, only the more specific concept (son) is considered.

New terms, used in the messages but not present in the ontology, are stored for future analysis.

4.3 The Ontology

A domain ontology is a description of "things" that exist or can exist in a domain (SOWA, 2002). It is a formal and explicit definition of concepts (classes or categories) and their attributes and relations (NOY & McGUINNESS, 2002).

A domain ontology is similar to a *thesaurus*. In fact, a *thesaurus* is a kind of ontology, but this will not be discussed in this paper (see GILCHRIST, 2003, for a detailed discussion). According to FOSKET (1997), *thesaurus* is a device to control terms in texts. *Thesauri* provide knowledge maps, representing concepts or ideas of the application domain and indicating relations among them. A *thesaurus* also defines terms used to describe a concept.

In the proposed system, the ontology is implemented as a set of concepts in a hierarchical structure (a root node, fathers and sons). Each concept has associated a list of terms and their respective weights that help to identify the concept in texts (messages or electronic documents). Weights are used to state the relative importance or the probability of the term for identifying the concept in a text. The relation between concepts and terms is many-to-many, that is, a term may be present in more than one concept and a concept may be described by many terms.

The ontology is used to identify subjects in textual messages and to automatically classify items of the digital library.

A software tool is used to manage and configure the ontology, including functions to visualize the structure of concepts and the list of terms, to insert a new concept and its respective list of terms, to insert/remove terms and to modify the weights. A group of administrators is responsible for creating and updating the ontology. New terms, found by the Text Mining module, may be added as a new concept, or inside an existing concept or the new term may be added to the stopword list.

Currently, the system uses a domain ontology for Computer Science, but others can be used. For this purpose, the domain ontology has a root node called "ontology". Under this node, other ontologies may be aggregated.

Concepts and the hierarchy were based on the ACM classification for Computer Science. Software tools supported people in identifying terms for each concept. Terms and weights were defined using the supervised learning strategy (for machine learning): experts selected texts about a concept (approximately 100 per concept) and a software tool identified the most important terms for each class. The texts where extracted from the Research Index database (www.researchindex.com). A probability measure was used to define the weight of each term in a concept. Stopwords (prepositions and others) were removed.

Furthermore, experts reviewed the ontology eliminating terms present in many concepts and adding word variations with the same weight as the principal. This last task was important since texts were written in English and Portuguese. So, the terms used in the ontology come from these two languages.

4.4 The Digital Library

The Digital Library used by the recommender system is a repository of electronic documents.

The inclusion (upload) of items in the Digital Library is responsibility of authorized people and can be made offline in a specific module.

A module for upload of items from non-authorized people is being developed, so that these items can be approved or rejected later by referees.

The classification of the electronic documents is made automatically by software tools, using the same text mining method used in the Text Mining Module. A difference is that a document may be related to more than one concept. A threshold is used to determine which concepts can be related to one document. Thus, the relation between concepts and documents in the Digital Library is many-to-many. The relationship degree is also stored.

When inserting a item in the base, the user have to designate whether the item is basic, intermediate or advanced. This information will be used later by the recommender module to avoid suggesting basic items to experienced users.

4.5 Recommendations

The goal of the Recommender Module is to offer electronic documents, stored in the Digital Library, to the chat participants. The module uses a content-based technique, where only items classified in the concept identified in the message are recommended.

The action of this module starts when it receives a concept from the Text Mining Module. Then, it searches the Digital Library for items classified in the same concept. Each time the Text Mining Module identifies a concept in a message, it sends this concept to the Recommender Module. Similarly, the searches happen online, that is, immediately when the Recommender Module receives a concept.

Since the discussion in the chat is synchronous, recommendations should not interrupt the users. So, indications are given in a separate frame and not inside the chat window. Recommendations are particular of each user. Thus, each user receives a different list of suggestions in the screen.

The Recommender Module uses a Profile Base for registering information about each user as demographic characteristics and his/her history in the system, like items read from, uploaded to or downloaded from the Digital Library.

The system also uses a punctuation schema to identify interesting areas for each user and to determine his/her degree of knowledge about subjects (represented by concepts of the ontology).

Each operation of the user inside the system gives him/her some points in the profile. Operations that punctuate are: download and upload of items in the Digital Library, opening a PDF file, participating in a discussion, sending messages and the authorship of electronic documents stored in the Digital Library.

The knowledge degree is used for not recommending basic items for advanced users and to determine authorities in each subject.

Additionally, the system must not recommend:

a) the same item twice in the same section;
b) items already associated to the user (downloaded, uploaded or read);
c) items marked by the user to not be recommend (button "never show me again").

As the recommendation list may increase without limit, there are options for the user to reduce the overload in the recommendation frame. For example, there is an option to eliminate a item from the session list ("*don't show me again in this session*") and other option to set in the user profile that the item should not be recommended again ("*never show me again*").

Furthermore, the user can access details of the item being recommended as title, authors, keywords, abstract and the related concepts with the degree of relationship (this information is stored in the digital library).

Figure 3 shows a snapshot of the system in a real situation. There is an area where the nicknames of the participants of the chat appear (*users*), an area where the messages can be viewed (*discussion*), an area where the recommendations appear individually for each user (*recommendations*) and another area for the user writing the messages (*message*).

The prototype system is accessible at http://gpsi.ucpel.tche.br.

5 CONCLUSIONS

This work presented a recommendation system to support knowledge exchange and acquisition in a Web chat discussion. The system allows proactive information retrieval because people receive suggestions of electronic documents without having to search the digital library

The main advantage of the system is to free the user of the burden to search for information during the online discussion. Users do not have to choose attributes or requirements from a menu of options, in order to retrieve items of a database. The system decides when and what information to recommend to the user. This proactive approach is useful for non-experienced users that receive hits about what to read in a specific subject. With the system, user's information needs are discovered naturally during the conversation.

This system points to a new kind of support for users in online discussions. Furthermore, Digital Libraries get a new facet with the additional help of recommender systems. Feigenbaum (*apud* DAVIES, 1989) compares the current libraries with future ones. The existing ones consist of a warehouse of passive objects. In the other side, libraries of the future will be a collection of active documents, helping people to discover new knowledge through providing connections and associations previously unknown, analogies and new concepts, without people having to state clearly their information needs.

At the moment, the ontology only concerns a small set of the Computer Science area. It is possible to extend this ontology or even to create and use other domain ontologies with concepts from other subject areas. Similarly, the current digital library only has items related to Computer Science. For other communities using the system, the digital library has to be populated with items from other areas.

Some experiments were carried out with the recommendation system. Results reveal a promising strategy, reducing the charge for the user find useful information, especially when dealing with great volumes of documents in Digital Libraries.

However, yet some poor recommendations are being generated. This is due in great part to the ontology. We detected some problems like, for example, the use of generic words (like "software") appearing with high weight in different concepts; lack of word variations and lack of specific concepts (some concepts are too much broad, including different items).

Quality of the recommendations can be improved by developing better the ontology. For example, we plan to use a stemming algorithm to treat word variations. We are also investigating how to automatically divide concepts in sub-concepts in order to accommodate the specificity of the sub-groups. A software tool is being developed to identify ambiguous terms (that appear in many concepts) and to automatically correct their weights.

Furthermore, recommendation quality is also dependent on the quality of the digital library. For example, few documents may cause poor recommendations independently of the methods used. We are studying an automatic software that will search the Web for electronic documents under concepts present in the ontology. After finding the documents, other software will extract document information as authors, title, abstract and keywords, besides the existing automatic classification of the document in the ontology concepts.

Regarding performance, it is possible to say that the recommendations are made very quickly (in milliseconds).

As a great number of recommendations may be sent to users, the system allows the user to set a threshold and only the items related to the concept with a degree above this threshold will be presented to the user.

The punctuation schema is still being tested. We have to determine how many points will be given for each operation. An ongoing implementation will determine initial points for the user profile according to his/her publications as specified in the *curriculum vitae*.

Other future works include the implementation of other recommendation techniques (like collaborative filtering) and techniques to minimize information overload due to the great number of items recommended. A special study is being conducted to use relevance feedback to narrow the list of recommendations. Users would read some items of the list and rate them, and the system would use this information to eliminate items from the list or to reorder the items in a new ranking.

An interesting future research is to use context in interpreting each message. Currently, each message is analyzed independently of others. This can lead to mistakes in the subject identification. For example, if one message has the words "*neural nets*" and the next one has "*learning*", the system should recognize that "*learning*" is related to "*machine learning*", since it is assumed that the discussion would not deviate from the context. However, in the current state, the system may identify "*learning*" in a context like "*Computers in Education*". We are studying a solution that considers the context of a discussion, that is, the system will assume that discussions do not move far from one subject to other. Using the hierarchy of concepts, it is possible to identify the distance of each movement from one message to another, and this will be used to disambiguate a message: when two or more subjects are identified in a message, the nearest one must be used.

ACKNOWLEDGEMENTS

This work is partially supported by CNPq, an entity of the Brazilian government for scientific and technological development.

REFERENCES

N. J. BELKIN; R. N. ODDY; H. M. BROOKS, ASK for information retrieval: part I - background and theory, in: K. SPARCK-JONES and P. WILLET, Peter, eds., Readings in Information Retrieval (Morgan Kaufmann, San Francisco, 1997).

P. BRUSILOVSKY, Methods and techniques of adaptive hypermedia, User Modeling and User Adapted Interaction 6 (2-3) (1996) 87-129.

V. CHOUDHURY and J. L. SAMPLER, Information specificity and environmental scanning: an economic perspective, MIS Quarterly 21 (1) (1997) 25-50.

R. DAVIES, The creation of new knowledge by information retrieval and classification, Journal of Documentation 45 (4) (1989) 273-301.

D. J. FOSKET, Theory of clumps, in: K. SPARCK-JONES and P. WILLET, eds., Readings in Information Retrieval (Morgan Kaufmann, San Francisco, 1997).

A. GILCHRIST, Thesauri, taxonomies and ontologies – an etymological note, Journal of Documentation 59 (1) (2003) 7-18.

D. GOLDBERG et al., Using collaborative filtering to weave an information tapestry, Communications of the ACM 35 (12) (1992) 61-70.

J. A. GULLA et al., An abductive, linguistic approach to model retrieval, Data & Knowledge Engineering, 23(1) (1997) 17-31.

L. J. KOMOSINSKI et al., The usage of agents to support dialog mediation between students via Internet, in: Proc. RIBIE'2000 5th Iberoamerican Conference on Informatics in Education, Viña del Mar, Chile, December 2000. (*in portuguese*)

R. D. LAWRENCE et al., Personalization of supermarket product recommendations, Journal of Data Mining and Knowledge Discovery 5 (1/2) (2001) 11-32.

D. D. LEWIS, Naive (bayes) at forty: the independence assumption in information retrieval, in: Proc. European Conference on Machine Learning, Lecture Notes in Computer Science, v.1398 (Springer, Berlin, 1998) 4-15.

S. LOH; L. K. WIVES; J. P. M. OLIVEIRA, Concept-based knowledge discovery in texts extracted from the Web, ACM SIGKDD Explorations 2 (1) (2000) 29-39.

M. MONTANER; B. LOPEZ; J. L. de la ROSA, Improving case representation and case base maintenance in recommender agents, in: Proc. 6th European Conference on Case Based Reasoning (2002).

H. NAKANISHI; I.B. TURKSEN; M. SUGENO, A review and comparison of six reasoning methods, Fuzzy Sets and Systems 57 (3) (1993) 257-294.

I. NONAKA and T. TAKEUCHI, The knowledge-creating company: how japanese companies create the dynamics of innovation. (Oxford University Press, Cambridge, 1995).

N. F. NOY and D. L. McGUINNESS, Ontology Development 101: a guide to creating your first ontology (2002) Available at http://protege.stanford.edu/publications/

D. W. OARD and G. MARCHIONINI, A conceptual framework for text filtering. University of Maryland, Technical Report EE-TR-96-25 (1996). Available at: <http://www.ee.umd.edu/medlab/filter/>. Access in June 1998.

H. RAGAS and C.H.A. KOSTER, Four text classification algorithms compared on a Dutch corpus, in: Proc. SIGIR'98 International ACM-SIGIR Conference on Research and Development in Information Retrieval (ACM Press, Washington, 1998) 369-370.

P. RESNICK et al., GroupLens: an open architecture for collaborative filtering of Netnews, in: Proc. Conference on Computer Supported Coooperative Work (1994) 175-186.

P. RESNICK and H. VARIAN, Recommender systems, Communications of the ACM 40 (3) (1997) 56-58.

E. RILOFF and W. LEHNERT, Information extraction as a basis for high-precision text classification, ACM Transactions on Information Systems 12 (3) (1994) 296-333.

J. J. ROCCHIO, Document retrieval systems - optimization and evaluation, Ph.D. Thesis, Harvard

Computation Laboratory, Harvard University, Report ISR-10 to National Science Foundation (1966).

J. B. SCHAFER et al., E-commerce recommendation applications, Journal of Data Mining and Knowledge Discovery 5 (1/2) (2001) 115-153.

J. F. SOWA, Building, sharing, and merging ontologies (2002). Available at http://www.jfsowa.com/ontology

K. SPARCK-JONES and P. WILLET, Eds., Readings in Information Retrieval, San Francisco, Morgan Kaufmann (1997).

D. R. SWANSON and N. R. SMALHEISER, An interactive system for finding complementary literatures: a stimulus to scientific discovery, Artificial Intelligence 91(2), (1997) 183-203.

L. TERVEEN and W. HILL, Beyond recommender systems: helping people help each other, in: J. CARROLL, ed., Human computer interaction in the new millennium (Addison-Wesley, 2001).

L. A. ZADEH, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man and Cybernetics SMC-3 (1) (1973) 28-44.

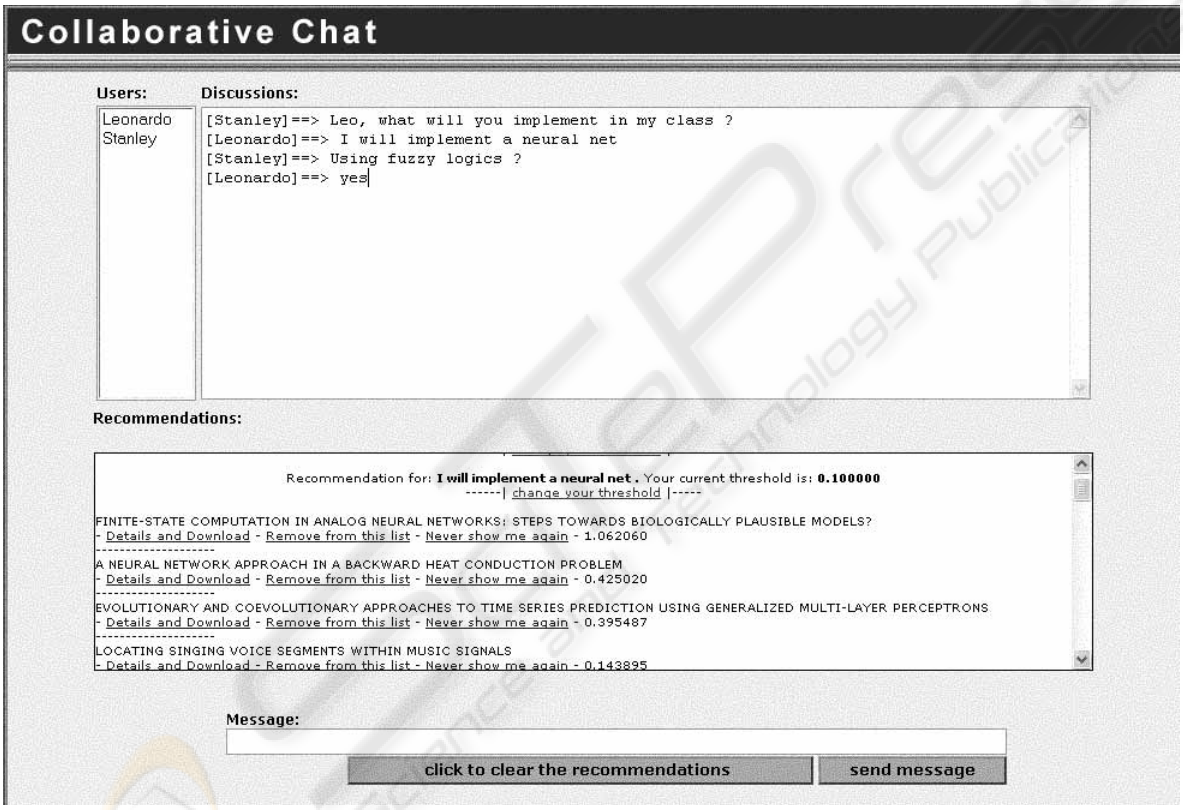

Figure 1: A snapshot of the system in a real situation