A Multi-Agent System for Generating a Personalized Newspaper Digest

Georg Veltmann
Daimler-Benz AG, Research Center Ulm
P.O. Box 23 60, 89013 Ulm, Germany
georg.veltmann@dbag.ulm.daimlerbenz.com

Abstract

This paper describes the *PZ* system, an assistant system that generates a personalized newspaper digest. To cope with the manifold user interests and the dynamic nature of the newspaper domain, the system is built as a multi-agent system. Each agent models a different facet of the user's interest. Working together in a economy where only useful agents survive, the agents learn from the feedback the users provide as they read the digest. The use of implicit feedback was found to be sufficient for generating a good digest. Experiments with simulated users and experience with real users are reported.

Introduction

This paper describes a system for generating a personalized digest from German newspapers which are available on the WWW. A learning *interface agent* (Maes 1994) for this task faces two main problems:

- The agent has to model the various facets of the user interest (e.g. stock prices, foreign affairs, and soccer) without putting too much mental load on the user.
- The newspaper domain is very dynamic. User interest may change from one topic to another over time either slowly (called *interest drift*) or suddenly (*interest shift*). Moreover, new topics arise every day. The agent should adapt quickly to these changes.

The *PZ* system (*PZ* for "Persönliche Zeitung," German for "personal newspaper") was designed with these problems in mind. The core of the system is a multi-agent system consisting of different agent types[1]. Each agent covers a given aspect of the user interest. Together they cooperatively model the complete interest. The *PZ* system does not force the user to express explicitly her/his interest in particular newspaper articles. Instead, the user interface logs the user's actions while reading the digest. Reading an article denotes interest whereas ignoring it expresses disinterest. This *implicit feedback* together with optional *explicit feedback* (user pushes a feedback button) provides learning input for the system. To adapt to changes each agent learns locally, but the complete agent system also learns by shifting weights between the agents' suggestions, introducing new agents and deleting no longer useful ones.

Related work

Personalized newspapers are already available on the Internet. Most services such as POINTCAST[2] simply allow users to choose among predefined categories. Other non-adaptive systems such as *Paperboy*[3] apply IR-like keyword-oriented queries to incoming articles. The cognitive load for constructing good queries is placed on the user.

Newt by Sheth (Sheth 1994) was one of the first learning systems for generating a digest. *Newt* uses a genetic algorithm where each individual in the population represents a query. Using crossover and mutation together with local learning by the individuals, *Newt* is able to track user interest drifts and shifts. Still, users have to control many parameters to tailor the system to their needs.

The effect of using implicit rather than of explicit feedback for deduction of user interest was examined in the *ANTAGONOMY* system (Sakagami & Kamba 1997).

Other research concentrates on improving classification accuracy, see (Yang 1997) for a comparison of different algorithms. For each part of the interest the user has to label articles as examples, and the system then builds user profiles from them.

Description of the *PZ* system

The *PZ* system consists of several parts (see Figure 1) which communicate via a central *Blackboard* where information is exchanged. The *Indexer* fetches HTML pages from the newspaper homepages and inserts articles as term vectors into the blackboard. Next, *Clustering* is performed on the new collection of articles. After user feedback has been written on the blackboard by the *User Interface* component, the *Supervisor* starts the learning phase of each agent in the *Multi-Agent System* (MAS). Then each agent computes recommendations, each with an associated confidence value, and writes them on the blackboard. The *Ranking Combination* joins all recommendations to form a final ranking. Now

[1]The term "agent" is not very clear. The *PZ* system as whole is an *interface agent* from the user's point of view, but is constructed as a *multi-agent system.* For a discussion of this problem see (Moukas & Zacharia 1997).

[2]URL: http://www.pointcast.com

[3]URL: http://www.paperboy.net

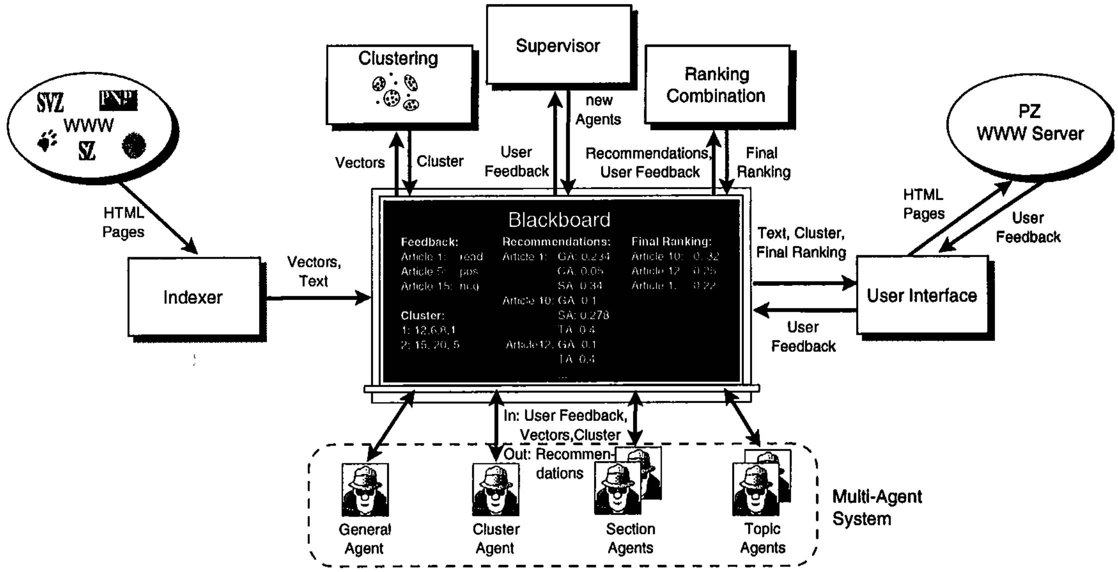

Figure 1: Overview of the *PZ* system

the new digest can be displayed via the User Interface. Most system components are explained in more detail below.

Indexing

Since *PZ* has to deal with HTML pages in different formats from the newspapers with much administrative content, the first step cuts out the interesting parts of a newspaper article such as headline, subheadline, date and the body of the article. *PZ* uses GERTWOL (Haapalainen & Majorin 1994), a morphological analysis system including part-of-speech tagging and stemming, to reduce the number of features. Only the stems of nouns, verbs, and adjectives are used as indexing terms. This cuts the number of features down to 60% of the original number. TF/IDF weighting is used to compute term vectors for each article.

Clustering

The aim of the *PZ* system was to generate a newspaper digest. It would be useful to group articles belonging to the same topic together in the display. Therefore the articles are clustered. We use a bottom-up hierarchical clustering method with a group average metric as the clustering criterion (Willet 1988). The cosine between article vectors is used as the simililarity metric. The resulting cluster tree is pruned at a predefined threshold of inter-cluster similarity (currently 0.13), and only clusters which contain more than 2 articles are preserved.

Multi-Agent System

To model the different facets of user interest, the *PZ* system utilizes different kinds of agents working together in the MAS. The characteristics of the four agent types and the learning method for the whole MAS are explained below.

Agent types The *General Agent* (GA) monitors all articles and learns from all user feedback. *Section Agents* (SA) focus their attention on one newspaper section such as economics or science. For special topics of user interest (expressed by giving positive explicit feedback for an article), *Topic Agents* (TA) are created. These three agent types use Rocchio's algorithm (Rocchio 1971) for learning and updating a profile vector.

Since the goal is to generate a digest containing selected articles and not to rank the whole collection, agents insert recommendations in the blackboard only if their confidence measured by the cosine coefficent between the article vector and the profile vector is above a given threshold θ, which is adjusted in every learning phase.

While it is clear what articles are positive or negative examples for GA and SA learning, TA learning is different. Only articles related to the topic should influence the profile vector update in order to maintain specificity. Therefore only articles with a cosine greater than Θ (here set to 0.08) with respect to the old TA profile vector are used as learning examples. This technique is similar to *query zoning* as described in (Singhal, Mitra, & Buckley 1997).

When the *PZ* system is used over a long period of time, the profile vectors tend to grow. Interest drift causes the inclusion of new terms to the vectors and makes old terms obsolete. Therefore each entry in the profile vectors has a time stamp with the date of its last use for calculating the cosine. If a term has not been accessed for τ days it is deleted from the user profile (τ was set to 14 days in the experiments).

The *Cluster Agent* (CA) does not model the user interest in topics, but rather the user's interest in up-to-date articles. The size of an article cluster gives a measure of actuality and importance the newspaper editors have assigned to the topic of the cluster. For each cluster, the article closest to the cluster centroid is recommended to the user.

Learning in the MAS Each agent is supplied with a intial energy. Whenever an agent receives positive feedback it gets a reward increasing his energy. Analogically, negative feedback decreases energy. Every agent is also charged an amount of energy for making a recommendation. Thus, the number of possible recommendations is limited by the energy of the agent. An obligatory everyday charge ensures that agents which never recommend articles do not clog the system. If the energy of an agent drops below zero, the supervisor deletes it from the MAS. This economic system retains useful agents and discharges ineffective ones.

Each agent marks the feedback it has used for learning. After all agents have learned, the supervisor checks whether there are any articles which come from a section without assigned SA and have positive feedback, or any articles with explicit positive feedback on topics which have no assigned TA. In this case new agents are created using the particular article as their inital learning example.

User Interface

In order to make the *PZ* system accessible for many users, the user interface was implemented as a WWW server. When a user connects to the *PZ* server, she is identified and her personalized newspaper digest is displayed (Figure 2). A list of newspaper sections is shown on the left while the right frame displays the text of the article. Below the text, all other articles belonging to the same cluster or the same section are listed. The server counts reading an article as implicit positive feedback. In the lowermost frame, explicit positive or negative feedback can be provided.

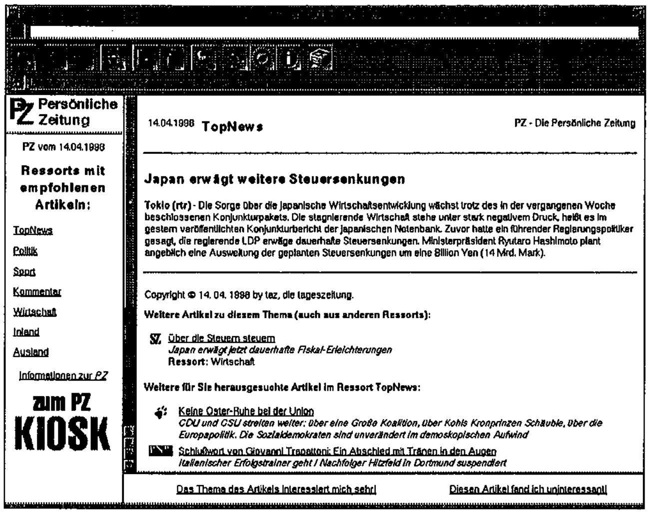

Figure 2: User interface in web browser window

The *PZ* system focuses the user attention on a small selection of articles. New topics are introduced only by the cluster agent. To allow exploration, users can browse all articles at the “Kiosk” (German for “newsstand”). Here the articles are sorted by newspaper or by section, and the systems logs only explicit positive feedback.

Experiments

Evaluating a information filtering system is based mostly on test collections of documents rated as relevant or irrelevant. The systems ability to adapt to changes in user interest can be tested by defining simulated users as introduced by (Sheth 1994). In addition to this, the *PZ* system was used over a period of six months by researchers at Daimler-Benz AG to obtain real user feedback.

Tests with simulated users

Over a two and a half month period 17304 articles from 4 different newspapers were collected. Fourteen topics such as UNO, soccer, and movies, were identified by the occurrence of certain keywords in the article. To test the *PZ* system, two different interest profiles were defined:

T14: This profile assumes a user with various interests. All defined topics are included. 2444 articles have been marked with implicit feedback. The first article of each topic was marked as having explicit positive feedback to allow topic agents to be initiated.

T6-T7: This profile simulates an interest shift. At first, only six topics are of interest. After 33 days the interest changes to seven other topics. No negative explicit feedback was provided for old interest articles after the shift, but each topic starts with an article having explicit positive feedback as in T14. Altogether 1362 articles were marked.

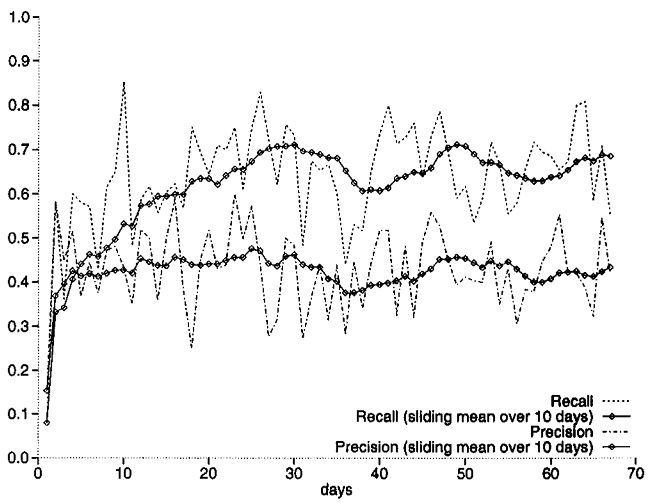

Figure 3: Precision and Recall on T14 data

The results for T14 are shown in Figure 3. *Precision* and *recall* are used as evaluation metrics. We calculated a sliding mean over 10 days to smooth the curve. The average precision was 0.44, while average recall was 0.65. The recall curve clearly shows that the system slowly learns the interests of the user, whereas precision is good from the start. On the average 36 articles were recommended, which means that 86% of the articles were filtered out. The results for T6-T7 (Figure 4) clearly show the interest shift on day 34. Both recall and precision drop significantly but rise quickly to their previous or (in the case of recall) to an even higher value. The smaller number of relevant articles caused a loss in precision (average 0.35) and recall (average 0.6).

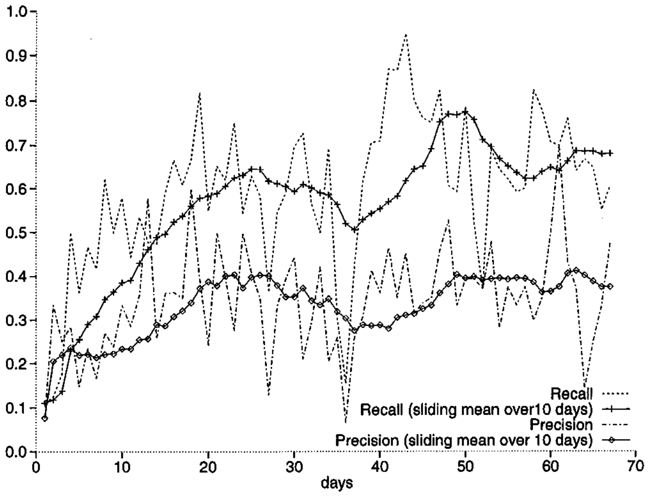

Figure 4: Precision and Recall on T6-T7 data

Experiences with real users

For experimentation the *PZ* system was accessible for 6 months to a group of researchers. Only five of them used the system on a regular basis. Due to other work the digest was not read every day. Sometimes only the most interesting parts of the digest were examined. Therefore the precision varied even more than in simulated user experiments.

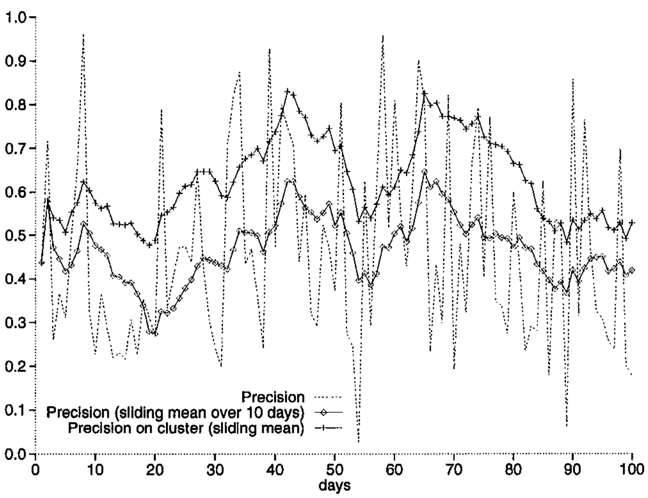

Figure 5: Precision for user U1

In Figure 5, the results for one user are given for those days the digest was read. The average precision was 0.46. This means that nearly half of the recommended articles were read. Scores for other users were similar. Since real users do not provide feedback on unread articles, no recall values can be calculated.

The *PZ* system inundates its users from time to time with up to ten articles belonging to one cluster. Not all articles normally get read, since the user is already very well informed after having read five articles. If we count all articles of a cluster belonging to a article already read as also being read, then the precision rises on the average to 0.62 (see top curve in Figure 5). On the average 39 out of 270 articles in a collection were presented.

Discussion and future work

The *PZ* system was developed as an interface agent to guide the user to interesting newspaper articles. The approach chosen deals sucessfully with the problems of the newspaper domain. Local and global learning enables the system to cope with changes in user interest and topics. The various facets of user interest were modeled with different agent types. Tests with real users yielded similar performance than experiments with simulated users.

Further work on the filtering effects of the different agent types is needed. Tools for tracking topics and agents have to be developed. Moreover, new agent types could be added to the MAS. Oddity agents recommend peculiar articles to enhance exploration of the article collection. Collaborative filtering agents (Resnick & Varian 1997) select articles based on recommendations from other similarly interested users.

References

Haapalainen, M., and Majorin, A. 1994. *GERTWOL: Ein System zur automatischen Wortformerkennung deutscher Wörter.* Lingsoft, Inc.

Maes, P. 1994. Agents that Reduce Work and Information Overload. *Communications of the ACM* 37(7):31–40.

Moukas, A., and Zacharia, G. 1997. Evolving a Multiagent Information Filtering Solution in Amalthea. In *Proceedings First International Conference on Autonomous Agents.*

Resnick, P., and Varian, H. R. 1997. Recommender Systems. *Communications of the ACM* 40(3):56–58.

Rocchio, J. J. 1971. Relevance Feedback in Information Retrieval. In Salton, G., ed., *The SMART Retrieval System: Experiments in Automatic Document Processing*. Engelwood Cliffs, NJ: Prentice Hall. 313–323.

Sakagami, H., and Kamba, T. 1997. Learning Personal Preferences On Online Newspaper Articles From User Behaviours. In *Proceedings of the Sixth International World Wide Web Conference.*

Sheth, B. D. 1994. A Learning Approach to Personalized Information Filtering. Master's thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA.

Singhal, A.; Mitra, M.; and Buckley, C. 1997. Learning Routing Queries in a Query Zone. In *Proceedings of SIGIR '97*, 25–32.

Willet, P. 1988. Recent Trends in Hierarchic Document Clustering: a Critical Review. *Information Processing & Management* 24(5):577–597.

Yang, Y. 1997. An Evaluation of Statistical Approaches to Text Categorization. Technical Report CMU-CS-97-127, Carnegie Mellon University, Pittsburgh, PA.