Using Genetics in Information Filtering

Max Höfferer, Bernd Knaus, Werner Winiwarter
Institute of Applied Computer Science and Information Systems
Liebiggasse 4/3, A-1010 Vienna, Austria
e-mail: {mh, bk, ww}@ifs.univie.ac.at

Abstract

Adaptive filtering of information involves modelling the user's behavior and learning from feedback. This paper describes the design of a cognitive information filtering system that applies (1) linguistic analysis to obtain consistent representations of the contents of messages, (2) an evolutionary algorithm for prioritizing morphologically parsed e-mails and (3) a monitor to simulate cognitive user behavior.

1. Introduction

More and more users of *e-mail* and other on-line communication systems are faced with the problem of selecting relevant information in a space of different information sources. Participants of on-line conferences, members of office-information systems, and network-diary makers are confronted with an endless stream of messages. To overcome this *information overload problem* [1] information filtering techniques have been developed to deliver information to users who really need it.

State of the art IFS, e.g. *Information Lens System* [2], *EDS Template Filler* [3] or *Isceen* [4] offer a static behavior. A cognitive information system should have the ability to learn and be adapted to the user's behavior. The system presented - *Cognitive Information Filtering System* (*CIFS*) - applies genetic adaption to learn from user feedback and user behavior. CIFS distills e-mails from the input stream depending on the user's interests and evaluation judgements which are used to rank e-mail information.

With regard to efficiency the linguistic analysis of the e-mails is performed by use of a cascading architecture. The filtering proceeds by stepwise refinement of analyzing techniques leading to a consecutive reduction of the problem

space. Therefore, only a very small fraction of the incoming e-mails must be analysed by sophisticated linguistic methods.

This paper is organized as follows. In Section 2 and Section 3 we will introduce the basic concepts of cognitive models and information filtering. The linguistic analysis applied consisting of the two modules indexer and parser will be presented in Section 4. Section 5 describes computational models of evolutionary processes. Finally, Section 6 gives a view of the architecture of the cognitive information filtering system.

2. Cognitive models

In each electronic information source there can be so much detail that the information presented to the reader may be of lower quality and less relevant than traditional approaches. The ability to select relevant information for a user is essential to the viability of such services and requires an individual user model [5]. In our approach we incorporate the following cognitive aspects to improve our system's ability for filtering e-mails.

Given the diversity of IFS users, the fact that they have not the same problems or needs and that the user's level of expertise and interests are likely to change over time, it is desirable that *profiles* should be able to be adapted to and support the requirements of individual users. Monitoring data collection techniques, think-aloud protocols, tape recording of interaction, interviews, and questionnaires are helpful to understand the filtering process of an individual user [6]. The user's behavior is mapped to the behavior of the system.

The system must also have a model of its own to foresee its actions in a possible future state and thus be able to choose the best way to perform. Therefore we apply techniques similar to those used for debugging to give us a trace of system actions during the filtering process. If no user interrupt occurs, in case of a relevant message, the system analyses any observations so that it can choose what to do with the next similar incoming message. Observation is needed to find out system actions in the course of the filtering process. The observation may be passive or active depending on whether it memorizes what happened or makes experiments to find out autonomous new

topics that may be interesting for the user. In the first prototype we only use passive observation.

As long as the system does not notice abnormal behavior, it does exactly the same without observing itself but simultaneously creates a trace of its actions and what it gets; it may correct immediately what goes wrong or may analyse it later. This observation may be continuous or occasional. In this approach we use *continuous observation*. At first, the user may specify a catalogue of relevant topics (*user-profile*) or the system uses a *standard-profile* that determines what can be observed. On getting more and more incoming messages the system learns by analyzing the traces and the user's reaction to the output.

A complete cognitive user model has to represent the user's cognitive style and personality factors, the user's goals and plans, the user´s capabilities and preferences, and the user's beliefs and knowledge. These characteristics are represented in a way that fits our genetic approach.

3. Information filtering

Information Filtering (IF) describes the processes of distribution and delivery of information to users of communication systems. Information filters assist users in finding relevant information but are also used to target information to potentially interested users. Information Filtering Systems (IFS) primarily handle primarily unformatted textual data like documents, semi-structured such as electronic-mail messages (*e-mail*), NetNews articles, newswire stories or more complex structures like hypertext documents containing voice, graphics, and pictures. IFS process streams of incoming data based on descriptions of a single user or groups of users. These user "profiles" typically describe long-term interests [7] and individually depend on the fact how the user reacts to an incoming stream of information. The user can either select information items (positive kind of filtering) or remove items (negative kind of filtering).

IF is closely related to *Information Retrieval* (IR) which is concerned with the representation, storage, organisation, and accessing of information items such as documents [8]. The fundamental problem in IR is to identify the relevant documents from nonrelevant ones according to a particular user's request.

There are three domains classifying IR research: indexing, retrieval and evaluation.

The *document representation* or *indexing* process performs the task of assigning information items to documents for retrieval purposes. An indexing language maps the contents of documents to a textual representation.

The three main retrieval models in IR - *boolean, vectorspace* and *probabilistic model* - differ with respect to the matching process between user queries and document representations.

The *Boolean* model [8] compares queries and document descriptions by exact matching of index terms with the help of boolean operators. A disadvantage of the exact match model is that the entire document space is divided into two sets of relevant and nonrelevant documents with a ranking of documents according to a query.

In the *Vector Space* model [8] queries and documents are represented as vectors in a multidimensional space and compared with the help of statistical methods e.g. the Cosine, Dice or Jaccard function [9].

The *Probabilistic IR* model estimates the probabilities of a document's relevance by using the Bayes' theorem. The model is based on the *probabilistic ranking principle* (PRP) [10] which states that optimum retrieval is achieved when documents are ranked according to decreasing values of their probability of relevance with respect to the current query.

	Information Filtering	***Information Retrieval***
System input	dynamic datastream	static database
User goals	long-term periodic desires	short-term intentions
User behavior to incoming data	reacts to	actively searching
Information processing	removing	finding (selection)
Information flow	distribution and organization	representation and organization
Use of the system	repeated	single
Representation of user interests	profiles	queries
Environment	more or less privacy	more or less public
User-groups	undefined	well-defined

Table 1: Information filtering vs. information retrieval

Differences between IF and IR are described in Table 1. The retrieval models described above are applied to IF [7], [11]. Simple *Keyword Matching* determines whether the user's information interests match the incoming information items of the system.

It is most important to the process of filtering that the indexing component consists of:

- a lexical scanner,
- a morphological component, and
- a component for generating postings.

4. Linguistic analysis

Within CIFS linguistic analysis consists of two separate modules: the indexer and the parser. Additionally, a pre-filter reduces the amount of relevant e-mails. For this purpose, the pre-filter contains a set of keywords and phrases that initially describe the user's current interests. These descriptors weed out e-mails that are not about a *topic* of interest.

Indexer

As basis for our computation we first transform the document text into a sequential word list and remove all stop words occurring. By making use of a lemmatising module these sequential lists are then converted into a word index. The lemmatising module was designed to handle the following four important morphological phenomena [12]:

- spelling errors
- vowel-gradation
- inflexions
- suffixes

Although we applied approximative methods not taking into account any kind of morphological surface pattern, we achieved a very high accuracy without significant loss of speed.

Parser

Due to the requirements of information filtering with regard to processing time, natural language analysis can only be performed by *information extraction* and not by *text understanding* [13]. This implies a cascaded architecture which does not perform a complete linguistic processing for the whole document but narrows the scope by first retrieving text segments of special interest which can then be analysed more carefully.

Therefore, based on the result of the indexation process, the index entries are in the first step matched with the *user profile*. Only if the document is estimated relevant, the contexts of the retrieved *trigger words* are further analysed by use of a simple parsing algorithm to detect syntactic constructs (e.g. noun phrase, verb phrase) or special patterns.

According to the *user profile*, the interesting pieces of information contained in the context are mapped to the slots of a frame used as semantic representation scheme [14]. Finally, all resulting frames are merged to obtain one consistent representation of the contents of the document [15]. Once more it must be stressed that the final semantic representation will only model that part of contents which is relevant to the user, all other aspects will be filtered out.

5. Evolutionary algorithms

Evolutionary algorithms (EA) use computational models of evolutionary processes as key elements in the design and implementation of problem-solving systems [16]. A variety of evolutionary computational models were proposed. They share a common conceptional base of simulating the evolution of individual structures via processes of *selection, mutation,* and *reproduction.* The processes depend on the perceived performance of the individual structures as defined by the environment. EAs maintain a population of structures which evolve according to rules of selection and other operators that are referred to as *search (genetic) operators* such as *recombination* and *mutation.* Each individual in the population receives a measure of its *fitness* in the environment. Reproduction - creation of a new individual from two parents - focuses attention on individuals of high fitness, thus exploiting the available fitness

information. Evolutionary computation (EC) includes research in *genetic algorithms* [17], *evolution strategies* [18], *artificial life* [19], and so forth.

```
Procedure EA;
    (1) Start with an initial time
        t := 0;
    (2) Initialize a random population of individuals
        initpopulation P (t);
    (3) Evaluate the fitness of all initial individuals in the
        population
        evaluate P (t);
    (4) Test for termination criterion (time, fitness, etc.)
    (5) While not done do
        (a) increase the time counter
            t := t + 1;
        (b) select a subpopulation for offspring production
            P' := selectparents P (t);
        (c) recombine the genes of selected parents
            recombine P' (t);
        (d) perturb the mated population
            mutate P' (t);
        (e) evaluate its new fitness
            evaluate P' (t);
        (f) select the survivors from actual fitness
            P := survive P, P' (t);
        end while;
end EA;
```

Figure 1: Evolutionary algorithm

The fitness of an individual measures how well an individual solves a task. This is important for the *genetic operators* to act. The crossover operator forms a new chromosome by combining parts of each of two parent chromosomes. The selection operator evaluates the fitness of each chromosome and the mutation

operator forms a new chromosome by making alterations to the values of genes in a copy of a single parent chromosome.
EAs are sufficiently complex to provide robust and powerful *adaptive search mechanisms*. We use the EC approach to adapt descriptions of e-mails.

6. Cognitive Information Filtering System

Figure 2 shows the architecture of CIFS. The central component of the system, the cognitive information filter contains a population of information objects (words, user usage patterns, phrases) called e-mail agents, a monitor, and a knowledge base.

The input to the filter module are the semantic representations of the e-mails. They form the initial population of the evolutionary algorithm, the so-called e-mail agents. E-mail agents obtain the following attributes: the user's evaluation, a bid in the range [+1, -1], a bid learning rate, and counters for frequency, recency and spacing.

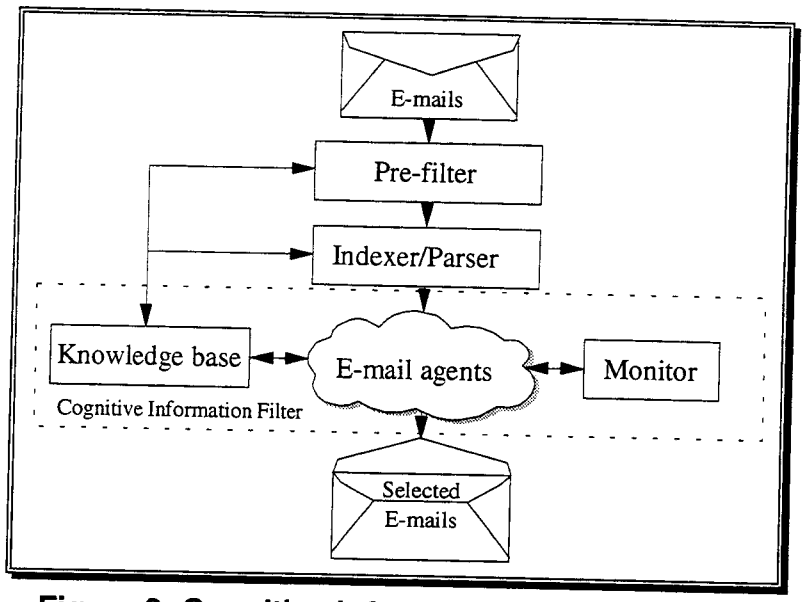

Figure 2: Cognitive information filtering system

The learning algorithm in Figure 3 describes the genetic adaption where e-mail agents cooperate and compete for correct evaluations of the user's actual interest rating. E-mail agents learn by adjusting their evaluation, thus moving it

closer to the user's evaluation. The payoff schema prevents the population from increasing.

Repeat

(1) For any particular set of e-mail agents, measure the fitness (strength S_i) of each of its descriptions.

(a) After an e-mail has been evaluated by a user, agents adjust their fitness:

Calculate for each competing agent the new strength S_i' after payoff:

$S_i' = (S_i + LR)/(1 + L)$ with L .. learning rate and R .. user rating.

(b) Enforce competition among n e-mail agents by the payoff function [9]

$P_i = E/n + 1/n - e_i - F$ $\quad F$ .. fee

error: $e_i = |R - S_i|$ $\quad E = \sum e_i$

(2) After each user evaluation randomly select e-mail agents for reproduction.

Cross over the fittest agents to produce offsprings.

Until a predefined number of generations is obtained

Figure 3: Genetic adaption of e-mail descriptions

Cognitive aspects are implemented as follows. E-mail agents that run out of strength leave the population for a calculated period of time - at that moment they are not relevant - and get incorporated into the pre-filter. Agents above the average fitness serve

- as 'new' keywords in the pre-filter, e.g. the system reacts to the fact that a user has not been interested in a topic for a peroid of time, or
- remain in the population and get a last chance to be useful.

The knowledge-base contains the semantic representation of the user profiles. Individual interests are mapped to frames. Their dynamic adaption is induced by e-mail agents .

The monitor component simulates a user's behavior and controls the agent's lifecycle. The monitor is a kind of feedback mechanism, measuring how effectively the history of usage patterns predicts current usage patterns and the probability that an item is needed given the history for such information [20].

Our system applies three time- and context-sensitive user preference functions: *Frequency* refers to the number of times an information is needed within a specified period of time. *Recency* counts the amount of time elapsed since the last need for an information item, and *spacing* refers to time distribution of the exposures to these information items. Equations using these user preference functions serve as predictors for the current need for an information item.

7. Conclusion

CIFS supports two general aspects:

- individual user preferences in daily operation with his/her e-mail system, and
- the actual contents of messages that are deemed interesting or uninteresting.

A further advantage is the time saving device to cope with the information overload problem.

References

[1] G. Fischer, C. Stevens. Information Access in Complex, Poorly Structured Information Spaces. *Proc. CHI Conf.*, pp.63-70, April 1991.

[2] T. Malone et al. Intelligent Information-Sharing Systems. *CACM* 30(5), pp.390-402, 1987.

[3] H.K. Shuldberg et al. Distilling Information from Text: The EDS Template Filler System. *JASIS* 44(9), pp.493-507, 1993.

[4] S. Pollock. A Rule-Based Filtering System. *ACM TOIS* 6(3), pp.232-254, 1988.

[5] R.B. Allen. User Models: Theory, Method, and Practice. *Int. J. Man-Machine Studies* 32, pp.511-543, 1990.

[6] J. Anderson. Cognitive Psychology and its Implications. *A Series of Books in Psychology,* R. Atkinson et al. (eds), New York: W.H. Freeman, 1985.

[7] N.J. Belkin, W.B. Croft. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *CACM* 35(12), pp.29-38, 1992.

[8] G. Salton, M.J. McGill. *Introduction to Modern Information Retrieval,* New York: McGraw Hill, 1983.

[9] T. Norault et al. A Performance Evaluation of Similarity Measures, Document Term Weighting Schemes Representations in a Boolean System. *Information Retrieval Research,* R.N. Oddy et al. (eds), pp. 57-71, 1981.

[10] S.E. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation* 33, pp. 294-304, 1977.

[11] P.S. Jacobs, L.F. Rau. SCISOR: Extracting Information from On-line News. *CACM* 33(11), pp.88-97, 1990.

[12] W. Winiwarter, A.M. Tjoa. Morphological Analysis in Integrated Natural Language Interfaces to Deductive Databases. *Proc. of the Fourth Int. Workshop on Natural Language Understanding and Logic Programming,* Sep. 1993.

[13] J.R. Hobbs et al. Description of the Fastus System for MUC-4. *Proc. of the 4th Message and Understanding Conf.,* pp.169-177, 1992.

[14] D. Ayuso et al. BBN: Description of the PLUM System Used for MUC-4. *Proc. of the 4th Message and Understanding Conf.,* pp.268-275, 1992.

[15] A. Meyers, D. de Milster. Description of the TexUS System Used for MUC-4. *Proc. of the 4th Message and Understanding Conf.,* pp.207-215, 1992.

[16] L.J. Fogel, J.W. Atmar (eds). *Proc. of the First Annual Conf. on Evolutionary Programming.* Evolutionary Programming Society, San Diego, CA, 1992.

[17] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley, Reading, Mass., 1989.

[18] H.P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Birkhäuser Verlag, Basel, Stuttgart, 1977.

[19] T.S. Ray. Is it Alive, or is it a GA. *Proc. of the 1991 Int. Conf. on Genetic Algorithms,* pp.527-543, Kaufmann, 1991.

[20] J.R. Anderson. *The Adaptive Character of Thought.* Lawrence Erlbaum Associates: Hillsdale, New Jersey, 1990.