PERGAMON

Information Processing and Management 37 (2001) 187–198

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Automatic keyword identification by artificial neural networks compared to manual identification by users of filtering systems

Zvi Boger [a], Tsvi Kuflik [b], Peretz Shoval [b,*], Bracha Shapira [c]

[a] *Nuclear Research Center – Negev, Optimal–Industrial Neural Systems Ltd, Beer-Sheva, Israel*

[b] *Information Systems Program, Department of Industrial Engineering and Management, Ben-Gurion University, 84105 Beer-Sheva, Israel*

[c] *Department of MSIS, School of Business, Rutgers University, NJ, USA*

Received 23 November 1999; accepted 14 June 2000

Abstract

Information filtering (IF) systems usually filter data items by correlating a vector of terms that represent the user profile with similar vectors of terms that represent data items. Terms that represent data items can be determined by experts or automatic indexing methods. In this study we employ an artificial neural network (ANN) as an alternative method for both IF and term selection and compare its effectiveness to that of ''traditional'' methods. In an earlier study we developed and examined the performance of an IF system that employed content-based and stereotypic rule-based filtering methods in the domain of e-mail messages. In this study, we train a large-scale ANN-based filter, which uses meaningful terms in the same database as input, and use it to predict the relevance of those messages. Our results reveal that the ANN relevance prediction out-performs the prediction of the IF system. Moreover, we found very low correlation between the terms in the user profile (explicitly selected by the users) and the positive causal-index (CI) terms of the ANN, which indicate the relative importance of terms in messages. This implies that the users underestimate the importance of some terms, failing to include them in their profiles. This may explain the rather low prediction accuracy of the IF system.

1. Introduction

Information filtering (IF) is a research area that offers tools for discriminating between relevant and irrelevant information. It provides personalized assistance for continuous retrieval of

* Corresponding author.

E-mail addresses: zvi@maxsw.com (Z. Boger), tsvikak@bgumail.bgu.ac.il (T. Kuflik), shoval@bgumail.bgu.ac.il (P. Shoval), shapira@business.rutgers.edu (B. Shapira).

0306-4573/01/$ - see front matter
PII: S0306-4573(00)00030-3

information in situations of information-overflow, in general, and on the Internet, in particular. IF combines tools from the field of artificial intelligence (AI), such as intelligent agents or software robots "softbots", guided by user profiles, with information retrieval (IR) methods geared to representing, indexing and retrieving content (Etzioni & Weld, 1994, 1995; Balabanovic' & Shoham, 1997). IF differs from traditional IR in that it deals with users who have long term interests (information needs) that are expressed by means of user profiles, rather than casual users whose needs are expressed as ad hoc queries (Belkin & Croft, 1992).

Agent technology provides a framework for automated information-gathering over the Internet. Indeed, quite a few applications have been developed for this purpose. Passive filtering of incoming messages, like e-mail and Usenet data, presents one such application (Maes, 1994). Active information seeking, like interesting web-site detection and browsing assistance, presents another application (Lieberman, 1995, 1997; Balabanovic' & Shoham, 1997; Joachims, Freitag & Mitchell, 1997). The heart of such an agent is the "user profile": a representation of user needs which is constantly updated according to user feedback. The performance of IR and IF systems (namely, their ability to retrieve or filter relevant information) depends on many factors; one of them is the selection of keywords that represent the user query or profile.

Artificial neural networks (ANN) have been used in recent years for modeling complex systems where either no explicit equations are known or the equations are too ideal to represent the real world. The ANN can form predictive models from data available from past history. Advanced algorithms can train large ANN models, with thousands of inputs and outputs. Analysis of the trained ANN may extract useful knowledge from it.

Keyword selection for the specification of user profiles or queries is an important, and sometimes frustrating, task. In this paper, we try to handle this task by training a large-scale ANN-based filter which uses all meaningful words in the document space (i.e., data items) as inputs and the user-given importance rating as output. Analysis of the trained ANN may achieve the automatic identification of important keywords. To test this technique, we use a rather small, user-ranked database of e-mail messages that was compiled for testing filtering methods by statistical techniques.

The rest of this paper is structured as follows: Section 2 reviews some essential concepts in IR and IF. Section 3 provides a brief introduction on ANN modeling techniques and Section 4 describes the possible application of ANN to IF. In Section 5, we present the results of a previous study on filtering e-mail messages, which combines content-based and rule-based sociological filtering, integrated with user stereotypes. This lays the ground for our implementation of the ANN approach to the same data (e-mail messages). Section 6 explains the ANN model building and compares the results of the ANN approach as an information filter to the results of the earlier study. Section 7 analyzes the ANN prediction of the importance of words in text and compares it to user evaluation of term importance. Section 8 concludes the paper and discusses further research issues.

2. Concepts in IR and IF

IR may be characterized as "leading the user to those documents that will best enable him/her to satisfy his/her need for information" (Robertson, 1981). This definition (among many others)

can be described in a model of IR in which the user seeks, by the use of queries, relevant information in some data space (e.g., a database of documents). Years of research in IR have yielded many useful results, among them document representing and indexing methods.

Interesting lessons learned from IR are in three main areas: text representation, retrieval techniques and acquisition of user information needs. The vector space model (Salton & McGill, 1983), according to which a document is represented by a (possibly weighted) vector of terms, is a very popular one among the research community in IR. A large variety of methods have been compared to the vector model, but the consensus seems to be that, in general, the vector model is either superior to or almost as good as the known alternatives. The user information interests (i.e., queries) can be represented as a vector of keywords in a similar way (Belkin & Croft, 1992; Oard & Marchionini, 1996; Aas, 1997). The main task of IR, given user queries and data representation, is to match the two vectors of terms and thus provide the user with relevant data items that best match the query.

There are several IR models for determining the weights of terms in documents or queries. The classic Boolean model considers index terms to be either present or absent in a document; thus, the index term weights are all binary. The Boolean model determines whether a document is *relevant* or *non-relevant*. There is no partial match to a query. Exact matching may lead to the retrieval of too few or too many documents. In the vector space model, *non-binary* weights are given to index terms in queries and documents. The weights reflect the importance of the terms to the query or the document. The weights are used to compute the *degree of similarity* between each document and the query; thus, partial matching is achieved, which is known to improve retrieval performance. One well-known method for the determination of term weights in documents is term frequency $*$ inverse document frequency (TFD $*$ IF) (Salton & McGill, 1983). It assigns a weight to a term in proportion to the number of its occurrences in the document and in inverse proportion to the number of documents in which it occurs at least once. This method is based on the statistical observation that the more times a term appears in a text the more relevant is the topic, and the more documents it appears in, the more poorly it discriminates between documents. Another model for the determination of term weights is a probabilistic model that uses the difference in the distribution behavior of words over all documents in a collection to guide the selection of index terms.

In IF systems, user needs are expressed as profiles. A profile represents his/her long-term information needs. There are two main distinct user profile approaches (Balabanovic' & Shoham, 1995; Oard & Marchionini, 1996; Aas, 1997):

- *Content based profile*: Represents the user's areas of interest by a set of terms. The profile can be defined "manually" (i.e., provided by the user), or generated automatically from a sample set of data items that are known to be of interest to the user.
- *Collaborative profile*: This approach matches users' information rating patterns. The main assumption is that people with similar rating patterns seem to like the same kind of information. Therefore, it makes sense to offer them information that was liked by people with similar rating pattern – "like minded people".

In order to support user needs, the user profile should be adaptable according to user reaction to the information provided to him/her, since user interests tend to change over time. This calls for incorporating learning mechanisms into user profiling (Billsus & Pazzani, 1998). A learning mechanism that suites user profiling is inductive learning: an initial profile is generated (either

manually or automatically, as explained) following some initial training from examples and then additional learning is done according to user feedback in order to continuously improve the user profile.

3. Brief introduction to ANN modeling

ANN modeling is done by learning from known examples. A network of simple mathematical "neurons" is connected by weights. Adjusting the weights between the "neurons" trains the ANN. The reader is referred to the many books and journal papers published on ANN modeling. A very good, monthly-updated source on any ANN subject is found on the Internet (Sarle, 2000).

Two main branches of ANN are in use, classified by their training methods: supervised and unsupervised. The *supervised ANN* branch uses a "teacher" to train the model, where an error is defined between the model outputs and the known outputs. Error back-propagation algorithms adjust the model connection weights to decrease the error by repeated presentations of input vectors (Rumelhart, Hinton & Williams, 1986). The *unsupervised ANN* branch tries to find clusters of similar inputs when no previous knowledge exists about the number of the desired classes. The best-known algorithms are the self-organized map (SOM) and adaptive resonance theory (ART).

In both cases, once the ANN is trained, and verified by presenting inputs not used in the training, it is used to predict outputs of new inputs presented to it. An example of SOM application to large database similarity finding can be found in Kohonen (1997) and Graupe and Kordylewski (1998). An ART application can be found in Hui and Lau (1997), while recent feed-forward ANN examples can be found in Creput and Caron (1997) and Kurbel, Singh and Teutenberg (1998).

There are several obstacles to applying ANN to large systems containing large numbers of inputs and outputs. Most ANN training algorithms need thousands of repeated presentations ("epochs") of the inputs to finally achieve small modeling errors. Large ANN's tend to get stuck in local minima during the training. As most ANN training starts from initial random connection weights sets, and the number of neurons in the hidden layer are usually determined by heuristic rules, many re-training trials are needed to achieve good models.

An efficient training algorithm, named PCA-CG, can easily train large-scale ANN models, as it pre-computes non-random initial connection weights from the manipulation of training data sets. A proprietary algorithm avoids and escapes local minima. A useful added feature of the PCA-CG algorithm is that it recommends the number of neurons in the hidden layer based on the number of principal components in the data. More details can be found in Guterman (1994) and Boger and Guterman (1997). Apart from these features, the ANN architecture used by the PCA-CG algorithm is the most common one – fully connected forwards only, one hidden layer, and sigmoid activation function. We use the conjugate gradient error back-propagation minimization search (Leonard & Kramer, 1990). The PCA-CG algorithm has been successfully used to train ANN models of industrial plants with hundreds of inputs and outputs (Boger, 1992, 1997). It has also been used for spectra and image analysis (Boger & Karpas 1994a,b; Lerner, Guterman, Dinstein & Ronen, 1995; Greenberg & Guterman, 1996).

Once trained, the ANN may be analyzed for knowledge extraction. One way is to estimate the relationships between the inputs and the outputs. A simple algorithm can calculate a causal index

(CI) that gives the relative magnitude and the sign of the influence of each input on each output (Baba, Enbutu & Yoda, 1990). The CI is calculated as the sum of the products of all "pathways" from each input to each output,

$$\mathrm{CI} = \sum_{j=1}^{h} W_{kj} * W_{ji},$$

where there are h hidden neurons, W_{kj} is the connection weight from hidden neuron j to output k and W_{ji} is the connection weight between input i and hidden neuron j. Although there is no rigorous theoretical basis for this algorithm, experience with large-scale plant modeling shows that the CI's found by this algorithm do indicate the known (and sometimes previously unknown) relationships in the data.

Another knowledge extraction technique is the ranking of inputs according to their relevance to the ANN prediction accuracy (Boger & Guterman, 1997). The least-relevant inputs may be discarded and the ANN re-trained with the reduced input set to give better prediction accuracy. The explanations for this possible improvement are:

(a) the elimination of noise or conflicting data in the non-relevant inputs;

(b) reduction of the number of connection weights in the ANN, which improves the ratio of the number of examples to the number of connection weights, thus reducing the chance of over-fitting.

The issue of over-training has troubled ANN model developers for a long time. It can be argued that the number of connection weights in the ANN has to be considerably smaller than the number of training examples. However, experience with real-world large systems indicates that this requirement is too conservative, as there are hidden relationships in the data (Boger, 1993, 1997). For instance, in a distillation column, there is a relation between the mixture composition, column pressure and boiling temperature. Similar hidden relationships may be found between words in a text.

4. The application of ANN to IF

The idea of matching the capabilities of ANN modeling to IR is not new. A search of the 1994–1998 INSPEC database with these terms yielded more than 60 papers dealing with this combination. Most of the papers use the SOM or the ART techniques to form clusters of textual documents based on the similarity of the keywords in the texts. Once trained, the ANN will classify new documents as belonging to one of these clusters.

The ability of the ANN to model non-linear, non-obvious relationships can be applied to the matching of textual features (inputs to the ANN) to user profile (ANN outputs). In contrast with the statistical methods used for the required modeling, no assumptions need to be made, such as normal distribution, subjective selection of the number of terms and the form of the model equations.

Of the two ANN training methods, supervised training should be preferred, as it is more adjustable to an individual user profile. The SOM may classify texts according to their similarity, but eventually the user will have to evaluate the number of clusters (too few or too numerous) and rank the clusters according to their degree of interest.

The most important feature of ANN modeling is that the user need not specify what features to extract from the text, such as keywords. If the ANN can use all the words in the text as inputs, the post-training ANN analysis should reveal which the more relevant words in the text are, according to the user profile. Thus, the subjective keywords selection process for queries is avoided, eliminating the frustration of getting too many responses to a general query, or the suspicion of missing important results from too narrow a selection of keywords. The trained ANN should then act as a filter, evaluating each additional text according to the ANN-predicted match to the users' profile requirements.

In Section 6, we try to prove the feasibility of ANN modeling and keyword extraction, employing a database used in a previous modeling of users' profiles by other techniques.

5. IF utilizing content-based and rule-based methods

Shapira, Shoval and Hanani (1999) have developed a dual-method model and system for filtering and ranking the relevance of information. One method is "content-based filtering", which is based on the correlation of two weighted vectors of terms, one representing the user profile and the other representing the data items. The other method is "sociological filtering" integrated with user stereotypes. Stereotypes are used to infer user preferences and habits as to their information needs, from their stereotypic belonging. Users are related to stereotypes according to their sociological parameters. Each stereotype is represented by a set of sociological parameters that are common to users who "belong" to the stereotype, and by a set of IF rules that are typical to the stereotype. The rules prescribe customary ways in which users who belong to a certain stereotype use or filter information. The rules refer to parameters in data items. For example, for an e-mail message, the parameters may include its goal (purpose), quality of source and length. A rule specifies the relevance of a data item to a user who belongs to the stereotype, with respect to a certain parameter. It has the following format:

$$\text{If } \langle \text{param} \rangle = \langle \text{value\_of\_param} \rangle \text{ then rank} \leftarrow \langle \text{value} \rangle,$$

where ⟨param⟩ is a parameter of the data item, such as its goal or length, ⟨value_of_param⟩ is a value from a set of possible values of the parameter and ⟨value⟩ is a 1–7 number that signifies, for each stereotype, the degree of relevance of a data item to the corresponding ⟨value_of_param⟩.

Here is an example of a rule: If (goal = 'conference') then rank ← 5.9. It determines that, for a certain stereotype, messages announcing conferences are of high relevance. Each filtering rule, if found relevant for a data item being evaluated, grants a relevance value to that item. The overall "sociological relevance" of the data item is the average relevance values of those rules. When the filtering system evaluates a data item for a user, it first identifies that user's stereotype (based on the similarity of the user's sociological profile to the respective profiles that represent each of the stereotypes). Then the filtering rules of that stereotype are applied and the relevance of the data item is calculated, as described above.

A prototype system was developed to test the applicability of the model for filtering e-mail messages and experiments were run to determine the effects of combining the two filtering methods in various filtering strategies. Ten users were asked to evaluate e-mail messages that they received and rank their relevance on a 7-point scale. The users were university people

professionally related to information science, namely, researchers, students, information specialists and technical staff. They evaluated messages that were sent to them from professional list-servers (such as the DBWORLD and ISWORLD) which they subscribed to. The content-based profile of each user was determined as follows: each of the ten participants received a proposed list of terms that was generated from a "training set" consisting of several dozens of his/her incoming e-mail messages. The list included the most frequently occurring terms in those messages. (It was prepared with the aid of a special software that extracts meaningful terms from messages, employing a stemming algorithm, look-up tables and a stop-list, and counts the frequency of the meaningful terms.) Each participant was asked to review the proposed list of terms, add or drop terms, and weigh each term for its degree of interest to him, using a 0–100 scale. The union of all those terms formed the terms database, which was used in content-based filtering. A stem look-up table was built with all English stemming variations of those terms.

The system evaluated the same e-mail messages several times, each time employing a different filtering strategy, namely, content-based alone, sociological filtering alone, both methods (parallel), where the final relevance rank was the average of both methods, content-based followed by sociological and sociological followed by content-based filtering. In each of the last two cases, the first (primary) method contributed 70% of the final relevance rank. In all cases, the relevance rank of the system was expressed on a 7-point scale, as used for user evaluations, to enable comparisons of results.

To enable content-based filtering, the system analyzed each message by stemming the words, eliminating "stop-list" terms and counting the frequency of the meaningful stems, thus generating a weighted vector of terms, which was correlated with the user's content-based profile. To enable sociological filtering, the system used the same word stems and employed specific algorithms attached to each rule to determine if and how relevant the rule is to the message. Based on that, it calculated the relevance rank for each rule and computed the overall sociological rank.

The performance of the system (i.e., the relevance ranks determined by the system), according to those various strategies, was compared to the user evaluations. The comparison's objective was to determine which filtering strategy generated evaluations that were mostly correlated with the user evaluations, and whether any filtering strategy was consistently more effective than the others for different user stereotypes. The comparison of filtering strategies was done within stereotypes because the experiments were meant to examine the effect of sociological filtering as integrated with stereotypes. For every filtering strategy, a vector that included the ranks of messages given by users who belonged to a certain stereotype was correlated (Pearson correlation) with the system-produced ranks for the same messages. This enabled conclusion of the extent to which the system satisfied users' needs and comparison between the various filtering combinations by computing the differences between correlations. Some relevant results of the experiments are summarized in Table 1. Each column refers to one of 4 stereotypes determined for the user population in that experiment. Each row shows the correlation (r) between the system's rankings of the messages and the users' rankings. (N is the number of messages evaluated by users within each stereotype.)

The results reveal that content-based filtering alone is usually more effective than sociological filtering alone, but that combinations of both methods yield better results than each method individually. The best filtering strategies are achieved when the two methods are used in parallel, or when content-based filtering is the primary method, followed by sociological filtering. At any

Table 1
Correlation (r) results per stereotypes

Filtering strategy	Stereotype 1 ($N=429$)	Stereotype 2 ($N=469$)	Stereotype 3 ($N=179$)	Stereotype 4 ($N=350$)
Content-based	0.58	0.50	0.47	0.41
Sociological	0.48	0.48	0.43	0.65
Parallel (both)	0.59	0.64	0.59	0.71
Content-based (70%) + sociological (30%)	0.61	0.44	0.62	0.53
Sociological (70%) + content-based (30%)	0.52	0.07	0.51	0.66

rate, correlation between system predictions and user evaluations are usually not very high, ranging (with one exception) between 0.4 and 0.7.

6. Training the ANN as information finder

The gathering and pre-processing of the training and testing data is the first phase (and, frequently, the most time-consuming one) of ANN modeling. In our case, most of the data were already collected and classified as described in the previous section – we had the 10 e-mail recipients' classification of the importance of the messages they received, on a scale of 1–7. The words in their messages were already stemmed and common words removed by a stop-list. We also had the content-based profiles, i.e., the user-given weights of each word as a subjective measure of relevance, on a scale of 0–100. Altogether, 1524 e-mail messages were used to form 10 user profiles, aiming to filter future e-mail messages according to their importance to the recipient.

In our case, the aim was twofold: (a) to predict relevance of the messages and (b) to evaluate the ability of the ANN to identify important keywords for the user profile. All the stemmed words in the messages were combined into one "keyword" list, 425 in length. Each e-mail message was transformed into a binary vector of 425 ones and zeros, the ones signifying the presence of the indicated word in the message. Data preprocessing to the form used in the ANN training consisted of changing the one and zero binary inputs to +1 and −1 values, respectively, and adding a small random noise value to them. This was done in order to avoid having an empty input column vector of constant −1 values. The 1–7 output range was transformed into the usual 0.1–0.9 range, which avoids asymptotic numerical problem during the training.

The PCA-CG algorithms trained 10 fully connected ANN models, one for each user. The models were trained with all available word vectors, with the 1–7 rating as a single output. Normally, some of the available examples are not used in the training, being left aside to check the generalization capacity of the trained ANN. In this case, all examples were used for the training, as our aim was to compare the ANN modeling with the previous models. The PCA-CG training algorithm suggested that 6 neurons (at most) in the hidden layer would be sufficient to achieve good prediction rates. The small number of hidden neurons compared with most other published ANN models is an important result of the PCA-CG algorithm we use, when we specify that the number of principal components chosen will explain 70% of the variance in the data.

Some of the ANN models (Users 6, 8 and 9) could not be trained to the desired accuracy. Some possible reasons for this are noted at the end of Section 3. Each "imperfect" ANN model was

Table 2
Prediction accuracy of ANN and "traditional" filtering methods (correlation between model prediction and user evaluation)

User	ANN	Content-based	Sociological	Parallel (both)	Sociological (70%) + content-based 30%)	Content-based (70%) + sociological (30%)
1	0.96	0.60	0.42	0.53	0.56	0.46
2	0.90	0.43	0.31	0.48	0.40	0.31
3	0.88	0.44	0.38	0.47	0.44	0.41
4	0.80	0.25	0.47	0.51	0.36	0.47
5	0.76	0.31	0.53	0.56	0.27	0.55
6	0.99	0.64	0.61	0.72	0.71	0.66
7	0.93	0.23	0.46	0.52	0.46	0.51
8	0.97	0.59	0.61	0.70	0.42	0.63
9	0.98	0.73	0.54	0.68	0.79	0.59
10	0.99	0.62	0.65	0.78	0.74	0.77

analyzed to identify a set of the more relevant inputs, as described in Boger and Guterman (1997). These ANN models were re-trained with the reduced keyword sets to give high model prediction accuracy. For example, reducing the number of inputs (words) for User 8 from 305 to 160 improved the standard deviation of the importance prediction of 90 examples from 0.214 to 0.035. Table 2 summarizes the results of the ANN approach, compared to the results of the "traditional" IF approaches. As can be seen, the ANN prediction correlation is very good, ranging between 0.76 and 0.99, while the "traditional" methods yield correlation of up to 0.79 at most. These good results are expected: ANN modeling should improve the filtering accuracy compared with linear modeling, as the number of adjustable connection weights in the ANN is, in this case, higher than the number of examples.

7. ANN as predictor of term importance

The next stage was to analyze the trained ANN to learn more about the real importance of the words in the e-mail message. The CI of each model was calculated, as described in Section 3. The magnitude relative to the other calculated pathways and the sign of the summed products may be interpreted as the overall degree and direction of influence of a particular input on a particular output. In our case, a large, positive CI means that the presence of this keyword in message tends to increase its relevance. A CI close to zero means that the keyword cannot be used for classifying the message relevance. The meaning of a large, negative CI is that messages containing this keyword tend to be less relevant. However, we do not analyze its meaning in this context; we refer only to the large, positive CI keywords.

As described in Section 5, the users gave their estimates of the importance of each keyword, on a scale of 0–100, to construct their content-based profiles. One possible explanation for the low accuracy of the previous models' predictions is that the users did not correctly identify the relative importance of the keywords to be included in their profiles. Thus, a comparison between the positive CI and the user-given importance rating of these words should be interesting. As spurious, small CI may result from the small random noise addition to the inputs, only CI values

Table 3
Correlation between users' rated importance and the CI

User	1	2	3	4	5	6	7	8	9	10
Pearson	0.17	0.19	0.07	0.20	0.15	−0.02	0.20	0.05	0.35	0.09

Table 4
Mean importance rating of positive CI keywords with zero rating

User	1	2	3	4	5	6	7	8	9	10
ANN	3.54	3.36	5.01	3.78	3.04	4.81	5.33	4.30	4.24	5.33
Overall	2.74	2.73	3.41	3.04	2.29	4.03	3.88	3.07	3.78	4.80

whose magnitudes were larger than 0.1 were used. The results are given in Table 3, using the Pearson statistic to find a correlation.

As can be seen from that comparison, no correlation was found. This suggested that the users' subjective importance ratings (i.e., the user-defined content-based profiles) were not good enough or the CI method was faulty.

To test which hypothesis was correct, we calculated the mean *user-given* importance rating of the e-mails in which words with positive CI values had a *zero* importance user rating, that is, those words that the users believed had no importance for them. For each ANN-based user profile, we randomly selected a list of 10 terms which had positive CI values, but for which the user assigned 0 importance for the content-based profile. For comparison, we also show the overall mean rating of each user's e-mail messages. The results are given in Table 4.

As can be seen from Table 4, the keywords that the ANN CI identified as important show up in e-mail messages whose importance rating is higher than the overall mean. Thus, it seems that users under-estimate the importance of some of the words in the e-mail messages, which may explain the rather low prediction accuracy based on the users' subjective rating.

8. Conclusions and suggestions for further research

The results presented in Sections 6 and 7 show that a large ANN model can successfully be trained from the non-trivial words in a text and give better predictions than statistically derived models. The ANN can be analyzed by the CI method to identify the more important keywords. The user-given rating of the keywords may be too subjective, or given without too much attention. It would be interesting to receive their evaluations of the CI-generated keyword importance. It has to be kept in mind, though, that the database is rather statistically small, and maybe the 1–7 rating is too detailed for exact classification by users.

Some of the usability features of the new ANN-based model need further study and will be the subjects of further research. One of them is the number of words in the user vocabulary, as related to the ability to train large scale ANN models. ANN models with several thousands inputs were successfully trained with the PCA-CG algorithm; thus, it is only a matter of large enough computer memory and training time. However, once a large ANN is trained, additional re-training

based on the ANN trained on an existing list of words is an easy task, if enough blank columns are used in the original training. To reduce the number of words in the user vocabulary, it will be necessary to use subject-specific thesauri, so that a new word will be checked to find if it can be represented by a synonymous word in the current ANN vocabulary.

Other usability issues are the number of training data needed to train a good predictive ANN and the need of periodical re-training to include new words and the user evaluation of new incoming e-mail messages.

An important feature is the ability to predict correctly the relevance of a data item that contains new words. One approach is to calculate some metric that will generate a "not sure" warning alongside the ANN prediction. This warning will prompt the user to read this data item and judge its relevance in the next updating of the ANN. Such a metric may be based on the average (absolute) CI values of all the words in the database (with the new words, having a zero CI, lowering the result). A "not sure" warning value will be generated for items having an average CI below some threshold.

References

Aas, K. (1997). A survey on personalized information filtering systems for the World Wide Web. Report No. 922, Norwegian Computing Center.

Baba, K., Enbutu, I., & Yoda, M. (1990). Explicit representation of knowledge acquired from plant historical data using neural network. In *Proceedings of the International Joint Conference on Neural Networks, vol. 3* (pp. 155–160). San Diego.

Balabanovic', M., & Shoham, Y. (1995). Learning information retrieval agents: experiments with automated Web browsing. In *Spring Symposium on Information Gathering from Heterogeneous Distributed Environments*. AAAI 95.

Balabanovic', M., & Shoham, Y. (1997). Fab: content-based collaborative recommendation. *Communications of the ACM*, *40* (3), 66–72.

Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, *35* (12), 29–38.

Billsus, D., & Pazzani, M. (1998). Learning collaborative information filters. In *Proceedings of the International Conference on Machine Learning*. Madison, WI: Morgan Kaufmann.

Boger, Z. (1992). Application of neural networks to water and wastewater treatment plant operation. *Transactions of the Instrument Society of America*, *31* (1), 25–33.

Boger, Z. (1993). Artificial neural networks for quantitative stationary spectroscopic measurements. In *Proceedings of the 10th Israeli Conference on Artificial Intelligence, Computer Vision and Neural Networks*, *AICVNN-93* (pp. 185–194). Ramat-Gan, Israel.

Boger, Z. (1997). Experience in industrial plant model development using large scale artificial neural networks. *Information Sciences – Applications*, *101* (3/4), 203–215.

Boger, Z., & Guterman, H. (1997). Knowledge extraction from artificial neural networks models. In *Proceedings of the IEEE International Conference on Systems Man and Cybernetics, SMC '97* (pp. 3030–3035). Orlando, Florida.

Boger, Z., & Karpas, Z. (1994a). Application of neural networks for interpretation of ion mobility and X-ray fluorescence spectra. *Analytica Chimica Acta*, *292*, 243–251.

Boger, Z., & Karpas, Z. (1994b). Use of neural networks for quantitative ion mobility spectrometric measurements. *Journal of Chemical Information and Computer Science*, *34*, 576–580.

Creput, J.-C., & Caron, A. (1997). An information retrieval system using a new neural network model. *Cybernetica*, *XL 2*, 127–139.

Etzioni, O., & Weld, D. (1994). A softbot-based interface to the Internet. *Communications of the ACM*, *37* (7), 72–76.

Etzioni, O., & Weld, D. (1995). Intelligent agents on the Internet: fact, fiction and forecast. *IEEE Expert*, *10* (4), 44–49.

Graupe, D., & Kordylewski, H. (1998). A large memory storage and retrieval neural network for adaptive retrieval and diagnosis. *International Journal of Software Engineering and Knowledge Engineering,* (6), 115–138.

Greenberg, S., & Guterman, H. (1996). Neural networks classifiers for automatic real-world image recognition. *Applied Optics*, *35*, 4598–4609.

Guterman, H. (1994). Application of principal component analysis to the design of neural networks. *Neural, Parallel and Scientific Computing*, *2*, 43–54.

Hui, S.C., & Lau, K.L. (1997). An application of neural networks in document retrieval. *The New Review of Applied Expert Systems,* (3), 69–80.

Joachims, T., Freitag, D., & Mitchell, T. (1997). A tour guide for the World Wide Web. In *Proceedings of IJCAI97*.

Kohonen, T. (1997). Exploration of very large databases by self-organizing maps. In *Proceedings of the IEEE International Conference on Neural Networks, vol. 1*. PL1-6.

Kurbel, K., Singh, K., & Teutenberg, F. (1998). Search and classification of 'interesting' business applications in the world wide web using a neural network approach. In K. Forcht, *Proceedings of the 1998 IACIS Conference* (pp. 75–81). Cancun, Mexico, 1–3 October.

Lerner, B., Guterman, H., Dinstein, I., & Ronen, Y. (1995). Human chromosome classification using multilayer perceptron neural network. *International Journal of Neural Systems*, *6*, 359–370.

Leonard, J., & Kramer, M. A. (1990). Improvement of the back-propagation algorithm for training neural networks. *Computers in Chemical Engineering*, *14*, 337–341.

Lieberman, H. (1995). Letizia: an agent that assist Web browsing. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Montreal.

Lieberman, H. (1997). Autonomous interface agents. In *Proceedings of the ACM conference on Computers and Human Interface, CHI 97*. Atlanta, Georgia.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, *37* (7), 31–40.

Oard, D. W., & Marchionini, G. (1996). A conceptual framework for text filtering. Technical Report EE-TR-96-25 CAR-TR-830 CLIS-TR-96-02 CS-TR-3643.

Robertson, S. E. (1981). The methodology of information retrieval experiment. In K. S. Jones, *Information retrieval experiment* (pp. 9–31). London: Butterworths (Chapter 1).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Salton, G., & McGill, W. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Sarle, W. S., (2000). Frequently asked questions, comp.ai.neural-nets Users Group, ftp://ftp.sas.com/pub/neural.

Shapira, B., Shoval, P., & Hanani, U. (1999). Experimentation with an information filtering system that combines cognitive and sociological filtering integrated with user strereotypes. *Decision Support Systems*, *27*, 5–24.